A heuristic approach for solving real world nurse rostering problems

H. Frøyseth, M. Stølevik and A. Riise

Received: date / Accepted: date

Keywords Nurse rostering · Personnel scheduling · Iterated local search · CSP

1 Introduction

Nurse rostering is the process of creating a working plan, showing working hours for the employees in the plan over a given planning horizon. Making rosters is an important activity in health care services. Today, highly qualified health personnel spend a lot of their time making and updating rosters by hand. By using good, automatic tools for this work, a considerable amount of time can be freed for working with patients. Fair and better solutions for the employees are other benefits.

Nurse rostering and other personnel scheduling problems have been the subject of extensive studies [2] [4] [3]. We describe a flexible model and solution approach for the nurse rostering problem. Both general and individual preferences are taken into consideration in addition to satisfying the Working Environment Act and system of agreements. The work described has been carried out for Gatsoft AS [1], a vendor of a commercial nurse rostering tool. Originally nurse rostering was a manual process in this system. The integration of our solution has enabled automatic roster generation.

2 Problem definition

A roster plan can be viewed as a matrix (table) where the employees are the rows, and the days are the columns:

Helle Frøyseth, Martin Stølevik and Atle Riise
SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, NO-0314 Oslo, Norway.
Tel.: +47 22 06 73 00
Fax: +47 22 06 73 50
E-mail: helle.froyseth@sintef.no, martin.stolevik@sintef.no, atle.riise@sintef.no

[1] Gatsoft AS, http://www.gatsoft.no/

	Day 1	Day2	...	Day D
Employee 1	$S_{1,1}$	$S_{2,1}$		
Employee 2	$S_{1,2}$	$S_{2,2}$		
...			$S_{d,r}$	
Employee E				

The problem definition consists of several soft and hard constraints. These have been determined in close collaboration with our commercial partner and nurses with long experience in creating rosters manually.

The hard constraints in the model:

- There must be one and only one assigned shift per day for each employee.
- The manpower plan must be covered. A manpower plan is a description of what shifts, and how many, that are required on each day of the planning period. The manpower plan may change over the weeks.
- The sum of working hours for each employee in the plan must be within $\pm$ 2 % of the employee's contracted hours.
- Shifts with a competence demand can only be assigned to an employee with this competence.
- There must be a given minimum number of hours off between shifts.
- There must be at least one day off each week.
- The sum of work hours for each employee in one week must be below a given maximum.
- Weekends off are assigned by user.
- Shifts may be locked to a given employee and day.

The soft constraints:

- Minimize the overrun of max number of consecutive shifts.
- Minimize the overrun of max number of consecutive day-shifts.
- Minimize the overrun of max number of consecutive evening-shifts.
- Minimize the overrun of max number of consecutive night-shifts.
- Minimize number of shifts not preceded or followed by a shift in the same category (Day, Evening and Night).
- Minimize deviation from recommended number of shifts in each category.
- Minimize deviation from employee's contracted hours.
- Minimize number of unwanted shift combinations.
- Maximize number of wanted shift combinations.
- Compile together days off.

The objective is a weighted sum of penalties for all the soft constraints. The weights can be adjusted by the user.

3 Solution method

Our approach uses a customized algorithm, based on CSP [6], to create the initial solution and Iterated Local Search [5] for iterative improvement. SCOOP (SINTEF Constrained Optimisation Package), a generic C++ class library for modeling and solving optimisation problems, was used to model the problem and create the construction algorithm and the iterative improvement algorithm. A heuristic approach is chosen due to the large size this problem can have in real life applications. The algorithm consists of the following steps:

1. Construction algorithm
2. Until all soft constraints are fulfilled, or the search is interrupted by user:
   - Improve solution until local optimum is found
   - Remove parts of the solution
   - Recreate a legal solution by reusing the construction algorithm

3.1 Construction algorithm

The task for the construction algorithm is to create a roster plan that satisfies all the hard constraints. We use one variable per day / employee combination. Each variable's domain is the available shift codes. The algorithm attempts to assign a value to each variable, ensuring that all constraints are satisfied, in the following way:

1. Select the most critical day. We experienced that weekend days were the most critical as the weekends off were assigned by user. After weekend days the remaining days are selected in sequence.
2. Select the shift to try to assign on this day. To do this we keep track of number of shifts of each different type left to assign on given day, and for each shift type number of employees with this shift in its domain on given day. We select the shift that is hardest to assign. For example if there are 5 N-shifts and 2 D-shifts left to assign on given day, and 8 employees have an N-shift and 8 employees have a D-shift in their domain on that day, we select an N-shift since this is the hardest to assign.
3. Decide which employee to assign the shift to. Based on position percentage and manning demand we keep track of the approximate number of shifts in each shift category that is left to assign to a employee if shifts in each shift category is distributed evenly. The selected shift is assigned to the employee needing this shift the most on the selected day if possible.

3.2 Iterative improvement algorithm

For the iterated local search algorithm we utilize 3 neighbourhood operators:

- 2-Exchange - Swap shifts between two employees on the same day.
- 3-Exchange - Swap shifts between three employees on the same day.
- 2-Double Exchange - Swap shifts between two employees on two days. Swap a shift with a free-shift on one day, and the opposite the other day. The two days are not necessarily consecutive.

The neighbourhoods utilize focal points to limit search. A focal point is a description of a place in the solution where a change will probably improve the quality of the solution with regard to the objective. We create one focal point for each place (employee and day) in the solution that is involved in a violation of one or more soft constraints.

After each local optimization we use a diversification mechanism. This is done to produce a suitable new starting point for the local search and escape local optima. Our diversification mechanism clears the roster plan for a random number of employees between 2 and 6. Some are selected randomly among employees with violation of soft constraints, and some are selected purely randomly. A legal plan is recreated by reusing the CSP-based construction algorithm. If a solution is not found within a given time limit, the algorithm retries with a different selection of employees. The time limit is equal to the time elapsed creating the initial solution.

4 Results

To evaluate the algorithm, several real world cases from Norwegian hospitals were used. Solutions to these cases were generated by using the method described above. The solutions were evaluated by nurses with long experience in creating rosters manually. The nurses' assessments of the generated rosters were all positive. The roster plans were considered ready to be used in real life.

The construction algorithm creates a feasible solution for 20 employees and 12 weeks in approximately 1 second. This is a common problem size for a Norwegian hospital department. A feasible solution for 120 employees for one year is created in approximately 30 seconds.

5 Conclusions and Future work

Real life nurse rostering is an important and complex problem. Prelimenary results show that our model and algorithm offer a flexible and efficient solution to this problem.

We are in the process of running tests on standard benchmarks [1]. When this is carried out we are also planning to test and compare our algorithm with other algorithmic approaches. Plans for future work also include extending the model and improving the algorithm.

There are other interesting problems related to nurse rostering we are considering making solutions to as well. Examples are staffing, rostering for the holiday periods and absence due to sickness.

We are currently carrying out a five year research project called HOSPITAL [2]. The main objective in the project is to further develop and disseminate world leading competence in high-performance, robust and adaptable optimization methods for decision support in health care planning software. The project is mainly funded by The Research Council of Norway. The industrial partners

[2] HOSPITAL(Health care Optimisation Software for PlannIng, rosTering, And scheduLing) http://www.sintef.no/hospital

Gatsoft AS and DIPS ASA are co-funding the project. ASAP [3] and CMA [4] are research partners in the project.

Acknowledgements This work is supported by the Research Council of Norway, and Gatsoft AS.

References

1. Personnel scheduling data sets and benchmarks, http://www.cs.nott.ac.uk/˜tec/nrp/.
2. Edmund Burke, Patrick De Causmaecker, Greet Vanden Berghe, and Hendrik Van Landeghem. The state of the art of nurse rostering. *Journal of Scheduling*, 7(6):441–499, 2004.
3. B. Cheang, H. Li, A. Lim, and B. Rodrigues. Nurse rostering problems - a bibliographic survey. *European Journal of Operational Research*, 151(3):447–460, 2003.
4. A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004.
5. F. Glover and G. Kochenberger, editors. *Handbook of Metaheuristics*, chapter 11, pages 320–353. Kluwer Academic Publishers, 2003.
6. Edward Tsang. *Foundations of Constraint Satisfaction*. Computation in cognitive science. Academic Press, 1993.

[3] The Automated Scheduling, Optimisation and Planning (ASAP) research group, http://www.asap.cs.nott.ac.uk/

[4] Center of Mathematics for Applications (CMA) at the University of Oslo, http://www.cma.uio.no/