Computer assisted nurse rostering, step by step.

R. Hope

Hedmark University College
2450 Rena
Norway
rafaelh@hihm.no

Rotating nurse rosters are commonly used in hospitals and other healthcare institutions where employees work round the clock, seven days a week. Nurse rosters include shiftwork, which can be health damaging. In this respect difficulties in social life and lack of sleep are two major problems [1], [2]. The construction of acceptable nurse rosters and thus the quality of the nurse rostering process is of vital importance.

Nurse rostering is very time-consuming [3]. There are a lot of common rules and constraints which must be considered [4]. In addition, it is not unusual that different wards even within the same hospital have their own adoption to these constraints. It is also common that nurses involved have their own individual rosters or special requirements which also must be considered in the rostering process. The amount of part-time employees with different working hours further complicates the nurse rostering process.

In the nurse rostering process it is important to have local knowledge about the current ward. This makes it almost impossible to construct a software system for automatic rostering [5]. There are several commercial shift scheduling systems. Some of these systems include scheduling algorithms [6], [7]. As a first step towards a commercial nurse rostering system, we are proposing a prototype of such a system. This prototype assists the user step by step with the more difficult calculations involved. In this way the user can concentrate on making an acceptable roster according to the given rules. The steps are based on experience presented in Norwegian literature on nurse rostering [8],[9]. The Prototype is built on analysis of ordinary nurse rostering described in UML [10]. The database in the system is constructed on analysis from Object Role Modeling [11]. The whole system is implemented by Object-Oriented Programming in C++. The system leaves the vital decisions to the user. The prototype is built for the purpose of testing the use of an active database, elements of the graphical user interface (GUI) and the basic functionality needed in a rostering system.

The prototype assists the user in the following ten steps towards an acceptable rotating schedule:

1: Basic user inputs:

The main input is the workforce matrix. This matrix shows, for each employee, the available weekly working hours.

2: User adjustments and acceptance of the computer-suggested workload matrix and basic shifts:

The calculation of the suggested workload matrix is based on the total sum of available weekly working hours and experiences working with real world nurse rosters. The basic shifts will be the main dayshift, afternoon shift and nightshift. The suggested times for these shifts are selected among commonly used shifts in nearby hospitals.

3: User input of individual basic rosters with computer assisted update of the workload matrix.

A basic roster includes only the basic shifts. The update of the workload matrix involves operations like EQUAL, ADD and SUB on workload matrices covering different number of weeks.

4: User input of pre designed teams with computer assisted update of the workload matrix.

A team consists of a workforce matrix and a corresponding workload matrix. In a team, the calculated number of weeks in the nurse roster must be less than a given constant “Maxweeks”.

5: Computer constructed teams with user altering and acceptance.

The user controls the constructions by an input parameter. This step is important if the calculated number of weeks in the nurse roster exceeds Maxweeks. The process splits of the original workforce matrix and workload matrix in disjunct subsets which eventually become the different teams. The parameter decides how the distribution of shifttypes between the different teams shall be.

6: Computer generated rotating basic roster with user altering and acceptance.

This step involves the main rostering algorithm. There are many reported methods solving this problem [12], but very few using an object-oriented approach. The algorithm first calculates and places the working weekends, continues with the remaining nightshifts. The algorithm uses a combination of counting and schearching before placing the days off and finish with placing the afternoon shifts and dayshifts.

7: Individualizing the basic roster in accordance to the difference in available working hours among the nurses involved.

Experience shows that a team often consists of employees with different available working hours. The process in this step involves moving shift from employees who have few available working hours to employees who have more available working hours. This can be done either by the user with computer assistance or by the computer with user altering and acceptance.

8: Individualizing the basic roster in accordance to individual wishes and needs.

This must, for the time being, be done by the user with only standard assistance by the computer. This step together with steps 4-7, allows teamrostering and to some extent selfrostering [13].

9: Individualizing the basic roster to the real roster in accordance to each nurse available working hours.

This can be done either by the user with computer assistance or by the computer with user altering and acceptance.

10: Roster accepted by collective agreements.

Testing the prototype on real world problems shows that the basic functionality should include methods for step 8 which will more or less cover step 3. In addition, in step 6 the setting of days off after working weekends possibly needs some small adjustments. The use of a database makes the system difficult to install and test on other machines. The current GUI was mainly constructed for the purpose of testing the functionality. A new system is under construction without the database and hopefully a more intuitive GUI.

[1] P. Totterdell, E. Spelten, J. Barton, L. Smith, and S. Folkard, "On-shift and daily variations in self-report and performance measures in rotating-shift and permanent night nurses," *Work and Stress*, vol. 9, pp. 187-197, 1995.

[2] C. M. Poissonnet and M. Véron, "Health effects on work schedules in healthcare professions," *Journal of Clinical Nursing*, vol. 9, pp. 13-23, 2000 January.

[3] H. C. Lau, "On the complexity of manpower shift scheduling," *Computers & Operations Research*, vol. 23, pp. 93-102, 1996.

[4] G. Weil, K. Heus, P. Francois, and M. Poujade, "Constraint Programming for Nurse Scheduling," *Ieee Engineering in Medicine and Biology Magazine*, vol. 14, pp. 417-422, 1995.

[5] G. Laporte, "The art and science of designing rotating schedules," *Journal of the Operational Research Society*, vol. 50, pp. 1011-1017, 1999.

[6] N. Musliu, J. Gartner, and W. Slany, "Efficient generation of rotating workforce schedules," *Discrete Applied Mathematics*, vol. 118, pp. 85-98, 2002.

[7] S. J. Darmoni., A. Fajner., N. Mahé., A. Leforestier., M. Vondracek., O. Stelian., and M. Baldenweck., "Horoplan: computer-assisted nurse scheduling using constraint-based programming," *Journal of the Society for Health Systems*, vol. 5, pp. 41-54, 1995.

[8] K. Kluften, T. Kokaas, and J.-T. Strandås, *Arbeidsplanlegging: håndbok i oppsetting av skift- og turnusplaner*. Oslo: Tiden, 1999.

[9] S. A. Thrana, *Turnusplanlegging*. Oslo: Handels- og servicenæringens hovedorganisasjon, 2001.

[10] J. Rumbaugh, I. Jacobson, and G. Booch, *The unified modelling language reference manual*. Reading, Mass.: Addison-Wesley, 1998.

[11] T. A. Halpin, *Conceptual schema & relational database design*. Bellevue WA: WytLytPub, 1999.

[12] 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*, vol. 153, pp. 3-27, 2004.

[13] R. Silvestro and C. Silvestro, "An evaluation of nurse rostering practices in the National Health Service," *Journal of Advanced Nursing*, vol. 32, pp. 525-535, 2000.