An Integer Programming-Based Nurse Rostering System

John Thornton[1] and Abdul Sattar[2]

[1] School of Information Technology, Griffith University Gold Coast, Southport, Qld, 4217, Australia, email: j.thornton@griffith.edu.au

[2] School of Computing and Information Technology, Griffith University, Nathan, Qld, 4111, Australia, email: a.sattar@griffith.edu.au

1 Introduction

This paper considers a real-world rostering problem at the Gold Coast Hospital, Queensland. Nurse rostering is a *constraint satisfaction problem* (CSP) [3]. The task is to find a consistent allocation of shift values, for a group of nurses, over a fixed period of time, that satisfy a set of rostering constraints. These constraints include i) acceptable shift combinations (or *schedules*) for individual nurses, and ii) acceptable overall staffing levels for each shift. Previous work in the area has looked at *optimally* solving small or simplified problems [5] or using *non-optimal heuristics* to solve larger, more realistic problems [1]. The current research uses an optimising integer programming (IP) approach to solve a large rostering problem within reasonable time and computing resource bounds. The paper shows how the nurse rostering problem can be simplified by combining different modelling and decomposition techniques. Finally, the research demonstrates an IP model can capture "difficult" constraints *and* remain flexible in changing circumstances.

2 The Mathematical Model

There are two basic approaches to modelling nurse rostering: i) as a series of nurse/shift slots that have to be filled with appropriate shift values (the *nurse/shift* approach) [1] or ii) as a collection of possible schedules from which one has to be chosen for each nurse (the *possible schedules* approach) [5]. The possible schedules approach has the advantage that the many complex and subtle constraints that are involved in schedule construction can be implemented in a separate schedule generation program. The nurse/shift approach has the advantage that only a limited number of variables need be created in the model. By combining both modelling techniques the size of a rostering problem can be effectively reduced. This results in a faster execution time and a reduced memory requirement. A basic constraint in the model is of the form:

$$b_{\min(l)} \leq \sum_{i=1}^{n} \sum_{j=1}^{J_i} s_{il} a_{ij} X_{ij} + \sum_{k=n+1}^{m} s_{kl} y_k \leq b_{\max(l)}$$

where m = total nurses, n = total nurses with possible schedules, J_i = total number of possible schedules for nurse i, q = (number of days in roster) × (number of shift types), $\boldsymbol{a}_{ij}$ = 0/1 vector of size q representing which shift types are being worked on each day in possible schedule j for nurse i, l = the nurse seniority level over which the constraint applies, $\boldsymbol{b}_{min(l)}$ = a vector of size q representing the minimum number of staff of level l that should be on duty for each shift in the roster ($\boldsymbol{b}_{max(l)}$ represents the maximum staff required), s_{il} = 0/1 vector of size q representing whether nurse i works at level l for each shift of the roster, X_{ij} = 0/1 decision variable which equals one if schedule j is worked by nurse i, zero otherwise and $\boldsymbol{y}_k$ = a vector of q 0/1 decision variables representing which type of shift nurse k is to work on a particular day. Other constraints are defined to ensure that each nurse works only one schedule and that nurses using the nurse/shift approach work acceptable schedules. There are various options for constructing objective functions within the model: a two-phase approach can be used to first minimise the deviations away from desired constraint levels (requiring further $\boldsymbol{b}_{desired(l)}$ type constraints) and then minimising nurse *dissatisfaction* with schedules (based on attaching a "hate point" score to each possible schedule [2]). The problem can also be *decomposed* by considering only night and day shifts in the initial solution. This solution is then used in a second IP model to allocate the different types of day shift (in this case an early and a late shift [4]). Finally, for modifying an existing roster, an objective function can be defined which minimises changes to the existing solution.

3 Results and Conclusions

The IP model was used to solve 52 roster problems taken from the Gold Coast Hospital archives. All problems were successfully solved in an average time of 43 minutes on a 486 PC. Using quantitative measures of roster quality, the IP solutions were found to be significantly better than those produced by human experts [4]. The study concludes that the IP model presented is an effective way to solve realistic nurse rostering problems.

References

1. K. P. Chow and C. K. Hui. Knowledge-based system for rostering. *Expert Systems with Applications*, 6:361-375, 1993.
2. M. Kostreva and K. Jennings. Nurse scheduling on a microcomputer. *Computers and Operations Research*, 18(8):731-739, 1991.
3. A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99-118, 1977.
4. J. R. Thornton. *An Enhanced Cyclic Descent Algorithm for Nurse Rostering*. Honours Thesis, Faculty of Engineering and Applied Science, Griffith University Gold Coast, Australia, October 1995.
5. D. M. Warner. Scheduling nursing personnel according to nursing preference: A mathematical programming approach. *Operations Research*, 24(5):842-856, 1976.