Case-based reasoning in employee rostering: learning repair strategies from domain experts

Sanja Petrovic, Gareth Beddoe, and Greet Vanden Berghe

{ sxp | grb }@cs.nott.ac.uk
greetvb@kahosl.be

Abstract. The inherent difficulties in eliciting domain knowledge from experts are often encountered when applying artificial intelligence techniques to real-world problems characterised by multiple conflicting constraints. Definitions of optimal solutions are often subjective and highly dependent on the opinions and work practices of individual experts. We developed a case-based reasoning approach to capture concepts of optimality through the storage, reuse, and adaptation of previous repairs of constraint violations. The technique is applied to the problem of rostering nurses at the *Queens Medical Centre, Nottingham.* An iterative roster repair system is presented that learns repair techniques from nurses with rostering experience.

Keywords: artificial intelligence; decision support; personnel scheduling; case-base reasoning

1 Introduction

Automated problem solving in complex real-world domains requires the abstraction of rules and objectives from the work-practices and experience of experts. The acquisition of knowledge in the form of "IF THEN" rules is difficult and tedious and can lead to the development of inflexible and incomplete domain models[16]. One such domain is that of the real-world scheduling problems, in particular the rostering problems defined by Wren [23] as 'the placing, subject to constraints, of resources into slots in a pattern'. These assignment problems are subject to numerous conflicting constraints and are difficult to solve. The practices of manual rostering experts are difficult to represent systematically. In fact these practices can vary considerably from person to person with respect to overall rostering objectives and management of unavoidable constraint violations and personnel replacement. We aim to develop a system to capture the individual style of rostering experts by means of non-explicit representation.

Previous work on the personnel rostering problem has been carried out using a variety of different techniques. The particular problems investigated and levels of generality vary from author to author but a trend in the approaches used can be observed. Over the years there has been a distinct shift in the aims and objectives of automated rostering methods. Initittial attempts focused on finding optimal solutions using a variety of mathematical techniques including linear and integer programming [2, 14, 20]. A number of different optimisation criteria were defined that reflected both staff or management viewpoints. To solve these problems multi-criteria goal progamming methods [3] were developed. All of these mathematical approaches required problems to be simplified to a large degree in order to be computationally tractable. They consequently lacked the flexibility necessary to solve most real-world problems.

More recently, the recognition that most rostering problems are NP-hard has led researchers to reject traditional optimising methods in favour of methods that can deliver good solutions in a reasonable time. Constraint programming techniques have been successfully used to solve the problem in [1, 8, 13, 12]. Meta-heuristic methods [6, 5, 4, 10, 9] have been employed to search solution spaces defined by evaluation functions developed using extensive domain knowledge and experimental trial and error. The methods developed have been very successful at solving very generalised problems but still require the formal definition of evaluation criteria and search-neighbourhoods. It is an aim of this research to move away from explicit representations of these concepts.

Case-based reasoning (CBR) [11] is an artificial intelligence methodology that has been used successfully in a large number of application domains. CBR systems attempt to imitate human-style decision making by reasoning about new situations using past experience of similar situations. In a problem solving setting we can apply CBR methodology under the premise that 'similar problems will require similar solutions'. Previous problems and solutions are stored in a *case-base* and accessed during reasoning using identification, retrieval, adaptation and storage phases. The identification and retrieval phases search the case-base for cases containing problems that are the most similar to the current problem using a set of descriptive features. The solutions to these similar problems are then adapted to the the current situation.. If the new solution might be useful for solving future problems then it is stored as a new case in the case-base. Key to the success of any CBR system are the definitions of similarity, used to retrieve problems from the case-base, and the techniques used to adapt retrieved solutions into the context of new problems.

Previous research using CBR for solving nurse rostering problems involves combining CBR with constraint logic programming (CLP) techniques [18]. This approach stores sets of efficient shift patterns to generate an initial, infeasible, roster which is then repaired using CLP. In our research we use CBR to capture scheduling knowledge implicit in the behaviour of senior nurses by storing examples of previous repairs. A similar approach is used to tackle job-shop scheduling problems in [15, 16] with the development of the interactive scheduler CABINS. CBR has also been used for the educational time-tabling problem [7] and for general production scheduling [17].

This paper describes the research carried out into a CBR based nurse rostering system. It describes a new approach to the repair of constraint violations within a roster. CBR methodology is used as a framework within which previous violations and corresponding repairs are stored in a casebase of rostering experience. This information is then adapted and reused to repair violations in new rosters. A particular nurse rostering problem from a UK hospital, which drove the development of our approach, is described in Section 2. Section 3 describes the new methods for storage and retrieval of constraint and repair information. Some empirical results and analyses are given in Section 4 and we conclude with remarks about the future direction of the research.

2 Problem Description

We investigated the problem of rostering nurses at the Ophthalmology ward of the Queens Medical Centre University Hospital NHS[1] Trust in Nottingham, UK (refered to herein as the QMC). This ward illustrates a nurse rostering problem that the approach described here attempts to solve. However, our intent is to investigate the applicability of this research to problems in other wards and institutions.

The ward consists of between 30 and 35 nurses and cover is required on a 24 hour basis. By NHS standards it is considered to be a medium sized ward with high demand predictability. The nurses are divided into between 4 and 6 teams on a monthly basis. These teams cover different areas of the ward although there is considerable overlap in their responsibilities. They are also used in the first step of the rostering process described in Section 2.2.

Nurses have one of four qualification levels: registered, enrolled, auxiliary, and student. Registered and enrolled nurses have formal nursing qualifications. Registered nurses are the most senior and generally hold a supervisory role. Auxiliary nurses and student nurses are untrained or in the process of receiving further training. These various nurse types can be classified hierarchically for descriptive purposes (Figure 1) although this does not necessarily give an indication of replacement suitability.

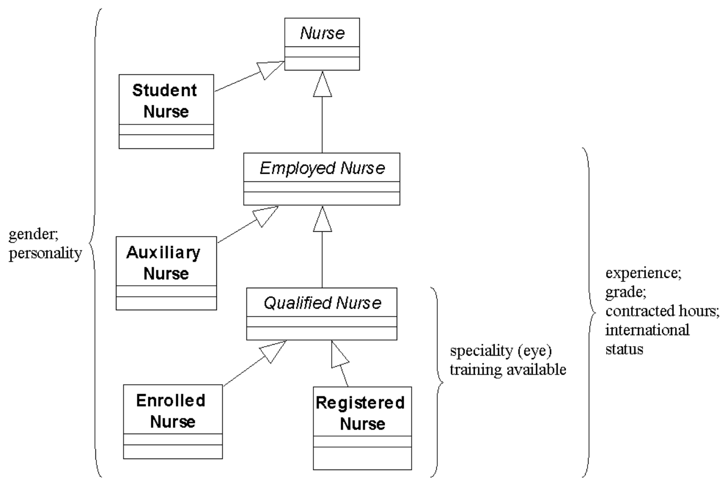

Figure 1: Classification of nurses

In addition to the basic qualifications, specialised training specific to the ward (here called *eye training*) is completed by certain registered and enrolled nurses. A nurse with eye training can take on a wider range of responsibilities within the ward, including the supervision of non-eye trained staff. Nurses are further classified by a grading system, common to NHS hospitals, determined by a number of different factors. In addition to qualifications, training, and

[1]National Health Service

nurse grade, characteristics such as gender, nationality, contracted hours, and even personality are considered when making staffing decisions.

We shall define the set NS as the set of all T nurses to be rostered. A set of descriptors ND_i is assigned to $nurse_i$ ($1 \leq i \leq T$) containing information about qualifications, experience, eye-training and other characteristic features.

For each day in the rostering period a nurse is assigned one of a number of different shift types. Four basic types (early 'E' , late 'L', night 'N', and unassigned 'U') are used for the majority of assignments. A considerable number of other disjoint shift types are in practice used to accommodate part-time nurses and nurses with contract-stipulated working hours. However, for this paper we shall only consider the four basic types.

Shift assignments over a planning period of length p days are represented by a set of decision variables for each nurse (called *nurse rosters*):

$$\forall nurse_i \in NS \quad \exists \quad NR_i = \{s_{ij} \in \{E, L, N, U\} : 1 \leq j \leq p\} \quad (1 \leq i \leq T)$$

We can then define a roster as being the set of nurse rosters over the planning period p:

$$R_p = \{NR_i : 1 \leq i \leq T\}$$

The feasibility of a roster will be determined by its satisfaction of certain constraints as defined in the following section.

2.1 Constraints

It is common throughout the literature to divide constraints in scheduling problems into sets of *hard* and *soft* constraints [3, 6, 13]. Hard constraints represent legal and management requirements and allow the definition of the feasibility of a roster. Where a particular characteristic of a roster is desirable but not essential it is described as a soft constraint. These are used in some meta-heuristic techniques as evalutation criteria by which to describe the equality of a roster [5, 10]. The classification of constraints in this way is very useful for modelling purposes although in practice the demarkation between the two sets varies considerably and is often fuzzy. Constraints that are described as hard are in practice sometimes ignored in extreme circumstances.

A number of different hard constraints have been identified at the QMC ward. These can be broadly categorised as:

Cover constraints define the skill mix required for a particular shift;

Working Hours constraints describe the maximum working hours allowed over a particular period. These can be defined for all nurses of a specific type as well as for individual nurses;

Rest Period constraints ensure that there is sufficient gap between shifts of various types.

Cover constraints are the most numerous and the most difficult to satisfy during the repair stage. They are described using combinations of the qualifications, training, grade, and descriptions of the required nurses. These are different for each shift type and depend on predicted patient levels over the planning period. Figure 2 shows a typical skill mix requirement. For simplicity the required skill mix is decomposed into a number of cover constraints. The example given represents four constraints (4 qualified, 2 eye-trained, 1 registered, and 1 auxiliary).

- 4 Qualified Nurses such that
    - at least two are eye-trained
    - at least one is a registered nurse
- 1 Auxiliary Nurse

Figure 2: Example skill mix requirement

There are many soft constraints that have been described by the nurses at the QMC. The most important of these are the preferences of each of the nurses. Other soft constraints include the specification of 'good' shift patterns, rules governing weekend allocation, and fairness of allocation over the ward. It is intended that these soft constraints are not explicitly stated but are instead implicit in the rostering behaviour captured by the developed technique. Hence, in this paper, they will not be strictly defined as constraints in the specification of the problem or the for the identification of violations to be repaired.

2.2 Self Rostering

Roster production in the QMC ward is a three stage process involving all nurses. The *self rostering* planning approach is used to give employees greater involvement in the rostering process. A comprehensive survey of the use of this approach in NHS hospitals can be found in [19]. This approach recognises that nurses are professionals who will fulfill their responsibilities without excessive administrative intervention. It also provides an efficient means by which staff can indicate preference information.

The three stages are:

1. Nurses are assigned to teams (according to a particular skill mix).
2. Nurses produce partial rosters (called *preference rosters*) for the planning period in consultation with other members of their teams.
3. Partial rosters are combined to produce the ward roster which is probably infeasible. Any constraint violations are repaired by senior staff members.

Preference rosters represent individual nurse's requests to work particular shifts on particular days. If they have no preference on a particular day then they can leave it blank. Preference rosters vary considerably between nurses with regards to the amount of detail included and individual flexibility.

The third stage is the most time consuming in the process and burdens a senior nurse with several hours of extra work every month. The constraint violations present in the roster must be repaired whilst maintaining as much information from the preference rosters as possible. In some extreme circumstances, when preference information has been severely damaged, certain constraint violations can be ignored (at the discretion of the senior nurse). It is knowledge and experience about how these constraint violations are repaired that we wish to capture using the technique described here.

3 Learning Repair Strategies

In this paper we investigate the automation of the third stage of the rostering process and develop a framework for capturing and re-using the experience of experts in rostering. The focus is specifically on imitating expert's repairs of infeasible rosters. We make the initial assumption that repairs of constraint violations are independent of one another. Longer term repair strategies (consisting of more than one individual repair) are not considered - instead a direct correspondence is drawn between a constraint violation and a suitable repair.

CBR methodology is an ideal basis for the framework as it removes the need to generate explicit rules describing this violation/repair correspondence. Such rules are difficult to establish in complex domains such as this [22]. Case-based techniques have been used successfully in the past to learn 'styles' of decision making from experts [11, 16]. This characteristic makes CBR well suited to the nurse rostering problem, where the subjective decisions of experts determine the nature of the final results. Consequently, the methods of quantitative roster evaluation and the criteria of optimality need not be defined here.

The hypothesis that 'similar constraint violations within similar rosters, require similar repairs' extends the standard premise of CBR methodology. The system stores information about how a constraint violation within a roster was repaired. When a similar problem is encountered in the future this information is extracted and used to generate a similar repair. It is clear that the definition of similarity in the various contexts is key to the success of the technique.

3.1 Framework

A set of hard constraints, C, is provided along with the preference roster PR containing information from stage 2 of the self rostering process. The set of constraints are then used to determine a set of violations, V, of the current roster R (Note that before the first repair has been applied $R = PR$). When a violation is chosen for repair (either randomly or from a list) a focus case is built and similar cases identified from the case base. Repairs are adapted from these cases by finding the most similar nurses given the existing shift assignment information in R and PR. The repair chosen by the expert is then applied to the roster and the focus case, including the repair, is stored in the case base. Figure 3 gives a flowchart of a single iteration of the method.

The case base, CB, is a set of cases each with the structure given in Figure 4. These cases consist of four different sets of data. *Violation* contains the information about the type and extent of the violation (e.g. the shortfall in nurses or amount of overtime). *Local* and *Global Features* are statistical characteristics of R in the region of the violation and over the whole roster respectively. These three sets form the indices of the case and are used during retrieval to determine similarity between the problems in the case-base and the focus case. The fourth

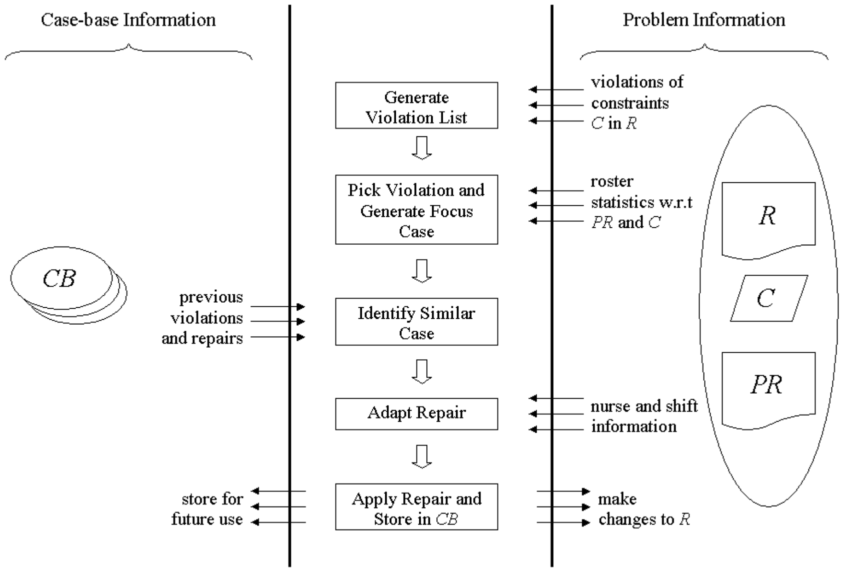

Figure 3: A single iteration of the method

set, *Repair*, describes the repair that was used including statistical information about the nurses and shifts involved.

The global features describe the state of the roster when the violation was repaired. The following statistics are used:

- total number of violations
- percentage of shift preferences that are satisfied
- percentage of available hours already assigned
  - over the whole planning period
  - over the week within which the violation occurs

Total number of violations gives a measure of the level of infeasibility of the roster at the time of the repair. Similarly the *percentage of shift preferences satisfied* indicates how 'bad' the roster is in terms of staff satisfaction. It must be emphasised that these two indices should not be regarded as evaluation criteria - we are not trying to optimise anything here in the numerical sense. These measures should only be interpreted as characteristics of the roster. The final two indices give the flexibility available for making rostering decisions given the current state of R.

Local features describe the roster in the region of the violation. This information is similar to the global feature information but is restricted to the type of nurse that is involved in the hard constraint violation. They consist of:

- specific nurse type index
- percentage of nurse specific shift preferences that are satisfied

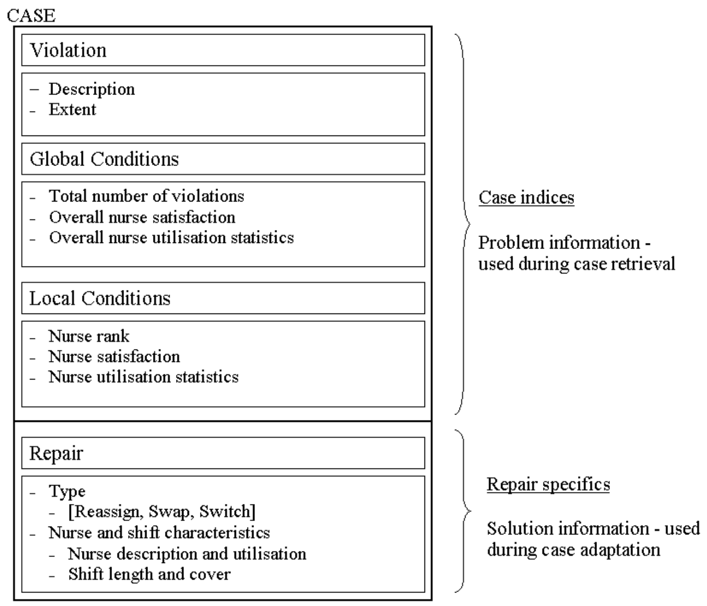

Figure 4: Structure of an individual case

- percentage of nurse specific available hours already assigned
    - over the whole planning period
    - over the week within which the violation occurs

The *nurse type index* is a value given to type of nurse specified in the violated constraint based on seniority, grade, and skill level.

In order to retrieve cases from the case-base we must define what is meant by the word 'similar'. A similarity measure is a function that takes a pair of cases and returns a value indicating how similar they are. To define this we represent a case as an ordered pair consisting of the violation description v_γ, and a set F_γ of I feature indices:

$$case_\gamma = \langle v_\gamma, F_\gamma \rangle, \text{ where } F_\gamma = \{f_{\gamma_i} : 1 \leq i \leq I\}$$

Retrieval of cases from the case base is a two step process. An initial search is performed to identify those cases that describe a compatible violation type. This is necessary largely because the objects (nurses or shifts) involved in constraints differ. A cover constraint, for example, considers the nurses in NS who are assigned a particular shift on a particular day - it is calculated from the perspective of this shift. Legal working hours constraints, however, are approached from the perspective of individual nurses. They are calculated by considering the values $s_{i,j} \in NR_i$ for a particular $nurse_i$ ($1 \leq i \leq T$). Nothing will be gained by trying to apply a repair used originally to decrease a nurse's hours when trying to fix a shift with insufficient cover. Removing incompatible cases from the search ensures that a case describing the repair of the violation of one constraint type will not be adapted to repair a violation of a different type.

The second part of the retrieval process identifies similar cases from the remaining set. A similarity measure is applied to the feature indices of the case which include the violation extent, and the local and global features of R.

The similarity between two cases with respect to their feature indices is calculated using a *nearest neighbour* similarity measure [21] as follows:

$$S(F_a, F_b) = \frac{1}{\frac{1}{I}\sum_{i=1}^{I} distance(f_{a_i}, f_{b_i})}$$

If the denominator happens to be zero then this is detected during calculation and the similarity set at ∞. Here *distance* is a function that normalises the difference between two feature values using the maximum and minimum values of the feature over all cases in the casebase.

$$distance(f_{a_i}, f_{b_i}) = \left| \frac{f_{b_i} - f_{a_i}}{f_{max} - f_{min}} \right|$$

Then the most similar case, $case_{nearest} = \langle v_{nearest}, F_{nearest} \rangle$, to a focus case, $case_{foc} = \langle v_{foc}, F_{foc} \rangle$, is defined:

$$case_{nearest} = case_{ret} \in CB \text{ such that } v_{ret} \equiv v_{foc} \text{ and}$$
$$S(F_{ret}, F_{foc}) = \max\Big[\{S(F_\gamma, F_{foc}) : case_\gamma \in CB\}\Big]$$

Here the equivalence relation is used to denote compatibility between violations. We use the k-nearest neighbour approach (with $k = 3$) to allow different possible repairs to be presented to the expert. The best of these 3 is selected by the expert and then applied to R and stored in the case base. If none of the repairs are acceptable then the expert can manually specify an alternative.

3.2 Adaptation

The identification of a similar case from the case base provides information about how a similar violation, in a similar roster, was repaired. This information then needs to be adapted to generate a repair of the violation being considered in the current roster. It consists of the type of repair operation that was used and feature information and statistics about the types of nurses and shifts involved.

Three basic repair operations have been identified for nurse rosters (see Figure 5). The simplest of these is *Reassign* which involves changing the assignment of a single nurse on a particular day. *Switch* interchanges the assignments of two shifts for a nurse on two different days, whereas *Swap* does the same for two nurses on a single day. These repair operations do not given any indication of the nurses and shifts involved but they do dictate exactly which information is stored with the repair operation type in the case.

Some information about a repair is determined immediately by the constraint violation. For example, a cover violation is a shortage of nurses for a particular shift on a particular day. So the shift and day are already set and must be part of any repair performed. However, a total hours violation is an excess of shifts assigned to an individual nurse and it is this nurse who must somehow be involved in the repair. To complete the adaptation of the repair

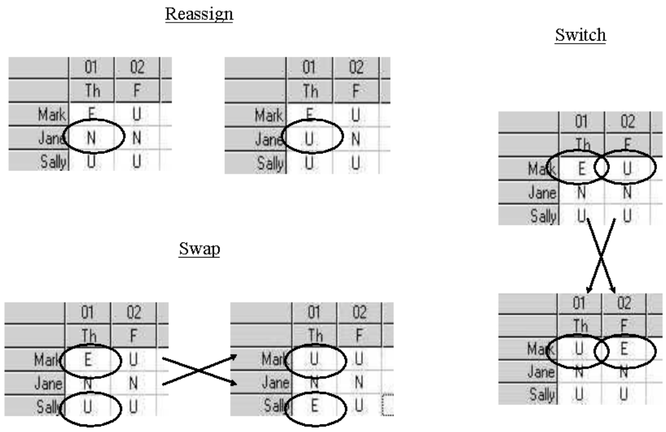

Figure 5: Basic repair operation types

from the retrieved case we need to generate a new repair, of the type specified (ie. reassign, switch, or swap), by incorporating the predetermined information from the violations with additional nurses and/or shifts to fill the remaining roles. We fill these by choosing the most similar nurses and shifts in the current problem to those that were used in the retrieved repair. Each nurse required to fill a particular role in the repair is considered separately.

The method used to identify a similar nurse is similar to that used during case retrieval. The first step is to restrict the search to only those nurses in NS who are of the same type to the nurse used in the repair. Statistics about a nurse's assigned shifts before and after the repair are then used to identify the nurse with the most similar shift assignments.

The following statistical information is stored in the a case for each nurse involved in the repair:

- percentage of contracted time used
- cover of the shift assigned before the repair
  - for all nurses
  - for type specific nurses
- cover of the shift after the repair
  - for all nurses
  - for type specific nurses
- length of the shift assigned before the repair
- length of the new shift

Using these indices and the nurse descriptor sets we can represent a nurse as the ordered pair:

$$nurse_\gamma = \langle ND_\gamma, F_\gamma \rangle, \text{ where } F_\gamma = \{f_{\gamma_i} : 1 \leq i \leq I\}$$

Using the functions introduced in section 3.1 we define the most similar nurse $nurse_{nearest}$ to the description of a nurse from a case $nurse_{case}$ as:

$$nurse_{nearest} = nurse_\alpha \in N \text{ such that } ND_\alpha \equiv ND_{case} \text{ and}$$
$$S(F_\alpha, F_{case}) = \max\Big[\{S(F_\gamma, F_{case}) : nurse_\gamma \in N\}\Big]$$

Here the equivalence relation denotes compatibility between nurse descriptions.

It must be emphasised that the adaptation technique described here does not aim to exactly replicate repairs from cases in the case-base. This would not only be impossible in most situations but also incorrect. The exact individuals involved in a previous repair may not be the most suitable to choose in the context of the current roster. Indeed, it would not be possible if the nurse in question was not in the set of those to be rostered. The nurses chosen using this technique are the most similar to those originally used with respect to their descriptive features and their current utilisation levels.

4 Implementation and Evaluation

The method has been implemented using an object-oriented approach in Microsoft Visual C++. A graphical environment has been developed to facilitate interaction with the user. This is of particular benefit when a selection of repairs are suggested. The system is flexible in that any number of nurses of varying descriptions can be specified. Furthermore, the main hard-constraint types have been generalised so that the user can specify any number of different constraints.

For the purpose of evaluation of the method, an instance of a real world problem is adapted. Sample rosters and staff details from the QMC ward are used. All nurse information and initial preference rosters are preserved but the number of constraints is reduced so that only those implemented at the current stage of development are considered. The complexity of the problems investigated will increase significantly in later versions.

The problem consists of rostering 19 registered and enrolled nurses with varying levels of training, experience, and contracted hours. Preference rosters, produced during the second stage of the self-rostering process, are entered over two 4-week rostering periods (representing two planning periods at the QMC ward). In total 9 constraints are specified and these define the required cover for each of the three shift types and restrict the maximum number of working hours in a fortnight to 75.

Constraint violations are selected at random and repairs are suggested by the software for each. Using the original rosters produced by the senior nurse the suggested repair is evaluated. The verdicts are recorded for generated repairs as :

Exact Match: The repair generate is identical to the expert's repair;

Equivalent Match: The repair generated involves nurses and shifts of the same types as those used in the expert's repair;

Fail: The repair generated is not an exact or equivalent match, or no repair was generated.

In each run a total 100 constraint violations are repaired over the two 4-week periods. The case-base is initially empty and for each violation addressed the repair from the expert's roster is stored. During the first few iterations repairs can not be generated (and therefore fail verdicts are recorded). The case-base is usually ready for most repairs after about five cases are stored.

The results here are averages over five 100-repair runs (with the case-base reset for each run). Figure 6 shows the average cumulative number of exact matched repairs, and of equivalent or exact matched repairs, as the case-base size increases from 0 to 100. Figure 7 gives the percentage of guesses of each of the three types averaged over 25-repair intervals.

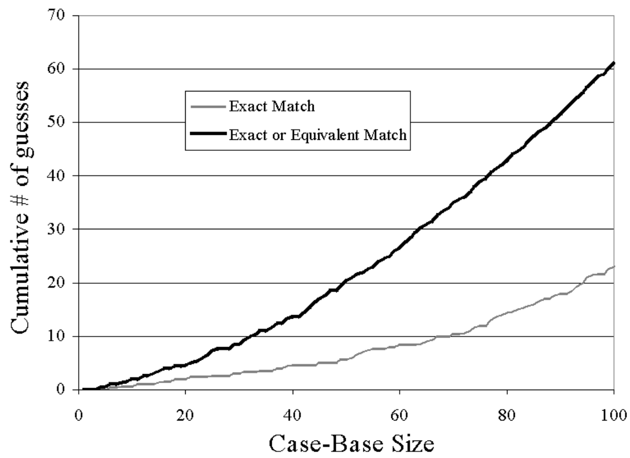

Figure 6: Cumulative guesses against case-base size

A steady increase in the gradient of the curves in Figure 6 shows that the quality of the suggested repairs is increasing as more experience is stored in the case-base. Note that the maximum possible gradient of these curves is 1. Although the average numbers of matched and equivalent repairs may not seem very high over the whole run, a more accurate evaluation can be made by considering the final 25 repairs in each run. These are repairs generated using a case-base containing between 75 and 100 cases. This relatively well trained (experienced) cases-base produces 41% exact matches and 93% equivalent or exact matches (Figure 7) and only 7% fail.

These experiments show that the system learns incrementally, and improves its performance throughout the use of the system. Generated repairs become more refined as the case-base size increases and the method produces a large number of 'good' (exact or equivalent) repairs for a reasonable case-base size. A number of other experiments on different problems from the QMC ward have produced similar results.

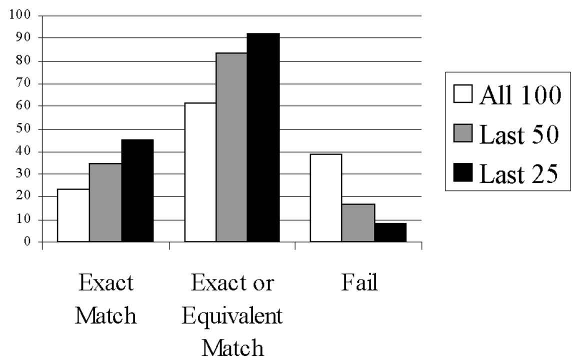

Figure 7: Percentage of repairs by guess verdict

5 Conclusion

This paper describes a new approach to the nurse rostering problem. The results of this initial research prove that the knowledge and experience of nurse rostering experts can be successfully captured in a case-base. The particular style and objectives of an individual expert are represented by the cases in the case-base.

The results produced so far, particularly with regard to exact matches, could be improved by implementing a more sophisticated case structure. The nature of the interactions between the variables used as indices, for both case retrieval and adaptation, needs to be more fully understood. This includes determining how the relative importance of the roster features affects the calculation of the similarities between cases. A system of adaptive weights for the case indices is being investigated. The selection of nurses for repairs could also be improved by incorporating information about surrounding shift patterns.

The application of this method within an automated rostering system is the subject of continuing research. A search technique, guided by the experience stored in the case-base, is proposed and a number of different meta-heuristic approaches are being considered. Instead of searching through a solution space by improving an objective or fitness function, we use the information memorised in the case-base. The implications of this present a number of fascinating challenges for future research in the subject.

References

[1] S Abdennadher and H Schlenker. INTERDIP – an interactive constraint based nurse scheduler. In *Proceedings of The First International Conference and Exhibition on The Practical Application of Constraint Technologies and Logic Programming, PACLP*, 1999.

[2] J Bailey and J Field. Personnel scheduling with flexshift models. *Journal of Operations Management*, 5:327–338, 1985.

[3] I Berrada, J A Ferland, and P Michelon. A multi-objective approach to nurse scheduling with both hard and soft constraints. *Socio-Economic Planning Sciences*, 30/3:183–193, 1996.

[4] E K Burke, P De Causmaecker, and G Vanden Berghe. A hybrid tabu search algorithm for the nurse rostering problem. In *Selected Papers from the 2nd Asia Pacific Conference on Simulated Evolution and Learning Volume*, volume 1585 of *LNAI*, pages 187–194. Springer Verlag, 1998.

[5] E K Burke, P De Causmaecker, S Petrovic, and G Vanden Berghe. Fitness evaluation for nurse scheduling problems. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Seoul, Korea*, pages 1139–1146.

[6] E K Burke, P I Cowling, P De Causmaecker, and G Vanden Berghe. A memetic approach to the nurse rostering problem. *Applied Intelligence*, pages 199–214, 2001.

[7] E K Burke, B MacCarthy, S Petrovic, and R Qu. Structured cases in case-based reasoning - re-using and adapting cases for time-tabling problems. *Knowledge-Based Systems*, 13:159–165, 2000.

[8] B M W Cheng, J H M Lee, and J C K Wu. A constriant-based nurse rostering system using a redundant modeling approach. Technical report, Department of Computer Science and Engineering at The Chinese University of Hong Kong, 1996.

[9] K A Dowsland and J M Thompson. Solving a nurse scheduling problem with knapsacks, networks and tabu search. *Journal of the Operational Research Society*, 51:825–833, 2000.

[10] Kathryn Dowsland. Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, 106:393–407, 1998.

[11] J L Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc., 1993.

[12] A Meisels, E Gudes, and G Solotorevski. Employee timetabling, constraint networks and knowledge-based rules: A mixed approach. In *Practice and Theory of Automated Timetabling, First International Conference*, pages 93–105. Springer, 1995.

[13] H Meyer auf'm Hofe. Solving rostering tasks as constraint optimization. In *Selected Papers from the 3rd international conference on Practice and Theory of Automated Timetabling (PATAT)*, Springer-Verlag Lecture Notes on Computer Science, pages 280–297. 2000.

[14] M L Miller, W Pierskalla, and G Rath. Nurse scheduling using mathematical programming. *Operations Research*, 24:857–870, 1976.

[15] K Miyashita, K Sycara, and R Mizoguchi. Modeling ill-structured optimization tasks through cases. *Decision Support Systems*, 17(4):345–364, 1996.

[16] K Miyashita, K Sycaras, and R Mizoguchi. Capturing scheduling knowledge from repair experiences. *International Journal of Human-Computer Studies*, 41:751–773, 1994.

[17] G Schmidt. Case-based reasoning for production scheduling. *International Journal of Production Economics*, 56-57:537–546, 1998.

[18] S Scott and R Simpson. Case-bases incorporating scheduling constraint dimensions - experiences in nurse rostering. In *Advances in Case-Based Reasoning - EWCBR98*, Lecture Notes in Artificial Intelligence. Springer Verlag.

[19] R Silvestro and C Silvestro. An evaluation of nurse rostering practices in the National Health Service. *Journal of Advanced Nursing*, 32(3):525–535, 2000.

[20] M Warner. Scheduling nursing personnel according to nurse preference: A mathematical programming approach. *Operations Research*, 24:842–856, 1976.

[21] I Watson. Case-based reasoning is a methodology not a technology. *Knowledge-Based Systems*, 12:303–308, 1999.

[22] I Watson and F Marir. Case-based reasoning: A review. *The Knowledge Engineering Review*, 9/4:327–354, 1994.

[23] A Wren. Scheduling, timetabling and rostering - a special relationship? In E Burke and P Ross, editors, *Practice and Theory of Automated Timetabling*, volume LNCS 1153 of *Lecture Notes in Computer Science*, pages 46–75. Springer Verlag, 1996.