Comparison of Algorithms for Nurse Rostering Problems

Sanja Petrovic[1*], Greet Vanden Berghe[2,3]

[1]*School of Computer Science and Information Technology*
University of Nottingham
Jubilee Campus, Wollaton Road
Nottingham NG8 1BB, UK

[2]*KaHo Sint-Lieven, Gebr. Desmetstraat 1, B-9000 Gent, Belgium*

[3]*K.U.Leuven, Department of Computer Science,*
E. Sabbelaan 53, 8500 Kortrijk, Belgium

Abstract
Despite decades of research into automated methods for nurse rostering and some academic successes, one may notice that there is no consistency in the knowledge that has been built up over the years and that many healthcare institutions still resort to manual practices. One of the possible reasons for this gap between the nurse rostering theory and practice is that often the academic community focuses on the development of new techniques rather than developing systems for healthcare institutions. In addition, methods suitable for one problem are usually not easily transferable to other problems. There is a lack of criteria for the comparison of approaches to provide a clear picture about their advantages and disadvantages and therefore their suitability to a problem in hand. This paper introduces seven criteria: expressive power, flexibility, algorithmic power, learning capabilities, maintenance, rescheduling capabilities, and parameter tuning. Two approaches to nurse rostering, which are of very different nature, are evaluated and compared against the introduced criteria. One approach is based on meta-heuristics, while the other employs case-based reasoning.

Key words: nurse rostering, comparison, employee timetabling, meta-heuristic, tabu search, case-based reasoning

*Contact author:

tel +44 (0)115 951 4222

fax: +44 (0)115 846 7813

e-mail sxp@cs.nott.ac.uk

1 Introduction

Nurse rostering has become a very attractive research area within the field of management science/operational research and artificial intelligence, especially since the 1990's. Personnel rostering is defined as the problem of placing resources into slots in a pattern, where the pattern denotes a set of legal shifts defined in terms of work to be done subject to given constraints (Wren, 1996). Healthcare institutions recognize that good rosters add to the quality of care and to the (mental) health and social well-being of large numbers of health workers. In constructing a roster, various legal, management, and staff requirements must be considered. They are often conflicting in their nature. For example, requirements for the cover, i.e. required mix of staff qualifications for a particular shift are often in conflict with the maximum working hours that are allowed for nurses, and may also be in conflict with individual staff preferences for that shift. In addition, hospital wards often have to deal with a lack of personnel, which makes the nurse rostering problem even more difficult. Being complex and highly constrained, nurse rostering problems have been the subject of interest within both the artificial intelligence and operational research communities. Consequently, a large number of papers have appeared presenting different approaches that often address one particular problem or a set of similar problem instances.

Classifying approaches to nurse rostering can be performed in different ways. Burke et al. (2004), offer a classification based upon the model that was developed. Some models were simple, such as assigning morning, late or night shifts to a group of equally skilled full time nurses over a limited period (Bellanti et al., 2004), while some were much more complex and addressed many requirements concerning shifts, work regulations, part time work, skill categories, legal constraints, personal requirements, etc (Meyer auf'm Hofe, 2001). Nurse rostering problems are often stated as optimisation problems. Many different objective functions have been defined that depended on the country, region, or healthcare institution, and even on the ward. More common objectives for nurse rostering include one single or a combination of the following objectives: minimising the number of constraint violations, minimising the number of nurses, minimising overtime, maximising the coverage, maximising satisfaction of personal preferences, etc. However, we may observe that the developed models have not evolved towards a more complete coverage of complex real-world situations. On the contrary, each new paper seems to introduce a particular set of constraints and a particular objective function and as such makes every comparison with previous research work almost impossible.

Another possible way of classifying is to take the method that was employed into consideration. Traditional operational research optimisation methods are still being employed to solve nurse rostering problems. The most recent publications include goal programming (Azaiez, Al-Sharif, 2005), column generation (Bard and Purnomo, 2005), a Lagrangian heuristic (Bard and Purnomo, 2007), etc. Meta-heuristic approaches have also been developed with a certain level of success. We refer to some of the most recent publications based on tabu search (Burke et al., 2006), scatter search (Maenhout and Vanhoucke, 2006), electromagnetism meta-heuristic (Maenhout and Vanhoucke, 2007), SAWing, Noising methods combined with Simulated Annealing (Parr and Thompson, 2007), ant algorithms (Gutjahr and Rauner, 2007), etc. Also, artificial intelligence techniques have been applied to nurse rostering, although to a less extent compared to meta-heuristics. Recent ones include case based reasoning (Beddoe et al., 2008), fuzzy logic (de la Asunción et al., 2007), multi-agent systems (Kaplansky and Meisels, 2007), etc.

Despite decades of research into automated methods for nurse rostering and some academic successes, many healthcare institutions still resort to manual practices. The research results obtained by the timetabling community still struggle to find their way to

implementations in healthcare institutions. Kellogg and Walczak, (2007) carried out a review on the application of academic results in healthcare environments. They discovered that, currently, only a small number of software systems that address the nurse rostering problem, rely on academic research results.

Although the problems that are tackled nowadays are larger and consider more requirements than previously published research work, one may notice that there is no consistency in the knowledge that has been built up over the years. The number of novel approaches (or novel to the field of nurse rostering) is very large indeed. Still, only a few comparisons have been carried out and they focus on the comparison of two or more approaches developed by the same authors. For a long time, there were no benchmark problem instances that would enable researchers to perform a more thorough comparison of the developed approaches for nurse rostering. Recently, an excellent Web page providing a collection of 13 employee timetabling problems, mostly nurse rostering ones, was generated (Burke et al, 2007, http://www.cs.nott.ac.uk/~tec/NRP/index.html). The problem instances have been derived from real-world problems and are available in XML format. The Web page also offers a parser and a set of solvers to tackle the problems through a nice graphical user interface. A track of the best results for each problem instance is kept. Another Web page (http://www.projectmanagement.ugent.be/nsp.php), named NSPLib, presents a complementary initiative in the sense that it offers a generator for nurse rostering problems. In contrast to the other benchmark site, the problems in NSPLib were not derived from real-world ones, but they were constructed automatically. A number of indicators were introduced which measure the complexity of problem instances. They were used to create a collection of problem instances that are as diverse as possible with respect to the complexity indicators. The NSPLib allows for proper statistical analysis of different approaches to nurse rostering.

The introduced benchmarks give researchers access to a variety of nurse rostering problems and enable them to compare the performance of their algorithms. Although, undoubtedly, the provided benchmark instances will enable advancements in the nurse rostering research, using them alone would not be enough to bridge the still existing gap between the nurse rostering theory and practice. One of the significant reasons for this is that, usually, methods suitable for one problem are not easily transferable to other problems. As pointed out, there are numerous differences in specifications of nurse rostering problems. The differences include the planning period, the number of possible shift types per day, the rules for replacement among different qualification categories, the variety of objective functions in use to measure the quality of solutions, etc. Usually, approaches work well only in environments that are very similar to the instance of the problem for which they were designed. Each new instance of the problem usually requires significant changes to the model, and to the algorithm itself (at least it requires tuning of parameters).

The comparison of methods for timetabling/scheduling, in general, and for rostering in particular, has traditionally been seen as a matter of their application to problems using the same objective function, while the required computational resources are eventually taken into consideration. However, this does not provide a full picture of the quality of a method and its suitability to other nurse rostering problems, different to the ones that the approach was developed for. The aim of this paper is to introduce criteria that will allow a thorough comparison of different approaches to nurse rostering. We will discuss these criteria using two developed approaches that are of very different nature:

1) ANROM (**A**dvanced **N**urse **Ro**stering **M**odel) together with a hybrid tabu search approach was developed for Belgian healthcare institutions (Burke et al., 2006, Vanden Berghe, 2002, Burke et al., 2001b, Burke et al., 2001a). In the rest of the paper, we will refer to the combination of model and algorithms by the same

name: ANROM. The developed approach was the core of a system that was in use for many years in over 40 Belgian hospitals. A new commercial software system has been recently developed that is based on ANROM with some adaptations made in order to meet today's requirements.

2) The Cabarost (**Ca**se-**ba**sed Reasoning **Rost**ering) approach that was developed for the complex nurse rostering problem faced by the Ophthalmology ward at the Queen's Medical Centre University Hospital NHS Trust (QMC) in Nottingham (Beddoe at al, 2008, Beddoe and Petrovic, 2007, Beddoe and Petrovic, 2006).

The first approach applies an optimisation method for constructing rosters and thus introduces an objective function to measure the quality of the generated rosters, while the second approach employs an artificial intelligence method in an attempt to mimic the way that personnel managers generate rosters.

The reminder of the paper is organised as follows. In Section 2, both nurse rostering problems are described together with the main components of the developed approaches. Criteria for comparing the approaches are discussed in Section 3. The developed approaches are evaluated against the introduced criteria. Some conclusions are given in Section 4.

2 Nurse rostering problems and approaches

The nurse rostering problem is characterized by the presence of a large set of constraints, which are usually divided into two categories: hard and soft constraints. Hard constraints are those that are rigidly enforced and should be met at all times. Violations of the soft constraints are to be avoided, if possible. Shifts have to be assigned to a set of nurses in such a way that at any time the coverage constraints imposed on personnel regarding the required skills are met. Assignments are at the same time subject to time related constraints imposed on individual nurse rosters. Each nurse is characterized by a skill category (or a set of skill categories) and a work regulation, which determines the percentage of her/his employment. Usually, problems with a large number of possible shift types, a large number of nurses and/or a long planning period, are more difficult to solve.

The list of constraints that can be associated with nurse rostering problems is very extensive. A subset that is considered by both approaches to be compared includes the following constraints:
- *Coverage*: defines the minimum number of nurses of particular qualifications that must be assigned to a particular shift type on a particular day. For example, the early shift requires 4 qualified nurses on a particular day.
- *Max (min) days on*: sets the maximum (minimum) number of days that nurses may (should) work in a row.
- *Max (min) hours*: sets the maximum (minimum) number of hours that nurses may (should) work over a period of time. For example, full time nurses may not work more than 75 hours in a fortnight;
- *Personal preference:* defines a request for a shift on/off (day on/off);
- *Single night*: the nurses prefer not to work a single night shift, but in blocks of two or more.
- *Succession*: defines illegal shift combinations for nurses. For example, it is not desirable to work a *Night* shift followed by an *Early* shift;
- *Weekend balance*: sets the number of weekends that nurses may work over a period. For example, nurses may not work more than 3 weekends out of 4, unless it is stipulated in their work regulations;

- *Weekends in a row*: sets the maximum number of weekends that nurses may work in a row.
- *Weekend split*: nurses prefer to work both days in a weekend, not a single day.

2.1 Meta-heuristic approach

2.1.1. Description of the problem

Belgian hospitals are characterised by a wide variety of constraints. Flexibility is a paramount issue and refers to both healthcare institutions and the individual nurses. A flexible institution needs to adapt its operations to the predicted daily or even hourly demands. Obviously, cyclic rosters cannot address these changing demands. The acyclic solutions to such nurse rostering problems require flexible work regulations that differ strongly from the well-known three-shift regime. On the other hand, nurses working in flexible environments are entitled to express their personal days off and holiday requirements. Even more, nurses nowadays have a say in determining the individual work regulations that match their private life best (e.g. part time 80%, never work on Wednesdays).

The coverage constraints are represented by intervals defining minimum and preferred coverage for each particular qualification. They are expressed either in terms of shift types or in terms of time intervals. The qualification requirements state that certain tasks/shifts can only be performed by nurses belonging to a particular skill category. Skills are associated with individual nurses. All the nurses have a primary skill, but they can also be assigned a list of secondary skills. It is not allowed to assign shifts to nurses that are not qualified at all for the particular shift type. This coverage constraint, together with the qualification requirements, constitutes the only hard constraints in the model. All the other constraints belong to the category of time related constraints and deal either with legal aspects, work regulations or personal preferences for one particular nurse. They are treated as soft constraints. Apart from the constraints described in the introduction to this section, additional constraints that are taken into consideration in ANROM include:
- *Maximum number of assignments in the planning period*
- *Minimum/maximum number of consecutive free days*
- *Maximum number of assignments per day of the week:* to express, for example, that a nurse needs to be free at least three out of four Fridays
- *Maximum number of assignments for each shift type*
- *Maximum number of a shift type per week*
- *Number of consecutive shift types:* allows to define acceptable sequences of a particular shift type, it is a generalisation of the *Single night* constraint (introduced earlier).
- *Assign 2 free days after night shifts*
- *Assign identical shift types during the weekend*
- *Maximum number of assignments on bank holidays*
- *Patterns enabling specific cyclic constraints:* e.g. a free Wednesday afternoon every 2 weeks, Monday morning is always free for a particular meeting, etc.
- *Counters:* allow to count, for example, assignments, assignments of particular shift types, assignments on particular days of the week, working hours, free days, requested days off, etc. over periods that do not necessarily match the planning period
- *Balancing the workload among personnel:* any of the above counters can be balanced

- *Nurses who should or should not work together:* For example, a trainee should only work when her/his supervisor is at work, personnel that have unique expertise should preferrably not work at the same time, etc.

Constraints can be set differently for particular work regulations. A global cost parameter is associated with each constraint. The cost parameter is considered global because it cannot take different values for different nurses. The quality of each individual nurse's roster is determined by the value of its objective function. We define the quality as the weighted sum of the number of violations of time related constraints, where weights are determined by global cost parameters. The overall quality of a roster is defined as a sum of individual nurse rosters. More details about this model can be found in (Burke et al., 2001b, Vanden Berghe, 2002, Burke et al., 2006).

2.1.2. Method

The expectations for decision support varied strongly among the hospital wards that were considered. Some personnel managers do not find time to be an important factor in rostering and prefer to wait until an algorithm produces a high quality roster, whilst others expect the system to instantly respond with a solution of reasonable quality. It depends on the situation in which the system is to be used, from long term personnel rostering to determine staff occupancy, to instant rescheduling in order to cope with a sudden personnel absence. Meta-heuristic approaches appear to be most suitable to address these diverse requirements.

The main components of the developed approach are presented in Figure 1. A tabu search meta-heuristic, which iteratively searches several neighbourhoods, forms the core of the method. Before the actual optimisation phase can start, a few pre-processing steps are required, some of which will be briefly explained. A `consistency check' procedure was developed that assists the personnel manager in setting the hard constraints in a way that renders the problem feasible. The `freezing' option prevents some parts of the roster from being modified. Freezing generally makes the problem more complex to solve, since it reduces the possibility to address certain time related constraints. Within the boundaries that have been set so far, the initialisation algorithm is applied to generate the initial roster. That could be: a copy of the roster from the previous planning period, the current roster (supposed that some attempt has been carried out before) or a completely empty roster. Neither of the three approaches is likely to lead to a feasible solution. The next step in the procedure is to split the problem into sub problems per skill category, which leads to a considerable complexity reduction. For each skill category, the initial solution is made feasible by randomly adding or removing assignments until all the hard constraints are met. The hard constraints can initially include the minimum coverage or the preferred coverage. The post-plan option denotes that, after the meta-heuristics have generated a solution, additional assignments can be allowed in order to better address the preferred requirements. Without the application of the consistency check, feasibility would not always be obtainable.

After the pre-processing steps are finished, a tabu search algorithm explores the space of the feasible solutions, mainly by re-assigning shifts. The parameters of the tabu search approach (such as the tabu list length, the tabu characteristics of visited solutions, the neighbourhoods, accepting the best-improving or first-improving move, the stop criteria, etc.) were set after rigorous experimentation with various test sets. The main neighbourhood applied by the algorithm is the `single-shift' neighbourhood, in which particular shifts are moved from one nurse's roster to another one on the same day. The application of that neighbourhood alone leads to a fast improvement of the (initially random) roster. However, it turned out to lead to solutions that could, in certain circumstances, be improved manually by experienced personnel managers. A few of the

neighbourhoods were particularly modelled after rigorous observation of experienced personnel managers. Although they appear to be quite computationally expensive, they allow for presenting solutions that strongly resemble rosters that were manually obtained, but with a lower number of constraints violated (Burke et al., 2001a). Examples of such neighbourhoods are:

1. the shuffle neighbourhood in which the assignments on a few consecutive days are swapped between the nurse with the worst schedule and another one,
2. the greedy shuffle neighbourhood in which for each couple of nurses, the shuffle neighbourhood is applied,
3. the weekend neighbourhood which particularly addresses weekend constraints and attempts to solve them without much consideration about the other constraints.

The exploration of the search space moves on to another neighbourhood when the stop criterion for the current neighbourhood is met.
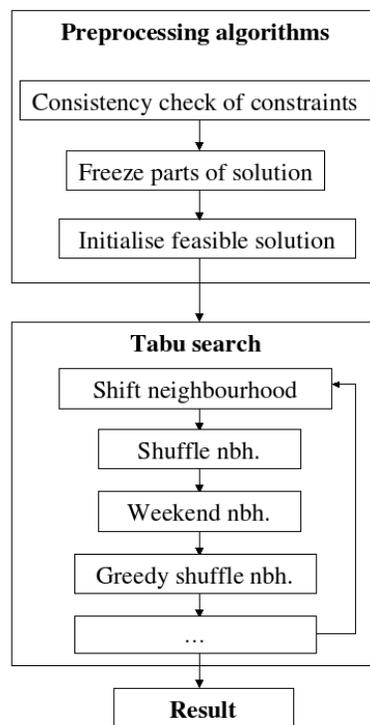

Figure 1. Main components of the tabu search algorithm for nurse rostering

2.2 Case-based reasoning approach

2.2.1 Description of the problem

The nurse rostering problem faced by the Ophthalmology ward at Queen's Medical Centre (QMC) motivated the development of a novel case-based reasoning approach. There are between 30 and 35 nurses in the ward. The rosters are non-cyclical. The ward uses the relatively new self-rostering human resource management methodology, which enables the nurses to express their requests and preferences for working/non-working certain shifts. Nurses in the ward have different characteristics which include qualifications, specialty training, experience, gender, etc.

Nurses can work in 3 shifts: *Early* (07:00-14:45), *Late* (13:30-21:15) and *Night* (21:00-07:15).

All the constraints defined in the introduction to this section, apply to the ward at QMC. A distinction is made between two types of requests.

- *Hard request* that defines a constraint that has to be respected in the roster and
- *Soft request*: defines the preferred shift expressed by a nurse that is desirable to respect, but can be violated in the roster;

Some typical values for a few of the constraints are given below:

- *Max (min) days on*: In general, in the QMC ward full time nurses may work maximum 6 days and minimum 2 days in a row;
- *Max (min) hours*: For example, full time nurses may not work more than 75 hours in a fortnight;
- *Maximum weekends in a row*: In the QMC ward, this is usually 3 weekends.

2.2.2 Method

A new Case-based Reasoning (CBR) methodology was developed that captures the rostering practice of a personnel manager in the ward. CBR relies on previous knowledge and experience gained in solving problems of a given type, rather than on formal models or rules (Kolodner, 1993), (Leake, 1996). In CBR, the description of the actual problems and their solutions are memorised as cases and organised in a case-base. Given the description of a new problem, the case that is most similar to the new problem is retrieved from the case-base and employed in solving the problem.

In the developed CBR approach named CABAROST, the main idea is to capture the knowledge/experience of personnel managers used in solving constraint optimisation problems by storing a history of constraint violations and actions used to repair them. A personnel rostering problem is represented by the list of constraints that have to be satisfied. A case contains a description of the constraint violation and the repair of the violation performed by the personnel manager. A large number of cases were obtained through interviews with a head nurse in the Ophthalmology ward in QMC. Each constraint is described by its type, the type of nurses that are involved in this constraint and a list of parameters.

A repair consists of the type of repair, a list of parameters required in the repair and a list of new violations which are caused in the roster by applying the repair. There are three types of repairs that are used:

- *Reassign*: assigns a shift to a nurse;
- *Swap*: interchanges shift assignments of two nurses on a particular day;
- *Switch*: interchanges shift assignments of two nurses on two days.

The constraint violations and repairs have to be independent of any individual problem instance, so that the experience stored about repairing the violations in one roster could be used to handle violations (of the same type) in any other roster with different nurses (employees). This is achieved by the generalisation process which ensures that information stored in the cases is relevant to rostering, independently of particular nurses. For example, it stores a nurse type rather than a specific nurse involved in the constraint violation and repair. The steps involved in CBR are illustrated in Figure 2. When a new violation is identified in the roster, the case (one or more) containing the most similar violation in the case-base is retrieved. The retrieval process consists of two phases. In the

first phase, the violation is generalised in order to enable the search of the case-base for similar violations. A set of cases containing violations that match the current problem in terms of the violation type and parameters is retrieved from the case-base. In the second phase, the cases retrieved in the first phase are ranked with respect to their distance to the new problem (distance is analogous to the similarity measure). Features that are used in the similarity measure are grouped into three groups:

1. Features that give statistical information about the roster, such as: the percentage of the number of hours in the roster that are still available for assignment, the total number of constraint violations, the magnitude of the violations, etc.
2. Features that give coverage information, i.e. record the number of nurses of each qualification over a period relevant to the type of constraint violation.
3. Shift pattern features that record the assignment of the nurses in the "vicinity" of the day on which the constraint involving the nurse occurred.
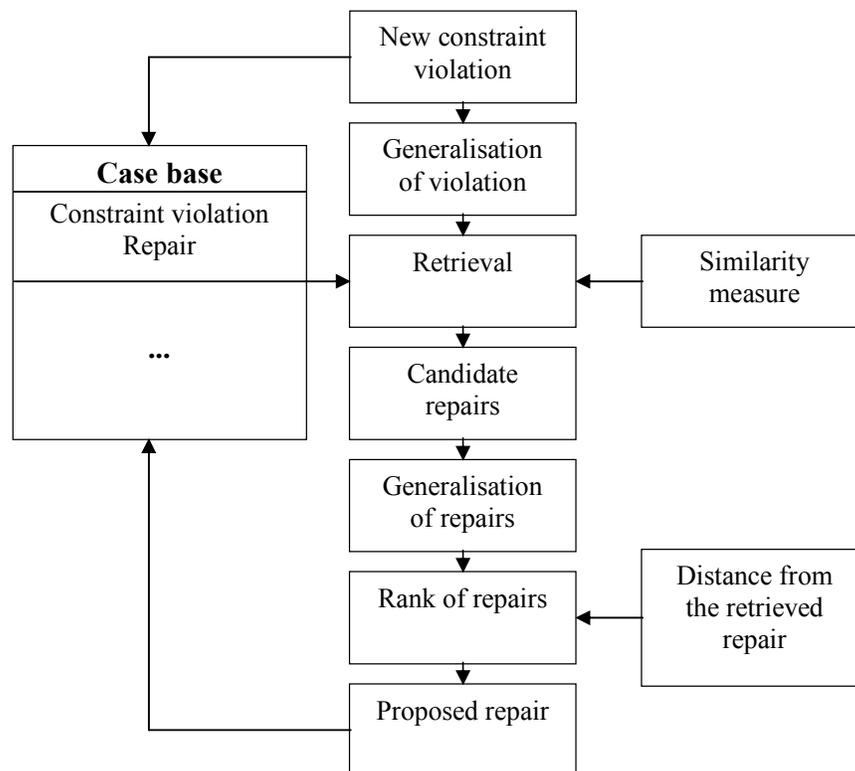
Figure 2. Main components of the case-based reasoning approach to nurse rostering

The repair proposed by the retrieved case has to be adapted to fit the current problem instance. The adaptation process also consists of two phases. In the first phase, the repair from each of the retrieved cases is used to generate a set of all possible candidate repairs that can be performed on the roster using particular nurses. The retrieved cases suggest the repair type to be used together with the type of nurses and shifts involved. In the second phase, the candidate repairs are generalised ("cleaned" from any data relevant for this particular roster) and then ranked with respect to their distance from the repair contained in the retrieved case. This distance takes into consideration the state of the

roster before and after the repair and also the new violations caused by the performed repair.

The constraint violations are repaired iteratively in the roster. However, the order of the repairs greatly affects the final roster quality. A memetic algorithm was developed which searches for optimal sequences of repairs (Beddoe et al., 2008). The results of experiments on the data from the QMC indicate that this hybridisation provides an excellent tool for solving nurse rostering problems.

CABAROST was implemented using Visual C++. Although CABAROST was developed to deal with a QMC ward, the system was designed to be as general as possible and could be used to solve a wide variety of the employee timetabling problems described in the literature.

As an illustration, let us consider a relatively simple constraint violation and the way that CABAROST handles it. Let us suppose that a violation of the *Coverage* constraint type occurred because in the current roster there is a lack of a registered nurse (the most qualified nurse who had extensive training in both the practical and managerial aspects of nursing) on the *Early* shift of the third day of the planning period.

In the retrieval process, this violation is generalized to extract the information needed to compare it to other cases in the case-base (the generalized violation contains the information that a registered nurse is missing from an early shift, while the particular nurse and the day of the violation is not important). The case-base is searched for all the cases that refer to the *Coverage* constraint type involving a registered nurse. The retrieved cases are sorted based on the distance measure. The similarity measure for this constraint includes the following features:

- The magnitude of the cover violation (the difference between the required number of registered nurses and the currently assigned number).
- The number of hours assigned to nurses of any type on the day of violation (the third day of the planning period).
- The number of hours assigned to registered nurses on the day of violation.
- The number of unassigned hours on the day of violation (takes into consideration any nurse that could be assigned to a shift on that day).
- The number of unassigned hours on the day of the violation for registered nurses.

Each of these cases suggests a possible repair of the *Coverage* violation. Let us suppose that the most similar case suggests a *Reassign* repair that was stored in a generalised form specifying that an *Unassigned* shift of any registered nurse is changed to *Early* on the day of the constraint violation. The repair process generates all possible reassignments taking into consideration the registered nurses from the roster whose shift was *Unassigned* on the third day of the roster (a specified day of the violation) and set it to *Early*. These repairs are referred to as the set of candidate repairs. The distance between each of the candidate repairs and the repair in the retrieved case is calculated. This distance involves features describing the state of the roster before and after the repair including the number of nurses of any qualification and also the number of registered nurses that had *Unassigned (Early)* on the third day in the roster, the percentage of contract hours already assigned to the nurse involved in the repair, etc. It also includes the measurement of new violations of constraints caused by the repair. The repair with the smallest distance is selected and applied to the roster.

3 Comparison of the algorithms

Facing two nurse rostering approaches that are of very different nature, the question arises how to compare them. A straightforward approach would be to apply each of the developed approaches to the other problem instances, namely to apply ANROM to the QMC problem instances and to apply CABAROST to the Belgian ones. However, the application of CABAROST to nurse rostering problems addressed by ANROM was not possible because personnel managers were not available to provide the relevant case-by-case examples of their rostering practice. On the other hand, ANROM could not be applied directly to the problem at QMC, because it would re-introduce the modelling deficiencies, which CABAROST was designed to avoid, i.e. it would require the definition of the objective function together with the weights of constraints and a rigid distinction between hard and soft constraints.

Instead of applying each approach to the other problem instances, we focused on the identification of advantages and disadvantages of each of them in solving real-world nurse rostering problems. We introduce seven criteria that we find important for the comparison of approaches for nurse rostering: expressive power, flexibility, algorithmic power, learning capabilities, maintenance, rescheduling capabilities, and parameter tuning. In the remainder of the section, we will evaluate the two described approaches against these criteria.

3.1 Expressive power

Expressive power refers to the ability of the model/system to represent a wide variety of real-world constraints and other characteristics, such as the possibility to define multiple skill categories, multiple shift types, different work regulations or enrollments, personal preferences, flexible coverage constraints, etc. Translating real-world requirements into mathematical models is not an easy task, especially in highly constrained problems such as nurse rostering. Consequently, approaches reported in the literature are often detached from the reality in real-world healthcare institutions and are developed for simplified nurse rostering problems.

Both developed systems consider real-world problems with all their complexities. Therefore, they can be assessed as being of high expressive power. Interestingly, they deal with the constraints and preferences of the personnel managers in different ways. ANROM covers a wide variety of constraints that are present in Belgian hospitals. In the developed meta-heuristic approach, most of the knowledge about the problem is put in the objective function which measures the constraint violations. It is a weighted linear combination of the magnitude of the violations of each constraint. Weights represent the relative importance of the constraints, thus enabling the personnel manager to express his/her preference, but also to obtain a roster with the desired characteristics. A larger weight means that a higher importance is given to a violation of the constraint. The user is allowed to interactively change the weights until the algorithms produce a high quality roster. The ANROM approach also attempts to capture some practice of the personnel managers through the definition of neighbourhoods. Some of the neighbourhoods have been generated after careful observation of personnel managers while creating or modifying a roster. It is assumed that these neighbourhoods should lead to a better performance of the method, without being tailored to specific cases.

Similarly to ANROM, CABAROST facilitates the definition of complex personnel characteristics. They are hierarchically organised (opposite to the ANROM approach which models individual personnel characteristics), but at the same time CABAROST allows their overlapping, and includes such details as qualifications, specialty training, experience, and gender. A variety of constraint types can be defined and applied to all

nurses or to a subset of nurses with certain characteristics. Many of the preferences/decisions made by personnel managers are of a personal, subjective nature, and are therefore difficult to model explicitly. The development of strict principles used in rostering in the form of IF-THEN rules would be a tiresome, if not impossible task. CABAROST enables elicitation of rostering practice from personnel managers in a very natural manner. It captures the preferences of the personnel manager regarding different staff requests, the importance of constraints, etc., on a case by case basis; thus preferences are implicitly contained in the case-base. CABAROST requires intensive consultations with the personnel managers in order to collect their practice. On the other hand, this helps in the acceptance of such a system by the user who can recognize his/her rostering practice in the case-base.

3.2 Flexibility

While various approaches have been satisfactorily applied to a wide range of personnel rostering and scheduling problems, a question arises concerning the applicability of a particular implementation when it is faced with different problem specifications or data characteristics. In the literature on scheduling, the term flexibility has a variety of meanings. We refer to two definitions of particular importance for nurse rostering. Le Pape defines flexibility as the ability to add/remove constraints at any time (Le Pape, 1993), while Cavalieri et al. refer to it as the ability to address effectively the changing circumstances in the scheduling environment (Cavalieri et al., 2007).

Both of the systems are very well suited for tackling different nurse rostering problems, and other timetabling problems. However, the addition of a new constraint requires some modifications in the method.

Depending on the character of new constraints, the objective function of the ANROM approach requires modification to a smaller or larger extent. The simplest situation occurs when a new constraint matches one of the constraint types that already exist in the model. In that case, only a few lines of extra code are required for modelling, while the evaluation function remains the same. The steps of the meta-heuristic itself are independent of the constraints, unless new neighbourhoods are required.

The inclusion of a new constraint in CABAROST requires a reasonable amount of effort. It requires a formal definition of the constraint which includes the type of constraint, the type of nurses that may be involved in the constraint and a set of parameters relevant for this violation. The action(s) that the personnel manager will perform on the particular roster in order to repair the violation has (have) to be also elicited. They have to be stored in the cases in a generalised form so that they can be useful for any roster with different staff. The description of a repair includes the set of parameters relevant for the repair and a set of new violations that this repair caused. It is also useful to provide repairs to new violations caused by the original repair.

Scalability is an important dimension of flexibility. The number of nurses, the number of shift types, and the length of the planning period are the problem dimensions to consider with respect to scalability. Neither of the two systems has set an upper limit to any of the dimensions. However, the larger the problem size the more computational time is required by both ANROM and CABAROST to solve it.

3.3 Algorithmic power

We consider efficiency and effectiveness to be the main factors that determine the algorithmic power. In general, effectiveness is defined to measure the degree to which

goals are achieved (Turban et al., 2008). Therefore, it takes into consideration the output of the system. Efficiency measures the use of inputs (or resources) to achieve outputs (Turban et al., 2008). As discussed in the introduction, the comparison of scheduling systems reported in the literature is based either on effectiveness alone or on both effectiveness and efficiency. Effectiveness is reflected in the achieved value of the objective function, while the time required to generate the roster reflects the efficiency of the system.

In general, meta-heuristics approaches evaluate a large number of solutions to the problem throughout the search of the solutions space. Therefore, the efficiency of the ANROM approach mostly depends on the evaluation speed of individual nurse rosters. A lot of attention was paid to the development of a sophisticated evaluator that could handle a variety of constraints (Burke et al., 2001a). It contributes significantly to the efficiency of the approach. Experiments were carried out in which rosters manually constructed by the personnel managers are compared with the ones obtained by the system. The computed rosters outperformed the original (manual) ones with respect to the achieved value of the objective function in all experiments.

In general, the CABAROST effectiveness depends on the proper definition of the similarity measure which enables the retrieval of an "appropriate" case, while the system efficiency is determined by the speed of the retrieval process. One of the possible ways to evaluate the quality of the retrieval process is to compare the repair action suggested by the system and by the personnel manager (Petrovic et al., 2002). Three outcomes were identified: exact match when the CABAROST repair was the same as the manager's one, equivalent match when the generated repair involved nurses of the same type and the same shifts as those used in the manager's one, and fail otherwise. As its size increases, CABAROST produces more exact or equivalent repairs, namely already with 120 cases it generates more than 80% of exact or equivalent repairs. In order to enable an efficient retrieval process, CABAROST searches only the subset of the case-base, namely it looks for the cases of a particular constraint violation type. In addition, it organises cases in a multi-dimensional binary search tree (called kd-tree) to avoid searching all the cases of a given type (Beddoe, 2005).

3.4 Learning

Learning can be defined as the capability of the system to improve its performance by gaining new knowledge/experience over time.

In general, there is no learning in methods for optimisation. ANROM is capable of generating high quality solutions, but it starts always from scratch. An enhancement of methods for optimisation with a sophisticated learning component (other than a tabu list) which would navigate the search of the solution space based on the results from previous iterations remains to be an interesting research issue within the optimisation community.

By definition, a CBR system is designed to enable learning by memorizing the successfully solved problems that contribute with the new knowledge/experience to an expansion of the case-base. The main motivation of using CBR in nurse rostering was to use the experience that was gained from repairing previous violations. In addition, CABAROST enables learning from failures by penalising cases whose repairs caused reappearance of the same violation in the roster. Penalisation is performed by the means of weights associated with cases that are used in the retrieval process. Penalised cases have smaller chances to be retrieved for a similar type of constraint violation for which they were not successful. In such a way, the repetition of the same mistake is avoided in the future. Such a case weighting system enables cases, which repeatedly generated

repairs that were not of high quality to be removed from the case-base, thus improving the performance of the case-base. At the current stage of development this is done manually by the user who can detect if a case is frequently being penalised.

3.5 Maintenance (redundant/conflicting constraints and cases)

We define maintenance in the context of rostering as the ability of the system to update the knowledge of the problem.

The ANROM approach detects, in certain circumstances, potentially conflicting constraints and handle them accordingly. As an example, let us consider both the cyclical patterns and the personal preferences. Both constraints are perceived and modelled to be dominant to the simpler and more general constraints. Conflicts which involve both of them are automatically detected and result in ignoring the later ones. For example, a cyclical pattern with an *Early* shift every Monday due to a weekly meeting conflicts with any constraint that restricts the number of Mondays worked. However, the automated detection of conflicting constraints should be taken with precaution, because it can lead to the loss of flexibility.

Redundancy is explicitly allowed in ANROM. For example, it is possible to model the constraint *Maximum two shifts per week* and at the same time *Maximum 16 hours per week*. In most situations the duration of shifts is 8 hours and therefore these two constraints are redundant. Evaluation of redundant constraints has a negative effect on the efficiency, but it turned out that it increases the effectiveness as it may guide the search away from parts of the solution space that are not particularly interesting.

In CBR systems maintenance includes issues regarding the size of the case-base and handling redundant/conflicting cases. The quality and coverage of the case-base affects the functioning of the CBR system to a high extent. However, it would be very difficult to recommend the size of the case-base that would enable an adequate capture of the problem space and a high quality performance of the system. This question is yet to receive the necessary attention in the artificial intelligence community, although some results have been reported in the literature (see, for example, Santamaria and Ram, 1996). A combination of on-going training and case weighting in CABAROST enables the maintenance of the case-base in terms of the size of the case-base (Beddoe et al., 2008). A case acceptance threshold is introduced to enable the user to control the repairs proposed by the system. The threshold is set for the acceptable distance between the retrieved repair and the generated one. The user can set this threshold depending on the amount of input he/she wishes to have in the process. Initially, this threshold can be set to a small value enabling the user to monitor the repairs generated by the system. If the personnel manager is not satisfied with the repair, he/she can change either the parameter involved in the repair (for example, change the nurse) or use a different repair. Later on, if the personnel manager user becomes confident that the system generates the repairs that are appropriate the value of the threshold could be increased. The weighting of cases that enables learning (described in Section 3.4) also enables the unsuccessful cases to be removed from the case-base, thus reducing the case-base size and leading to better and faster retrieval process. Redundant cases are not handled in CABAROST. However, it is an interesting issue, as redundant cases can affect the efficiency of the system.

3.6 Rescheduling

In our research we refer to robustness as the ability of a system to react to unforeseen disturbances in the real-world environment. In healthcare institutions, very often, there is a need to reschedule some resources in an already existing solution in response to expected/unexpected events, such as the request of a nurse for holidays or days off, a

request to work a certain duty on a certain day, leave due to illness, maternity leave, temporary secondment to another ward, sudden increase in demand due to an epidemic, etc. Often in practice, such cases are dealt with manually, either calling nurses "at the last minute" or borrowing nurses from an existing pool. This often leads to new constraint violations that may remain unresolved.

In ANROM, a new search triggered by small requirement changes will not start from scratch but from the previous solution (such a solution is no longer a local optimum due to the new input data). The algorithm parameters remain the same and the search continues with the exploration of the neighbourhood of the previous solution. Therefore, the chance that the local search approach comes up with a completely different solution is moderate indeed. That is exactly what the personnel managers expect from the system. CABAROST enables personnel managers to reschedule (add/remove) one or more nurses in response to a changing environment using the previous experience, and also to handle new constraint violations in a systematic manner.

The question that arises is how to measure the quality of the new schedule modified to respond to the events. While in the scheduling literature there are definitions of stability measures defined in the context of manufacturing environments, to our knowledge there is no measure of stability defined for nurse rostering. A possible way to do that is to measure how many shifts assignments in the original roster differ from those in the roster created as a result of rescheduling; or how many nurses are rescheduled. We believe that this presents an interesting direction for research in the employee timetabling community.

3.7 Parameter tuning

In order to achieve a high quality performance the parameters employed in the algorithm usually have to be carefully tuned.

The meta-heuristics applied in the ANROM approach have been tuned in order to address a wide range of problem instances. In that respect, further tuning of the algorithms is not required when addressing a new problem. However, to reach that level of generality a very time-consuming effort was required to set the meta-heuristic parameters, such as stop criteria, tabu list length, solution characteristics that determine the tabu status, etc. Experiments on a large number of different problem instances led to parameter settings that are general but depend upon some problem characteristics. For example, stop criteria have been defined in accordance with the size of the problem, which was determined by the number of nurses, the number of shift types and the length of the planning period. We distinguish between the global stop criterion and the local ones that are applied to the whole search and each neighbourhood search, respectively. Also, the appearance of certain constraints determines whether or not particular neighbourhoods will be explored during the search. In addition to the tuning of algorithm specific parameters, the performance of the approach also strongly depends upon setting and tailoring the constraints and tuning their corresponding weights. The difficulty of that task seems to be strongly underestimated by the researchers. The large number of possibly conflicting and redundant constraints, together with the rather intractable search procedure, makes it difficult to predict the behaviour of the approach under certain parameter settings. For example, it was noticed that personal preferences were often violated in the solution that was generated by the meta-heuristics. As a remedy, the personnel manager attempted to increase the weight of the particular constraint. Unexpectedly, by doing so, the obtained roster became even worse. Intuitively, we believe that a very large weight for a constraint can sometimes put up barriers that are difficult to cross by the local search algorithms.

CABAROST requires tuning the parameters that are of different nature than ANROM ones. Parameters that require tuning are as follows:

1. Features in the similarity measure: In CABAROST a case consists of sets of features that describe the type of constraint violation and the performed repair. The initial set of features was determined through consultation with the personnel managers at QMC. A genetic algorithm was developed to refine this large initial set of features, by selecting relevant features and at the same time determining their relative importance. The objective defined in the genetic algorithm was to maximise the classification accuracy of the retrieval process (Beddoe and Petrovic, 2006). By reducing the number of features that is necessary to store in each case, the search time of the system (efficiency) is significantly decreased. In addition, this provided us with an insight into the nature of rostering decisions by identifying those features of the roster which were relevant to the repairs of constraint violations.
2. Weights of the cases: The cases that were not useful for repairing the constraint violation for which they were retrieved are penalized by reducing their weights (described in Section 3.4). The rate of the weight reduction has to be tuned. A large reduction may be too severe thus preventing the retrieval of potentially useful cases, while too small reductions obstruct the system to learn from failure.
3. Thresholds for acceptance of the repair suggested by the retrieved case: This is useful in the training of the system and affects the quality of the cases in the case-base (explained in more detail in Section 3.5).

5 Conclusions

This paper addresses an issue of qualitative comparison of nurse rostering approaches that has not been investigated thoroughly by the timetabling community yet. Naturally, the comparison of methods is stated as a multicriteria problem, in which seven criteria relevant for the comparison are identified, including the expressive power, the flexibility, the algorithmic power, the learning capabilities, the maintenance, the rescheduling capabilities, and the parameter tuning. These criteria would enable comparison of different approaches, as is the case with ANROM and CABAROST, that would be difficult to compare based on the quality of the produced rosters since their evaluation differs completely.

The purpose of this paper is not to put forward any particular algorithm for nurse rostering. Our aim is to demonstrate the need for a more thorough analysis of nurse rostering problems and algorithms developed for their solving. We hope that it will ease further comparison for situations in which qualitative results are incommensurable. More importantly, we believe that it will contribute to bridging the gap that still exists between the timetabling theory and practice, and that it will make the implementation in real-world settings simpler. Although the criteria introduced for comparison are discussed in the context of nurse rostering, they are applicable to employee timetabling approaches in general.

References

Azaiez MN, Al-Sharif SS (2005) A 0-1 goal programming model for nurse scheduling. Computers and Operations Research, 32(3): 507-491

Bard JF, Purnomo HW (2005) Preference scheduling for nurses using column generation. European Journal of Operational Research 164: 510-534

Bard JF, Purnomo HW (2007) Cyclic preference scheduling of Nurses Using A Lagrangian-Based Heuristic. Journal of Scheduling 10(1): 5-23

Beddoe G. (2005) Case-based Reasoning in Personnel Rostering, PhD thesis, University of Nottingham.

Beddoe G, Petrovic S, Li J (2008) Hybrid Metaheuristic Case-based Reasoning System for Nurse Rostering, accepted for publication in the Journal of Scheduling.

Beddoe G, Petrovic S (2007) Enhancing case-based reasoning for personnel rostering with selected tabu search concepts. Journal of the Operational Research Society (JORS) 58(12): 1586-1598.

Beddoe G, Petrovic S (2006) Selecting and weighting features using a genetic algorithm in a case-based reasoning approach to personnel rostering. European Journal of Operational Research (EJOR) 175 (2): 649-671.

Burke E.K., Curtois T., Qu R., and Vanden Berghe G (2007) A Time Pre-defined Variable Depth Search for Nurse Rostering. Technical report, University of Nottingham.

Burke EK, Cowling P, De Causmaecker P, Vanden Berghe G (2001a) A Memetic Approach to the Nurse Rostering Problem. Applied Intelligence 15(3): 199-214

Burke EK, De Causmaecker P, Petrovic S, Vanden Berghe G (2001b) Fitness Evaluation for Nurse Scheduling Problems. In: Proceedings of Congress on Evolutionary Computation, CEC2001, Seoul, IEEE Press, pp 1139- 1146

Burke EK, De Causmaecker P, Petrovic S, Vanden Berghe G (2006) Metaheuristics for handling Time Interval Coverage Constraints in Nurse Scheduling. Applied Artificial Intelligence. 20(9): 743-766

Burke, EK, P De Causmaecker, Vanden Berghe G, Van Landeghem H (2004) The State of the Art of Nurse Rostering. Journal of Scheduling 7(6): 441-499

Cavalieri S, Terzi S, Macchi M (2007) A Benchmarking Service for the Evaluation and Comparison of Scheduling Techniques. Computers in Industry 58: 656-666

de la Asunción M, Castillo L, Fernández-Olivares J, García-Pérez O, González A, Palao F, (2007) Handling fuzzy temporal constraints in a planning environment. Annals of Operations Research 155(1): 391-415

Gutjahr WJ, Rauner M S (2007) An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. Computers & Operations Research 34(3): 642-666

Kaplansky E, Meisels A (2007) Distributed Personnel Scheduling – Negotiation among Scheduling Agents. Annals of Operations Research 155(1): 227-255

Kellogg, DL; Walczak, S (2007) Nurse scheduling: From academia to implementation or not? Interfases 37(4): 355-369

Kolodner J (1993) Case-based Reasoning, Morgan-Kaufmann.

Leake D (eds.) (1996) Case-Based Reasoning, Experiences, Lessons & Future Directions AAAI Press.

Le Pape C, (1993) Classsification of Scheduling Problems and Selection of Corresponding Constraint-based Techniques. In: Proceedings of the IEE Colloquium on Advanced Software Technologies for Scheduling, London, UK:1.1-1.3.

Maenhout B, Vanhoucke M (2007) An Electromagnetism meta-heuristic for the nurse scheduling problem, Journal of Heuristics 13 (4): 359-385

Maenhout B, Vanhoucke M (2006) New Computational Results for the Nurse Scheduling Problem: A Scatter Search Algorithm. In Gottlieb J, Raidl GR (eds.): EvoCOP, LNCS 3906, Springer, pp 159-170

Meyer auf'm Hofe H (2001) Solving Rostering Tasks as Constraint Optimization. In: Burke, E, Erben W (eds.) Practice and Theory of Automated Timetabling III, LNCS 2079, Springer, pp. 191-212

Parr D, Thompson JM (2007) Solving the multi-objective nurse scheduling problem with a weighted cost function. Annals of Operations Research 155(1):,279-288

Petrovic S, Beddoe G, Vanden Berghe G (2003) Storing and adapting repair experiences in personnel rostering. In: Burke EK, De Causmaecker P (eds) Practice and Theory of Automated Timetabling IV, LNCS 2740, Springer, pp 149-166

Santamaria, JC, Ram A (1996) Systematic Evaluation of Design Decisions in Case-Based Reasoning Systems. In (Leake 1996), pp 199-233

Turban E, Aronson J, Liang T-P, Sharda R (2007) Decision Support Systems and Business Intelligence Systems, Eight Edition, Prentice Hall.

Vanden Berghe G (2002) An Advanced Model and Novel Meta-heuristic Solution Methods to Personnel Scheduling in Healthcare. PhD thesis, University of Gent

White C, Nano E, Nguyen-Ngoc D-H, White G (2007) An Evaluation of Certain Heuristic Optimization Algorithms in Scheduling Medical Doctors and Medical Students. In: Burke EK, Rudová H (eds), Practice and Theory of Automated Timetabling IV, LNCS 3867, Springer, pp. 105-115

Wren A (1996) Scheduling, timetabling and rostering - a special relationship? In: Burke E, Ross P. (eds), LNCS 1153 Springer, pp. 46–75