Novel Meta-heuristic Approaches to Nurse Rostering Problems in Belgian Hospitals

Edmund Kieran Burke*, Patrick De Causmaecker†, and Greet Vanden Berghe†

*Automated Scheduling, Optimisation and Planning Research Group
School of Computer Science & IT
University of Nottingham, Jubilee Campus, Nottingham NG81BB, UK
†KaHo Sint-Lieven, Information Technology
Gebroeders Desmetstraat 1, 9000 Gent, Belgium

Abstract

Constructing timetables of work for personnel in healthcare institutions is a highly constrained and difficult problem to solve. In this chapter, we will present an overview of our development of the algorithms that underpin a commercial nurse rostering decision support system that is in use in over 40 hospitals in Belgium. As such, we are particularly concerned with the real world regulations and requirements of Belgian institutions. We have concentrated upon short term rostering which involves assigning tasks to nurses in a hospital ward. Of course, the over-riding requirement of our algorithms is to assure a permanent level of care for the patients. However, our approaches are also required to consider administrative requirements, the nurses' contracts and their personal preferences.

In this chapter, we will describe a general model for non-cyclical rostering that copes with a large set of constraints and with the varying objectives that are encountered in practice. The model includes minimum and preferred coverage levels, self-definable (overlapping) qualifications and shift types, different contracts with modifiable constraints and other features.

We present a solution framework that applies a modular evaluation function. It also provides many options for initialising and for formulating various objectives and meta-heuristics for searching solutions. Different local search heuristics are applied in different neighbourhoods and we demonstrate the importance of introducing problem specific characteristics into the algorithms.

We will conclude by giving an overview and evaluation of the meta-

heuristics that have been investigated for the problem and we will outline some future research directions.

1 Introduction

There is an increasing pressure of work in healthcare organisations in Belgium. This continues to remain a serious problem in spite of recent significant technological advances. One potential way of easing this pressure is to develop better nurse rostering decision support systems that can help to produce rosters which employ resources more efficiently. However, there is more than just one goal when generating personnel rosters in hospitals. Resource efficiency is important but so is the satisfaction level of patients. Personnel rosters also affect the organisational structure of the hospital and they directly influence the private lives of the staff. It is therefore important to provide an interactive system that generates high quality scheduling solutions within a reasonable amount of computing time. Such schedules should cover the hospital requirements while avoiding patterns that are detrimental to the nurses' and patients' priorities.

The presented approach concentrates on the short-term problem of assigning specific tasks to a sufficient number of qualified nurses. This problem is often referred to as 'rostering', 'scheduling' or 'timetabling'. A part of the problem data, such as the number of personnel in a ward, the required qualifications, the definition of shift types, etc is determined at the strategic level. Although these settings are determined at the strategic level. Although these settings are not usually considered to be part of the nurse rostering problem, some longer term strategic decisions can affect the solution strategies. The model described in this chapter therefore provides several possibilities for flexible problem setting. Examples are: shift types can be divided over several nurses, personnel demands can be expressed in terms of shorter intervals than shift length, night shifts can be assigned to a special category of night nurses, possibilities exist for creating part time work, people can temporarily be assigned to different wards in order to address emergencies, personnel members can express certain preferences for particular times in the planning period, etc.

The problem of finding a high quality solution for the personnel timetabling problem in a hospital ward has been addressed by many scientists, personnel managers and schedulers over a number of years. In recent years, the emergence of larger and more constrained problems has presented a real challenge for researchers. A flexible planning system should incorporate as much knowledge as possible to relieve the personnel manager or head nurse from the unrewarding task of setting up objective schedules that attempt to satisfy a range of conflicting objectives.

This chapter presents each of the different steps in developing the system: a comparison with related publications (Section 2), the modelling of the nurse

rostering problem (Section 3), the setting up of a solution framework (Section 4), and the development of appropriate search techniques (Section 5).
We also investigate the applicability of a multi criteria approach for solving the nurse rostering problem (Section 6). The possibility of assigning weights to certain criteria or conditions guides the search through a different set of solutions and produces interesting results of a very good quality.
We compare the results of the developed algorithms, summarise their benefits and drawbacks and end with a general discussion in Section 7.

2 Related Literature

Employee scheduling covers staffing, budgeting and short-term scheduling problems. Although these fields have variable time horizons, they are strongly interrelated. Scheduling of hospital personnel is particularly challenging because of different staffing needs on different days and shifts. Unlike most other facilities, healthcare institutions work around the clock.
Until recently, nearly all personnel scheduling problems in Belgian hospitals were solved manually. Planners had no automatic tool to test the quality of a constructed schedule. They made use of very straightforward constraints on working time and idle time in a recurring process.

2.1 Categories of nurse scheduling

We distinguish between different categories of nurse or hospital scheduling:
Hospital **staffing** involves determining the number of personnel of the required skills in order to meet predicted requirements [20, 23, 42, 47, 51, 55, 56]. Factors that make this task complex are the organisational structure and characteristics, personnel recruitment, skill classes of the personnel, working preferences, patient needs, circumstances in particular nursing units, etc. Another significant staffing decision is to define work agreements for part time workers, to decide whether substitution of skill classes is allowed and for which people.

Two major advantages of **centralised scheduling** [23, 43, 47, 51] are fairness to employees through consistent, objective, and impartial application of policies and opportunities for cost containment through better use of resources.
We refer to **unit scheduling** when head nurses or unit managers are given the responsibility of generating the schedules locally [1, 7, 21, 29, 30, 44]. It is sometimes considered to be an advantage because nurses get more personalised attention. Conversely, it can be the case that personnel members see their schedule as a punishment or suspect the head nurse to be giving preferential treatment to certain people.

Self-scheduling is a manual process [31, 40]. The technique is more time consuming than automatic scheduling but it has the advantage that the nurses co-operate and are asked for advice. Generally it is performed by the personnel members themselves and co-ordinated by the head nurse of a ward. Self-scheduling is so common that complete automation is not recommended.

Cyclical scheduling concerns organisations in which each person works a cycle of a number of weeks [3, 24, 35]. Staff know their schedule a long time in advance and the same schedule patterns are used over and over again. There are significant benefits [51] but cyclical schedules unfortunately lack generality. They cannot address (without major changes) many of the flexible features and personal preferences that are part of the modern problem.

2.2 Complexity factors

When only considering short term scheduling, we distinguish between two main goals: meeting the coverage requirements [9, 14, 29, 41] and satisfying time related constraints [5, 32]. Approaches that solve all time related constraints generally consider a low number of such constraints.

Many models allow limited violations of coverage constraints and consider them in the evaluation function [16, 30]. These violations are not allowed in our model, unless they are explicitly approved of as a relaxation [19]. The approach provides different planning options for precisely defining the desired personnel coverage.

The personnel scheduling literature has examples with strictly separated skill categories [2, 27, 34, 37, 54]. Other approaches apply hierarchically substitutable qualifications (in which higher skilled people can replace less experienced colleagues) [1, 21]. User definable substitution [9, 14, 32, 53] is particularly well suited to real world practice.

In simplified research examples, problem definitions have the same constraints for all the personnel members [2, 16, 54]. The assignment of schedules to people is then very arbitrary. More realistic examples take part time contracts into account and provide flexibility to define personal work agreements [14, 17, 30, 41]. Some approaches generate schedules that consist of days off and days on [31, 34]. The assignment of actual shifts to people, is then left for a head nurse to carry out manually. Algorithms developed for use in practical healthcare environments do not usually work with three strictly distinct shift types (like in [1, 5, 49]). The activities in hospitals are so diverse that a large number of user-definable shifts is often necessary [17, 29, 30, 36, 41, 54].

The personnel requirements are nearly always expressed as a number of people required per shift type or even per day [1, 9, 14, 21]. In a more flexible approach [12, 19], the number of possible shift types is higher than in other problems. However, but also the idea o shift types is higher than in other problems. However, the idea of composing a schedule with different combinations of shift types, through time interval requirements has not been well studied in the nurse rostering literature.

We have compared a large number of time related soft constraints with those that are implemented in our model [50]. Some models apply set values for the constraints [1, 26, 36] whereas they are user definable in more advanced approaches [5, 9, 14].

2.3 Nurse Rostering Approaches

Since the 1960's, publications on various aspects of healthcare personnel scheduling appear regularly. We categorise approaches into optimisation, heuristic and artificial intelligence techniques.

Most mathematical scheduling approaches make use of an objective function which is optimised subject to certain constraints. Nearly all of the earlier papers in the literature (e.g. [32, 49, 52, 53]) mention mathematical optimising techniques for their linear models. Simplifications of the real data are unavoidable. Real world problems are too complex to be *optimised* and many authors employ heuristics (e.g. [6, 28, 45, 46]). It is also common to define more than one objective [2, 5, 16, 26, 34, 37].

Although cyclical schedules are generally considered to be less difficult to generate, most of them are constructed with heuristic [45, 6, 28, 29] techniques. In the 1980's and later, artificial intelligence techniques for nurse scheduling (declarative approaches [18], constraint programming [15, 36, 29, 30], expert systems [16, 45, 37], case based reasoning [4, 39] etc) were investigated. Some of these approaches are still relevant to today's research issues [15, 17, 30].

Many of the most recent papers (1990's and later) tackle the problem with metaheuristic approaches such as simulated annealing [8, 25], tabu search [5, 21] and evolutionary algorithms [1, 9, 22, 27, 48].

There are many advanced models for practical personnel scheduling but none of them is suited for the problem that we discovered in Belgian hospitals. Strong arguments for the importance of the model and solution methods presented in this chapter are the flexibility of the approach, the applicability in practice and the generic problem formulation.

3 The nurse rostering problem

3.1 Problem Dimensions

This research arose from the need for automated rostering assistance in Belgian healthcare organisations. We developed a general model for the nurse rostering problems and refer to it as ANROM (Advanced Nurse Rostering Model). A software package based on the model and the solution framework was first implemented in a hospital in 1995 but the system is still evolving to cope with the new and more complex real-world problems that keep appearing. So far, over 40 hospitals in Belgium, of which some have about 100 wards, have replaced their time consuming manual scheduling by this system.

Although the problem is user-defined to a large extent, the software has to be efficient in different settings. Every specific hospital ward should be able to formulate its problem within the restrictions of the model described in the following sections.

The main goal of the system is to create a schedule by assigning shift types to skilled personnel members, in order to meet the requirements in a certain planning period. Personnel have work regulations limiting their assignments.

		Start	End
M	morning shift	06:45	14:45
L	late shift	14:30	22:00
N	night shift	22:00	07:00

Table 1: Example of shift types

Hospitals are organised in wards with fixed activities, usually a settled location, and, for the most part, they have a permanent **team of nurses**. Although practical situations often allow people to be moved to another ward whenever a personnel shortage is unsolvable, this personnel rostering problem concerns a group of personnel belonging to the same ward.

Personnel members in a ward belong to **skill categories**. The division into categories is based upon the particular level of qualification, responsibility, job description, and experience of the personnel. Rather than employing strictly disjoint skill categories or hierarchical substitutability, we opted for a solution that is closer to the reality in hospitals. The problem of replacing people is solved in ANROM by assigning *alternative* skill categories to certain people. People with more experience or who have taken some exams, can be substitutes for higher skill categories.

Hospital personnel have **work regulations** or contracts with their employer. Several job descriptions such as part time work, night nurses' contracts and weekend work are possible. The regulations involve different constraints but they can make the schedules much more flexible. Moreover, very personal arrangements like 'free Wednesday afternoons' or 'refresher courses' at regular points in time, can easily be set. It is not unlikely to have personalised contracts for the majority of personnel members in Belgian hospitals.

A **shift type** is a predefined period with a fixed start and end time in which personnel members can be on or off duty. Different part time contracts require a large variation in start and end times and in duration. Table 1 presents a simplified example of a set of shift types. It is common that hospital schedulers define shift types according to their needs.

Planning periods for nurse rostering vary from a couple of days to a few months. Since cyclical rosters are not common at all, it is important for individual employees to know their schedule some time in advance. Long term scheduling, on the other hand, should not be too detailed because the personnel requirements and preferences fluctuate and are not predictable in the long term.
Short planning periods enable the search algorithms to find good quality results much faster than longer planning periods. However, guaranteeing fairness among personnel members is restricted when the planning period is short.

The roster, in which the shift assignments are stored, is called the *schedule*. We define assignment units as entities of minimum allocation in a schedule. They are mainly introduced to express and evaluate the soft constraints on the

personnel's schedules. Each shift type corresponds to an assignment unit in practice.
We illustrate the meaning of assignment units with a simple example. A fragment of a possible personnel roster is presented in Fig. 1. We notice that there are five people in the ward, and the shift types correspond to Table 1. Fig. 2 presents the *schedule* that corresponds to the roster of Fig. 1. Each column in the schedule represents an assignment unit. For every day of the planning period there are 3 columns, one for each shift type. The assignment units

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
P1	M	M	L	L	N		
P2	N		N	L	L		
P3	M	M	M	M	M	M	M
P4	M		L	N	N	N	
P5	M L	L	L	L			

Figure 1: Roster example for 5 people (P1,. . . , P5) and 1 week; M, L, and N being the shift types introduced in Table 1

are ordered according to the start times of the shift types that they represent. When two shift types have the same start time, the first assignment unit will match the shift type with the earliest end time.

3.2 Hard and Soft Constraints

Hard constraints are those that must be satisfied at all costs. Soft constraints are those which are desirable but which may need to be violated in order to generate a workable solution. We call a *feasible* solution one that satisfies the following hard constraints:

- all the shift types specified in the personnel requirements have to be assigned to a personnel member
- one person cannot be assigned twice to the same shift on the same day
- shifts can only be assigned to people of the right skill category.

	Schedule Example																				
P1	*	_	_	*	_	_	_	*	_	_	*	_	_	_	*	_	_	_	_	_	_
P2	_	_	*	_	_	_	_	_	*	_	*	_	_	*	_	_	_	_	_	_	_
P3	*	_	_	*	_	_	*	_	_	*	_	_	*	_	_	*	_	_	*	_	_
P4	*	_	_	_	_	_	_	*	_	_	_	*	_	_	*	_	_	*	_	_	_
P5	*	*	_	_	*	_	_	*	_	_	*	_	_	_	_	_	_	_	_	_	_

Figure 2: Schedule corresponding to the roster in Fig. 1: '*' denotes that there is an assignment in the schedule, '-' denotes that the schedule is free.

The real-world situation addressed in this research incorporates a high number of **soft constraints** on the personal schedules. The soft constraints will preferably be satisfied, but violations can be accepted to a certain extent. It is highly exceptional in practice to find a schedule that satisfies all the soft constraints. The aim of the search algorithms is to minimise the penalties due to violations of these constraints.
The model includes exceptions for the evaluation in addition to certain corrections which are required in holiday periods or periods of illness absence. Boundary conditions at the beginning and end of the planning period have an important impact on the evaluation. No penalty is generated when a violated constraint can still be satisfied by scheduling suitable shifts in the next planning period.
The list of soft constraints can be divided into 3 categories.

1. Certain constraints hold for the entire hospital. Examples include:
    - Minimum time between two assignments
    - Allow use of an alternative skill category in certain situations
2. Another set of soft constraints is the same for all the people with the same contract (full-time, half-time, night nurses, etc). Values are set by the users. Examples include:
    - Maximum number of assignments in the planning period
    - Minimum/Maximum number of consecutive days
    - Minimum/Maximum number of hours worked
    - Minimum/Maximum number of consecutive free days
    - Maximum number of assignments per day of the week
    - Maximum number of assignments for each shift type
    - Maximum number of a shift type per week
    - Number of consecutive shift types
    - Assign 2 free days after night shifts
    - Assign complete weekends
    - Assign identical shift types during the weekend
    - Maximum number of consecutive working weekends
    - Maximum number of working weekends in a 4-week period
    - Maximum number of assignments on bank holidays
    - Restriction on the succession of shift types
    - Patterns enabling specific cyclic constraints (e.g. a free Wednesday afternoon every 2 weeks)
    - Balancing the workload among personnel
3. When individual personnel members have an agreement with the personnel manager or head nurse, then certain constraints can be implemented. Examples include:
    - Day off; shifts off
    - Requested assignments

- Tutorship (people not allowed to work alone)
- People not allowed to work together

For more details about these soft constraints, we refer to [50].

4 Solution Framework

4.1 The Evaluation of Solutions

The search heuristics for solving the ANROM model are driven by an evaluation function that estimates the quality of schedules. Since hard constraints have to be satisfied at all costs, only soft constraint violations contribute to that quality. In this section, we briefly introduce the method that was developed to model and evaluate the complex time-related constraints. It obviously takes care of boundary constraints imposed by previous planning periods.
The evaluation makes use of a simple algorithm and requires very little memory and computation time. This is useful for evaluating intermediate solutions while exploring the large search space. Since hard constraints have to be satisfied at all costs, only soft constraint violations contribute to the quality of a solution. The procedure tackles the characteristics of the soft constraints (cost parameters, tolerable deviations, restrictions on consecutiveness) in a modular way. It is easily extendible and provides a very structural technique for incorporating new constraints that appear in real world problems. The modular nature of the approach allows the system to provide some feedback. This functionality assists the user with the interpretation of the quality of the result.

4.2 Planning Procedures

The general meta-heuristics for ANROM are supplied with a few planning options for varying objectives. They have been developed as separate procedures in order to split typical hospital goals from general search algorithms. Fig. 3 schematically demonstrates the order in which these planning procedures (presented in bold) appear in the total planning process.
Boxes that are on the same level indicate alternatives. There is one general box for meta-heuristics because every algorithm that respects the hard constraints can be plugged into the framework.

4.2.1 Consistency check on available people

When the hard constraints are so strict that no feasible solution exists, planners can opt to relax them. The planning system is not developed for handling infeasible problems. In most cases, the hard constraints are so strong that it is obvious, after a preliminary check, that some of the soft constraints cannot be satisfied.
The procedure that we developed can handle shift type as well as time interval requirements. Apart from an obvious check on the hard constraints, the users

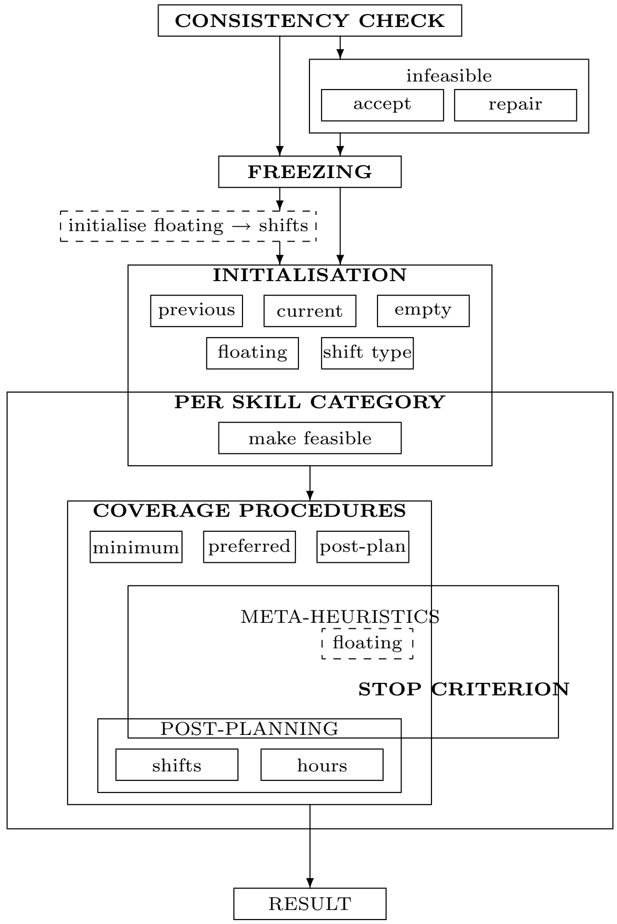

Figure 3: Overview of the solution framework

demanded an extra check on some 'precedence' soft constraints, namely on patterns, personal requests for days and shifts off, and requested shift assignments. Either the user can accept the remedy by relaxing the personnel requirements when necessary or he/she can deliberately choose to violate the soft constraints that were checked. Both options are demonstrated in the overview of Fig. 3.

4.2.2 Freezing parts of the schedule

The search space for rostering algorithms can be restricted for several reasons. We distinguish between 3 arguments to prevent algorithms from modifying certain parts of the schedule:

- Some hospitals prefer interaction and they schedule the timetables of certain nurses manually.
- The start and end date of personal contracts do not necessarily coincide

with the start and end times of the planning period. Also, people can temporarily be assigned to other wards.
- In case urgent rescheduling is required, it is recommended not to alter already existing personal schedules completely but to freeze certain periods in time.

No matter which parts of the schedule are frozen, the evaluation procedure considers the entire schedule.

4.2.3 Initialisation

The initialisation of the scheduling algorithm consists of two phases for constructing a feasible initial solution. In the first phase, the input is loaded (after a coverage procedure has been selected). Planners can make a choice among a random initial schedule and an initial schedule that is copied from the previous or current planning period (if that exists).
The second phase makes the schedule feasible by randomly adding or removing assignments until the hard constraints are satisfied [19].

4.2.4 Planning order of skill categories

Each skill category is scheduled separately in this model. Dividing the rostering into sub problems reduces the search space. The number of personnel belonging to each skill category is often considerably smaller than the entire staff in the ward. The number of shifts to be assigned (whether or not translated from floating requirements) is also lower.
In the case when people have permission to carry out shifts for alternative skill categories, there are a few difficulties. After the planning for a skill category has stopped, the algorithm moves on to the next skill category and temporarily freezes the already assigned shifts.

4.2.5 Time interval requirements

Time interval requirements provide an alternative way of expressing the coverage needs. This idea was put forward by users of the initial version of the model. It is now part of the solution framework and it can be combined with any of the algorithms.

Hospitals define a large number of shift types that match different activities and enable many kinds of part time work. Personal schedules are always set up with shift types although it is quite complex for planners to express the daily personnel coverage in terms of shift types. With this new option, we concentrate on an advanced formulation of requirements in terms of intervals of personnel coverage. It often meets the habits and needs that occur in practice more appropriately, it reduces non-productive attendance at work and it results in much more flexible timetables by splitting and recombining shift types. Details of this model for formulating coverage constraints can be found in [12].

4.2.6 Coverage Procedures

In practice, the number of required personnel on a certain day is not completely strict. Experienced planners know very well when it is reasonable to plan more or less personnel than is supposedly *required*. However, there exist no clear rules for decisions like this. Planners using ANROM can optionally choose among different coverage strategies. We call some of them post-planning algorithms. Examples include:

- **Minimum - preferred requirements.** Each of these two kinds of requirement can be set as hard constraints for the search algorithms.
- **Plan towards preferred requirements.** Instead of strictly setting the hard constraints, this option allows a range in which the hard constraints are considered to be satisfied. The algorithm to organise this option first takes the minimum requirements as hard constraints. After a result has been calculated by the scheduling algorithms, the system searches possibilities for adding shifts to the schedule whenever this does not involve an extra violation of soft constraints.

  When personnel requirements are expressed as a number of personnel needed per time interval (and not per shift type, as is mostly the case), a slightly different approach is required [12]. Instead of searching the best candidate to assign an extra shift or an extra pair of shifts for, we have to add switches from a shift to a longer shift, from a pair of shifts to a single shift with a longer entire duration or from a single shift to a pair of shifts that last longer (see [12]).
- **Adding hours.** In the personal schedules that do not have enough hours assigned, this planning option assigns extra shifts. The only condition is that the preferred requirements for the new shift types are not zero and that the extra assignment does not generate additional violations.

5 Meta-heuristics and Hybrids

5.1 Variable Neighbourhood Search

In this section, we present a method that applies the problem characteristics to dynamically modify the environments of the search heuristics. The exploration of the search space can be improved by combining short sighted search with greedy search in a wider environment. The main concepts of the variable neighbourhood approach are explained in [33].

5.1.1 Environments for personnel rostering

The meta-heuristics that we developed to solve nurse rostering problems change neighbourhoods when they cannot find better solutions during a number of iterations. All the solutions in an environment must satisfy the hard constraints. Although the cost function is the driving force for the heuristics, it may remain blind for improvements if it cannot interpret certain problem characteristics.

A few of the environments were designed only for the purpose of finding such improvements. Examples include:

- **Single shift-day neighbourhood.** The simplest environment contains all solutions that differ from the current solution by moving one assignment to another person's schedule. Fig. 4 illustrates this neighbourhood for an example with 4 nurses and 4 different shift types. Nurse A can replace the head nurse (HN), the other 2 regular nurses cannot. The arrows in the figure show the only possible moves that do not violate the hard constraints. Moving the head nurse's D shift to nurse B does not result in a feasible solution. Neither does moving nurse A's E shift to the schedule of the head nurse.

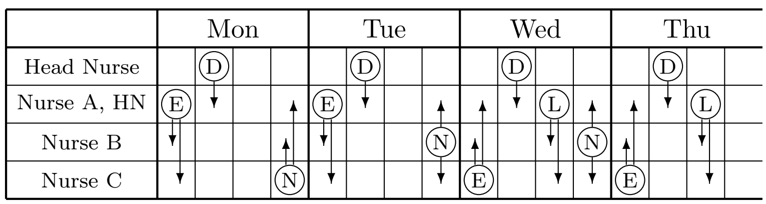

Figure 4: Possible moves in a 'single shift-day' neighbourhood. Each day is divided into columns that represent the shifts. Shifts are Early (E), Day (D), Late (L), and Night (N)

- **Soft constraint related neighbourhoods.** Planners with practical experience were often more concerned about violations of certain soft constraints than about the overall solution quality. This resulted in a particular set of environments that consider the search space in a completely different way. They are blind for the global effect of a move but attempt to ameliorate the quality of a roster by only considering the soft constraints that are being addressed. Of course, this approach is completely opposed to the idea of the general evaluation function that abstracts from individual constraints. Therefore, the entire search procedure should not end immediately after exploring this neighbourhood.
  Examples of such environments look at weekend constraints, overtime, alternative qualifications, personal preferences, etc. A special neighbourhood only looks at the most violated constraint (no matter which constraint that is).
- **Swapping large sections of personal schedules.** Unlike the previous two, in which neighbouring solutions only differ in the position of one single shift type, this set of neighbourhoods looks at schedules that differ substantially from the current solution.
  The 'shuffle' environment considers switches from a part of the worst personal schedule, in terms of the evaluation function, with any other schedule. Instead of moving single shift types, all the assignments in a longer period (that can be one day or a number of days that equals half the plan-

ning period) are switched. A similar environment is called the 'greedy shuffle'. It consists of all the possible shuffles between any set of two people.
Reallocating larger groups of shifts at once is often less harmful for the quality of a schedule than moving shifts around. The drawback of applying this category of neighbourhoods is that the number of neighbouring solutions is very large, and iterations thus require a large amount of computation time.

5.1.2 Algorithms and search order

Different scenario's are possible in variable neighbourhood search. For the experiments that we carried out in [13], we only applied steepest descent and tabu search as local search algorithms. The results mainly provided insight into the characteristics of the search space and into the importance of the order in which to explore neighbourhoods. We discovered that the most efficient application of different neighbourhoods is to search them in order of increasing size (i.e. starting from the single shift-day environment and ending with swapping large sections). Each time a better solution is found, the search procedure restarts from the smallest size neighbourhood.
Some large size environments take the search procedure to a kind of end state. The resulting roster is (in general) very satisfactory, especially since it is nearly impossible for experienced planners to make manual improvements. This explains why it is not useful to modify such end states by applying another small size neighbourhood afterwards.

5.2 Hybrid Tabu Search

5.2.1 Tabu search for personnel scheduling

Hybrid tabu search algorithms are integrated in the software package that is based on ANROM (see [13, 14] for more details). One of the major benefits of tabu search is its ability to escape from local optima, by preventing the search from going through the same areas. As long as there are improvements, tabu search behaves like a steepest descent algorithm. There are aspiration criteria to prevent better solutions in a 'tabu' part of the search space from being ignored. The efficiency of the search can be increased by applying diversification.
The basic algorithm is the combination of tabu search with the single shift-day neighbourhood that was explained in Section 5.1. Certain characteristics of encountered solutions (and similar solutions in the neighbourhood) are put into the tabu list in order not to circle round.
Instead of carrying out random diversifications when the basic algorithm cannot generate better solutions during a number of iterations, we opted for implementing more problem specific steps. The different neighbourhoods that we introduced above are useful in this respect. They are particularly useful when considering the following:

- the soft constraint on full weekends
- improving the worst personal roster (shuffle)
- all the large improvements between two personal rosters (the greedy shuffle environment).

5.2.2 Hybrid algorithms for practical use

With the two following algorithms, we offer planners a choice between a fast algorithm that generates schedules of acceptable quality in a very short amount of time, and a more thorough one that is deployed when a final solution is required. The course of both options is illustrated in Fig. 5. That diagram fits into the meta-heuristics box of Fig. 3.

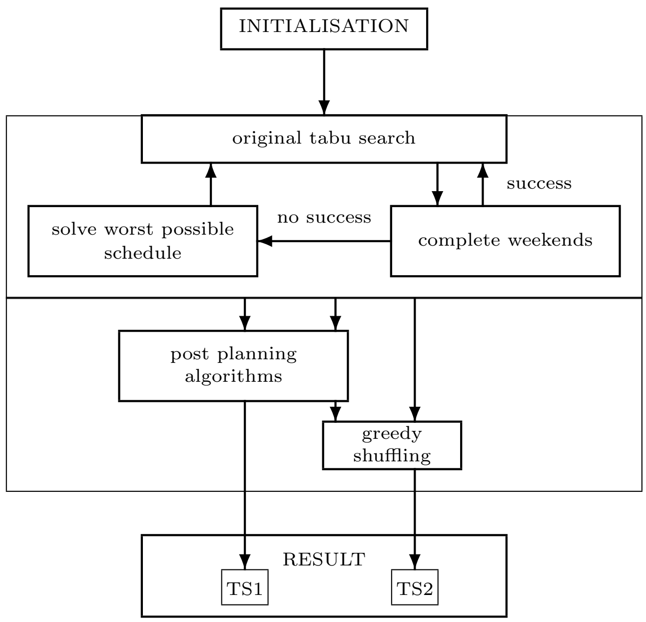

Figure 5: Diagram of the hybrid tabu search algorithms for the nurse rostering problem, plug-in for Fig. 3

We consider two versions of hybrid tabu search (see [13, 14] for more details):

- **TS1: Fast planning.** When the basic tabu search algorithm does not find improvements, the diversification consists of searching in the weekend environment, and after that in the shuffle environment. This process is iteratively repeated until the global stop criterion is reached.
- **TS2: Thorough planning** (TS1 + greedy shuffle). This combination requires more time but the results are of a considerably higher quality. Especially since the greedy step is carried out at the end of the calculations (and not iteratively in the process), the computation time is kept

under control. The confidence of the users is high because all the manual modifications make the obtained solution worse.

Hybrid algorithms (compared to single tabu search) are much better suited for the personnel rostering problems that we tackle. We refer to [14] for test results on real world problems. Application in practice reveals that the increased quality of solutions generated with TS2 compensates for the large amount of computation time required.

5.3 Memetic Algorithms

Population based techniques can overcome problems that occur when a single solution 'evolves'. We briefly introduce a set of memetic algorithms for personnel rostering, that make use of cross-over (in order to copy good characteristics of previous generations) and mutation (for random diversity). Memetic algorithms apply local search on every individual of a population. The computation time is large but the results demonstrate that memetic algorithms offer more to rostering than the sum of the component algorithms (see [9] for more details).

5.3.1 Evolutionary approaches for personnel rostering

It is quite complex to apply cross-over operators to ANROM because the standard operators do not maintain feasibility. Moreover, the identification of good roster features is particularly difficult since the nature of the constraints hardly allows any partial evaluations. Combining parts of nurse schedules seldom results in good quality. Fig. 6 demonstrates how the memetic algorithms for ANROM are composed. The diagram needs to be plugged into Fig. 3 in order to see the full process.

An initial population consists of N individual schedules that match the hard constraints. The quality of a roster is determined by the sum of the qualities of all individual rosters. Therefore, some cross-over operators will copy full personal rosters to the next generations. Others will only consider good assignments. In any case, nearly all the operators require repair functions that make the newly created individual solutions feasible. The procedure that was introduced in Section 4.2 to make initial schedules feasible, is also applicable here. We briefly discuss a few algorithmic variants:

- The **Simple Memetic Algorithm** applies steepest descent (in the single shift-day neighbourhood from Section 5.1) on each newly created individual. The planning order of qualifications (Section 4.2) remains unchanged. A simple tournament selects the best parents. The first child is obtained by copying the best personal schedule from the first parent, together with the best one (for another member of staff) from the other parent. A personal schedule is simply the work schedule for one particular person. All the other personal schedules are copied randomly from both parents. For the second child, we start from the best solution of the other parent. Both

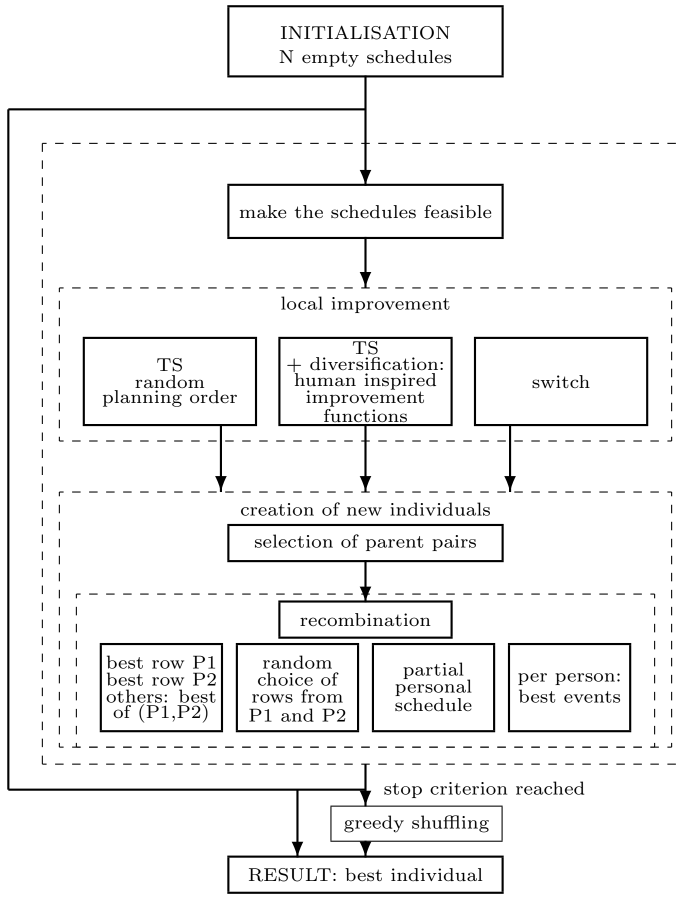

Figure 6: Diagram of the components of the genetic and memetic algorithms for the nurse rostering problem, (plug-in for Fig. 3)

new individuals are normally infeasible but they are repaired by randomly adding or removing assignments until the coverage constraints are met.

- The **Diverse Memetic Algorithm** is based on the Simple Memetic Algorithm but the planning order of skill categories is randomly chosen in the local search step.
- **Diverse Memetic Algorithm with Random Selection** applies the general features of the Diverse Memetic Algorithm but all the personal schedules are randomly selected from both parent pairs.
- **Memetic Algorithm with Cross-over.** A random assignment unit is generated for each person in the schedule and at that point, personal schedules from 2 parents are combined. Again, the schedules are made feasible and the steepest descent algorithm is carried out on each individ-

ual.

- **Memetic Algorithm that Copies the Best x Assignments.** For every personal schedule, the algorithm selects the assignments that induce the highest increase in violations when removed. In the case when the best assignments are the same in both parents, only x are copied to the next generation. The diversity is obtained by randomly making the schedules feasible. Afterwards, the local search algorithms improve the individuals.

5.3.2 A combination of tabu search and evolutionary algorithms

We have explored various algorithmic variants of combining tabu search and evolutioary methods (see [9] for more details). Approaches we have considered include:

- **Multistart TS with Random Planning Order** enables a direct comparison between tabu search, memetic algorithms and hybrids. Finally, the best solution is improved with the greedy shuffle.
- **Memetic Algorithm with Improved Local Search.** This option applies the TS2 algorithm from Section 5.2 to the best solution obtained by the Memetic Algorithm that Copies the Best 4 Assignments described in Section 5.3.1. The value x=4 turned out to produce the best results for a wide range of problems.
- **Switch.** All the previously introduced algorithms satisfy the user determined hard constraints but this approach accepts all solutions that remain within the minimum-preferred requirements interval. New generations are created by adding or removing assignments (within the feasible region) in the parent schedules.

5.3.3 Results

Copying entire parts of parent schedules (the Diverse Memetic Algorithm with Random Selection, the Memetic Algorithm with Cross-over, the Simple Memetic Algorithm and Simple Memetic Algorithm and the Diverse Memetic Algorithm) turns out not to be efficient at all. Steepest descent does individuals into acceptable solutions. In contrast, by copying very tiny partial schedules with good qualities (as in the MEx algorithms), we discovered that there was a considerably higher diversity and more freedom to improve the *children* with local search. We may assume that the best x assignments of a schedule influence the rest so much that solutions evolve towards good quality schedules.
Multistart TS with Random Planning Order cannot compete with the real memetic algorithms. This proves that longer computation time and variance in initial solutions does not lead to the quality that a hybrid approach can reach. The memetic-tabu hybrid Memetic Algorithm that Copies the Best x Assignments behaves extremely well for very complex problems (see [9] for a thorough discussion of the test results). It outperforms all the other algorithms, except Switch, and clearly demonstrates the benefits of hybrid approaches. Switch, which is a little bit less strict about the selected hard constraints, provides very

interesting results. It demonstrates that relaxing the hard constraints, to a controlled extent, is often not as harmful as expected [9].

6 Multi Criteria Approach

Users of the planning software that we have described [12, 14, 19] are confronted with the difficult task of expressing their needs in terms of the set of constraints and corresponding cost parameters. We have developed a multi criteria approach, which attempts to overcome some practical difficulties that planners deal with [10].

The evaluation function sums violations of constraints that are completely different. Yet, it is not desirable that hospital planners have to compare all these different measures. The lack of insight into the characteristics of the solution space often leads to very poor parameter settings. Experiments teach us, for example, that higher values of cost parameters can give rise to high barriers in the search space and make it even more difficult to find better solutions.

Multi criteria modelling allows users to express the importance of criteria according to their own preferences. The approach bears more resemblance to the customs in hospitals than the cost function approach, and it supports better control of compensation between constraints.

Nurse rostering problems are generally unsolvable in terms of satisfying all the constraints that planners initially state are hard - they usually have to 'soften' some of these constraints. There is considerable scope for the uptake of multi-objective methods which promote a reasonable approach for searching for a compromise between conflicting objectives.

6.1 A multi criteria model for nurse rostering

We implemented a multi criteria approach for nurse rostering in which the violations of one soft constraint are measured with one criterion. Constraints of a very different nature, that are expressed in different measures, can now be treated simultaneously (see [10] for more details).

We apply 'compromise programming', which is based on the concept of the distance to an ideal point. Each schedule is represented as a point in the criteria space. The number of soft constraints determines the dimensions of the criteria space. For real world problems, solutions that correspond to the ideal point do not usually exist. The anti-ideal point is represented by a schedule with the worst value for all the criteria. In order to treat all the criteria in dimensionless units, a preference space is created (Fig. 7). The model requires a best and a worst value for each of the criteria. It is acceptable to state that the value 0 (denoting no violations of the corresponding soft constraint), matches the best value, even if that is not feasible. Estimating the worst value of criteria is more complex. We opted for generating the most extreme values instead of determining realistic worst values in the feasible region. The obtained worst value for each criterion is mapped to the relative priority or the weight (w in Fig. 7) as-

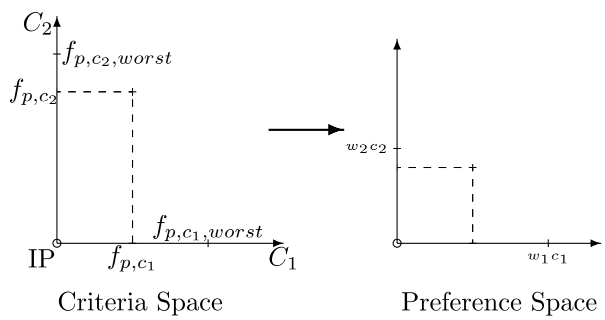

Figure 7: Mapping from the criteria space into the preference space for person p's schedule

signed to the constraint. A smaller distance to the ideal point in the preference space denotes a *better* schedule.

Each personal roster is evaluated separately by measuring its distance to the ideal point. We refer to [10] for more details about the metrics (called L_p) that we used. Small values of p allow for compensation between criteria. Good values for a criterion can make up for a worse evaluation of another one. By increasing the distance measure p to ∞ only the value of the largest co-ordinate contributes to the quality.

The sum of the distances of all the personnel members determines the quality of the entire schedule. For the experiments that we undertook, we applied the previously described meta-heuristics to generate better solutions. The evaluation function is replaced by the measurement of the distance to the ideal point. The meta-heuristics ensure that the search does not leave the feasible part of the search space.

6.2 Results

The proposed multi criteria approach is very promising for nurse rostering. Conclusions are drawn from experiments on real world test problems and are reported in [10]. The approach is particularly appropriate for treating completely different criteria. Users determine the weights for each of the soft constraints and thus set relative priorities. They can control compensation of these constraints by manipulating the distance measure.

In general, this multi criteria approach has the potential to more accurately reflect the daily real world situation in hospitals than the methods that employ a single evaluation function.

7 Conclusions

Nurse rostering is a very complex combinatorial problem, for which software assistance is not common (at least in Belgium). During our work it became clear that, especially in Belgian hospitals, any assistance for the head nurses or ward managers to automatically generate their monthly rosters could save a lot of time and effort. Several levels of decision making can be distinguished in nurse scheduling but the problem dealt with in this model is situated at the short-term timetabling level. Its main objective is to understand and automatically generate comfortable shift schedules for personnel members in order to meet the staff coverage. We captured an extensive set of realistic constraints, and integrated them, together with explicit and implicit objectives, in a general, flexible model.

The development of the solution framework, with modifiable evaluation tools and a large set of heuristics targeting specific objectives, constitutes a major improvement for hospital schedulers. However, the model often requires more insight into the characteristics of specific data than planners in practice can be expected to have. Some of the planning procedures already assist in setting feasible hard constraints or relaxing them when necessary. It will be beneficial for the model to elaborate on the consistency check and to take the idea of relaxing the rather strict distinction between hard and soft constraints further.

A set of meta-heuristics and hybrids are included in that solution framework, as the central search force for solving nurse rostering problems. We gained insight into the behaviour of applying heuristics and in making use of different problem specific neighbourhoods. The nurse rostering package based upon ANROM has become commercially available and is used in many wards in Belgian hospitals. This overview paper has summarised much of our research work in this area over recent years. A more detailed analysis of our approaches can be found in [9, 10, 11, 12, 13, 14, 19].

Future research will certainly build upon the promising early findings of testing our multi criteria approach on nurse rostering. It opens perspectives for releasing the planners from setting the cost parameters. It is more realistic and increases the flexibility in setting the weights and thus modifying the relative priority of the constraints.

By automating the nurse rostering problem for Belgian hospitals, the scheduling effort and time are reduced considerably as compared to the manual approach that was previously used. The time for automatic schedule generation can be tailored to suit the time available by selecting appropriate search heuristics. The proposed solution method provides an unbiased way of generating the schedules for all the personnel members. It enables simple verification of the constraints, helps redefine unrealistic hard constraints and thus leads to an overall higher satisfaction among the personnel, as is manifest in many applications.

Although the nurse rostering model was developed explicitly to address hospital personnel rostering, the techniques and methods developed as a result of this research are certainly adaptable to other personnel scheduling problems. Of course the presented algorithms deal with an extensive set of soft constraints, of which many are only valid in healthcare. Moreover, other sectors require the evaluation of constraints on locations, equipment, etc that are irrelevant in nurse rostering.

References

[1] U. Aickelin, K. Dowsland: Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem, Journal of Scheduling, Volume 3 Issue 3, 2000, 139-153

[2] J. Arthur, A. Ravindran: A Multiple Objective Nurse Scheduling Model, AIIE Transactions, Vol. 13, No. 1, 1981, 55-60

[3] S.E. Bechtold, L.W. Jacobs: Implicit modeling of flexible break assignments in optimal shift scheduling, Management Science 36, 1990, 1339-1351

[4] G.R. Beddoe, S. Petrovic, G. Vanden Berghe: Case-based reasoning in employee rostering: learning repair strategies from domain experts, working paper at the University of Nottingham

[5] I. Berrada, J. Ferland, P. Michelon: A Multi-Objective Approach to Nurse Scheduling with both Hard and Soft Constraints, Socio-Economic Planning Science 30, 1996, 183-193

[6] R. Blau: Multishift Personnel Scheduling with a Microcomputer, Personnel Administrator, Vol. 20, No. 7, 1985, 43-58

[7] D. Bradley, J. Martin: Continuous Personnel Scheduling Algorithms: A Literature Review, Journal of the Society of Health Systems 2, 1990, 8-23

[8] M.J. Brusco, L.W. Jacobs: Cost analysis of alternative formulations for personnel scheduling in continuously operating organisations, European Journal of Operational Research 86, 1995, 249-261

[9] E.K. Burke, P. Cowling, P. De Causmaecker, G. Vanden Berghe: A Memetic Approach to the Nurse Rostering Problem, Applied Intelligence special issue on Simulated Evolution and Learning, Vol. 15, Number 3, Springer, 2001, 199-214

[10] E.K. Burke, P. De Causmaecker, S. Petrovic, G. Vanden Berghe: A Multi Criteria Meta-heuristic Approach to Nurse Rostering, Proceedings of Congress on Evolutionary Computation, CEC2002, Honolulu, IEEE Press, 2002, 1197-1202

[11] E.K. Burke, P. De Causmaecker, S. Petrovic, G. Vanden Berghe: Fitness Evaluation for Nurse Scheduling Problems, Proceedings of Congress on Evolutionary Computation, CEC2001, Seoul, IEEE Press, 2001, 1139-1146

[12] E.K. Burke, P. De Causmaecker, S. Petrovic, G. Vanden Berghe: Floating Personnel Requirements in a Shift Based Timetable, working paper KaHo Sint-Lieven, 2003

[13] E.K. Burke, P. De Causmaecker, S. Petrovic, G. Vanden Berghe: Variable Neighbourhood Search for Nurse Rostering Problems, in Metaheuristics: Computer Decision-Making (edited by Mauricio G.C. Resende and Jorge Pinho de Sousa), Chapter 7, Kluwer, 2003, 153-172

[14] E.K. Burke, P. De Causmaecker, G. Vanden Berghe: A Hybrid Tabu Search Algorithm for the Nurse Rostering Problem, B. McKay et al. (Eds.): Simulated Evolution and Learning, 1998, Lecture Notes in Artificial Intelligence, Vol. 1585, Springer, 1999, 187-194

[15] P. Chan, G. Weil: Cyclical Staff Scheduling Using Constraint Logic Programming, E.K. Burke, W. Erben (Eds.): Practice and Theory of Automated Timetabling, Third International Conference, Konstanz, Springer, 2000, 159-175

[16] J.-G. Chen, T. Yeung: Hybrid Expert System Approach to Nurse Scheduling, Computers in Nursing, 1993, 183-192

[17] M. Chiarandini, A. Schaerf, F. Tiozzo: Solving Employee Timetabling Problems with Flexible Workload using Tabu Search, E.K. Burke, W. Erben (Eds.): Proceedings of the 3rd international conference on the Practice and Theory of Automated Timetabling, ISBN 3-00-003866-3, 2000, 298-302

[18] S.J. Darmoni, A. Fajner, N. Mahe, A. Leforestier, O. Stelian, M. Vondracek, M. Baldenweck: Horoplan: computer-assisted nurse scheduling using constraint based programming, Journal of the Society for Health Systems, Vol.5, 41-54, 1995

[19] P. De Causmaecker, G. Vanden Berghe: Relaxation of Coverage Constraints in Hospital Personnel Rostering, E.K. Burke, P. De Causmaecker (Eds.), Selected papers of 4th International Conference on Practice and Theory of Automated Timetabling, LNCS (accepted for publication)

[20] G. de Vries: Nursing workload measurement as management information, European Journal of Operational Research 29, 1987, 199-208

[21] K. Dowsland: Nurse scheduling with Tabu Search and Strategic Oscillation. European Journal of Operations Research 106, 1998, 393-407

[22] F. Easton, N. Mansour: A Distributed Genetic Algorithm for Employee Staffing and Scheduling Problems, Conference on Genetic Algorithms, San Mateo, 1993, 360-367

[23] F. Easton, D. Rossin, W. Borders: Analysis of Alternative Scheduling Policies for Hospital Nurses, Production and Operations Management, Vol.1, No. 2, 1992, 159-174

[24] R. Hung: A Cyclical Schedule of 10-Hour, Four-Day Workweeks, Nursing Management, Vol 22, No 5, 1991, 30-33

[25] M. Isken, W. Hancock: A Heuristic Approach to Nurse Scheduling in Hospital Units with Non-Stationary, Urgent Demand, and a Fixed Staff Size, Journal of the Society for Health Systems, Vol. 2, No. 2, 1990, 24-41

[26] A. Jaszkiewicz: A metaheuristic approach to multiple objective nurse scheduling, Foundations of Computing and Decision Sciences, Vol. 22, No. 3, 1997, 169-184

[27] H. Kawanaka, K. Yamamoto, T. Yoshikawa, T. Shinogi, S. Tsuruoka: Genetic Algorithm with the Constraints for Nurse Scheduling Problem, Proceedings of Congress on Evolutionary Computation, Seoul, IEEE Press, 2001, 1123-1130

[28] M. Kostreva, K. Jennings: Nurse Scheduling on a Microcomputer, Computers and Operations Research 18, 1991, 731-739

[29] A. Meisels, E. Gudes, G. Solotorevski: Employee Timetabling, Constraint Networks and Knowledge-Based Rules: A Mixed Approach, E.K. Burke, P. Ross (Eds.): Practice and Theory of Automated Timetabling, First International Conference Edinburgh, Springer, 1995, 93-105

[30] H. Meyer auf'm Hofe: ConPlan/SIEDAplan: Personnel Assignment as a Problem of Hierarchical Constraint Satisfaction, Proceedings of the Third International Conference on the Practical Application of Constraint Technology, London, 1997, 257-271

[31] H.E. Miller: Implementing Self Scheduling, The Journal of Nursing Administration, Vol. 14, March 1984, 33-36

[32] H.E. Miller, W. Pierskalla, G. Rath: Nurse Scheduling Using Mathematical Programming. Operations Research 24, 1976, 857-870

[33] N. Mladenović, P. Hansen: Variable Neighborhood Search, Computers & Operations Research, Vol. 24, 1997, 1097-1100

[34] A. Musa, U. Saxena: Scheduling Nurses Using Goal-Programming Techniques, IEEE, 1984, 216-221

[35] N. Muslija, J. Gaertner, W. Slany: Efficient Generation of Rotating Workforce Schedules, E.K. Burke, W. Erben (Eds.): Proceedings of the 3rd international conference on the Practice and Theory of Automated Timetabling, ISBN 3-00-003866-3, 2000, 314-332

[36] M. Okada: An approach to the Generalised Nurse Scheduling Problem - Generation of a Declarative Program to represent Institution-Specific Knowledge. Computers and Biomedical Research 25, 1992, 417-434

[37] I. Ozkarahan: A Disaggregation Model of a Flexible Nurse Scheduling Support System, Socio-Economical Planning Science, Vol. 25, No. 1, 1991, 9-26

[38] I. Ozkarahan: An Integrated Nurse Scheduling Model, Journal of the Society for Health Systems, Vol. 3, No. 2, 1991, 79-101

[39] S. Petrovic, G. Beddoe, G. Vanden Berghe: Storing and adapting repair experiences in personnel rostering, E.K. Burke, P. De Causmaecker (Eds.), Selected Papers of the 4th International Conference on Practice and Theory of Automated Timetabling, LNCS (accepted for publication)

[40] K.K. Ringl, L. Dotson: Self-Scheduling for Professional Nurses, Nursing Management, Vol. 20, 1989, 42-44

[41] A. Schaerf, A. Meisels: Solving employee timetabling problems by generalised local search, Proceedings Italian AI, 1999, 493-502

[42] D. Schneider, K. Kilpatrick: An Optimum Manpower Utilization Model for Health Maintenace Organisations, Operations Research, Vol. 23, No 5, 1975, 869-889

[43] S.P. Siferd, W.C. Benton: Workforce staffing and scheduling: Hospital nursing specific models, European Journal of Operational Research 60, 1992, 233-246

[44] D. Sitompul, S. Randhawa: Nurse Scheduling Models: A State-of-the-Art Review, Journal of the Society of Health Systems 2, 1990, 62-72

[45] L.D. Smith, D. Bird, A. Wiggins: A Computerised System to Schedule Nurses that Recognises Staff Preferences, Hospital & Health Service Administration, 1979, 19-35

[46] L.D. Smith, A. Wiggins: A Computer-Based Nurse Scheduling System, Computers and Operations Research, Vol. 4, No. 3, 1977, 195-212

[47] V. Smith-Daniels, S. Schweikhart, D. Smith-Daniels: Capacity Management in Health Care Services: Review and Future Research Directions, Decision Sciences, Vol.19, 1988, 891-919

[48] J. Tanomaru: Staff Scheduling by a Genetic Algorithm with Heuristic Operators, Proceedings of the IEEE Conference on Evolutionary Computation, 1995, 456-461

[49] V.M. Trivedi, M. Warner: A branch and bound algorithm for optimum allocation of float nurses, Management Science vol. 22, No 9, 1976, 972-981

[50] G. Vanden Berghe: An Advanced Model and Novel Meta-heuristic Solution Methods to Personnel Scheduling in Healthcare, PhD dissertation, University of Gent, 2002

[51] M. Warner: Nurse Staffing, Scheduling, and Reallocation in the Hospital, Hospital & Health Services Administration, 1976, 77-90

[52] M. Warner: Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Programming Approach. Operations Research 24, 1976, 842-856

[53] M. Warner, J. Prawda: A Mathematical Programming Model for Scheduling Nursing Personnel in a Hospital, Management Science 19, 1972, 411-422

[54] G. Weil, K. Heus, P. Francois et al.: Constraint Programming for Nurse Scheduling, IEEE Engineering in Medicine and Biology, 1995, 417-422

[55] H. Wolfe, J.P. Young: Staffing the Nursing Unit: Part I, Nursing Res. 14 (3), 1965, 236-234

[56] H. Wolfe, J.P. Young: Staffing the Nursing Unit: Part II, Nursing Res. 14 (4), 1965, 299-303