Ann Oper Res (2012) 194:33–57
DOI 10.1007/s10479-010-0804-0

Local search neighbourhoods for dealing with a novel nurse rostering model

Burak Bilgin · Patrick De Causmaecker · Benoît Rossie · Greet Vanden Berghe

Published online: 3 November 2010

Abstract A novel nurse rostering model is developed to represent real world problem instances more accurately. The proposed model is generic in the sense that it allows modelling of essentially different problem instances. Novel local search neighbourhoods are implemented to take advantage of the problem properties represented by the model. These neighbourhoods are used in a variable neighbourhood search and in an adaptive large neighbourhood search algorithm. The performance of the solution method is evaluated empirically on real world data. The proposed model is open to further extensions for covering personnel planning problems in different sectors and countries.

Keywords Nurse rostering · Hospital personnel planning · Variable neighbourhood search

1 Introduction

The nurse rostering problem, which constitutes the assignment of shifts to nurses according to several criteria, is a complex personnel planning problem (Burke et al. 2004b). The problem becomes more complicated in Belgian hospitals, where the planning periods are flexible, problem elements like shift types and skill types are user defined, legal restrictions and contractual agreements impose complex constraints and where cyclic assignments are

B. Bilgin (✉) · G. Vanden Berghe
Information Technology, KaHo Sint-Lieven, Gebroeders Desmetstraat 1, 9000 Gent, Belgium
e-mail: Burak.Bilgin@kahosl.be

G. Vanden Berghe
e-mail: Greet.VandenBerghe@kahosl.be

P. De Causmaecker
Computer Science and Information Technology, K.U. Leuven Campus Kortrijk, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
e-mail: Patrick.DeCausmaecker@kuleuven-kortrijk.be

B. Rossie
Products Division, SAGA Consulting NV, Weiveldlaan 41 bus 35, 1930 Zaventem, Belgium

not the standard practice (Burke et al. 2001a, 2004a; De Causmaecker and Vanden Berghe 2003). Furthermore, the problem has a dynamic nature due to the ever changing labour legislation, contractual agreements, and nurse preferences. New aspects and constraint types are introduced in the course of time. The models and solution methods of the automation tools have to comply with the latest state of the problem.

1.1 Literature review

Nurse rostering problems from several countries such as Belgium (Burke et al. 2001a, 2001b), the Netherlands (Brucker et al. 2005; Burke et al. 2008), the United Kingdom (Aickelin and Li 2007; Burke et al. 2003), the United States (Bard and Purnomo 2007), Norway (Frøyseth et al. 2008), and Italy (Bellanti et al. 2004) have been introduced in the scientific literature. For investigating the problem systematically, synthetic nurse rostering data have been created and examined in the nurse rostering literature (Maenhout and Vanhoucke 2008; Özcan 2007). The properties of the problems reported in the nurse rostering literature vary. De Causmaecker and Vanden Berghe (2009) have proposed a reference model to categorise the timetabling and rostering problems according to their properties such as the personnel environment, work characteristics, and optimisation objective. The categorisation of the problems will help researchers to study the complexity and hardness of the problem instances and the efficiency of the corresponding algorithms.

Numerous artificial intelligence and operations research approaches have been applied to the nurse rostering problem. These approaches include integer programming by Glass and Knight (2010), scatter search by Burke et al. (2009), a tabu search hyperheuristic by Burke et al. (2003), several hybrid methods by Bellanti et al. (2004), Brucker et al. (2005) and Burke et al. (2008), problem specific cross-over methods in a genetic algorithm by Maenhout and Vanhoucke (2008), and a population based metaheuristic inspired by electromagnetism by Maenhout and Vanhoucke (2007). In order to compare the solution methods from the literature, Petrovic and Vanden Berghe (2008) have proposed seven comparison criteria: "expressive power, flexibility, algorithmic power, learning capabilities, maintenance, rescheduling capabilities, and parameter tuning."

A real world nurse rostering data set is maintained online by the ASAP group of the University of Nottingham (the Nottingham benchmarks for short) (Curtois 2009). The problem instances in the benchmark data set have been subject of the nurse rostering literature and they were gathered from different countries. The idea behind the maintenance of a public benchmark library is to allow researchers to test their solution methods on problems with different properties, and compare their results to the state of the art (Brucker et al. 2010).

Burke et al. (2009) have applied a scatter search algorithm to the nurse rostering problem. The parameters of the algorithm have been selected in a way to keep the execution time under 15 minutes on a desktop computer in order to satisfy the preferences of the end users. The algorithm has been tested on 20 problem instances from the Nottingham benchmarks. Experimental results have indicated that the scatter search variant that deploys time predefined variable depth search for improvement performs efficiently considering the short execution time.

Glass and Knight (2010) have solved four instances from the Nottingham benchmarks to optimality. They have used integer programming and search space reduction based on the properties of the problem instances. Glass and Knight (2010) have also pointed out the differences between the formulation of the problems in the Nottingham benchmarks and the real world practice. The difference concerns continuity handling between two planning periods in the formulation of the Nottingham benchmarks.

Bourdais et al. (2003) have grouped the constraints in the nurse rostering problem into rule categories. The fact that coverage constraints and the *maximum number of assignments per nurse per planning period* constraint are considered hard, enables the reduction of the solution space by inference prior to search. Another constraint programming approach has been introduced by Pesant (2008). It softens the constraints and at the same time performs inference on them. Constraint-centered search heuristics are introduced in the same paper. These heuristics direct the search procedure to the areas of the search space with higher probabilities of being intensely populated by solutions (Pesant 2008).

The nurse rostering problem in Belgian hospitals has been the subject of many research papers. The problem has been tackled with a variable neighbourhood search (VNS) that takes advantage of the synergy between simple and greedy neighbourhoods by Burke et al. (2004a) and a hybridisation of this algorithm with a memetic approach deploying a steepest descent heuristic by Burke et al. (2001a). The contribution of coverage constraint relaxation to the production of higher quality schedules is investigated by De Causmaecker and Vanden Berghe (2003). Beliën and Demeulemeester (2007) compare and discuss the performance and modelling capabilities of two different problem decomposition strategies within a branch-and-price algorithm. The problem instance tackled is a trainee scheduling problem, which differs from the nurse rostering problem.

1.2 Contribution

The motivation for this study is twofold. (1) We aim at developing a generic nurse rostering model for allowing a broad range of real world problems to be defined accurately. The proposed model has an increased ability to reflect extra soft constraints in different hospitals, sectors and countries. (2) We develop an associated solution method, which includes neighbourhoods that take advantage of different properties of the problem model. These neighbourhoods, when deployed among other traditional neighbourhoods in VNS, have performed better than the traditional neighbourhoods alone.

(1) Nurses belong to different skill categories according to their job descriptions, qualifications, experience, and responsibilities. Every nurse has a primary skill type. Some of the nurses have secondary skill types, which means they can work as substitutes in order to fulfill other skill requirements. The coverage constraints restrict the number of nurses with a specific skill type that should be present at a given date and a given shift type (Aickelin and Li 2007; Burke et al. 2004a). A new hard constraint imposes that only the assignments that are defined in the coverage constraints can be added to or deleted from the roster by the solution method (see Sect. 2.4.5). However, exceeding the minimum and maximum thresholds of the coverage constraints is allowed but penalised in the objective function as a soft constraint violation.

    The time related constraints, considered as soft constraints in the model, restrict the assignments to a specific nurse (Burke et al. 2004b). Constraints defined by the employment contracts of the nurses, called *horizontal constraints*, are a subset of the time related constraints. The *horizontal constraints* are organised in three general types in this model. These are *counters*, *series*, and *successive series*, each with their own subjects and parameter sets. This organization is more flexible than the approach in (Burke et al. 2004a), where soft constraints were predefined with all or most of their parameters and then applied to all problem instances. *Counter* constraints restrict the number of occurrences of a subject over a counter period, such as *hours worked*, *days worked*, and *shift types worked*. *Series* is a general term for describing the constraints that restrict the number of consecutive subjects, like *days worked*, *days idle*, and *weekends worked*. In this context, *successive series* is a new formulation. *Successive series* restrict the succession of two series. An early example of a *successive series* constraint is *assign two free days after night shifts* constraint in (Burke et al. 2004a). Another novel aspect of the model is the *compatible shift types* concept. Some of the soft constraints involving shift types are defined for a set of *compatible shift types* instead of for a single shift type as in (Burke et al. 2004a).

(2) The second part of the study is the investigation of associated solution methods. VNS is a metaheuristic that systematically switches between the neighbourhoods of a predefined neighbourhood set during the local search (Hansen and Mladenović 2003). Adaptive Length Neighbourhood Search (ALNS) is an iterative method that operates on a feasible solution. At each iteration, a destroy and a repair neighbourhood is selected according to a roulette wheel selection. The probabilities of the selection of a neighbourhood depend on the previous performance of the neighbourhood (Pisinger and Ropke 2007). Neighbourhoods based on the proposed model are developed for application within the VNS and ALNS algorithms. The proposed neighbourhoods make use of problem properties like compatible shift types and secondary skill types.

In the remainder of the paper, we describe the empirical investigation of the solution methods. The experimental setup and the processing of the results are aimed to be carried out according to the principles given by Schaerf and Di Gaspero (2007). The experiments are carried out on two sets of benchmark data. The instances in the first benchmark set, the KaHo nurse rostering benchmarks, are gathered from six wards in two Belgian hospitals through our industrial partner. For each problem instance of the KaHo nurse rostering benchmarks (KaHo benchmarks for short), the input data and a sample solution with penalty details are published online (Bilgin 2008). The differences between the problem definitions of the KaHo and Nottingham benchmarks make an experimental evaluation of our algorithms on the Nottingham benchmarks impractical. Four problem instances from the Nottingham benchmarks are studied to present the differences between both problem definitions (see Sect. 4.1). Experimental results are also provided on these instances. Different algorithm settings are experimentally evaluated on the benchmarks and the results are statistically tested.

The research is carried out with an industry partner[1] producing a nurse rostering assistance system.[2] This is a computer aided nurse rostering tool for the hospital planners that keeps track of the constraint violations. The implementations of the system are utilised in many Belgian hospitals. The automation tool developed within this research project is integrated in the nurse rostering assistance software of our industry partner. The resulting system is being deployed in Belgian hospitals.

The problem definition and the model are described in Sect. 2. The solution method is presented in Sect. 3. The experiments and the experimental results are discussed in Sect. 4. The paper is concluded in Sect. 5.

2 Problem definition and model

The objective of the nurse rostering problem is to assign shifts to nurses in accordance with workforce requirements, legal and contractual restrictions, personal preferences, and

[1] SAGA Consulting NV.

[2] HCPS: Health Care Personnel Scheduling.

further criteria (Burke et al. 2004b). The problem that is addressed in this paper can be categorised as ASBI|RVNO|PLR using the notation introduced by De Causmaecker and Vanden Berghe (2009). We present the problem model by the search space, schedule, hard and soft constraints. The set of all possible solutions is represented by the search space, which contains all possible mappings from the shift types and skill types set to the set of nurses within the planning period. Any candidate solution in the search space corresponds to a schedule. A schedule is feasible only if it fulfills all the hard constraints. The degree of soft constraint satisfaction determines the quality of the schedules.

2.1 Search space

The search space of the problem is represented by the planning period, skill types, shift types, and nurses (Burke et al. 2004a).

2.1.1 Planning period

The planning period is defined by a start date and a variable period length given as a number of days. This definition makes the model more complicated than the approach that is based on fixed planning periods (Aickelin and Li 2007; Brucker et al. 2005; Burke et al. 2008). The planning period length differs between wards. Among the real world response groups, the most common planning periods are one month and four weeks. In some periods of the year, like Christmas holidays, shorter planning periods like two weeks can be considered. The bank holidays within the planning periods are also part of the problem parameters.

2.1.2 Skill types

In hospitals, tasks are distributed among nurses according to their job descriptions, qualifications, experience, and responsibilities. This division is formalised by skill types. In our model, each nurse has a single primary skill. Occasionally, a nurse can have enough experience or education for carrying out tasks that are not associated with her primary skill but with another skill type. The other skills are considered as the secondary skill types of the nurse in consideration. In practice an assignment to a secondary skill type of a nurse is accepted but not preferred. However, a nurse cannot be assigned with a skill type that she does not have as a primary or secondary skill type. The second rule about the skill types is considered as a hard constraint.

As an example, consider a regular nurse who is experienced enough to substitute a head nurse in case of absence. That means, she can work as a head nurse as the secondary skill type (soft constraint). However, a head nurse who is not a regular nurse as a secondary skill type is not allowed to be assigned as a regular nurse (hard constraint).

The skill categories are not fixed in the problem definition. Instead they are defined by the users for each problem instance. The hierarchical substitution, which means nurses of a higher rank can substitute nurses of a lower rank, are not implicitly foreseen in the model. In Belgian hospitals, nurses do not prefer to carry out the tasks defined for a lower skill type. Therefore only user defined secondary skill types are considered by the solution method.

2.1.3 Shift types

The daily assignments are made in terms of shifts. Shifts are time periods defined with specific start and end times, rest periods before the start and after the end time, and a net job time. The rest periods before the start and after the end time ensure that the nurses have enough rest times between two working days. The rest periods are especially necessary in the wards that work around the clock to avoid assignment sequences such as a night shift followed by an early shift the next day. The definition and number of shift types are taken as parameters. This property makes the model more complex than usual problem definitions with a fixed number of shift types having predetermined definitions (Bard and Purnomo 2007; Bellanti et al. 2004; Brucker et al. 2005; Burke et al. 2008).

2.1.4 Nurses

The set of nurses is a user-defined parameter. The nurse definition is generic in the model. It allows the user to represent the properties of each nurse accurately. An *employment contract* is defined with a start and end date, and a constraint set. The fact that each nurse can have her own *employment contract* increases the individuality of nurses and therefore the complexity of the model.

2.2 Schedule

The schedule is composed of a set of assignments that are defined as quadruples of (nurse, day, shift type, skill type). The skill type has to be specified explicitly in an assignment, because a shift type is not necessarily associated to a skill type and a nurse can have more than one skill type. An empty, full or partially full schedule can be input to the solution method.

2.3 Soft constraints

The satisfaction of the soft constraints is not necessary for the feasibility but for the quality of the solution. The quality of a schedule is measured by an objective function that is the linear combination of the number of violations of each soft constraint, similar to (Burke et al. 2001b). Coverage constraints, rest times, assignment to the primary skill constraints are global, meaning that they apply to the whole schedule. Requests and horizontal constraints are specific to nurses. Nurses with similar contracts have similar horizontal constraints.

The threshold values of the coverage constraints and horizontal constraints can be either a minimum, a maximum, or a range defined by a minimum and a maximum value. The weight set used in the objective functions are specific for each problem instance. In a problem instance, the weights for the *coverage constraints* and *horizontal constraints* are specific to each constraint, but they are global for the *rest times* and *assignment to the primary skill* constraints. The weights can be any positive integer value.

The composition of the most suitable weight set is a complex task for the planner because of the high number of possible combinations. Nevertheless it is crucial to get satisfactory rosters (Burke et al. 2008). The planners will have to experiment with different weight settings and build up experience over time in order to produce weight sets that result in rosters that satisfy their priorities. The weight sets of the KaHo benchmarks are composed according to the feedback received from the planners of each ward in the hospitals during demo sessions. This weight sets are unique to each problem instance and available in the respective XML files in the project web site (Bilgin 2008).

Table 1 Compatible shift types set of early shift types

Short early [08:00 12:00]	Assigned to part timers
Regular early [08:00 14:36]	Assigned to 70% time working nurses
Long early [08:00 17:00]	Assigned to full timers

Table 2 Compatible shift types set of undesired shift types

Early morning [05:00 11:00]	Starts too early
Day [11:00 20:36]	Covers the whole day
Late [17:00 02:00]	Ends too late

2.3.1 Compatible shift types

A new concept introduced in this research paper is called compatible shift types. The nurse rostering problems that we received from our industry partner and from the hospitals involve coverage constraints and horizontal constraints that are defined on a set of shift types instead of a single shift type. The shift type sets that are used as a parameter in constraints are called compatible shift type sets. As a hypothetical example, consider the coverage constraint that is defined for early shift types. Any of the different shift types given in Table 1 is counted by this coverage constraint. However, the shift types in this compatible shift type set differ from each other in duration and in the corresponding contract. Another example is the maximum shift types worked counter that is defined on the undesired shift types given in Table 2. The objective of this counter is to minimise the number of shift types that are generally not desired by the nurses. A shift type can belong to more than one compatible shift type set in a problem instance. The compatible shift types sets may differ among constraints.

2.3.2 Coverage constraints

The number of nurses needed for each day, skill type and a compatible shift types set are called *coverage constraints* (Burke et al. 2004b). Coverage constraints are considered as hard constraints in many problem instances (Aickelin and Li 2007; Brucker et al. 2005; Burke et al. 2008, 2004a). However, the over-constrained nature of the problem in Belgium makes relaxations to coverage constraints necessary. The relaxation methods were investigated and implemented in automated nurse rostering tools (De Causmaecker and Vanden Berghe 2003). The coverage constraints are considered to be soft constraints by the real world response groups.

Although compatible shift types sets increase the accuracy of modelling the real life problem, they should be handled with care. In some situations errors may occur, if the domains of two different soft constraint instances overlap. If the compatible shift types sets of two coverage constraints at the same day and for the same skill type have common elements, any assignment of such elements contributes to both coverage constraints. The compatible shift types sets should be disjoint in such cases to avoid errors.

2.3.3 Assignment to the primary skill

An assignment to one of the secondary skills of a nurse is considered a soft constraint violation.

2.3.4 Rest times

For each shift type a period of rest time is defined before the start and after the end time. An assignment of a shift that overlaps with the rest period of another assignment is considered a soft constraint violation.

2.3.5 Requests

In nurse rostering models, nurses are allowed to request specific assignments of free days or periods (Bellanti et al. 2004; Burke et al. 2004a). In this model, two types are defined: assignment and absence requests. Assignment requests are defined with a preferred shift type to be assigned to a specific nurse on a specific day. Absence requests are defined by a specific day and a specific period where any assignment is to be avoided for a specific nurse. Absence requests also have a job time field. The job time of the absence request is added to the hours worked counters of the nurse in case the absence request is granted.

2.3.6 Horizontal constraints

The soft constraints imposed by the employment contracts of the nurses are called horizontal constraints. Various instances of horizontal constraints are present in nurse rostering problems in different countries (Bellanti et al. 2004; Brucker et al. 2005; Burke et al. 2004a). In the proposed model, horizontal constraints are generalised in three categories, being *counters*, *series* and *successive series*. This is a generic approach that allows users to define horizontal constraints, with specific subjects and parameters.

2.3.6.1 Counters The horizontal constraints that restrict the number of specific instances over a period are called counters. The counter period is defined by a start time and a length, which is given as a number of days. It does not necessarily match the planning period. If the counter period starts before the planning period, the counter value at the start of the planning period is given as an input to the solution method. Minimum thresholds cannot induce violations if the counter period exceeds the planning period, because they can be met in the upcoming planning period (Burke et al. 2004a). There are six subjects for counters: *hours worked*, *shift types worked*, *days worked*, *days idle*, *weekends worked* and *weekends idle*. *Shift types worked* counters are defined for a compatible shift types set. Apart from *weekends worked* and *weekends idle*, all counters have the *day types* parameter. The *day types* parameter can have the value of either any, holidays or a set of week and weekend days.

2.3.6.2 Series The number of consecutive occurrences of specific instances are restricted by series. There are five subjects for series: *shift types worked*, *days worked*, *days idle*, *weekends worked* and *weekends idle*. *Shift types worked* series are also defined for a compatible shift types set. The algorithm not only checks the series that start and end within the planning period but also the series that start in the previous planning period and extend to the current planning period. Similar to the counters, the minimum threshold violations by series that can be compensated in the next planning period are not penalised (Burke et al. 2004a). The problem model also considers the previous planning period when series have started but not finished. Therefore, the schedule information of the associated parts of the previous planning period are taken as a part of the input by the solution method.

2.3.6.3 Successive series Another type of horizontal constraint is the restriction of the succession of two series, each with their own threshold values. Any occurrence of the first series implies the second series to follow. Deviations from the second series are penalised. Possible orders of series are *days worked—days idle*, *days idle—days worked*, *shift types worked—days idle*, *days idle—shift types worked* and *shift types worked—shift types worked*. Similar to the series and counters, successive series involving shift types are defined for a compatible shift types set. Again, the minimum threshold violations that can be covered in the next planning period are not penalised (Burke et al. 2004a). Similar to the series, successive series that have been started but not ended in the previous planning period are considered within the problem model and the associated schedule information is taken as input by the solution method.

2.3.6.4 Example constraint set The work presented in this paper does not focus on a problem instance encountered in a ward in a hospital. It presents a generic problem model and a solution method that enables addressing several problem instances from different wards in different hospitals. We experimented with several real world problem instances to assess the capabilities of the proposed model and solution method. The working time regulations and agreements are different in each problem instance of the KaHo benchmarks. In addition to that, the regulations and agreements vary between the nurses in the same department. Therefore it is not feasible to specify the constraint set of each nurse in each problem instance of the KaHo benchmarks. However, an example constraint set from a nurse in one of the input data instances from the KaHo benchmarks is presented in Table 3. The other constraint sets can be found in the XML files that we provide online (Bilgin 2008).

Table 3 The soft constraints that apply to the schedule of the nurse with ID 855 from the input file Hospital1-Emergency-Normal.xml

ID	Type	Weight	Details
37	series	100	subject: shift types worked/ shift types: 927, 928, 930, 931/ min: 6/max: 8
35	successive series	500	series 1—subject: shift types worked/ shift types: 922, 923, 924, 925, 926/min: 1 series 2—subject: shift types worked/ shift types: 915, 916, 917, 918, 919/max: 0
36	successive series	500	series 1—subject: shift types worked/ shift types: 927, 928, 930, 931/Min: 1 series 2—subject: shift types worked/ shift types: 915, 916, 917, 918, 919/max: 0
−1000855	counter	100	subject: hours worked/ start date: 2007-12-03/period: 28 days/ min: 547200 sec/max: 547200 sec
No ID	absence request	500	date: 2007-12-03/start time: 08:00/ end time: 16:06/job time: 27360 sec
No ID	absence request	500	date: 2007-12-04/start time: 08:00/ end time: 16:06/job time: 27360 sec

2.4 Hard constraints

The schedule needs to satisfy all the hard constraints in order to be feasible.

2.4.1 *Single assignment start per nurse per day*

For each nurse only one assignment start per day is allowed, similar to (Brucker et al. 2005; Burke et al. 2008).

2.4.2 *No overlap between assignments*

The assignment of two shifts with an overlap in the work periods to one nurse is not allowed, as opposed to the model in (Burke et al. 2001a). In some cases, a dummy shift type, called *free shift*, is assigned to a nurse in order to balance the extra hours worked. The nurses do not actually work during *free shifts* and therefore these shifts are the only exception to the *No Overlap between Assignments* constraint.

2.4.3 *Honour skill types*

An assignment is allowed only if the skill type matches one of the skill types of the nurse, either primary or secondary.

2.4.4 *Schedule locks*

In some real life situations the automation tool is prevented from making alterations to some specific parts of the input schedule. These can be simple cases like granting an absence request or more complicated situations like partial rescheduling due to an unforeseen absence of a nurse. In the latter case, the assignments of the absent nurse need to be redistributed among other nurses. However, due to the time related constraints, this operation may modify the unaffected parts of the planning period as well. On the other hand, for several reasons, alterations to an announced roster are avoided to the extent possible. To meet this criterion, the unaffected parts of the schedule are locked before it is given as an input to the algorithm. Schedule locks are defined with (nurse, day) pairs. The objective function always evaluates the complete schedule regardless of schedule locks.

2.4.5 *Operations on defined assignments only*

The shift types are not always relevant to all skill types. For example, in many wards, night shifts are not assigned to head nurses. Therefore coverage constraints are given with associated skill types (Sect. 2.3.2). The assignments are only allowed if they are defined in the coverage constraints. On the other hand, some input schedules contain assignments that are not mentioned in the coverage constraints. These are preassigned schedule parts which the solution method is not allowed to modify or delete, even if they are not locked. Suppose that a nurse is assigned a shift that is not relevant to the ward under study, but to another ward in the same hospital. This specific assignment is not defined in the coverage constraints of the current ward and is not allowed to be deleted. These restrictions are called *Operations on Defined Assignments Only*. They reduce the size of the feasible search space and prevent the solution method from searching the irrelevant parts of the search space.

3 Solution method

The solution method consists of an iterative improvement step preceded by a preprocessing step. We have experimented with two algorithms as the iterative improvement step: VNS and ALNS. The solution method does not schedule nurses with different skill types separately as in (Burke et al. 2004a). This allows exploiting the advantages offered by secondary skill types. The solution method is not allowed to make modifications that result in an infeasible schedule. The termination criterion involves the maximum execution time of the algorithm, without taking the preprocessing step into account. The pseudocode of the algorithm utilising VNS is presented in Algorithm 1.

Algorithm 1 Pseudocode of the solution method utilising VNS

```
S = Initial Schedule
Preprocessing
Add assignments randomly to S in order to meet the minimum coverage constraints
Variable Neighbourhood Search
CQ = Circular Queue of the Neighbourhoods
N = First Neighbourhood in CQ
BS = S
while Termination criterion not met do
   S* = Search (N(S)) with respect to the tabu search strategy
   if cost(S*) < cost(BS) then
      Decrement tabu length
      BS = S*
   else
      Increment tabu length
      if cost(S*) ≥ cost(S) then
         N = next neighbourhood in CQ
      end if
   end if
   S = S*
end while
return BS
```

3.1 Preprocessing

The solution method accepts an input schedule from the user. The input schedule can be an empty, a partial or a complete schedule. The preprocessing method tries to fulfill the minimum coverage constraints. A roster that satisfies the minimum coverage constraints is perceived as a complete roster by the planners. The aim of the preprocessing step is to provide a complete roster in a minimum amount of time. The roster can be improved iteratively in the following steps of the search algorithm. This way the planner will receive a complete roster no matter how short the execution time is. The execution time will only influence the quality of the roster.

3.2 Variable neighbourhood search

The VNS algorithm utilises several neighbourhoods and holds the parameters of the executed moves in a tabu list. At each iteration of the algorithm, a single neighbourhood is searched. The best move in the neighbourhood that complies with the hard constraints and is not tabu is executed. The schedule remains feasible throughout the execution of the algorithm. The exceptions to the strict steepest descent practice are discussed in the corresponding paragraphs about each neighbourhood.

Token-ring search (Gaspero and Schaerf 2002) is used to switch between the neighbourhoods in the VNS algorithm. In token-ring search, the neighbourhoods are held in a circular queue that determines their application sequence (Gaspero and Schaerf 2002). If the applied neighbourhood does not result in an improving move, the algorithm switches to the next neighbourhood in the queue.

3.3 Adaptive large neighbourhood search

Similar to VNS, ALNS utilises a set of neighbourhoods and explores one neighbourhood at each iteration. The neighbourhood is selected in a stochastic way using the roulette wheel method. The scores of the neighbourhoods are increased in three cases: if an overall best solution is found, if a solution is found that is better than the current solution, or if the solution found is feasible and not tabu. The scores of the neighbourhoods are updated regularly by putting more emphasis on the performance in recent iterations (Pisinger and Ropke 2007).

3.4 Tabu list

A tabu list is maintained in both iterative improvement algorithms: VNS and ALNS. The function of the tabu list is to avoid cycles of the algorithm around local optima. The parameters of the executed moves (nurse, day, shift type, and skill type) are kept in the tabu list in a hashed way. Prime numbers are used as values for the tabu tenures, in order to avoid hash collusions and cycling. The tabu tenure is variable during the execution. It is increased to the next prime number at each non-improving iteration and decreased to the previous prime number if there is an improvement. The tabu tenure varies between a lower and an upper bound. The lower bound is equal to seven and the upper bound is a parameter of the algorithm. Based on our preliminary experiments, we have decided to experiment further with two values, 97 and 199, as the upper limit. The aspiration criterion holds so that tabu moves that result in overall best candidate solutions are allowed.

3.5 Assign shift

Since an assignment is defined as a quadruple of (nurse, day, shift type, skill type) (Sect. 2.2), the Assign Shift neighbourhood operates on these quadruples. When an assignment is made, not only a shift type is assigned, but also the associated skill type for that assignment. High numbers of shift types are common in Belgian hospitals, and they result in a large Assign Shift neighbourhood. The evaluation of each candidate in such a large neighbourhood consumes a high amount of CPU time and results in inefficiencies in the algorithm. For each (nurse, day, skill type) triple, there is more than one relevant shift type assignment possible. The size of this neighbourhood is reduced by evaluating one random shift type among the relevant ones for each (nurse, day, skill type) triple. A move of the Assign Shift neighbourhood is encoded as a hash value of the (nurse, day, shift type, skill type) quadruple and inserted into the tabu list.

3.6 Delete shift

The deletion of an assignment is feasible only if two conditions are satisfied. First, the assignment (day, shift type, skill type) must correspond to a coverage constraint, i.e. a coverage must be required for the shift type, the day, and the skill (see Sect. 2.4.5). Second, the assignment should not be locked (see Sect. 2.4.4). Since the coverage constraints are not defined as hard constraints, deletions that violate coverage constraints are considered feasible as well (see Sect. 2.3). The Delete Shift neighbourhood consists of all the feasible deletion moves. A deletion move not only deletes the assigned shift type at that given timeslot, but also the skill type, since this is a property of the assignment as well. Similar to the Assign Shift neighbourhood, a move of the Delete Shift neighbourhood is encoded as a hash value of the (nurse, day, shift type, skill type) quadruple and inserted into the tabu list.

3.7 Single shift-day

An assignment is removed from a nurse's schedule and added to another nurse on the same day, if the second nurse has no assignment on that day and if she has the associated skill type (Burke et al. 2004a). A move of the Single Shift-Day neighbourhood consists of two moves: a delete and an assign move. Therefore the parameters of both, the delete and assign moves, are separately encoded as hash values and inserted into the tabu list. Since the following neighbourhoods are also composed of two moves, a delete and an assign move, their parameters are encoded and inserted into the tabu list in the same way as single shift-day.

3.8 Change assignment based on compatible shift type

The shift type of an assignment is changed to another compatible shift type defined in the coverage constraints for the associated day and skill type. One random shift type from the same compatible shift types set is considered for each assignment according to the same motivation as for the *Assign Shift* neighbourhood.

3.9 General assignment change

An assignment is changed to another shift type, while the skill type of the assignment remains the same. This neighbourhood does not necessarily consist of the compatible shift types from the same coverage constraint, as in *Change Assignment based on Compatible Shift Type* neighbourhood. Again a subset of the complete neighbourhood is considered, like in the *Assign Shift* neighbourhood. The subset simply consists of a single alternative random shift type for each assignment.

3.10 Change assignment based on skill type

This neighbourhood operates on the roster of a nurse with at least two different skill types. It deletes an assignment and adds another assignment to one of the nurse's other skill types.

4 Experiments

The solution method needs to cope with different situations and scenarios that may occur in the real world. Hospitals are organised in wards, each with different settings of problem variables: planning periods, nurse properties, shift types, skill types, and soft constraints. Variations and unexpected changes in the workload of hospital wards are not rare. Sample scenarios are *overload of work*, for example in case of an epidemic, and *unexpected absence of a nurse* in case of an illness. In the latter case, partial rescheduling of the complete roster is needed. Various solution method settings are tested with different scenarios in order to measure the performance of the method and to find the best performing algorithm setting.

4.1 Experimental settings

The first data set, the KaHo benchmarks involve the Emergency, Psychiatry, Reception, Meal Preparation, and Geriatrics wards from Hospital 1 and the Palliative Care ward from Hospital 2. For each of the wards, three different scenarios are considered. The first scenario has normal settings with an empty input schedule (*normal*). The second scenario considers an overload of work, involving higher values for the coverage constraints (*overload*). The last scenario is the unexpected absence of a nurse (*absence*). In that case the complete schedule is taken into account but only the affected parts are required and allowed to be modified.

Start dates and period properties of the planning periods per ward are given in Table 4. In the *absence* scenarios, the entire schedule is locked except the one week period that start at the date mentioned in Table 4. The number of shift and skill types for each ward is given in Table 5. Although some of the shift types in these wards have identical working times, they have different tasks attached. Therefore, they are treated within different compatible shift types sets in horizontal constraints and coverage constraints. The number of nurses with each skill type and the total number of nurses are presented in Table 5. It is clear from this table that several nurses have secondary skill types. The wards have different contract types according to their weekly job time. The weekly job time and the number of nurses for each weekly job time is given in Table 6. Some of the nurses change from one

Table 4 Planning periods and absence start dates

Ward	Start date	Period	Absence start date
Emergency	03/12/2007, Monday	4 weeks	10/12/2007, Monday
Psychiatry	01/12/2007, Saturday	1 month	10/12/2007, Monday
Reception	14/4/2008, Monday	6 weeks	5/5/2008, Monday
Meal Preparation	1/2/2008, Friday	1 month	18/2/2008, Monday
Geriatrics	25/2/2008, Monday	4 weeks	17/3/2008, Monday
Palliative Care	31/12/2007, Monday	13 weeks	4/2/2008, Monday

Table 5 Number of shift types and number of nurses with each skill type

Ward	Shift types	Skill 1	Skill 2	Skill 3	Skill 4	Total employees
Emergency	27	1	15	4	26	27
Psychiatry	14	1	17	1	–	19
Reception	19	1	1	3	15	19
Meal P.	9	1	31	–	–	32
Geriatrics	9	4	20	–	–	21
P. Care	23	1	21	4	1	27

Table 6 The weekly job time and the corresponding number of nurses

Ward	38 hours (100%)	34.2 hours (90%)	30.4 hours (80%)	28.5 hours (75%)	22.8 hours (60%)	19 hours (50%)
Emergency	24	–	–	3	–	–
Psychiatry	13	–	–	2	–	4
Reception	5	–	–	7	–	7
Meal P.	3	2	–	1	–	28
Geriatrics	9	–	–	9	1	3
P. Care	13	–	2	4	1	7

employment contract to another within the given planning period. Consequently, when that happens within one planning period, the contracts are considered serially. The second data set consists of four problem instances from the Nottingham benchmarks.

We want to point at one important difference with the KaHo instances, in which the assignments and the idle days in the previous planning period are considered. The relevant elements that determine the border conditions are included in the input files of the KaHo benchmarks. The problem definition of the KaHo benchmarks penalises the unresolved constraint violations at the end of the previous planning period in case they could be solved by the assignments or idle days in the current planning period.

Similarly, the assignments and idle days at the beginning of the current planning period are evaluated together with the relevant parts of the previous planning period. Some of the valid constructs at the beginning of the current planning period would be considered constraint violations if the planning period was handled in an isolated way. The evaluation of a roster according to the Nottingham benchmarks does not cover the real world requirements that we collected for the KaHo benchmarks. In the following paragraphs, we discuss some examples of differences between the evaluation methods of both problem definitions.

4.1.1 Valouxis-1

The definition of the problem instance Valouxis-1 involves the *no isolated working days* constraint with a penalty of 1000. The optimal solution published on the website of the Nottingham benchmarks contains three isolated assignments at the beginning of the current planning period but a fitness value of 20. The fitness value of this solution is greater than 3000 in our evaluation function. Our solution methods attempt to avoid such isolated assignments, even at the beginning of the current planning period.

4.1.2 BCV-3.46.2

The *max five consecutive working days* constraint is defined as the exclusion of the following pattern: six days worked in a row followed by an idle day. According to this definition, any number of consecutive working days greater than five is penalised by a fixed penalty. Suppose that there are eight working days scheduled in a row, followed by an idle day. This construction satisfies the pattern only once and it is penalised only once in the evaluation function. However, our evaluation function counts three violations in this construction and penalises it three times. The weight of this constraint is 5. According to the pattern definition, the series of 6, 10, and 20 assignments in a row all have the same penalty: 5. In the real world examples that we consider, it was necessary to distinguish between small and large violations. Consequently, our evaluation method penalises these series by 5, 25, and 75 respectively. Similar pattern definitions occur in BCV-4.13.1 as well, such as the *max three consecutive D shifts*, *max three consecutive V shifts*, and *max two consecutive L shifts* constraints.

Since no previous planning period information is provided in the Nottingham benchmarks, our algorithms consider the previous planning period to be empty and try to correct the violations to the *max seven consecutive free days* constraint by assigning shifts at the beginning of the current planning period. Consequently, this practice is balanced by the violations to other constraints. Similar pattern definitions occur in BCV-4.13.1 as well, such as the *max 3 consecutive free days*.

4.1.3 BCV-4.13.1

The pattern definitions of the *min two consecutive working days* and *min two consecutive free days* conflict at the beginning of the current planning period. The pattern definition of the first penalises a worked day followed by an idle day at the beginning of the current planning period. This practice presumes the last day of the previous planning period to be an idle day. The pattern definition of the second constraint penalises an idle day followed by a worked day at the beginning of the current planning period. In contrast to the first one, the second practice presumes the last day of the previous planning period to be a worked day. The definition of the latter constraint also differs from our evaluation method, which interprets the lack of the previous planning period information as idle days.

The solution methods studied in this paper have been developed for addressing the problem definition of the KaHo benchmarks. The differences in the problem definitions makes our algorithm to carry out the optimisation procedure with different priorities, which results in different fitness values. Under these circumstances, a comparison of our results on the Nottingham benchmarks with the results from the literature is not relevant.

The experiments were undertaken with 12 different solution method settings. There are three decisions to be made for setting the algorithm: the rule to switch between the neighbourhoods, the neighbourhood composition, and the maximum tabu tenure. The neighbourhood sets that we applied in the experiments are presented in Table 7. We experiment with different neighbourhood sets in order to measure the contribution of specific neighbourhoods to the search. The *assign shift* and *delete shift* neighbourhoods are complementary. They are utilised consecutively in the VNS algorithm. The remaining neighbourhoods are special compositions of *delete shift* and *assign shift* neighbourhoods, each with a specific rule to restrict the *delete shift-assign shift* sequence. In the VNS algorithm, the search starts with the *assign shift* and *delete shift* neighbourhoods, and continues with the *single shift-day* neighbourhood and the additional neighbourhoods in the order of appearance in Table 7. The ALNS algorithm is also tested in order to determine the succession of the neighbourhoods in an adaptive way. Two different upper bounds for the tabu tenure are 97 and 199.

Table 7 Neighbourhood sets

Set 1	assign shift, delete shift, single shift-day
Set 2	Set 1 + change assignment based on compatible shift type
Set 3	Set 1 + change assignment based on skill type
Set 4	Set 1 + general assignment change
Set 5	Set 1 + general assignment change + change assignment based on skill type

The model and the solution method is implemented in C#. The experiments are carried out in MS Visual Studio 2005 Professional Edition. The operating system is MS Windows Server 2003 Enterprise Edition SP 2 running on an Intel Pentium 4 CPU with 2.40 GHz and 2.00 GB of RAM.

Each algorithm setting is executed ten times for each problem instance. The execution time is 10 minutes for the Nottingham benchmarks, *normal* and *overload* scenarios of the KaHo benchmarks and one minute for the *absence* scenario of the KaHo benchmarks. For each problem instance, a group of best performing algorithm settings are identified using statistical methods. First the best performing setting is identified according to the average best fitness values of the algorithm settings over 10 runs. Second the results of each remaining algorithm setting are compared with the best performing algorithm setting using the Wilcoxon test with 95% confidence. If the result of a setting does not have a significant performance difference with the best performing setting, then this setting is also included in the best performing group for the problem instance under examination. For each problem instance; the input data, a sample solution obtained from the algorithm, and the penalty details of this sample solution are published online (Bilgin 2008).

4.2 Experimental results

The experimental results for each algorithm setting, problem instance couple are presented in Tables 8–17. The values in these tables are the average fitness values of the best solutions found and their standard deviations. In these tables, the algorithm settings that performed significantly better than the rest are highlighted with bold characters. Tables 8–13 also depict the best results of the normal scenarios of the KaHo benchmarks achieved by the human planners using the nurse rostering assistance system (mentioned in Sect. 1). Table 14 presents the results by algorithm settings with 97 as the maximum tabu tenure to give an overall view.

Table 8 Hospital 1 Emergency Results. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively

A. Setting		Normal		Overload		Absence	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
VNS-1	97	**11410.50**	**386.46**	27441.33	332.37	21849.67	88.86
VNS-1	199	14020.66	1566.69	27052.50	138.65	21864.67	45.72
VNS-2	97	11817.33	655.30	29231.67	2461.29	21862.17	38.55
VNS-2	199	13310.66	714.66	27055.33	203.29	21882.17	91.24
VNS-3	97	**11295.83**	**275.42**	27813.66	733.73	21412.67	246.77
VNS-3	199	11801.66	376.20	**26836.17**	**227.36**	21843.67	38.09
VNS-4	97	**11201.17**	**308.12**	27130.00	314.70	21532.67	186.01
VNS-4	199	12014.00	561.92	**26699.67**	**161.17**	21711.17	268.75
VNS-5	97	**11361.17**	**239.20**	27483.17	370.90	**21175.17**	**20.82**
VNS-5	199	**11285.50**	**201.21**	**26731.84**	**185.23**	21753.84	213.04
ALNS-5	97	11753.33	191.95	27679.17	216.29	21327.67	137.90
ALNS-5	199	11858.83	163.41	27325.17	123.33	24121.33	1598.55
Human Planner		49236.00			–		

Table 9 Hospital 1 Psychiatry Results. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively

A. Setting		Normal		Overload		Absence	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
VNS-1	97	9079.00	191.51	12339.00	311.11	13109.00	210.42
VNS-1	199	10236.00	255.13	14687.00	557.77	**12906.00**	**203.26**
VNS-2	97	9074.00	223.67	12429.00	234.92	12993.00	99.34
VNS-2	199	10160.00	430.86	14492.00	656.94	**12874.00**	**235.33**
VNS-3	97	9041.00	173.36	12207.00	205.32	**12946.00**	**206.30**
VNS-3	199	10363.00	218.84	14930.00	570.28	13022.00	124.26
VNS-4	97	8751.00	127.93	**10914.00**	**115.59**	**13001.00**	**201.19**
VNS-4	199	9184.00	256.31	12166.00	393.62	**12888.00**	**337.96**
VNS-5	97	8774.00	146.98	**10966.00**	**215.83**	**12850.00**	**172.43**
VNS-5	199	9292.00	208.74	12196.00	240.15	**13019.00**	**263.67**
ALNS-5	97	**8649.00**	**136.42**	**10916.00**	**238.15**	13790.00	473.15
ALNS-5	199	9210.00	219.95	11929.00	297.97	14247.00	1653.78
Human Planner		35480.00		–			

Table 10 Hospital 1 Reception Results. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively

A. Setting		Normal		Overload		Absence	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
VNS-1	97	22582.17	219.75	56302.17	606.76	28730.67	223.07
VNS-1	199	23786.67	238.13	56348.17	387.77	28821.17	120.93
VNS-2	97	**21720.67**	**173.39**	54122.67	378.08	28699.67	233.65
VNS-2	199	22555.67	183.41	53873.67	133.67	28778.17	105.38
VNS-3	97	22525.67	242.67	56336.67	382.11	28838.17	192.46
VNS-3	199	23884.67	289.83	56274.17	179.97	28850.67	205.76
VNS-4	97	**21739.67**	**139.77**	**53036.17**	**120.17**	28673.67	187.00
VNS-4	199	22698.17	238.81	53605.17	189.69	28768.17	137.22
VNS-5	97	**21812.17**	**136.88**	53122.67	145.43	28729.17	128.80
VNS-5	199	22624.67	215.23	53563.17	226.91	28786.17	117.70
ALNS-5	97	**21774.17**	**199.63**	**52975.17**	**120.72**	28786.17	364.21
ALNS-5	199	22534.67	264.43	53536.67	199.51	28658.67	372.63
Human Planner		48358.00		–			

4.2.1 The KaHo benchmarks

The experimental results indicate that the solution methods with 97 as the maximum tabu tenure perform better than the ones with 199 on the KaHo benchmarks. The solution methods with 97 as the maximum tabu tenure were among the best performers for 17 problem instances, while the ones with 199 were among the best performers only for seven of the KaHo benchmarks. The only exception to this observation is the Emergency Overload sce-

Table 11 Hospital 1 Meal Preparation Results. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively

A. Setting		Normal		Overload		Absence	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
VNS-1	97	**2904.10**	**25.40**	10949.10	22.49	**5342.83**	**185.31**
VNS-1	199	3048.80	31.76	11238.20	72.98	**5408.50**	**157.95**
VNS-2	97	**2894.67**	**21.82**	10942.30	36.74	**5348.67**	**157.48**
VNS-2	199	3064.63	53.68	11245.60	68.49	**5401.33**	**177.21**
VNS-3	97	**2896.33**	**14.08**	10944.80	37.97	**5346.33**	**186.36**
VNS-3	199	3049.70	40.02	11230.90	64.73	**5425.00**	**157.11**
VNS-4	97	3162.60	37.78	**10867.90**	**29.75**	**5326.33**	**99.71**
VNS-4	199	3121.50	51.75	10978.30	68.29	5516.17	133.61
VNS-5	97	3133.33	19.07	**10881.60**	**25.54**	**5350.17**	**82.24**
VNS-5	199	3104.83	59.53	10987.00	46.31	**5405.00**	**155.49**
ALNS-5	97	3107.87	28.20	11058.00	49.10	**5338.33**	**52.90**
ALNS-5	199	3124.27	39.96	11120.60	42.98	7193.17	460.98
Human Planner		22100.00		–			

Table 12 Hospital 1 Geriatrics Results. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively

A. Setting		Normal		Overload		Absence	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
VNS-1	97	**4301.00**	**134.88**	10898.50	303.58	**9194.83**	**260.34**
VNS-1	199	5295.33	633.63	12659.67	935.36	**9035.67**	**447.74**
VNS-2	97	**4208.67**	**117.93**	10914.33	285.87	**9138.83**	**204.83**
VNS-2	199	5170.67	364.91	13539.67	643.76	**9286.83**	**299.59**
VNS-3	97	4612.67	175.90	11417.00	414.39	9451.33	247.29
VNS-3	199	6177.50	1212.17	15225.50	1202.69	**9425.67**	**395.41**
VNS-4	97	4343.67	110.91	**10705.50**	**146.81**	**9180.17**	**358.61**
VNS-4	199	5647.67	381.44	12192.83	510.15	**9153.17**	**410.28**
VNS-5	97	4788.33	209.88	11131.17	257.38	**9400.17**	**390.50**
VNS-5	199	6038.00	642.87	13302.67	889.10	**9300.17**	**320.23**
ALNS-5	97	4657.67	199.62	**10897.00**	**330.73**	9984.33	554.00
ALNS-5	199	6945.50	805.99	13685.83	871.46	10282.00	706.24
Human Planner		28594.00		–			

nario. The VNS variants with neighbourhood sets three, four, and five and the maximum tabu tenure 199 are the best performing solution methods for the Emergency Overload scenario.

Among the algorithm settings with 97 as the maximum tabu tenure, VNS with neighbourhood set four is among the best performers for 13 instances, and neighbourhood set five for 12 instances. The remaining algorithm settings cannot compete with this performance. The exceptions to this observation are the experiments for the normal scenarios of the psychiatry, meal preparation, and geriatrics wards. ALNS is the best performing algorithm for the psychiatry normal scenario. The best performing algorithm settings for geriatrics and meal

Table 13 Hospital 2 Palliative Care. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively

A. Setting		Normal		Overload		Absence	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
VNS-1	97	**44392.50**	**1090.53**	55675.00	1169.25	**56388.00**	**363.93**
VNS-1	199	46348.25	1053.51	54399.00	1632.47	**56481.50**	**438.73**
VNS-2	97	**44515.75**	**973.50**	55553.25	965.17	56957.50	351.71
VNS-2	199	46539.25	721.09	55346.75	1711.53	57139.00	444.82
VNS-3	97	**44555.75**	**967.81**	55916.25	1828.05	**56659.50**	**642.87**
VNS-3	199	46662.25	856.09	54812.50	1198.47	**56355.00**	**323.37**
VNS-4	97	**44951.50**	**791.85**	**50632.25**	**647.25**	**56356.50**	**445.51**
VNS-4	199	46325.50	1004.18	51638.75	858.53	**56676.00**	**420.49**
VNS-5	97	**44952.75**	**725.61**	**50177.75**	**528.20**	**56655.50**	**443.67**
VNS-5	199	46145.25	917.36	51736.75	816.13	**56658.50**	**420.48**
ALNS-5	97	**44155.25**	**840.44**	51612.25	829.23	57713.75	721.77
ALNS-5	199	45646.75	607.29	52746.00	1032.53	61334.25	2523.93
Human Planner		183859.00		–			

preparation normal scenarios are similar: VNS with neighbourhood set one and two for the geriatrics, and VNS with neighbourhood set one, two, and three for the meal preparation.

VNS with neighbourhood set five outperforms VNS with neighbourhood set four only for the Emergency Absence. This result can be explained by the fact that the nurse rostering problem in the Emergency ward involves numerous nurses with secondary skill types and neighbourhood set five involves the *change assignment based on skill type* neighbourhood. As a conclusion, VNS with neighbourhood set four and maximum tabu tenure 97 is the most suitable one among the algorithms settings tested on the KaHo benchmarks.

The basic neighbourhood set (set one) is among the best performing neighbourhood sets on eight out of 18 problem instances. For these instances, the basic neighbourhood is not the unique best performer. The contribution of the problem specific neighbourhoods that take advantage of the problem properties like secondary skill types and compatible shift types are emphasised by this result.

4.2.2 The Nottingham benchmarks

We provide the fitness values obtained from our evaluation method as well as from the evaluator provided by the University of Nottingham (Curtois 2009) obtained in Tables 15–17. Since the statistical analysis is carried out on the results from the Nottingham evaluator, the best performing group is not indicated for the results from the evaluation method of KaHo.

The algorithm settings with 199 as the maximum tabu tenure are not among the best performing group for any of the problem instances of the Nottingham benchmarks. The algorithm settings with 97 as the maximum tabu tenure are more suitable for the Nottingham benchmarks.

There are no performance variances between the algorithm settings with 97 as the maximum tabu tenure. In this group, all algorithm settings were in the best performing group on two problem instances, except VNS with neighbourhood set three. The latter algorithm setting was among the best performing group only for the SINTEF instance.

Table 14 Overall Results on the KaHo benchmarks. The first row for each problem instance setting is the average fitness value of the solutions, the second row the standard deviation

		VNS-1-97	VNS-2-97	VNS-3-97	VNS-4-97	VNS-5-97	ALNS-5-97
Emergency	Normal	**11410.50**	11817.33	**11295.83**	**11201.17**	**11361.17**	11753.33
		386.46	655.30	**275.42**	**308.12**	**239.20**	191.95
	Overload	27441.33	29231.67	27813.66	27130.00	27483.17	27679.17
		332.37	2461.29	733.73	314.70	370.90	216.29
	Absence	21849.67	21862.17	21412.67	21532.67	**21175.17**	21327.67
		88.86	38.55	246.77	186.01	**20.82**	137.90
Psychiatry	Normal	9079.00	9074.00	9041.00	8751.00	8774.00	**8649.00**
		191.51	223.67	173.36	127.93	146.98	**136.42**
	Overload	12339.00	12429.00	12207.00	**10914.00**	**10966.00**	**10916.00**
		311.11	234.92	205.32	**115.59**	**215.83**	**238.15**
	Absence	13109.00	12993.00	**12946.00**	**13001.00**	**12850.00**	13790.00
		210.42	99.34	**206.30**	**201.19**	**172.43**	473.15
Reception	Normal	22582.17	**21720.67**	22525.67	**21739.67**	**21812.17**	**21774.17**
		219.75	**173.39**	242.67	**139.77**	**136.88**	**199.63**
	Overload	56302.17	54122.67	56336.67	**53036.17**	53122.67	**52975.17**
		606.76	378.08	382.11	**120.17**	145.43	**120.72**
	Absence	28730.67	28699.67	28838.17	28673.67	28729.17	28786.17
		223.07	233.65	192.46	187.00	128.80	364.21
Meal P.	Normal	**2904.10**	**2894.67**	**2896.33**	3162.60	3133.33	3107.87
		25.40	**21.82**	**14.08**	37.78	19.07	28.20
	Overload	10949.10	10942.30	10944.80	**10867.90**	**10881.60**	11058.00
		22.49	36.74	37.97	**29.75**	**25.54**	49.10
	Absence	**5342.83**	**5348.67**	**5346.33**	**5326.33**	**5350.17**	**5338.33**
		185.31	**157.48**	**186.36**	**99.71**	**82.24**	**52.90**
Geriatrics	Normal	**4301.00**	**4208.67**	4612.67	4343.67	4788.33	4657.67
		134.88	**117.93**	175.90	110.91	209.88	199.62
	Overload	10898.50	10914.33	11417.00	**10705.50**	11131.17	**10897.00**
		303.58	285.87	414.39	**146.81**	257.38	**330.73**
	Absence	**9194.83**	**9138.83**	9451.33	**9180.17**	**9400.17**	9984.33
		260.34	**204.83**	247.29	**358.61**	**390.50**	554.00
P. Care	Normal	**44392.50**	**44515.75**	**44555.75**	**44951.50**	**44952.75**	**44155.25**
		1090.53	**973.50**	**967.81**	**791.85**	**725.61**	**840.44**
	Overload	55675.00	55553.25	55916.25	**50632.25**	**50177.75**	51612.25
		1169.25	965.17	1828.05	**647.25**	**528.20**	829.23
	Absence	**56388.00**	56957.50	**56659.50**	**56356.50**	**56655.50**	57713.75
		363.93	351.71	**642.87**	**445.51**	**443.67**	721.77

Table 15 The Nottingham benchmarks: SINTEF, BCV-3.46.2. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively. The columns with (N) denote the results calculated using the evaluation method of Nottingham, and (K) the results calculated using the evaluation method of KaHo

A. Setting		SINTEF		BCV-3.46.2 (N)		BCV-3.46.2 (K)	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
VNS-1	97	**17.5**	**1.6**	946.2	16.8	947.3	17.2
VNS-1	199	106.1	94.0	1008.2	26.7	1009.8	26.3
VNS-2	97	**16.4**	**1.5**	967.2	21.1	967.6	21.2
VNS-2	199	125.7	94.5	1016.3	44.8	1017.3	44.8
VNS-3	97	**16.8**	**1.8**	962.0	33.3	962.5	32.6
VNS-3	199	93.1	91.3	996.9	41.6	998.5	41.5
VNS-4	97	21.2	2.3	1001.0	28.5	1001.7	28.2
VNS-4	199	35.8	6.6	1027.3	43.1	1029.0	43.2
VNS-5	97	22.4	2.5	995.5	29.3	996.2	30.0
VNS-5	199	37.0	4.6	1010.2	32.9	1011.7	32.9
ALNS-5	97	23.1	2.2	**930.6**	**8.7**	930.7	8.5
ALNS-5	199	127.6	120.2	950.0	14.3	949.8	14.0
Optimal Solution		0.0		894.0			

Table 16 The Nottingham benchmarks: BCV-4.13.1. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively. The columns with (N) denote the results calculated using the evaluation method of Nottingham, and (K) the results calculated using the evaluation method of KaHo

A. Setting		BCV-4.13.1 (N)		BCV-4.13.1 (K)	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.
VNS-1	97	**75.4**	**27.33**	73.8	27.3
VNS-1	199	237.4	92.95	235.8	92.8
VNS-2	97	**68.7**	**21.63**	67.5	21.6
VNS-2	199	162.8	82.15	163.5	82.5
VNS-3	97	76.3	19.84	75.6	20.0
VNS-3	199	158.1	57.26	157.4	57.3
VNS-4	97	**66.2**	**14.79**	66.4	15.7
VNS-4	199	1168.8	3130.76	1170.8	3129.8
VNS-5	97	**62.1**	**24.37**	64.8	28.0
VNS-5	199	157.3	46.08	157.7	44.8
ALNS-5	97	**52.2**	**16.92**	50.9	17.9
ALNS-5	199	165.9	28.17	164.8	29.0
Optimal Solution		10.0			

Table 17 The Nottingham benchmarks: Valouxis-1. A. Setting, N. Set, T. Limit, St. Dev. denote algorithm setting, neighbourhood set, upper bound for the tabu list length, and standard deviation respectively. The columns with (N) denote the results calculated using the evaluation method of Nottingham, and (K) the results calculated using the evaluation method of KaHo

A. Setting		Valouxis-1 (N)		Valouxis-1 (K)	
N. Set	T. Limit	Average	St. Dev.	Average	St. Dev.
VNS-1	97	2796.0	1085.5	3586.0	880.7
VNS-1	199	6872.0	875.3	7946.0	1165.3
VNS-2	97	3226.0	710.7	3726.0	458.0
VNS-2	199	7052.0	1839.8	7590.0	1543.5
VNS-3	97	2892.0	671.1	3494.0	708.9
VNS-3	199	8186.0	938.3	8618.0	1091.2
VNS-4	97	**1718.0**	**597.4**	2618.0	437.4
VNS-4	199	4910.0	995.5	5778.0	786.6
VNS-5	97	**1604.0**	**640.2**	2516.0	454.3
VNS-5	199	4420.0	1198.8	5338.0	615.2
ALNS-5	97	2766.0	977.8	3638.0	492.1
ALNS-5	199	6110.0	1159.8	6928.0	1064.2
Optimal Solution		20.0			

5 Conclusions

A new nurse rostering model is proposed to accurately represent the current situation of the problem in real world environments. This model is also open to extensions with extra soft constraints encountered in different hospitals, other sectors and different countries. To cope with the new properties of the problem, associated neighbourhoods are defined and utilised within a VNS algorithm. These neighbourhoods exploit the problem properties like compatible shift types and secondary skill types.

The experimental results show that the algorithm settings that deploy problem specific neighbourhoods in addition to the basic neighbourhood set perform significantly better than the algorithm settings with the basic neighbourhood set only. The experiments are also carried out on four instances from the Nottingham benchmarks. The differences in the problem definitions between the KaHo and Nottingham benchmarks make a meaningful comparison between our results and the results from the Nottingham website impractical. The differences between the problem definitions offer opportunities to study the modelling aspects of the nurse rostering problem in a greater depth.

The exploitation of the extendibility of the model by adding extra soft constraints and other properties to cover real world problems encountered in other hospitals, sectors and different countries, is a natural future research direction. The application of other optimisation techniques, like hyperheuristics, can also be investigated. In this case heuristics that are specialised to tackle different aspects of the scenarios can be studied and deployed within hyperheuristics.

Acknowledgements This research project was funded by IWT (Institute for the Promotion of Innovation by Science and Technology in Flanders) within the projects IWT 060376, and IWT 060353 (RAP: Framework for Automated Personnel Rostering).

References

Aickelin, U., & Li, J. (2007). An estimation of distribution algorithm for nurse scheduling. *Annals of Operations Research*, *155*(1), 289–309.

Bard, J. F., & Purnomo, H. W. (2007). Cyclic preference scheduling of nurses using a Lagrangian-based heuristic. *Journal of Scheduling*, *10*(1), 5–23.

Beliën, J., & Demeulemeester, E. (2007). On the trade-off between staff-decomposed and activity-decomposed column generation for a staff scheduling problem. *Annals of Operations Research*, *155*(1), 143–166.

Bellanti, F., Carello, G., Della Croce, F., & Tadei, R. (2004). A greedy-based neighborhood search approach to a nurse rostering problem. *European Journal of Operational Research*, *127*(1), 28–40.

Bilgin, B. (2008). Project web page of automation of nurse rostering in Belgian hospitals. http://allserv.kahosl.be/~burak/project.html.

Bourdais, S., Galinier, P., & Pesant, G. (2003). HIBISCUS: A constraint programming application to staff scheduling in health care. In *Lecture notes in computer science: Vol. 2833* (pp. 153–167). Berlin: Springer.

Brucker, P., Qu, R., Burke, E. K., & Post, G. (2005). A decomposition, construction and post-processing approach for a specific nurse rostering problem. In *Proceedings of multidisciplinary international conference on scheduling: theory and applications*, Aug, 2005, New York (pp. 397–406).

Brucker, P., Burke, E. K., Curtois, T., Qu, R., & Vanden Berghe, G. (2010). A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristics*, *16*(4), 559–573. doi:10.1007/s10732-008-9099-6.

Burke, E. K., Cowling, P. I., De Causmaecker, P., & Vanden Berghe, G. (2001a). A memetic approach to the nurse rostering problem. *Applied Intelligence*, *15*(3), 199–214.

Burke, E. K., De Causmaecker, P., Petrovic, S., & Vanden Berghe, G. (2001b). Fitness evaluation for nurse scheduling problems. In *Proceedings of the 2001 congress on evolutionary computation CEC2001* (pp. 1139–1146). New York: IEEE Press.

Burke, E. K., Kendall, G., & Soubeiga, E. (2003). A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, *9*(6), 451–470.

Burke, E. K., De Causmaecker, P., & Vanden Berghe, G. (2004a). Novel Metaheuristic Approaches to Nurse Rostering Problems in Belgian Hospitals. In *Handbook of scheduling: algorithms, models and performance analysis* (pp. 44.1–44.18). Boca Raton: CRC Press.

Burke, E. K., De Causmaecker, P., Vanden Berghe, G., & Van Landeghem, H. (2004b). The state of the art of nurse rostering. *Journal of Scheduling*, *7*(6), 441–499.

Burke, E. K., Curtois, T., Post, G., Qu, R., & Veltman, B. (2008). A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research*, *188*(2), 330–341.

Burke, E. K., Curtois, T., Qu, R., & Vanden Berghe, G. (2009). A scatter search approach to the nurse rostering problem. *Journal of the Operational Research Society* (accepted for publication).

Curtois, T. (2009). The nurse rostering benchmark data set of the university of nottingham. http://www.cs.nott.ac.uk/tec/NRP/.

De Causmaecker, P., & Vanden Berghe, G. (2003). Relaxation of coverage constraints in hospital personnel rostering. In *Lecture notes in computer science: Vol. 2740* (pp. 129–147). Berlin: Springer.

De Causmaecker, P., & Vanden Berghe, G. (2009). *Categorisation of personnel rostering problems*. Working Paper K.U. Leuven.

Frøyseth, H., Stølevik, M., & Riise, A. (2008). A heuristic approach for solving real world nurse rostering. In *The 7th international conference on the practice and theory of automated timetabling*, PATAT 2008, Montreal (p. 5).

Gaspero, L. D., & Schaerf, A. (2002). Multi-neighbourhood local search with application to course timetabling. In E. K. Burke & P. D. Causmaecker (Eds.), *Lecture notes in computer science: Vol. 2740. PATAT* (pp. 262–275). Berlin: Springer.

Glass, C. A., & Knight, R. A. (2010). The nurse rostering problem: a critical appraisal of the problem structure. *European Journal of Operational Research*, *202*(2), 379–389.

Hansen, P., & Mladenović, N. (2003). Variable neighborhood search. In *Handbook of metaheuristics* (pp. 145–184). Berlin: Springer.

Maenhout, B., & Vanhoucke, M. (2007). An electromagnetic meta-heuristic for the nurse scheduling problem. *Journal of Heuristics*, *13*(4), 359–385.

Maenhout, B., & Vanhoucke, M. (2008). Comparison and hybridization of crossover operators for the nurse scheduling problem. *Annals of Operations Research*, *159*(1), 333–353.

Özcan, E. (2007). Memes, self-generation and nurse rostering. In *Lecture notes in computer science: Vol. 3867* (pp. 85–104). Berlin: Springer.

Pesant, G. (2008). Constraint-based rostering. In *The 7th international conference on the practice and theory of automated timetabling, PATAT 2008*, Montreal (p. 11).

Petrovic, S., & Vanden Berghe, G. (2008). Comparison of algorithms for nurse rostering problems. In *The 7th international conference on the practice and theory of automated timetabling, PATAT 2008*, Montreal (p. 18).

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*(8), 2403–2435.

Schaerf, A., & Di Gaspero, L. (2007). Measurability and reproducibility in university timetabling research: discussion and proposals. In *Lecture notes in computer science: Vol. 3867* (pp. 40–49). Berlin: Springer.