A stepping horizon view on nurse rostering

Fabio Salassa · Greet Vanden Berghe

Received: date / Accepted: date

Abstract The present paper introduces a *stepping horizon* approach to optimisation problems whereby data can be considered static within a limited time horizon. This implies that problem instances need to be solved at certain moments in time, while imposing constraints on the subsequent period's instance. Nurse rostering can be identified as an optimisation problem for which a stepping horizon approach is recommended, whereas a static approach is suitable for academic algorithm development objectives.

In order to support this claim, the paper focuses on the *sprint* instances from the 2010 Nurse Rostering Competition. These instances represent a sufficiently realistic set of constraints while still being solvable to optimality with a general purpose solver. Two different sets of experiments are presented. First, it is shown that a static horizon approach runs the risk of generating unbalanced rosters regarding some so called *counter* constraints. A second set of experiments points at the benefits of a stepping horizon approach with respect to constraints of the so called *series* type. These two are general constraint types used as clarifying examples supporting the need for a *stepping horizon* approach. In both experimental setups, lower bounds are computed for rosters spanning more than one time horizon. The stepping horizon approach yields rosters that violate fewer constraints than those obtained in a static setting.

Keywords Nurse Rostering · Series Constraints · Counters Constraints · Stepping Horizon

F. Salassa
DAI - Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
Tel.: +39 0110907195
E-mail: fabio.salassa@polito.it

G. Vanden Berghe
CODeS - KAHO Sint-Lieven, Gebroeders De Smetstraat 1, 9000 Gent, Belgium
KU Leuven-Kulak, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
Tel.: +32 92658610
E-mail: greet.vandenberghe@kahosl.be

1 Introduction

Nurse Rostering Problems (NRPs) are encountered in almost every hospital around the world. Despite possible differences between countries due to contractual and operational regulations, the core problem is always the same: assign a working shift or a day off to each nurse of a ward on each day of a defined planning horizon (often one month) taking into account a set of constraints. The assignment problem's objective function, assessing the quality of generated rosters, is usually based on constraint violations. According to [3], nurse rostering constraints can be divided into counters and series. The counters denote all the constraints that can be evaluated by counting the appearance of certain assignments in a roster. The series constraints correspond to restrictions on successive assignments, e.g. successive working weekends, successive morning shifts, etc. The literature presents a large number of different approaches devoted to NRPs covering many aspects [8]. Some work has focused on generic approaches providing a sufficient quality level over a class of instances [3, 6]. Very fast and accurate heuristics [1] and recently also hyperheuristics [2] have been developed. In addition, exact methods are available [15], some of which exploit intrinsic peculiarities or specific knowledge about the problem [10], while others combine a metaheuristic framework and a Mixed Integer Linear Programming (MIP) solver [9]. The winning approach of the Nurse Rostering Competition (NRC) 2010 [22] is an excellent example of a hybrid mathematical approach. [14] obtained very good results for the competition instances with an adaptive neighborhood search thereby borrowing some ideas from SAT solvers.

We noticed, in the nurse rostering literature, that almost all the effort was spent on solving problems with a single compounded time horizon rather than on improving the perceived quality of rosters over a long period. It is a natural approach in academia to consider a restricted time horizon within which the information is complete. However, real hospital applications are strongly influenced by the inertia of previous periods. The working history has been modelled in [24], where a balance is made between the quality of the nurses' previous rosters and their preferences. [7, 12] model constraints induced by the previous planning period. A few roster schemes stretching out over more than one planning horizon are visualized in both papers. In addition, some data considering future timeslots (e.g. requests for days off) may have an impact on the attainable roster quality within the present planning horizon.

It appears that the rostering horizon is a crucial element to be considered when designing NRP approaches. Suppose that an optimal approach can generate a solution in limited time. It is not unlikely that this solution suffers from an imbalance in workforce assignment. Workload balancing has only rarely been identified as an objective of nurse rostering problems [19]. In case the planning period is isolated, this workload imbalance will probably be repeated when addressing future planning periods. Such results obviously prevent automated nurse rostering approaches from being acceptable in practice. A similar example is represented by the impact of shifts assigned at the end of the previous planning period w.r.t. the first days of the current planning period. Constraints on consecutive assignments are strongly affected by the presence of assignments in the period preceding the current one. Regardless of the quality of the applied algorithm, approaches based on these restricted models do not correspond to a hospital's requirements. Nevertheless, they are common in academic environments [5, 11, 23]. This consideration may, to some extent, explain the gap between academic and applied approaches to nurse rostering [13].

Some practical aspects should be included in an NRP model, whether clearly stated or not, in order to generate a repeatedly applicable automatic timetabling procedure. Considering simplified models for nurse rostering, and in particular models with an isolated planning horizon, we demonstrate some risks and show how they can be overcome without modifying the algorithm.

2 Stepping horizon

The present paper introduces the keyword *stepping horizon*, identifying the class of problems with a static time horizon, yet subject to inertia from previous periods and characterized by data concerning future timeslots that will be disclosed only as time proceeds. *Stepping horizon* approaches differ from *rolling horizon* methods [20] in that they consider a fixed time horizon and fixed data. *Rolling horizon* approaches are characterized by data uncertainty. As soon as data become available (partial) rescheduling of the current solution is performed thereby adapting the time horizon, if necessary, and the solutions to the newly available information. The ideas of a *static* versus a *rolling* and *stepping horizon* are depicted in Figures 1, 2 and 3.

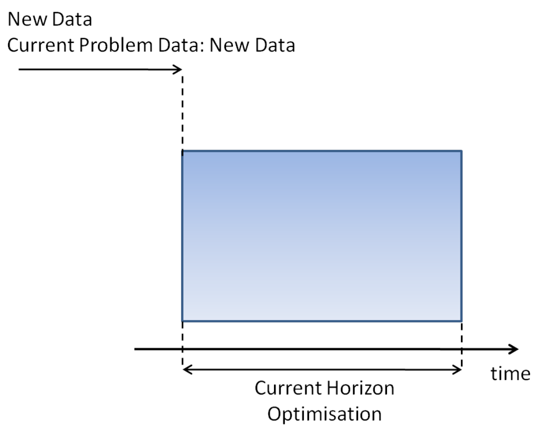

Fig. 1 Static horizon approach example

Fig. 1 presents the simple situation of a problem that is completely determined by the data corresponding to its planning horizon. The new data available are the only information used to optimise the problem. It corresponds to most of the nurse rostering instances that are publicly available for research purposes.

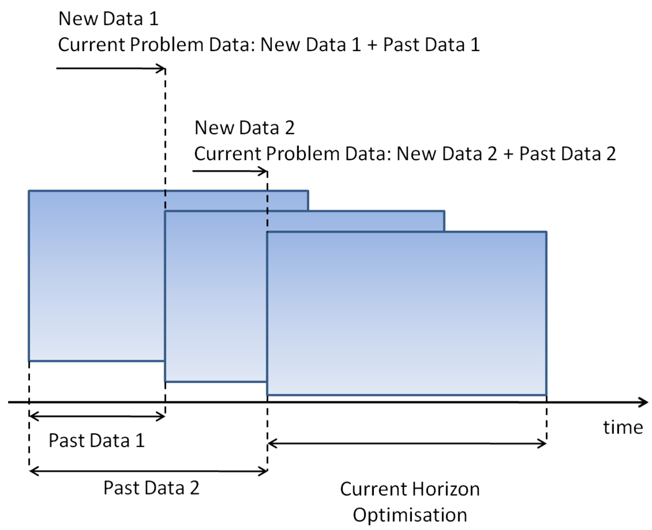

Fig. 2 Rolling horizon approach example

Fig. 2 shows that a new static problem, denoted by a rectangle, is delineated each time new information becomes available. The time horizons are overlapping. Hence, subsets of the problem's variables will take part in a number of consecutive instances to be solved. This situation is not likely to occur in nurse rostering environments but it is very common for production scheduling [20].

Fig. 3 denotes a different approach in which the information does not change as rapidly as in rolling horizon situations. The problem presented by a rectangle can be treated as a static problem. The approach should provide some mechanisms for improving the computed solution if it appears no longer valid due to data modifications. More importantly, the solution obtained for the first planning horizon, corresponding to the leftmost rectangle, imposes restrictions on the second one [7].

The problem addressed in this paper compares to Fig. 3 in that it will not adapt the time horizon when new information becomes available. Rather, it considers whichever information from the previous planning period in order to generate a roster that violates as little constraints as possible. Obviously, in case of any data disruption, e.g. an unexpected absence, the rostering algorithm should be called for sorting out the problem, without extending the original rostering horizon. As a matter of fact, reducing the planning horizon is

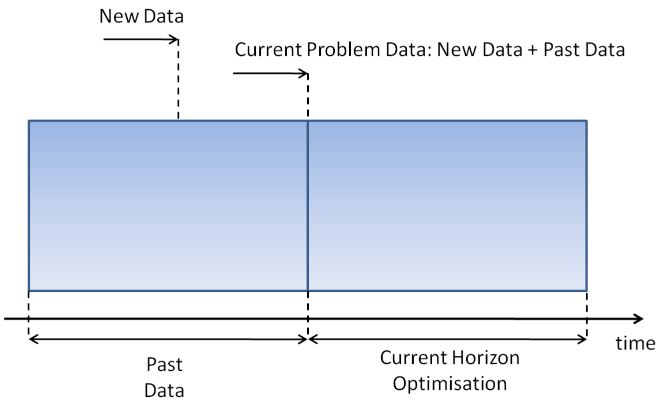

Fig. 3 Stepping horizon approach example

a more common way of dealing with disruptions [4]. Moz and Pato labelled the requirement that computed rosters sometimes need a revision due to external circumstances as the nurse rerostering problem. They modelled it as a multicommodity flow problem [17] while their later work focuses on evolutionary approaches [18]. The same authors continued developing new algorithms to the nurse rerostering problem, e.g. a bi-objective approach in [21]. Also Maenhout and Vanhoucke [16] present a genetic approach for rerostering nurse schedules. All these recent nurse rerostering approaches would fit well in a stepping horizon model.

In what follows, the importance of the stepping horizon approach is illustrated with some clear examples. The first set of experiments indicates the danger of imbalanced solutions in the long term. This is illustrated by focusing on a counter constraint and showing the potential long term effect of a small imbalance in a static roster. While balancing constraints may not be explicitly part of the problem, the results show that only limited effort is required for considering a better workload balance. In the second set of experiments, we point at the issue that series constraints can be evaluated consistently across time horizon boundaries. The results of static rosters are misleading because they appear to be better than the results of the stepping horizon approach, whereas the long term effect is again not acceptable.

3 Problem description

The problem considered here is a classical Nurse Rostering Problem where a working shift or a free day should be assigned to each nurse on each day of the planning horizon according to several contractual and operational requirements. Please note that free days are modelled as a special shift and hence shifts are of five kinds:

Late	14:30 – 22:30
Day	08:30 – 16:30
Early	06:30 – 14:30
Night	22:30 – 06:30
(Off)	not explicitly requested but needed to model the problem.

The instances are based on a given number of nurses, i.e. 10 for the *sprint* instances addressed in this paper. All the *sprint* instances have their planning horizon set equal to 28 days, roughly referring to a period of one month.

A set of hard constraints must be satisfied, otherwise solutions would be infeasible. The hard constraints include

– demand cover: all the shifts demanded on a day of the planning period must be assigned to the exact number of nurses
– exactly one shift (working or free) must be assigned to each nurse on each day.

The instances considered incorporate a large number of soft constraints that, when violated, contribute to the objective function value by weighted penalties. The problem's objective function should be minimised. The soft constraints of the problem belong to either the counter or the series category [3]. A limited selection of the soft constraints is presented below.

Counters

– maximum and minimum number of shifts that can be assigned to nurses
– maximum and minimum number of free days
– day off or shift off requests

Series

– maximum and minimum number of consecutive working days
– unwanted patterns (such as a Night shift followed by an Early shift).

For the complete problem definition and a detailed description of the constraints, refer to [11][1]. The computational results and instance files are available at `www.kuleuven-kulak.be/nrpcompetition`.

The problem considered at the competition can be modelled as an Integer Linear Problem. Indeed, with n nurses, m days in a planning horizon and s different shifts, it is sufficient to introduce a set of 0/1 variables $x_{i,j,k}$ ($i = 1..n$, $j = 1..m$, $k = 1..s$) indicating if nurse i is assigned to shift k on day j of the roster horizon. Correspondingly, sets of integer variables represent the different penalties that can be associated with soft constraint violations. As an example two different constraints of the problem model are discussed in detail. First, the constraint related to the maximum number of assignments is an example of how other counter type constraints are modelled. Second, the constraint determining the maximum number of allowed consecutive working

[1] `www.kuleuven-kulak.be/~u0041139/nrpcompetition/nrpcompetition_description.pdf`

days represents all other constraints of the series type. Let $MaxAssignments$ be the maximum number of assignments for a nurse in the considered period and $PenaltyMA_i$ the integer (variable) penalty caused by the total number of exceeding assignments over the horizon for a nurse i.

Let W be the set of working shifts and D the set of days in the considered time horizon. Let lim be the limit on the number of consecutive working days defined by the problem instance and let $PenaltyMW_{i,j}$ be the binary (variable) penalty caused by a working day on day j exceeding the limit for a nurse i. This leads to the following inequalities:

$$\sum_{j}^{D} \sum_{k}^{W} x_{i,j,k} \leq MaxAssignments + PenaltyMA_i \qquad \forall i = 1, \ldots, n$$

$$\sum_{k}^{W} \sum_{t=0}^{lim} x_{i,j+t,k} \leq lim + PenaltyMW_{i,j} \quad \forall i = 1, \ldots, n, \quad j = 1, \ldots, m - lim$$

The first constraint determines a maximum number of assigned working shifts. When single time horizons are considered, constraints like the one described above can be tricky because nurses can have excess assignments over successive planning periods. The second linear inequality conditions the maximum number of consecutive working days. A penalty proportional to the excess value is issued whenever that value is exceeded. Clearly, these constraints will not be violated at the beginning of a time horizon if no data about previous periods is considered, while with the stepping horizon approach past assignments do have an effect on the current time horizon.

The objective function to be minimised eventually is the weighted sum of penalties over the entire set of nurses and constraints. All instances have been implemented with the XPRESS MOSEL modelling language. XPRESS (v. 21.01.06) has been used to solve problem instances on an Intel Core2 Duo CPU @ 2.13 GHz with 4 GB of RAM memory.

The experimental setup serves the purpose of indicating the potential drawbacks of static nurse rostering approaches, which are very common in the academic literature, compared to the stepping horizon approaches we advocate. In order to provide a clear example of the presented issues, the ideas were tested on a few instances from the Nurse Rostering Competition [11]. This choice is motivated by the fact that 1) these instances have become benchmarks for nurse rostering research and 2) some of the instances are fairly easy in that they are solvable, in less than 120 seconds, with a MIP solver. The optimality of these instances can thus be certified by the solver and validated by the evaluation algorithm provided by the competition organizers.

The stepping horizon idea is simulated by solving each instance and making sequences of the roster solutions obtained for one time horizon, here corresponding with 28 days, into a multi-period roster. In real hospital environments, the availability of the nurses and the personnel demand cannot be considered constant over the entire period. Nevertheless, the results of these simple experiments are convincing and support the stepping horizon approach.

4 Numerical example

The present section provides numerical results of the *stepping horizon* approach, applied to nurse rostering instances. The basic idea is to test the impact of both series and counters contraints on the solutions' quality when more than one time horizon is considered. This is simulated by considering preceding timeslots as key inputs to the current optimization problem. In other terms, results of past optimisations are added trying to adhere more to real-life applications.

4.1 Counter constraints, balanced workload

The focus of the first set of experiments is on workload balancing for which evaluations of counter constraints provide sufficient information. Table 1 presents results for the 10 sprint instances. The result obtained for one instance was copied into a large roster 12 times the original time horizon's size. The imbalance of working shifts between nurses is denoted by a very simple quality indicator, namely the largest difference of the total number of assigned working shifts, measured among all the nurses. Assume for example that the optimal solution assigns k working shifts to nurse i and l working shifts to nurse j, then the difference between these two nurses' assignments equals $|k - l|$. Although it is not always explicitly requested, we assume that a balanced number of working shifts among nurses is desirable and contributes to a balanced overall workload. Table 1 shows that the optimal solution for *sprint*01 is a roster in which one or more nurses have 96 assignments over a year, while at least one other nurse has 216 assignments. The term *maximum imbalance* is introduced. It refers to the maximum difference between the work assignments of nurses over a given period. We would like to underline that we are here considering the total number of working shifts assigned to nurses, not yet taking into account the understandable preferences between the different shifts for nurses. The maximum imbalance of a solution to the *sprint*01 instance is 10 shifts for a monthly roster, which produces an imbalance of 120 shifts when replicating the solution over 12 consecutive months.

In the second set of experiments, an additional hard constraint was added to the problems so that the maximum imbalance between the working shifts of any two nurses within one month is at most 3. This means that the nurse with the heaviest workload has to work at most 3 shifts more than the least active nurse, considering one rostering period. As a consequence, the instances are no longer the same as the original ones. Nevertheless, the solutions are evaluated with the same objective function. Alternatively, the maximum imbalance constraint could have been modelled as a soft constraint. Without understanding how the other constraints' weights were set, it would be hard to set an appropriate value to the new imbalance constraint's weight. The authors opted to avoid search space distortion by modelling the new

INSTANCE	MIN (working shift)	MAX (working shift)	Objective over one year	Maximum imbalance
sprint01	96	216	672	120
sprint02	96	228	696	132
sprint03	108	240	612	132
sprint04	144	204	708	60
sprint05	120	252	696	132
sprint06	144	204	648	60
sprint07	108	240	672	132
sprint08	144	240	672	96
sprint09	108	228	660	120
sprint10	132	228	624	96

Table 1 Optimal solutions for the sprint instances (12 replicated monthly rosters) with indications of workload imbalance.

constraint as hard. In future work, the impact of a soft versus a hard balance constraint is an interesting subject to investigate.

Table 2 shows the computational results for the same instances. As was expected, the imbalance is reduced considerably within one roster. The result obtained for *sprint*01 reveals that people assigned to the least number of working shifts perform 13 shifts per 28 days, whereas the people working most perform 16 shifts per 28 days. When replicated 12 times, the overall imbalance equals 36 which is much better than the imbalance of 120 resulting from the experiments in Table 1. The introduction of the balance constraint has a limited negative effect on the overall roster quality. The value of the yearly objective function increased from 672 to 744 for *sprint*01. From a computational point of view, adding a hard constraint such as the one we have introduced, makes these *sprint* instances more difficult to solve to optimality. However, the computation time never exceeds 60 seconds.

INSTANCE	MIN (working shift)	MAX (working shift)	Objective over one year	Maximum imbalance
sprint01	156	192	744	36
sprint02	156	192	780	36
sprint03	156	192	624	36
sprint04	156	192	708	36
sprint05	168	204	708	36
sprint06	168	192	648	24
sprint07	156	192	672	36
sprint08	156	192	672	36
sprint09	156	192	684	36
sprint10	168	204	636	36

Table 2 Optimal solutions for the sprint instances subject to an additional constraint restricting the maximum difference between people's shift assignments to 3 working shifts per month (12 replicated monthly rosters).

The experiments reported in Table 3 go beyond the previous ones in that the balance constraint is much stricter. The overall results of this previous set of experiments cannot be considered completely satisfactory. It is, in fact, clear that in the worst case a nurse can work 36 shifts more than the "luckiest" one, which roughly corresponds to a difference of almost two full time months of work. This is definitely unwanted. It seems not to be sufficient to introduce a simple working imbalance constraint because the imbalance can still be very large over a period of one year. The new constraint is formulated such that the maximum shift assignment imbalance between personal rosters is 3 within a roster horizon as well as over the entire period of 12 repetitive roster horizons. Again, the results are obtained by solving the problem once for one roster period and repeating it 12 times, which is computationally a small effort. Considering the illustrative example of *sprint*01, it can be noticed that the yearly shift assignment imbalance between people is at most 3, which is an excellent result. The drawback is that the overall roster quality over a 12 month period is 840, which is worse than the result of the experiments conducted with a monthly imbalance constraint only. Clearly, other constraint violations compensate for a better balance of the number of working shifts.

INSTANCE	MIN (working shift)	MAX (working shift)	Objective over one year	Maximum imbalance
sprint01	181	184	840	3
sprint02	181	184	875	3
sprint03	181	184	684	3
sprint04	181	184	757	3
sprint05	181	184	766	3
sprint06	181	184	683	3
sprint07	181	184	722	3
sprint08	181	184	706	3
sprint09	181	184	733	3
sprint10	181	184	716	3

Table 3 Optimal solutions subject to an additional constraint restricting the maximum difference within a single roster horizon and over all 12 replicated roster horizons to be at most 3 working shifts

4.2 Series constraints

Series constraint restrict the number of consecutive working days, free days, working weekends, etc. Similar to the experiments reported in Section 4.1, solutions are obtained with the static problem definition of the NRC instances as well as with the stepping horizon approach. The latter incorporates data from previous and future roster horizons into the problem to be solved. Little effort was spent on developing an efficient MIP model. Some of the constraints incorporated in the competition's instances were hard to model and to verify. The number of variables appears huge for problems considering multiple planning horizons at once. It is definitely worth concentrating on improving the model in future research. Given the straightforward MIP model from Section 3, instances of limited size can be loaded by the XPRESS solver. They correspond to period stretches of five months, which are all solvable within 10 seconds.

Preliminary experiments with the MIP solver generated a memory exception for planning horizons exceeding five months. We therefore restricted the horizon to five consecutive periods, without losing generality. Table 4 reports the results covering these five consecutive periods, from now on referred to as months. Particularly, column 2 in the table shows objective values of solving a planning period of five months as a whole. In other words, a planning period of five months has been considered instead of a single month with the objective function calculated over the complete horizon. Column 3 shows the results achieved in terms of objective function values, replicating five times the optimal value of a single month. These solutions were generated without taking into account constraints overlapping different months, such as the series constraints. Therefore the overall objective value is worse than in the previous case even for solutions that are optimal for a single month. The last column provides the achievements when previous periods are considered fixed. These solutions have been generated by optimally solving one month but considering the inertia of past periods as follows. First a problem with a planning horizon of one month is solved to optimality. Then the second month is again solved to optimality but the time horizon considered is now two months, of which the first one is represented by the solution achieved in the past step. In practice a model considering two months is generated and the variables related to the first one are fixed to the values obtained in the previous step. This procedure is repeated up to five months each time considering all the previous months. The depicted values reveal that the best would be to solve a large horizon in one go. This is almost impossible because data are available only as timeslots pass. Even if suboptimal, a more interesting procedure than solving only single time horizons, is the stepping horizon approach. A fairly good solution can be generated when also considering constraints overlapping months. That is obtained by solving single time horizons while basing the current solution on what has happened in the previous periods. In the authors' opinion this procedure should always be conducted when optimising nurse rostering instances.

INSTANCE	Obj. Fun. one go	Month by month Obj over 5 months	Stepping horizon
sprint01	276	332	287
sprint02	286	306	297
sprint03	251	287	262
sprint04	284	315	292
sprint05	290	310	297
sprint06	266	314	272
sprint07	280	312	287
sprint08	276	296	280
sprint09	271	307	281
sprint10	264	304	271

Table 4 Optimal solutions subject to series constraints

5 Conclusion and future work

Academic problems tend to concentrate on static instances representing an isolated planning period. While a static approach is very common in nurse rostering research, the present paper focused on drawbacks over a long term work stretch. The idea of a stepping horizon was introduced in order to model problems in such a way that they compare better to real practice in hospitals. A stepping horizon incorporates characteristics from a static as well as from a rolling horizon.

A set of simple experiments was set up so as to indicate the strength of a stepping horizon approach. Instances have been taken from the first International Nurse Rostering Competition. The smallest *sprint* instances of that competition were solvable to optimality with a straightforward MIP approach and these optimal solutions allowed to make strong quality claims.

The experiments concentrated on two sets of roster qualities. Static horizons often proved to induce significant imbalance between individual nurses' assignments. This set of experiments concentrated on counter type constraints. The introduction of an additional balance constraint showed not to be sufficient to cope with the intrinsic imbalance of splitting a long term problem into isolated small problems. The stepping horizon approach provides an alternative in that its long term effect on balanced workload is advantageous, at the expense of potentially reducing the quality within the present planning horizon.

Besides counter constraints, a second set of experiments demonstrated that series constraints can also have a strong impact on the roster quality of subsequent rosters, when optimising static rosters only. In practical applications of nurse rostering it is inevitable that series constraints will overlap the monthly planning horizons. Hence, static horizon approaches are inadequate while a stepping horizon approach offers a manner to cope with series constraints across planning horizon borders.

Both sets of experiments produced somewhat poorer results within a single time horizon, whereas the long term effect was significantly better.

The tests were conducted on the smallest NRC instances only for computational reasons. Exactly the same experiments can be translated to the larger instances or to complex real experiments, for which dedicated algorithms are more appropriate than MIP solvers. The positive effect of the stepping horizon approach is expected to be stronger in case of a large set of complex constraints.

Future work will be dedicated to investigating the impact on other counter constraints than the total number of assigned shifts. The implications of the stepping horizon approach on the overall objective including all the counter and series constraints will be investigated too. In addition, the design of appropriate objective functions will be investigated so that results of a stepping horizon approach generate the best possible long term effect. These examinations will preferably be conducted using the mathematical solver, if the model can be improved sufficiently. Otherwise, heuristics for nurse rostering are a reasonable alternative.

Another interesting aspect to be studied is the benchmark of the stepping horizon approach compared to other ways of dealing with long term horizons, such as shift rotation schedules.

Sets of real rostering problems with a given working history will be collected to support future research.

References

1. J.F. Bard and H.W. Purnomo. Real-time scheduling for nurses in response to demand fluctuations and personnel shortages. In E.K. Burke and M. Trick, editors, *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, PATAT, pages 67–87, Pittsburgh, August 2004.
2. B. Bilgin, P. Demeester, M. Misir, W. Vancroonenburg, and G. Vanden Berghe. One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics*, 18(3):401–434, 2012.
3. B. Bilgin, P. De Causmaecker, B. Rossie, and G. Vanden Berghe. Local search neighbourhoods to deal with a novel nurse rostering model. *Annals of Operations Research*, 194(1):33–57, 2012.
4. E.K. Burke, P. Cowling, P. De Causmaecker, and G. Vanden Berghe. A memetic approach to the nurse rostering problem. *Applied Intelligence, Special issue on Simulated Evolution and Learning*, 15:199–214, 2001.
5. E.K. Burke, T. Curtois, R. Qu, and G. Vanden Berghe. A scatter search approach to the nurse rostering problem. *Journal of the Operational Research Society*, 61:1667–1679, 2010.
6. E.K. Burke, T. Curtois, R. Qu, and G. Vanden Berghe. A time pre-defined variable depth search for nurse rostering. *INFORMS Journal on Computing*, to appear.
7. E.K. Burke, P. De Causmaecker, S. Petrovic, and G. Vanden Berghe. Fitness evaluation for nurse scheduling problems. In *Proceedings of the Congress on Evolutionary Computation (CEC2001)*, pages 1139–1146, Seoul, Korea, May 27-30 2001. IEEE Press.
8. E.K. Burke, P. De Causmaecker, G. Vanden Berghe, and H. Van Landeghem. The state of the art of nurse rostering. *Journal of Scheduling*, 7(6):441–499, 2004.
9. F. Della Croce and F. Salassa. A variable neighborhood search based matheuristic for nurse rostering problems. Technical report, Politecnico di Torino – http://dl.dropbox.com/u/24916303/TR-01-02-2012.pdf, 2012.
10. C.A. Glass and R.A. Knight. The nurse rostering problem: A critical appraisal of the problem structure. *European Journal of Operational Research*, 202:379–389, 2009.
11. S. Haspeslagh, P. De Causmaecker, M. Stolevik, and A. Schaerf. The first international nurse rostering competition 2010. *Annals of Operations Research*, 194(1):59–70, 2012.
12. A. Ikegami and A. Niwa. A subproblem-centric model and approach to the nurse scheduling problem. *Mathematical Programming*, 97(3):517–541, 2003.
13. D.L. Kellogg and S. Walczak. Nurse scheduling: From academia to implementation or not? *Interfaces*, 37(4):355–369, 2007.
14. Z. Lu and J.K. Hao. Adaptive neighborhood search for nurse rostering. *European Journal of Operational Research*, 218(3):865 – 876, 2012.
15. B. Maenhout and M. Vanhoucke. Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of Scheduling*, 13:77–93, 2010.
16. B Maenhout and M Vanhoucke. An evolutionary approach for the nurse rerostering problem. *COMPUTERS & OPERATIONS RESEARCH*, 38:1400–1411, 2011.
17. M. Moz and M. Pato. An integer multicommodity flow model applied to the rerostering of nurse schedules. *Annals of Operations Research*, 119:285–301, 2003.
18. M. Moz and M. Pato. A genetic algorithm approach to a nurse rerostering problem. *Computers & Operations Research*, 34:667–691, 2007.

19. D. Ouelhadj, S. Martin, P. Smet, E. Özcan, and G. Vanden Berghe. Fairness in nurse rostering. Technical report, University of Portsmouth, 2012.
20. D. Ouelhadj and S. Petrovic. A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12:417–431, 2009.
21. M. Pato and M. Moz. Solving a bi-objective nurse rerostering problem by using a utopic pareto genetic heuristic. *Journal of Heuristics*, 14:359–374.
22. C. Valouxis, C. Gogos, G. Goulas, P. Alefragis, and E. Housos. A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research*, 219(2):425 – 433, 2012.
23. M. Vanhoucke and B. Maenhout. NSPLib - a nurse scheduling problem library: A tool to evaluate (meta-)heuristic procedures. *Operational research for health policy: Making better decisions*, pages 151–165, 2007.
24. M. Warner. Nurse staffing, scheduling, and reallocation in the hospital. *Hospital & Health Services Administration*, pages 77–90, 1976.