A Variable Neighborhood Search based Matheuristic for Nurse Rostering Problems

Federico Della Croce · **Fabio Salassa**

Abstract A practical nurse rostering problem, which arises at a ward of an Italian private hospital, is considered. In this problem, it is required each month to generate the nursing staff shifts subject to various requirements. A matheuristic approach is introduced, based on a set of neighborhoods searched by a commercial integer programming solver within a defined global time limit. Generally speaking, a matheuristic algorithm is a heuristic algorithm that uses non trivial optimization and mathematical programming tools to explore the solutions space with the aim of analyzing large scale neighborhoods. The solutions computed by the proposed procedure are compared to the solutions achieved by the pure solver within the same time limit. The results show that the proposed solution approach outperforms the solver in terms of solution quality.

Keywords Variable neighborhood search · Matheuristics · Timetabling · Nurse Rostering Problem

1 Introduction

The paper pertains to a nurse rostering problem, which occurs at a private hospital located in Turin, Italy. The problem consists in optimally assigning a working shift or a day off to each nurse, on each day of a month, according to several contractual and operational requirements. The problem belongs to the family of timetabling problems [5], [6] and [16]. Many works have been published on the nurse rostering problem since the pioneering works of Warner [17] and Miller [14] and the proposed approaches are mainly based on constraint programming and metaheuristic procedures (see for instance [4] and [10]). In [1], a nurse rostering problem similar to the one considered here

F. Salassa
DISPEA - Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
Tel.: +39 0110907206
Fax: +39 0110907299
E-mail: fabio.salassa@polito.it

F. Della Croce
DAI - Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
Tel.: +39 0110907059
Fax: +39 0110907099
E-mail: federico.dellacroce@polito.it

was tackled by means of a metaheuristic approach. In particular, in the metaheuristic framework, a simple and effective metaheuristic for combinatorial and global optimization, called variable neighborhood search (VNS) has been successfully applied to solve both general mixed integer programming problems (MIPs) [15] [11] and timetabling problems [3].

Combinatorial optimization problems, such as timetabling problems, usually may be formulated in different ways. Mathematical programming formulations such as integer programs are very popular, since they make possible the use of general-purpose Integer Linear Programming (ILP) solvers, independently of problem specific properties. However, in some hard cases, for instance when the number of variables becomes too large, solvers might not be an adequate choice [12].

Metaheuristic approaches, instead, rely on formulations adapted to be able to handle the special concrete combinatorial optimization problems that have to be faced thus loosing the advantage of working in a generic modelling framework, in fact, even slight changes of the problem description can cause a complete redesign of data structures and algorithms. Indeed, the adaptation of the approach proposed in [1], even if the constraints set was quite similar, proved to be quite tricky. On the other hand, the improvement of ILP solvers in the recent years have made them already competitive (by simply adding a time limit) to metaheuristic approaches in the search of suboptimal solutions within limited CPU time. Further, whenever an ILP model is available, the addition of new constraints or the modification of the objective function is often straightforward.

Recently [9], [13] a new topic has attracted the attention of the community of researchers, the so called Matheuristics. Matheuristics are heuristics algorithms made by the inter operation of metaheuristics and mathematical programming techniques. An essential feature is the exploitation in some part of the algorithms of features derived from the mathematical model of the problem of interest [2].

The aim of the present work is, then, to demonstrate the applicability, presenting comparison results, of a VNS based matheuristic for solving hard timetabling problems. As ILP solver we use XPRESS IVE version 1.19 from Fair Isaac Corporation.

The article is organized as follows. In Section 2 the problem is described. In Section 3 the matheuristic solution approach is described. Section 4 is devoted to computational testing and comparison. Section 5 concludes the paper with final remarks.

2 Problem Description

The problem considered here is based on a real situation encountered in a ward of a private hospital in Turin. At the end of each month, the nurses' shifts for the following month must be scheduled. The ward is made up of a given number of hired nurses, but the timetable of these nurses is not always sufficient to cover the legal minimum of personnel presence in every shift. For that reason the hospital management can make use of freelance nursing staff to cope with the demand of personnel inducing however an extra cost. The problem consists, thus, in optimally assigning a working shift or a day off to each nurse, on each day, according to several requirements in order to reduce the outsourced work. In the considered case all nurses have the same skills' level except for newly recruited personnel. This issue is resolved imposing a period of shadowing between specific nurses or setting the incompatibility in the same shift for specific workers (the newly recruited nurses).

Table 1 Shift types and demand

	Demand						
Shift type	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Morning	3	3	3	3	3	3	3
Afternoon	3	3	3	3	3	3	3
Night	2	2	2	2	2	2	2

In that hospital, the nurses timetabling was generated manually every month by the hospital management, but it was too time consuming leading often to unfair timetables and to an overuse of freelance staff. Indeed the hospital management wanted to try other approaches to overcome those issues and decided then to test an automated timetabling generation procedure in order to better manage the hired personnel and to reduce, as much as possible, the number of outsourced shifts. We were then asked to realize a tool to cope with these requirements.

2.1 Types of shifts

There are five different kinds of shifts considered in the hospital: three of them are working shifts and two are off-duty shifts. During the last week of each month, the nurses can provide to the management their requests of off-duty shifts. In general, no request of specific working shifts are accepted by the hospital management. The working shifts, all lasting 8 hours, are:

- Morning shift (M)
- Afternoon shift (A)
- Night shift (N)

The off-duty shifts are:

- Rest (R)
- Off (O)

The difference between Rest and Off shifts is basically that a number of Rest shifts can be both requested by staff and assigned by the management, Off shifts can only be requested by the nursing hired staff.

2.2 Constraints Description

Hereafter is presented the complete list of constraints integrated in the support tool realized for the hospital.

(C1) The number of R shifts per month must be equal to a predefined value provided by the management.
(C2) The nurses' requirements related to O shifts and R shifts must be satisfied.
(C3) A nurse cannot work consecutively for more than D days.
(C4) Specific shifts types must be allocated in sets of minimum L_k and maximum S_k consecutive days for each shift type k.

(C5) After a set of N shifts, there must be a given number E of R shifts.
(C6) An interval of at least P days (working or not) must occur between two N shifts sets.
(C7) A minimum number of nurses must be guaranteed for each working shift. This parameter, provided by the management, may differ from shift to shift and from day to day. An example of personnel demand is summarized in Table 1.
(C8) Forbidden sequences of shifts (e.g. N-R-N, N-M, N-A, N-O, etc.).
(C9) A balanced assignment of M, A and N shifts must be guaranteed among the nurses.
(C10) Working shifts and days off during weekends mus be evenly assigned.
(C11) At least two off-duty weekends for each nurse/per month.
(C12) All constraints must be respected considering the last days of the previous month.
(C13) Some nurses must (not) work together in particular set of days and for particular shifts (shadowing/incompatibility period).

The considered problem can be easily formulated as an ILP model. Indeed, with n nurses and m days in a month, it is sufficient to introduce a set of 0/1 variables denoted $x_{i,j,k}$ ($i = 1, ...n$, $j = 1, ...m$, $k = 1, ...5$) indicating if nurse i is assigned to shift k ($k = 1$: morning, $k = 2$: afternoon, $k = 3$: night, $k = 4$: rest, $k = 5$: off-duty) on day j. Correspondingly, an integer variable $y_{j,k}$ is introduced indicating the number of freelance nurses used on working shift k on day j. The objective function is then

$$\min \sum_{j=1}^{m} \sum_{k=1}^{3} y_{j,k}.$$

We do not present here the complete ILP formulation (see [7] for details) but just present as an example the formulation of constraint C7 related to the morning shift. If we denote by M_j the requirement of nurses for the morning shift of day j, we have

$$\sum_{i=1}^{n} x_{i,j,1} + y_{j,1} \geq M_j \quad \forall j = 1, ...m.$$

For the considered hospital ward, we have $n = 10$. Correspondingly, there are approximately 90 integer variables $y_{j,k}$ and 1200 binary variables $x_{i,j,k}$ (the off-duty shifts are pre-determined). Also, there are approximately 2000 constraints.

3 The VNS Matheuristic Based Approach

The pure ILP model solved by means of the ILP solver (XPRESS) already gave satisfactory results so that the hospital management decided after the very first results to stop creating nurse timetables manually, considering the results of the ILP solver much better in terms of required outsource shifts (that does not consume personnel time with respect to the generation of the timetable). However, the hospital management wanted also to generate more than one timetable, for the ward under analysis, considering, for example, different weekly demands of nurses.
During the testing of that feature for the support tool we found that, sometimes, the considered ILP solver was not performing sufficiently well as for some problems the

gap between upper bound (the feasible solution obtained within the considered time limit) and lower bound (the bound given by the solver at the end of the considered time limit) was significant. Hence, a new matheuristic approach based on VNS was proposed to be compared to the ILP solver.

3.1 Description of the matheuristic approach

The core of our approach is made by two further constraints: the first one states that the number of variables $x_{i,j,k}$ that can change their value from the current solution is less than a parameter value K, that is the neighborhood size is controlled by parameter K. Indeed, K is the maximum Hamming distance between the current solution and all feasible solutions of the neighborhood, or, in other words, K is the maximum number of changing shifts in the new solution with respect to the incumbent solution.
The second constraint states the structure of the neighborhoods. In particular we have defined 12 types of neighborhoods, for the considered problem, each of different size. For each of these subproblems, the ILP solver used as a black box is applied searching for the optimal solution (limiting though the search by means of a given time limit). In order to escape from possible local minima, we integrate in our approach, as an additional feature, the use of neighborhhod structures not only limited by the value K but also by the value 2*K, expanding, in that way, the search space thus generating globally 24 different neighborhoods.

We have used the variable fixing method for generating our neighborhoods defining $N_{J_g,T}(x)$ as the structure of neighbors of x.
In particular $g = 1..3$ and $J = 1..30$ days of the current month where $g = 1$ represents the first 10 days of the month, $g = 2$ the second decade and $g = 3$ the last ten days. $T = 1..5$ represents the shift types and in our matheuristic are taken into account with the sequence order:

- fix all T shifts of all i nurses for a period J_g of days and K value
- fix all T shifts of all i nurses for a period J_g of days and K2 value
- fix $T = 1$ shift of all i nurses for a period J_g of days and K value
- fix $T = 2$ shift of all i nurses for a period J_g of days and K value
- fix $T = 3$ shift of all i nurses for a period J_g of days and K value
- fix $T = 1$ shift of all i nurses for a period J_g of days and K2 value
- fix $T = 2$ shift of all i nurses for a period J_g of days and K2 value
- fix $T = 3$ shift of all i nurses for a period J_g of days and K2 value

We decided not to include the off-duty shifts because they are mainly requested by nurses so a priori fixed in the model.
The mixing of these two constraints generates a sequence of increasing size neighborhoods at each iteration. Note that after adding one (or several) constraints the resulting MIP has the same structure as the original MIP but a smaller solution space.
The strength of a matheuristic procedure can be seen also in another relevant aspect: the effort spent on the generation of different neighborhoods and on the analysis of their quality it is noticeably lower than the majority of pure heuristics or metaheurisics approaches as we had to add two constraints to the original model to implement our neighborhood structure. Hereafter is the pseudo code of our matheuristic procedure.

Pseudocode

```
Begin
(1) choose an initial feasible solution z
(2) iter:=1
(3) while iter <=24 do
  (4) |z - x|<=K(iter)
  (5) set the variables belonging to N(iter) to their corresponding values in z
  (6) solve subproblems within local time limit
  (7) if f(x) < f(z) then do
     (8) update solution
     (9) iter:=1
  (10) else do
     (11) iter:=iter+1
  (12) end if
  (13) if the overall CPU time is greater than 3600 s. or iter>24 then
     (14) EXIT
  (15) end if
(16) end-do
End
```

3.2 Implementation

It is important to underline that the difference between the current solution and the incumbent solution, as our decision variables are binary, corresponds to the Hamming distance between two strings of [0,1] values and, in our approach, that distance must be lower than K. It is possible to model this feature in a very compact way by taking advantage of the linear programming formulation. For instance, by denoting with $z_{i,j,k}$ the value of the $x_{i,j,k}$ variables in the incumbent solution, the Hamming distance constraint can be modeled as follows.

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{5} z_{i,j,k}(1-x_{i,j,k}) + (1-z_{i,j,k})x_{i,j,k} \leq K.$$

3.3 Parameters Settings

The parameters present in our matheuristic are the value K, and time T for each iteration. The K values used in all our tests are K1= 50 and K2=100. Even though these values of K could appear very large, they were the ones that gave best results in preliminary testing. This can be explained by the strong structure of the ILP models related to timetabling problems. The global time stopping criterion, as told before, has been set to 3600 seconds which has been considered by hospital management a reasonable time window to wait for solutions of hard instances. In order to evaluate the quality of our approach, we decided to generate a number of instances similar to the real ones. We also decided to generate different size instances for testing in order to widen the possibilities of application of the approach. For that reason we used different time limit at each iteration with relation to the different sizes of the models.

1. Instances with 10 hired nurses - time limit = 60 seconds for each iteration

2. Instances with 20 hired nurses - time limit = 120 seconds for each iteration
3. Instances with 30 hired nurses - time limit = 240 seconds for each iteration

3.4 The Instances Generation Process

This paragraph presents a short description of the rules used to randomly generate instances[1]. The random generation concerned:

1. the last 5 days of the previous month, respecting all constraints, i.e. without any schedule violation;
2. the requests of R shifts and O shifts of the current month are generated nurse by nurse: for each nurse the requests are drawn from a uniform distribution according to the following probabilities:
   - 60% no requests
   - 5% 15 days of O shifts
   - 10% seven days of O shifts
   - 20% three days of R shifts
     - 50% three single R shifts
     - 50% two consecutive R shifts and a single R shift.
   - 5% two consecutive R shifts
3. shadowing or incompatibility constraints (from 0 to 4 w.r.t. nurses number);
4. different nurses demand structures (3-3-2; 4-3-2; 3-3-3; 3-2-1 mixing also weekday and weekend demands) multiplied by the size of the instances (e.g. 30 nurses, request 9-9-6).

4 Computational results

The proposed procedure was tested on 20 instances generated as mentioned before. The approach was implemented directly in XPRESS and tested on a Pentium IV Quad Core at 2.4 GHz. The results and the comparison with XPRESS with the time limit of 60 minutes are presented in Table 2. In the table, the first column depicts the instance size and name, the second column depicts the lower bound computed by XPRESS after 60 minutes of CPU time and the third and fourth column indicate the cost function values (the total number of outsourced shifts) of the solutions computed by MathVNS and the pure XPRESS solver respectively within a time limit of 60 minutes.
From this table we notice that the MathVNS procedure reaches solutions that are better than or equal to those of the pure Xpress solver in all cases but one. These results indicate that the proposed MathVNS approach is a viable option in handling nurses rostering problems.

5 Concluding Remarks

A VNS based matheuristic approach was proposed for a real life nurse rostering problem. The local search steps work on a smaller solution space specifying that at most

[1] Instances are available upon request from authors

Table 2 Comparing MathVNS vs XPRESS

Instance size/name	LP	MathVNS (1h)	Xpress (1h)
10n*1	43	54	54
10n*2	71	71	72
10n*3	9	9	9
10n*4	21	27	27
10n*5	72	80	80
10n*6	44	53	53
10n*7	24	28	30
10n*8	40	43	46
10n*9	21	26	27
10n*10	74	81	82
20n*1	51	68	71
20n*2	141	141	141
20n*3	56	64	67
20n*4	88	102	108
20n*5	35	46	50
30n*1	94	103	107
30n*2	139	158	161
30n*3	73	83	85
30n*4	35	45	45
30n*5	135	161	160

K decision variables can be complemented and fixing iteratively subsets of them. To do so we have exploited the strength of the mathematical programming formulation making use of a compact modellization of the Hamming distance between two strings and drawing advantage from the capability of generating and analyzing different neighborhoods' structures and sizes in a very short time. The proposed procedure shows a very good behavior in terms of solutions quality with CPU time limit of 60 minutes as presented by the achieved results. A software has been developed which is currently under testing in the hospital ward. This software can handle different nurse planning scenarios with different personnel requirements and different nurses' requests at the same time.

References

1. Bellanti F., Carello G., Della Croce F., Tadei R., A greedy-based neighborhood search approach to a nurse rostering problem, European Journal of Operational Research, 153 , pp. 28-40, (2004).
2. Boschetti M.A., Maniezzo V., Roffilli M., Bolufe Rohler A., Hybrid Metaheuristics, 6th International Workshop, pp., 171-177, Spriger,(2009).
3. Burke E., De Causmaecker P., Petrovic S., Vanden Berghe G., Variable neighborhood search for nurse rostering problems. In Metaheuristics: Computer Decision-Making, Applied Optimization, 86, Kluwer, pp.153-172, (2004).
4. Burke E. K., De Causmaecker P., Vanden Berghe G., Van Landeghem H., The State of the Art of Nurse Rostering, Journal of Scheduling, 7, pp. 441-499,(2004).
5. De Werra D., An introduction to timetabling, European Journal of Operational Research, 19, pp. 151-162, (1985).
6. De Werra D., The combinatorics of timetabling, European Journal of Operational Research, 96, pp. 504-513, (1997).
7. Della Croce F., Salassa F., A VNS based Matheuristic Procedure for Nurse Rostering Problem, Internal Report, D.A.I. Politecnico di Torino, (2009).
8. Ernst A.T., Jiang. H., Krishnamoorthy M., Sier D., Staff scheduling and rostering: A review of applications, methods and models, European Journal of Operational Research, 153, pp. 3-27, (2004).

9. Fischetti M., Lodi A., Local Branching, Mathematical Programming B, 98, 23-47, (2003).
10. Grobner M., Wilke P., Buttcher S., A Standard Framework for Timetabling Problems in Practice and Theory of Automated Timetabling IV, pp. 24-38, Springer, (2003).
11. Hansen P., Mladenovic N., Variable neighborhood search: Principles and applications, European Journal of Operational Research, 130, pp. 449-467, (2001).
12. Hansen P., Mladenovic N., Urosevic D., Variable neighborhood search and local branching, Computers and Operations Research, 33, pp. 3034-3045, (2006).
13. Maniezzo V., Stutzle T., Voss S., Matheuristics: Hybridizing Metaheuristics and Mathematical Programming, Annals of Information Systems, 10, Springer,(2009)
14. Miller H.E., Nurse scheduling using mathematical programming, Operations Research, 24, pp. 857-870, (1976).
15. Mladenovic N., Hansen P., Variable neighborhood search, Computers and Operations Research, 24, pp. 1097-1100, (1997).
16. Nanda R., Browner J., Introduction to Employee Scheduling, Van Nostrand Reinhold, New York, (1992).
17. Warner D.M., Scheduling nursing personnel according to nursing preference: A mathematical programming approach, Operations Research, 24, pp. 842-856, (1976).