An introduction to Nurse Rostering Problem heuristics methods

Guilherme Ramon Rodrigues da Silva [a].

[a] Escola de Engenharia de São Carlos and Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brazil, guilhermerrs@usp.br.

Abstract. The Nurse Rostering Problem (NRP) is a highly constrained combinatory problem. Mathematical Modelling approaches may successfully solve small instances, but their performance is limited by the number of decision variables and time available for more complex instances and real-world scenarios. Heuristics methods can tackle the problem through a different perspective: decomposing models, creating sub-Mixed-Integer Problems (MIPS) and generating feasible solutions quickly. In this article, we will review some constructive heuristics and local search methods such as relax-and-fix, fix-and-optimize, and proximity search. The purpose is to provide a trustful source to understand these methods that can be merged to tailor better schedules, not only for NRP but for other personnel rostering problems.

Keywords. Nurse Rostering Problem, Heuristics, Relax-and-fix, Fix-and-optimize, Proximity Search.

1. Introduction

The Nurse Rostering Problem (NRP) defines the days and shifts of work of nurses in health institutions. A feasible solution for this NP-hard problem must obey working regulations, the minimum personnel to attend the hospital, and respect the rest and preferences of the nurses [1].

A low-quality schedule decreases the well-being of the nurse team, assigning unnecessary overtime, increasing the costs [2, 3], and negatively affecting the population that depends on this service. Different approaches have been developed to improve the quality of these scheduling.

This problem can be addressed by manual algorithms, requiring a nurse officer for this task [4]. Mathematical modeling, in turn, provides better schedules appealing to computational power. Gurobi® and COIN-OR are solvers that handle those Mixed-Integer Problems (MIP). However, the results provided by them may take too long to be obtained and yet be of low quality. A heuristics - a non-exact strategy that does not guarantee the solution's optimality [5] – can get better results faster [6].

The importance of NRP and its heuristics methods are also shown by competitions such as the International Rostering Competition, which ran in 2010 [7] and 2015 [8]. The vast literature addressing the problem then stresses the need to develop better solution strategies for this problem.

Section 2 will discuss the mathematical modeling of the problem; in Section 3, constructive heuristics methods are presented. Section 4 shows the improvement methods, and Section 5 the outcomes of the research.

2. Problem definition

NRP is constituted by a scheduling horizon in which nurses work with different skills in a set of shifts. The nurses are grouped by working contracts that define the problem's parameters, such as the workload [4]. Other benchmarks, however, may consider each nurse as a single contract; an example is [9].

A feasible solution must consider the contract parameters (minimum and maximum workload, number of consecutive working/free days) and others (hospital demand of nurses, a penalty for the solution in case of under/over coverage) [8] according to the different needs of the health institution.

The NRP can be dealt with as non-cyclical – every nurse can work in any shift since it respects shift rotance [9] – or cyclical – the schedule must try to follow the same pattern of shifts on the horizon.

This approach may require more constraints [10].

2.1 Decision Variables

Even though heuristics methods may not rely on Mathematical Models (MM) as in [11], understanding the constraints provides a deeper view of the problem and a basis to comprehend different solution methods.

A nurse-day view can represent the NRP schedule. In this approach, a decision variable is defined for each nurse, day and shift, so the value of this variable x_{nds} indicates whether the nurse works on not in a specific shift of a day. The nurse-shift pattern view simplifies the schedule, o each nurse can only be assigned to a shift pattern p, so, x_{np} indicates if the nurse works in a specific shift pattern or not [4].

2.2 Constraints

The constraints can be divided into hard (a feasible solution must obey) and soft (generates a penalty for the objective function). Even though they and their kind vary among the models [10], it is possible to highlight: nurse workload, consecutive minimum/maximum working/free days, holydays, unwanted shift patterns, shift rotation, previous assignments, the maximum number of weekends, whether a nurse should work whole weekends, nurses the must work together, and more [4, 7, 8, 9].

2.3 Objective Function

The objective function tries to minimize penalties, such as the incurred by over/under coverage, disrespect to the shift preference of the nurse [11], the break of a shift pattern (for cyclic scheduling), and more. The complexity depends on the modeling and may help to guarantee feasibility [4].

3. Constructive Heuristics

In order to obtain better solutions faster, heuristics tend to mix different approaches, usually a quick constructive method and then an improving algorithm that may run until the time limit [11]. Among the constructive heuristics, there are relax-and-fix, greedy algorithms MIP-depended and not dependent, decomposition of the model, and others.

3.1 Relax-and-Fix

The planning horizon of the schedule for high instances may demand a lot of computational power from a solver to obtain a feasible solution for the Integer Programming (IP) MM. The relax-and-fix heuristics divides this horizon into partitions - for example, groups of weeks – and relaxes them; that is, the domain of decision variables that is integer becomes real. A solution of the relaxed MIP will not be feasible but is computationally less expensive, providing a good bound for the problem [12].

The algorithm takes a partition and sets its domain back to integer, solves the MIP, and fixes the integer values. If there is any partition with all variables relaxed, the procedure is repeated until none is left. Notice that the number of free decision variables decreases as the algorithm runs. Meanwhile, the number of fixed variables gets higher, constraining the problem more and more [13]. Figures 1 to 4 illustrate the algorithm.

	Week 1	Week 2	Week 3	Week 4	Week 5
Nurse 1	Real	Real	Real	Real	Real
Nurse 2	Real	Real	Real	Real	Real
Nurse 3	Real	Real	Real	Real	Real
Nurse 4	Real	Real	Real	Real	Real
Nurse 5	Real	Real	Real	Real	Real

Fig. 1 – Relax-and-fix algorithm: First, all variables are relaxed.

	Week 1	Week 2	Week 3	Week 4	Week 5
Nurse 1	Integer	Integer	Real	Real	Real
Nurse 2	Integer	Integer	Real	Real	Real
Nurse 3	Integer	Integer	Real	Real	Real
Nurse 4	Integer	Integer	Real	Real	Real
Nurse 5	Integer	Integer	Real	Real	Real

Fig. 2 – Relax-and-fix algorithm: The domain of set of weeks (partition) is set to integer. The MIP is solved.

	Week 1	Week 2	Week 3	Week 4	Week 5
Nurse 1	Fixed	Integer	Real	Real	Real
Nurse 2	Fixed	Integer	Real	Real	Real
Nurse 3	Fixed	Integer	Real	Real	Real
Nurse 4	Fixed	Integer	Real	Real	Real
Nurse 5	Fixed	Integer	Real	Real	Real

Fig. 3 – Relax-and-fix algorithm: If the whole integer partition is fixed, the relax-and-fix doesn't have overlapping, otherwise, it does. The overlapping means that the fix partition is smaller than the integer partition (also called windows). It may help avoid infeasibility due to the higher levels of constraint the algorithm gets while runs.

	Week 1	Week 2	Week 3	Week 4	Week 5
Nurse 1	Fixed	Integer	Integer	Real	Real
Nurse 2	Fixed	Integer	Integer	Real	Real
Nurse 3	Fixed	Integer	Integer	Real	Real
Nurse 4	Fixed	Integer	Integer	Real	Real
Nurse 5	Fixed	Integer	Integer	Real	Real

Fig. 4 – Relax-and-fix algorithm: The domain of the next integer window is set to integer and the algorithm is repeated.

Due to its simplicity in implementation, relax-and-fix heuristics can be used to provide feasible solutions for the NRP [14] and other problems such as high school timetabling [15] and multi-level lot-sizing [13]. It is essential to mention that relax-and-fix may return no feasible solution: while the program runs, the initial partitions are fixed according to a MIP in which the final partitions assume real values, which may lead to invalid combinations. Different approaches have been tested to solve it, for example, running a proximity search in a sub-IP with only the fixed and integer partition, disturbing the current invalid solution [15].

For models without coverage constraints [9], another way to avoid infeasibilities is to consider each nurse a single sub-problem. Depending on the available time and size of the benchmark's instance, relax-and-fix can be applied to each of the nurses, or the sub-problem can be sent to a solver. The schedule is the sum of all nurses (notice this last approach is relax-and-fix inspired, it is not a relax-and-fix heuristic because at no time is the problem where relax).

3.2 Random greedy algorithm

Another way to tackle the problem is using a simple greedy algorithm like in [11]. The method proposed by the author only works if there is no coverage constraint in the MM: the schedule is started without any assignment, a random nurse is chosen, and then all the working days are assigned for this nurse. Randomly, the shifts for these working days are set. If the solution is valid, the schedule is saved; else, the solution is destroyed, and a new one is generated by the same method.

```
Pre-process problem data;
Create an empty schedule for each nurse;
foreach nurse do:
    while nurse schedule is not feasible do:
        Assign working days of nurse;
        Randomly assign shifts of nurse;
        Evaluate constraints;
        if schedule not valid then:
            Destroy nurse schedule;
        end
    end
end
return [all nurse schedules]
```

Fig. 5 – Random greedy algorithm [11]: The idea of this constructive method is divide-and-conquer the model and benefit from randomness to generate quickly feasible solutions.

Notice, that different from relax-and-fix heuristics, this method is not MIP dependent, so the assignments are made without the help of a solver.

3.3 Decomposing model heuristics

Like the random greedy algorithm, the sub-model approach divides the problem into smaller pieces to obtain a feasible final solution. Some NRP models don't apply coverage constraints over the schedule. This leads to the possibility of treating the nurse as a single entity. Then, each nurse's schedule is a sub-problem that must be solved. Summing all the schedules and creating each of them without any global view leads to poor quality solutions; meanwhile, the algorithm's performance may improve with fewer decision variables.

[9] model constraints can be separated into constraints that define the rules of working days (minimum/maximum consecutive working/free days, maximum number of weekends, workload, and holiday) and assigning shifts (shift rotation, maximum number of days working in a shift).

In [16], for the lot sizing and scheduling problem on multiple heterogeneous production lines with perishable products, the initial solution is obtained using a decomposition of the model. The same idea can be applied to the NRP. For each nurse, an IP runs with the working day rules constraints, then another IP runs with the shift assignment rules constraints, defining the shifts for the working days for the solution of the first IP. If the solution is feasible, it is saved, else, destroyed.

4. Improve Heuristics

Although they are feasible, the solutions obtained by constructive heuristics may not be the best ones to use. Better scheduling can be obtained by improving heuristics, such as fix-and-optimize and proximity search. It is essential to highlight that the quality of the initial solution affects the local search heuristics. Reasonable initial solutions, for example, can lead the algorithm to a local minimum that may be hard to leave [11, 17].

4.1 Fix-and-Optimize

Fix-and-optimize is a heuristic that tries to improve the quality of the overall schedule through slight improvement in partitions of the problem. As in the relax-and-fix, the planning horizon can be partitioned. Fix-and-optimize considers all variables fixed; then, only one partition is released to be optimized - if a partition is released, it can assume any value of its domain. The IP is solved and if the solution is better, the schedule is updated; else, the algorithm tries again in another neighborhood (partition) [18, 19].

	Week 1	Week 2	Week 3	Week 4	Week 5
Nurse 1	Fixed	Fixed	Fixed	Fixed	Fixed
Nurse 2	Fixed	Fixed	Fixed	Fixed	Fixed
Nurse 3	Fixed	Fixed	Fixed	Fixed	Fixed
Nurse 4	Fixed	Fixed	Fixed	Fixed	Fixed
Nurse 5	Fixed	Fixed	Fixed	Fixed	Fixed

Fig. 6 – Fix-and-optimize algorithm: An initial feasible solution is loaded. The decision variables are all fixed.

	Week 1	Week 2	Week 3	Week 4	Week 5
Nurse 1	Free	Fixed	Fixed	Fixed	Fixed
Nurse 2	Free	Fixed	Fixed	Fixed	Fixed
Nurse 3	Free	Fixed	Fixed	Fixed	Fixed
Nurse 4	Free	Fixed	Fixed	Fixed	Fixed
Nurse 5	Free	Fixed	Fixed	Fixed	Fixed

Fig. 7 – Fix-and-optimize algorithm: A partition is released, so the decision variables can assume any value on their domain.

	Week 1	Week 2	Week 3	Week 4	Week 5
Nurse 1	Fixed	Free	Fixed	Fixed	Fixed
Nurse 2	Fixed	Free	Fixed	Fixed	Fixed
Nurse 3	Fixed	Free	Fixed	Fixed	Fixed
Nurse 4	Fixed	Free	Fixed	Fixed	Fixed
Nurse 5	Fixed	Free	Fixed	Fixed	Fixed

Fig. 8 – Fix-and-optimize algorithm: If a better solution is found, the new schedule is saved and the algorithm proceeds to the next partition until none of them improves or the time limit is reached.

4.2 Proximity Search

The Proximity Search is a MIP-dependent heuristic that consists of finding a better solution near the current solution. Basically, we add a constraint that forces the output solution to be better than the initial – constraint (1) – and the objective function tries to guide the search for a new schedule near the current one.

$$f(x) \leq f(\tilde{x}) - \theta \qquad (1)$$

Constraint (1) states that the objective function *f(x)* of the schedule must be θ smaller then de initial objective function $f(\tilde{x})$. A high value of theta may lead to better performance due to the more aggressive cut, but significant improvement can be taken from a small theta – a bigger queue of simple IPs is solved. Both approaches have their advantages and drawbacks [20]. Function (2) shows the objective function of proximity search:

$$Min\ \Delta(x, \tilde{x}) = \sum_{\tilde{x}_i \in J_0} x_i + \sum_{\tilde{x}_i \in J_1} (1 - x_i) \qquad (2)$$

In function (2), xi is a binary variable that defines whether the nurse works (value 1) or not (0). x ~_i is the initial solution, J0 the days the nurse rest in the initial solution and J1 the working days. The objective function, then, tries to minimize the variation between the initial and new solution.

5. Conclusion

After the literature review, it is possible to notice the vast number of heuristics methods to solve NRP and other operational research problems. The research showed us different metaheuristics and AI methods to solve personnel rostering problems. These topics can be better studied in future research, stimulating the development of new methods for the NRP and, consequently, positively affecting health institutions, nurses, and the population in general.

6. Acknowledgement

The author would like to thank the following funding agencies for the financial support: Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP) via CEPID No. 2013/07375-0.

7. References

[1] Glass C. A., Roger A. K. The nurse rostering problem: A critical appraisal of the problem structure. *European Journal of Operational Research*. 2010; 202(2):379-389.

[2] Shin S., Park J., Bae S. *Nurse staffing and nurse outcomes: A systematic review and meta-analysis*. *Nursing outlook*. 2018; 66(3):273-282.

[3] Neto, T. M, Junior D. P. A., Gonçalves M. B., Coelho A. S. Modelo de Programação Linear como ferramenta para otimizar a capacidade de atendimentos nos serviços de saúde em UBSS de Manaus - AM. *XXXI Encontro Nacional de Engenharia de Produção*. 2011.

[4] Cheang B., Li H., Lim A., Rodrigues B. Nurse rostering problems – a bibliographic survey. *European journal of operational research*. 2003; 151(3):447-460.

[5] Talbi E. *Metaheuristics from design to implementation*. Wiley, New Jersey; 2009; 1 p.

[6] Rahimian E., Akartunalı K., Levine J. A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems. *European Journal of Operational Research*. 2017; 258(2):411-423.

[7] Haspeslagh S., De Causmaecker P., Schaerf A., Stølevik M. The first international nurse rostering competition 2010. *Annals of Operations Research*. 2014; 218(1):221-236.

[8] Ceschia S., Dang N., De Causmaecker P., Haspeslagh S., Schaerf A. The second international nurse rostering competition. *Annals of Operations Research*. 2019; 274(1):171-186.

[9] Curtois, T., & Qu, R. Computational results on new staff scheduling benchmark instances. *tech. report*. 2014.

[10] De Causmaecker P., Vanden Berghe G. A categorisation of nurse rostering problems. *Journal of Scheduling*. 2011; 14(1):3-16.

[11] Rahimian E., Akartunalı K., Levine, J. A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems. *European Journal of Operational Research*. 2017; 258(2):411-423.

[12] Arenales M., Armentano V., Morabito R., Yanasse H. *Pesquisa Operacional*. Elsevier, Rio de Janeiro; 2011; 165-166, 271 p.

[13] Toledo C. F. M., da Silva A. M., Hossomi M. Y. B., França P. M., Akartunalı K. A relax-and-fix

with fix-and-optimize heuristic applied to multi-level lot-sizing problems. Journal of heuristics. 2015; 21(5):687-717.

[14] Gomes R. A., Toffolo T. A., Santos H. G. Variable neighborhood search accelerated column generation for the nurse rostering problem. *Electronic Notes in Discrete Mathematics*. 2017; 58:31-38.

[15] Saviniec L., Santos M. O. dos, Costa A. M. A parallel multi-start iterated local search and a proximity relax-and-fix heuristic for high school timetabling problem. *Pré-anais. Rio de Janeiro: SOBRAPO*. 2017.

[16] Soler W. A., Santos M. O., Akartunalı K. Decomposition based heuristics for a lot sizing and scheduling problem on multiple heterogeneous production lines with perishable products. *Pesquisa Operacional*. 2021:41.

[17] Burke E., Causmaecker P. D., Berghe G. V. A hybrid tabu search algorithm for the nurse rostering problem. *Asia-Pacific Conference on Simulated Evolution and Learning, Springer, Berlin, Heidelberg*. 1998; 187-194.

[18] Turhan A. M., Bilgen B. A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem. *Computers & Industrial Engineering*. 202; 145:106531.

[19] Santos H. G., Toffolo T. A., Gomes R. A., Ribas S. (2016). Integer programming techniques for the nurse rostering problem. Annals of Operations Research. 2016; 239(1):225-251.

[20] Fischetti M., Monaci M. Proximity search for 0-1 mixed-integer convex programming. *Journal of Heuristics*. 2014; 20(6):709-731.