Healthcare Transformation with Informatics and Artificial Intelligence
J. Mantas et al. (Eds.)

doi:10.3233/SHTI230511

Nurse Scheduling with State-of-the Art Open-Source Tools

Dominik J AESCHBACHER[a], Jessica MEISSNER[a] and Murat SARIYAR[a,1]
[a]*Bern University of Appl. Sciences, Dept. Medical Informatics, Switzerland*

Abstract. Nurse scheduling is still an unsolved issue, as it is NP-hard and highly context-dependent. Despite this fact, the practice needs guidance on how to tackle this problem without using costly commercial tools. Concretely, we have the following use case: a Swiss hospital is planning a new station designed for nurse training. The capacity planning is finished, and the hospital wants to assess whether shift planning with known constraints leads to valid solutions. Here, a mathematical model is combined with a genetic algorithm. We trust the solution of the mathematical model more, but if it does not provide a valid solution, we try out an alternative. Our solutions indicate that actual capacity planning together with the hard constraints cannot lead to valid staff schedules. The central conclusion is that more degrees of freedom are necessary and that open-source tools OMPR and DEAP are valuable alternatives to commercial products such as Wrike or Shiftboard, in which the degree of freedom of customization is reduced in favor of easiness of use.

Keywords. Mixed-integer programming; genetic algorithms; staff scheduling.

1. Introduction

For medical informaticians in the data science domain, a different expertise is required when compared to bioinformaticians or clinical statisticians. Besides their knowledge of clinical processes, their familiarity with analytical problems regarding organizational issues imbue them with a differentiating skill set, which could be subsumed under the term 'biomedical operations research'. One such organizational problem is staff scheduling or rostering, which has been investigated since many decades and still actual [1–3]. Even though, this issue seems frequently less attractive than analyzing clinical data, it has the potential to increase efficiency and contentment of staff, which in turn might have an impact on the clinical outcomes. Hence, the topic should receive more attention, especially in the era of various forms of flexible working models.

The nurse scheduling problem (NSP) concerns the optimal assignment of nurses and other clinical staff to shifts, while considering constraints, such as "a nurse does not have more than shift per within 24 hours" [4]. All those constraints that have to be adhered to are called hard constraints, and any violations of them makes the solution invalid. In addition to that, there are soft constraints, reflecting aspects such as shift preferences, that do not affect the validity of the solution but allow assessing the overall quality of the solution. NSP is often the operative part of a broader personnel strategy comprising four levels [5]: (i) capacity and recruitment planning, (ii) rostering in the sense of shift and

[1] Corresponding Author, Murat Sariyar, Bern University of Applied Sciences, Quellgasse 21, CH2502 Biel/Bienne, Switzerland; E-mail: murat.sariyar@bfh.ch.

absence planning, (iii) real-time shift assignment with ad-hoc adjustments and (iv) task assignment. Only this broader strategical perspective allows addressing issues such as work-life balance and fairness adequately. Here, we focus on (ii), which has a bridging function between the long-term and short-term perspectives.

Concretely, we have the following use case: a Swiss hospital is planning a new station designed for nurse training. The capacity planning is finished, and the hospital wants to assess whether shift planning with known constraints leads to valid solutions. If not, the capacity planning must be adapted accordingly before starting the recruitment. One reason for this approach is the awareness of the hospital that the hard constraints considered for NSP allow an objective assessment of the validity of their personnel planning, which would otherwise rely just on experiences and probable outcomes. Following research question arises in this connection: what is a state-of-the-art approach to NSP for the nurse training station using open-source tools alone?

In the next section, we describe our approach of combining (mixed integer) linear programming tool with genetic algorithms for tackling NSP, and the tools used for that task. The Result section presents the outcome of applying our approach for the nurse training station, without presenting all we have done due to the limitation of space. In the Discussion section, we take up both research questions, and discuss related topics.

2. Methods

Combining a mathematical model with a probabilistic one is done with the following reasoning: If the former model class generates a valid solution only for one set of constraints in contrast to a superset of it, it is an indication that the search space gets too complicated. Maybe, there is still a valid solution, and this should be investigated by applying a probabilistic model. In other words, we trust the solution of the mathematical model more, but if it does not provide a valid solution, we try out an alternative. Using the alternative right from the start would lead to an increased uncertainty regarding its potential of generating suboptimal solutions. For the probabilistic model, a genetic algorithm is used, due to the success of such this method in different applications and its highly intuitive justification for adapting solution candidates in an evolutionary manner. Formulation of the constraints is still the first step, but they are not used for mathematically deriving a solution, but for assessing whether a proposal is valid or not.

Due to our knowledge of the R package OMPR (Optimization Modeling Package R [12]), it was a natural choice for the linear programming model. OMPR is inspired by the Jump project in Julia [6], which uses the GNU Linear Programming Kit (GLPK), written in C, to compute a solution. GLPK contains implementations of the simplex method for linear programming and the branch-and-bound method for mixed-integer programming. For the implementation of a genetic algorithm, the Python Package DEAP (Distributed Evolutionary Algorithms in Python) was used [7]. The reason for not staying in the R environment, for example, by using the package GA [8], is due to the higher maturity, better documentation, and bigger community of DEAP. DEAP has four blocks: 1) defining the type of the problem, 2) initialization of the chromosomes, including the number of chromosomes to consider, 3) choosing operators for perturbing & selecting the chromosomes, and 4) stacking all together into an algorithm to optimize an objective function. For example, we chose in 1) a minimization problem with the sum of penalties as the objective function (violation of constraints are penalized), in 2) 300 as population size, and in 3) tournament selection, flip-bit mutation and 2-point crossover strategies.

3. Results

From the capacity planning, we received following information: there are 35 staff members from 7 categories, for which different hard constraint must be considered for a one-week plan. 14 of these constraints were included in our approach. Main constraints were the minimum numbers of the staff members that have to work at each shift, the prohibition of consecutive shifts, and legal requirements such as apprentices under 18 are not allowed to work on Sundays. In the OMPR package the constraint for the maximal number of shifts per week has this form (i: staff index, j: day index, k: shift index, `x[i,j,k]` is a binary variable):

```
ompr::add_constraint(sum_expr(x[i,j,k], j = 1:numOfDays,
k =1:numOfShifts) <= 6, I = 1:35)
```

while in DEAP, this constraint has a more complex formulation (shift_dict is an array resulting from a matrix that contains 2 rows for each staff member, representing the 2 possible shifts per day, and the columns are the days; self refers to the DEAP object):

```
def conShiftsPerWeek(self, shift_dict):
    violation = 0
    for value in shift_dict.values():
        index = 0
        while (index < len(value)):
            if sum(value[index:index + 14]) > 6:
                violation += 1
            index += 14
    return violation
weeksPerWeekViolations = self.conShiftsPerWeek(staffShiftsDict)
```

We quickly found out that the OPMR model could not generate a valid solution when more than 10 constraints were considered. Therefore, we skipped several constraints in the order of their assumed relevance, especially those related to the required nurse type mixture. Even then, the valid solutions were inefficient in the sense of assigning too many staff members to some shifts (in one case 20 instead of the minimal 4), due to the many constraints. This indicated that the model was at its limits. As soon as we reduced this minimal number from 4 to 2, the solutions were efficient (not more staff than necessary were assigned to each shift).

Applying the DEAP model with the same constraints as in the OMPR model, led to valid solutions for a weekly and even for a bi-weekly plan. As the solutions were efficient, we assumed that the model was not as its limit and added further constraints. First, the constraint that minors are not allowed to work on Sundays was added. Second, the number of minimal shifts per day was adjusted to different values for different nurse types. In that scenario, solutions were found for a weekly and a bi-weekly schedule. For the latter, the population size had to be increased to 500. However, further constraints such as the prohibition of consecutive shifts increased the complexity in such a manner that a violation was unavoidable. The best result consisted in a fitness of 4, which is the number of constraints that were violated. These violations were always related to the minimal number of shifts per week for different nurse types. To prevent such a violation, the minimal number of workdays per week was reduced from 4 to 3. Then, the DEAP-model was able to find a valid solution, which is just one example of many further scenarios we have investigated. Due to the limit of space, they had to be omitted here.

4. Discussion

Our approach using OMPR and DEAP showed that especially the minimum staff members per day constraint is an issue. The central conclusion is that more degrees of freedom are necessary, which can be achieved either by increasing the size of the staff members or by softening some of the constraints. With our results, we increased the transparency regarding the feasibility of current capacity planning and could provide hints regarding the trade-offs involved. The models should not be regarded as ways to take related decision off, but only for preparing such decisions. Such an increase in transparency might not be what hospitals want. Our insights rather point to a delicate legal matter in practice, as one can infer from our use case that schedules may not even meet the requirements prescribed by law. From our perspective, this should be discussed openly, not in order to blame hospitals but to find a general solution backed by politics. For example, educational institutions should be involved in the scheduling process for deciding how degrees of freedom can be increased.

Combining linear programming and genetic algorithm proved to be useful for assessing the complexity of the NSP problem and the solutions generated, even or especially when no valid solution could be found. Non-experts are frequently cautious when a heuristic such as genetic algorithms are proposed. One should explain to them that the mathematical model also uses a heuristic (branch-and-bound), and that the genetic algorithm is only used in case the mathematical model generates invalid or inefficient solutions. Both are representatives of the soft computing approach, which generated approximate models that give solutions to complex real-life problems (in contrast to exact models such as logic reasoning and numerical modelling). In our use case, genetic algorithms were able to generate valid and efficient solutions for more constraints than the mathematical model. In addition to that, the random component led to different schedules in different runs, thereby guaranteeing that individuals were not always assigned together to the same shifts. Hence, the open-source tools OMPR and DEAP are valuable alternatives to commercial products such as *Wrike* or *Shiftboard* and have more degree of freedom in terms of customization.

References

[1] Abdalkareem ZA, Amir A, Al-Betar MA, et al. Healthcare scheduling in optimization context: a review. Health Technol. 2021;11:445–469.
[2] Chaieb M, Sassi DB, Jemai J, et al. Challenges and solutions for the integrated recovery room planning and scheduling problem during COVID-19 pandemic. Med Biol Eng Comput. 2022;60:1295–1311.
[3] Chen P-S, Tsai C-C, Dang J-F, et al. Developing three-phase modified bat algorithms to solve medical staff scheduling problems while considering minimal violations of preferences and mean workload. Technol Health Care Off J Eur Soc Eng Med. 2022;30:519–540.
[4] Burke EK, De Causmaecker P, Berghe GV, et al. The State of the Art of Nurse Rostering. J Sched. 2004;7;441–499.
[5] Ernst AT, Jiang H, Krishnamoorthy M, et al. Staff scheduling and rostering: A review of applications, methods and models. Eur J Oper Res. 2004;153:3–27.
[6] Dunning I, Huchette J, Lubin M. JuMP: A Modeling Language for Mathematical Optimization. SIAM Rev. 2017;59:295–320.
[7] Fortin F-A, De Rainville F-M, Gardner M-AG, et al. DEAP: evolutionary algorithms made easy. J Mach Learn Res. 2012;13:2171–2175.
[8] Scrucca L. GA: A Package for Genetic Algorithms in R. J Stat Softw. 2013;53:1–37.