Integer Programming for Nurse Rostering: Modelling and Implementation Issues

Erfan Rahimian, Kerem Akartunalı, John Levine

Abstract Integer Programming is often a suitable candidate for solving real-world Nurse Rostering Problems. However, there are some less-investigated aspects in modelling the problem and solving it using a general IP solver, which might lead to better performance, if they are carefully considered. In this paper, we present some discussions for a common Integer Programming model of the problem, in which we compare a number of scenarios and report some insightful results for each case.

Keywords Nurse Rostering · Integer Programming · Modelling and Implementation

Introduction

Although there is a variety of papers in the literature regarding how to solve the Nurse Rostering Problem (NRP) [8,1,5], only a few papers are dedicated to the modelling and implementation aspects of this complicated class of problems in real-world settings [9]. Since NRP is NP-hard [2], and real-world instances of this problem are highly-constrained, they can be computationally challenging to solve, thereby having an efficient model and implementation of the problem is as essential as having a robust solution method. In this paper, we present some practical issues emerged from our attempts in designing and developing a general realistic model of the problem, and a hybrid solution method. We explain how some minor adjustments in modelling and implementation of the problem may affect the computational results, and how an understanding of the way the applied optimization tool works may help to get better insights into the search algorithm. In this paper, we model a Nurse Rostering Problem using Integer Programming

Erfan Rahimian, Kerem Akartunalı
Dept. of Management Science, University of Strathclyde, Glasgow, G4 0QU, UK
John Levine
Computer And Information Sciences, University of Strathclyde, Glasgow, G1 1XH, UK
Tel.: +44-141-548-4361
Fax: +44-141-552-6686
E-mail: {erfan.rahimian, kerem.akartunali, john.levine}@strath.ac.uk

and benchmark it by using two popular commercial IP solvers, and some published problem instances in the literature [3,6]. Particularly, we focus on the classical index-based IP model of the problem, and describe how *hardness of constraints*, *domains of variables*, and *upper and lower bounds* might affect the performance of the IP solver, and therefore, the quality of the obtained solutions. In fact, we demonstrate how the proportional number of hard (or soft) constraints in the IP model, the size of domains of decision variables, and the upper and lower bounds associated with the objective function value of the problem using the studied IP model might influence the performance of the IP solver. For each case, we explain the issue and present a number of scenarios, and then benchmark them using some instances exist in the relevant literature, and finally, we report some concluding remarks.

It should be noted that the main purpose of this paper is not to demonstrate some heuristic rules or implementation tips for the studied IP model, which can be useful to solve the problem more efficiently. Indeed, the aim of this paper is to draw attention to some less-investigated aspects of the problem modelling and implementation, which provide insightful information to researchers and practitioners, and might lead to better performance while using a commercial IP solver. In other words, implementing an IP model of the problem using some off-the-shelf IP solvers, one might get better performance by investigating the presented issues in this work apart from gaining a deeper insight into the problem.

The Issues

In the NRP literature, the constraints are often categorized as *Hard* and *Soft* constraints [5]. Hard constraints must be satisfied and are essential for the feasibility of the obtained solutions. Soft constraints are desired to be satisfied and the degree to which they are often met indicates the quality of the obtained solutions. In a branch-and-bound algorithm, which is applied to most of the commercial IP solvers [7], the more the number of hard constraints, the less the number of auxiliary variables, and hence, the smaller the size of the IP model, which often leads to better performance. However, due to the highly-constrained nature of NRPs, obtaining a feasible solution (i.e., satisfying all the hard constraints) is very difficult. On the other hand, if the number of soft constraints increases, the IP solver is able to obtain a feasible solution sooner but it might take a longer computational time to prove the solution quality. Therefore, the proportional number of hard and soft constraints alter the performance of the associated IP solver.

The domain of a variable is defined as all the possible values which can be taken by the variable according to the problem constraints. In Integer Programming, smaller domain of a variable often leads to better performance in the branch-and-bound algorithm, due to the fewer number of branches required on the variable. Furthermore, a smaller domain of variables might result in stronger bounds on the involved variables, and better performance of the applied heuristics during the search process. Apart from the inherent nature of the problem, the size of the domain of variables is directly relevant to the art of modelling and the applied solution method.

In the branch-and-bound algorithm, pruning based on the upper and lower bound of a problem is one of the key underlying concepts [10]. In fact, during the search process, extra branches on variables can be pruned using the problem upper and lower bound, and therefore, implicitly enumerate all the possible solutions for a particular problem. In this algorithm, a branch on a variable can be pruned if the solution to the relevant relaxed IP problem (lower bound) is less than the global lower bound, or if the relevant obtained feasible solution (upper bound) is larger than the global upper bound, assuming that the optimization problem is minimization. Having said that, there is a huge amount of effort such as finding problem-specific cuts or designing tailored heuristics in the relevant literature [8, 4], and in the commercial IP solvers (before or during the branch-and-bound algorithm) to tighten the problem bounds. Because it increases the efficiency of the IP solver, thereby reducing the computational time for obtaining and proving the optimal solution.

Computational Experiments

In this paper, we present some important aspects of modelling and implementation in Integer Programming by using off-the-shelf IP solvers. In our ongoing computational results, we define a number of scenarios for each issue in order to provide some insightful information regarding the behaviour of the applied IP solvers under different circumstances. We benchmark each scenario according to the latest instances published in the relevant literature [3], and the instances introduced in the First International Nurse Rostering Competition (INRC-I) [6]. We will publish the results and some discussions at the conference.

References

1. Burke, E.K., Curtois, T.: New approaches to nurse rostering benchmark instances. European Journal of Operational Research **237**(1), 71–81 (2014). DOI 10.1016/j.ejor.2014.01.039
2. Chuin Lau, H.: On the complexity of manpower shift scheduling. Computers & Operations Research **23**(1), 93–102 (1996). DOI 10.1016/0305-0548(94)00094-O
3. Curtois, T., Qu, R.: Computational results on new staff scheduling benchmark instances. Tech. Rep. 06-Oct-2014, ASAP Research Group, School of Computer Science, University of Nottingham (2014)
4. Dowsland, K.A., Thompson, J.M.: Solving a Nurse Scheduling Problem with Knapsacks, Networks and Tabu Search. Journal of the Operational Research Society **51**(7), 825–833 (2000). DOI 10.1057/palgrave.jors.2600970
5. 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**(1), 3–27 (2004). DOI 10.1016/S0377-2217(03)00095-X
6. Haspeslagh, S., De Causmaecker, P., Schaerf, A., Stolevik, M.: The first international nurse rostering competition 2010 (2014). DOI 10.1007/s10479-012-1062-0
7. IBM: IBM ILOG CPLEX Optimizer (2015). URL `http://www.ibm.com/software/integration/optimization/cplex-optimizer/`
8. Rahimian, E., Akartunali, K., Levine, J.: A Hybrid Constraint Integer Programming Approach to Solve Nurse Scheduling Problems. In: Proceedings of the Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA 2015 (2015)
9. Smet, P., Bilgin, B., De Causmaecker, P., Vanden Berghe, G.: Modelling and evaluation issues in nurse rostering. Annals of Operations Research **218**(1), 303–326 (2014). DOI 10.1007/s10479-012-1116-3
10. Wolsey, L.A.: Integer programming. Wiley (1998)