European Journal of Operational Research 000 (2016) 1–13

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A hybrid Integer Programming and Variable Neighbourhood Search algorithm to solve Nurse Rostering Problems

Erfan Rahimian[a,*], Kerem Akartunalı[a], John Levine[b]

[a] Department of Management Science, University of Strathclyde, 130 Rottenrow, Glasgow G4 0GE, UK
[b] Computer and Information Sciences, University of Strathclyde, 26 Richmond Street, Glasgow G1 1XH, UK

ARTICLE INFO

Article history:
Received 24 January 2016
Accepted 19 September 2016
Available online xxx

Keywords:
Timetabling
Nurse Rostering
Hybrid algorithm
Integer Programming
Variable Neighbourhood Search

ABSTRACT

The Nurse Rostering Problem (NRP) is defined as assigning a number of nurses to different shifts during a specified planning period, considering some regulations and preferences. This is often very difficult to solve in practice particularly by applying a sole approach. In this paper, we propose a novel hybrid algorithm combining the strengths of Integer Programming (IP) and Variable Neighbourhood Search (VNS) algorithms to design a hybrid method for solving the NRP. After generating the initial solution using a greedy heuristic, the solution is further improved by employing a Variable Neighbourhood Descent algorithm. Then IP, deeply embedded in the VNS algorithm, is employed within a ruin-and-recreate framework to assist the search process. Finally, IP is called again to further refine the solution during the remaining time. We utilise the strength of IP not only to diversify the search process, but also to intensify the search efforts. To identify the quality of the current solution, we use a new generic scoring scheme to mark the low-penalty parts of the solution. Based on the computational tests with 24 instances recently introduced in the literature, we obtain better results with our proposed algorithm, where the hybrid algorithm outperforms two state-of-the-art algorithms and Gurobi in most of the instances. Furthermore, we introduce 11 randomly generated instances to further evaluate the efficiency of the hybrid algorithm, and we make these computationally challenging instances publicly available to other researchers for benchmarking purposes.

1. Introduction

Nurse Rostering (also referred to as *Nurse Scheduling*) is the process of creating a schedule by assigning some nurses to different shift types, e.g. day, and night, during a predetermined planning horizon, where many limitations such as hospital regulations and employee contracts as well as management and individual preferences are taken into account. The output of this process is a roster of working shifts for all the involved nurses, which is expected to result in an increase of job satisfaction and staff utilisation while reducing stress and outsourcing cost (Burke, De Causmaecker, Berghe, & Van Landeghem, 2004; Ernst, Jiang, Krishnamoorthy, Owens, & Sier, 2004a; Ernst, Jiang, Krishnamoorthy, & Sier, 2004b). Real-world Nurse Rostering Problems are very difficult to solve and comprise many challenges for the people involved in the preparation process, e.g. personnel managers, and head nurses (Ernst et al., 2004b).

Many studies have been accomplished for the Nurse Rostering Problem (NRP) over the last few decades, with a variety of methods and algorithms applied to solve this problem in real-world settings. The proposed approaches are mainly based on meta-heuristic algorithms (Blum & Roli, 2003; Glover & Kochenberger, 2003; Talbi, 2009), which are straightforward and effective for many practical problems. These range from Variable Neighbourhood Search (Della Croce & Salassa, 2014; Stølevik, Nordlander, Riise, & Froyseth, 2011) and Tabu Search (Burke, Causmaecker, & Berghe, 1999) to Genetic Algorithms (Aickelin & Dowsland, 2004) and tailor-made heuristics (Lu & Hao, 2012; Valouxis, Gogos, Goulas, Alefragis, & Housos, 2012). However, meta-heuristic algorithms are not as efficient for problem instances where the structure of the problem is very complex, making it challenging to find a good-quality (or even a feasible) solution in a reasonable runtime. On the other hand, there is also some research employing exact approaches such as Integer Programming (IP) (Beaumont, 1997; Dowsland & Thompson, 2000) and Constraint Programming (CP) (Bourdais, Galinier, & Pesant, 2003; Cipriano, Gaspero, & Dovier, 2006), which are very powerful at dealing with complex structures. Nevertheless, they are not efficient enough for solving

* Corresponding author.
E-mail addresses: erfan.rahimian@strath.ac.uk (E. Rahimian), kerem.akartunali@strath.ac.uk (K. Akartunalı), john.levine@strath.ac.uk (J. Levine).

http://dx.doi.org/10.1016/j.ejor.2016.09.030
0377-2217/

many medium- to large-scale problem instances in practice, even though there are some very powerful and mature commercial solvers applying these methods such as Gurobi (Gurobi Optimization, 2015) and IBM CP Optimiser (IBM, 2015). Having said that, in recent years, some researchers have focused on combining these two approaches to utilise their complementary strengths in order to solve highly-constrained real-world NRPs efficiently (Burke, Li, & Qu, 2010; Qu & He, 2009; Rahimian, Akartunali, & Levine, 2015; Stølevik et al., 2011).

In this paper, we propose a novel hybrid Integer Programming and Variable Neighbourhood Search (VNS) algorithm to solve the Nurse Rostering Problem in modern hospital environments. We employ IP not only to diversify the search process, but also to improve the quality of the obtained solutions from the VNS algorithm in a creative way. First, a greedy heuristic is used to generate an initial solution, and then the generated solution is further improved using a VNS algorithm until a stopping criterion is met. To further enhance the efficiency of the VNS algorithm, IP is employed iteratively during the running of the algorithm as a neighbourhood structure to improve the quality of the incumbent solution using a *ruin-and-recreate* framework (Stølevik et al., 2011). In this framework, the high-penalty components of the solution are destroyed according to a generic scoring scheme, and then they are created again by an IP solver. Finally, IP is applied once more to the best-found solution to improve it globally as much as possible until the overall time limit is reached. The proposed algorithm is designed to perform efficiently when only short computational times are available, so that many practical problems can be tackled.

The novelty of our approach is to embed IP as a neighbourhood structure through a ruin-and-recreate framework in the VNS algorithm to improve the quality of the obtained solution and diversify the search process at the same time. Our method of hybridisation is entirely different from the similar algorithms reported in the literature (Burke et al., 2010; Qu & He, 2009; Stølevik et al., 2011). In fact, there are various hybridisation schemes in order to combine different approaches together (Raidl, Puchinger, & Blum, 2010). For example, Qu and He (2009) applied CP to generate an initial solution by decomposing the problem to various subproblems, and then applying VNS to improve the generated solution. Stølevik et al. (2011) applied an Iterated Local Search framework for generating an initial solution and employed VNS and CP in order to improve the solution and diversify the search process, respectively. Burke et al. (2010) employed IP to generate a solution satisfying all hard constraints, and then improve it using VNS to satisfy the remaining soft constraints. In most of the mentioned approaches, IP or CP is used to generate a solution satisfying some constraints of the problem (or parts of the problem), and then a meta-heuristic algorithm is applied to further improve the generated solution. However, in our approach, we employ VNS as the main local search framework and then embed IP as a neighbourhood structure to intensify and diversify the search process in an iterative manner considering all the constraints. Indeed, we use IP through a ruin-and-recreate strategy to escape from local optima and at the same time, improve the quality of the obtained solution. Having said that, incorporating IP in our hybrid algorithm, we also allow the search process to traverse the infeasible space by allowing all the constraints to be violated in order to find out the latent feasible solutions. Moreover, we hybridise IP through VNS in a lower level compared with the approaches reported in the literature (Talbi, 2009) and therefore we exploit the complementary strengths of both methods in a more sophisticated and effective way. In addition, we have applied a scoring scheme to evaluate the quality of the obtained solution according to the associated underlying elements such as nurses or days, which empower the hybrid algorithm to focus on parts of the solution having the most likelihood of gaining a better solution. The proposed algorithm also works with a pre-determined time limit in which the algorithm tries to generate the best solution.

The rest of this paper is organised as follows. We first describe the studied Nurse Rostering Problem and present the relevant IP formulation in Section 2. Next, we elaborate on the solution method and different components of the proposed hybrid algorithm in Section 3. Finally, in Sections 4 and 5, we present our computational results, and draw some conclusions and potential future research directions, respectively.

2. Problem description and IP formulation

In this section, we provide a brief description of the studied problem and the relevant constraints, and present a mathematical formulation. For further information regarding the problem, we refer interested readers to Curtois and Qu (2014), where the detailed description of the problem as well as some instances are presented.

The NRP is defined as assigning a number of nurses to different shifts (e.g. early, late) during a specified planning period, where some regulations (e.g. employee contracts) and preferences (e.g. individual requested days off) are taken into account. Most NRPs including the studied problem are $\mathcal{NP}$-hard (Chuin Lau, 1996; Osogami & Imai, 2000) and computationally challenging, and have a very complex structure even when the problem size is relatively small. Tackling this problem in real-world settings, the constraints of the problem are often classified as hard and soft constraints. Hard constraints are necessary to be satisfied under any circumstances, and therefore, make a problem feasible when they are met. Soft constraints, on the other hand, are those we would prefer to be met (but are not crucial), and define the quality of a generated roster according to the degree to which they are satisfied. Therefore, the objective is to reduce the number of violations associated with the soft constraints as much as possible, i.e. increase the quality of the roster. In the following, the hard and soft constraints of the problem (denoted by prefixes *HC* and *SC*, respectively) are explained:

- *HC1*: nurses cannot be assigned more than one shift on a day.
- *HC2 [Shift rotations]*: the shift assignment of nurses on two consecutive days must comply with the pre-defined set of shift patterns (rotations). The shift patterns prevent forbidden shift sequences.
- *HC3 [Maximum number of shifts]*: the maximum number of shift types that can be assigned to each nurse within the planning period.
- *HC4 [Maximum total minutes]*: the maximum amount of total time in minutes that can be assigned to each nurse within the planning period.
- *HC5 [Minimum total minutes]*: the minimum amount of total time in minutes that can be allocated to each nurse within the planning period.
- *HC6 [Maximum consecutive shifts]*: the maximum number of consecutive shifts, which are allowed to be worked within the planning period.
- *HC7 [Minimum consecutive shifts]*: the minimum number of consecutive shifts, which are allowed to be worked within the planning period.
- *HC8 [Minimum consecutive days off]*: the minimum number of consecutive days off, which are allowed to be assigned within the planning period.
- *HC9 [Maximum number of weekends]*: the maximum number of worked weekends (a weekend is defined as being worked if there is a shift on Saturday or Sunday) within the planning period.