MISTA 2015

Variable neighbourhood search for rich personnel rostering problems

Pieter Smet · Greet Vanden Berghe

1 Introduction

Assigning shifts to employees is a problem solved daily in various organisations in health care, logistics, manufacturing, etc. Despite the many academic advances in the field of automated personnel rostering, this task is often still done manually. One reason for this research-application gap is that academic models for personnel rostering often fail to capture characteristics that cannot be neglected in practice.

The present contribution introduces a variable neighbourhood search (VNS) algorithm for solving feature-rich personnel rostering problems. The general object model introduced by Smet et al (2014) presents a flexible approach for modelling a large variety of rostering problems. This model was developed as part of a research project, and has been implemented in a commercial software system for staff scheduling.

Variable neighbourhood search is a well known metaheuristic which alternates between two phases to balance intensification and diversification (Mladenovic and Hansen, 1997). Bilgin et al (2012) applied VNS to a real world nurse rostering problem, illustrating the algorithm's flexibility and robustness on instances with varying characteristics. The present contribution introduces new neighbourhoods for the general model of Smet et al (2014), and presents new best results for a benchmark dataset.

2 General object model

Through reusable components, the model addresses common, yet complex problem characteristics which are often neglected in academic models. The model can represent problems in the class ASCNI|RVNO|PL, according to the $\alpha|\beta|\gamma$ notation of De Causmaecker and Vanden Berghe (2011).

Shift types are characterised by a start and end time, a net job time and a minimum rest time required before and after an assignment of the shift. Due to breaks, the difference between the start and end time often does not correspond to the actual job time. Therefore, the net job time is specified for each shift, which is used when counting

Pieter Smet, Greet Vanden Berghe
KU Leuven, Department of Computer Science, CODeS & iMinds - ITEC
E-mail: {pieter.smet, greet.vandenberghe}@cs.kuleuven.be

the total number of hours an employees has worked. According to the $\alpha|\beta|\gamma$ notation, the number of shifts is variable (N), and they can be overlapping (O).

Employees are characterised by a set of skill types, contracts and requests. The number of skills for which an employee is qualified and the organisation's skilling structure can vary significantly between different application contexts. Typically, one employee has multiple skill types, sometimes coupled with a level of experience. The general model allows the definition of such problems, modelling the complex interactions between employees that occur, for example, in a hierarchical skilling structure. According to the $\alpha|\beta|\gamma$ notation, the number of skills is variable (N), and can be defined individually (I).

Contracts consist of a number of time-related constraints and a period in which the contract is valid. The rules making up the contract are considered soft constraints, and determine the cost of an individual's roster through a weighted sum of violations (personnel objectives (P) in the $\alpha|\beta|\gamma$ notation). The threshold of each constraint is defined as a range, i.e. both a minimum and maximum value are specified. The time-related constraints are classified into three general types: counters, series and successive series (availabilities (A) and sequences (S) in the $\alpha|\beta|\gamma$ notation).

- **Counters** restrict the number of specific roster items in a certain period, defined by a start date and a number of days. This *counter period* does not need to match the scheduling period. Seven subjects can be restricted by counters: *hours worked*, *shift types worked*, *days worked*, *days idle*, *weekends worked*, *weekends idle* and *domain*.
- **Series** restrict consecutive occurrences of specific roster items. These constraints are valid in the period corresponding to the period of the contract. They are defined for five subjects: *shift types worked*, *days worked*, *days idle*, *weekends worked* and *weekends idle*.
- **Successive series** restrict two consecutive series; if the first series appears in a roster, it must be immediately succeeded by the second series. Similar to series, a successive series' period corresponds to the contract's period. Five successions of series are considered: *days worked* $\rightarrow$ *days idle*, *days idle* $\rightarrow$ *days worked*, *shift types worked* $\rightarrow$ *days idle*, *days idle* $\rightarrow$ *shift types worked* and *shift types worked* $\rightarrow$ *shift types worked*.

The time-related constraints are restricted to specific parts of the roster using *domains*. A domain is defined by a day set, shift type set and skill set, each delineating part of the problem in a different dimension. For example, the domain of the *maximum number of consecutive shift types worked* constraint is a restricted shift type set to indicate which shift types should be constrained. The domain counter constraint is included in the model as a very general type of counter constraint, which uses domains to identify the roster items whose occurrence is limited.

Apart from the time-related constraints, different scheduling constraints further restrict the assignments. This class of constraints contains three types: coverage requirements, collaboration preferences and training constraints. Coverage constraints express the number of employees required for each shift on each day as a range (fluctuating (V) and ranged coverage (R) in the $\alpha|\beta|\gamma$ notation). The model allows for a flexible definition of these constraints through skill type requirements and an associated minimum level of experience. Collaboration preferences model situations in which it is desirable to have a minimum or maximum number of employees from a subset of employees to be working together. Finally, training constraints allow for expressing guidelines regarding the training of personnel, e.g. employees who are less experienced in a skill should work together with more experienced employees. These latter two constraints are chaperoning constraints (C) in the $\alpha|\beta|\gamma$ notation. All scheduling constraints have a weight, so that their violations can also be included in the weighted sum objective function (coverage objectives (L) in the $\alpha|\beta|\gamma$ notation).

Finally, the model includes elements which enable consistent constraint evaluation at the boundaries of the scheduling period. For all counter constraints, start and end values are defined for each employee individually. To allow consistent evaluation of the other constraint types, assignments from other scheduling periods can also be included.

3 Solution approach

Due to the large variety of soft constraints in the model, there are many neighbourhoods that need to be explored in order to reduce the number of constraint violations. As standard local search algorithms provide little support for multiple neighbourhood definitions, VNS was chosen to manage the different move operators.

In the first phase, Variable Neighbourhood Descent (VND) is used to reach locally optimal solutions. Several VND neighbourhoods are proposed which make both small and significant changes to the current solution. These neighbourhoods include commonly used ones, such as swapping one or multiple consecutive assignments between employees. However, due to the constraint definitions in the model, additional neighbourhoods are required that delete assignments or make new assignments. In the spirit of the original formulation of VNS, the order in which these neighbourhoods are explored is linked with their complexity.

The shaking phase attempts to escape local optima by perturbing the current solution. The neighbourhoods used in this phase make large changes to a solution, which cannot be achieved by only applying the VND move operators. Strong perturbations are obtained by swapping a single or all assignments between employees, and by applying a ruin-and-recreate heuristic. These neighbourhoods are ordered in a nested structure.

A computational study is performed to determine suitable parameter settings for the algorithm. Furthermore, the contribution and relative importance of the VND and shaking neighbourhoods are analysed. Computational results on instances based on real world data will be presented at the conference.

Acknowledgment: This research was carried out within the IWT 110257 project.

References

Bilgin B, De Causmaecker P, Rossie B, Vanden Berghe G (2012) Local search neighbourhoods for dealing with a novel nurse rostering model. Annals of Operations Research 194(1):33–57

De Causmaecker P, Vanden Berghe G (2011) A categorisation of nurse rostering problems. Journal of Scheduling 14(1):3–16

Mladenovic N, Hansen P (1997) Variable neighborhood search. Computers & Operations Research 24(11):1097 – 1100

Smet P, Bilgin B, De Causmaecker P, Vanden Berghe G (2014) Modelling and evaluation issues in nurse rostering. Annals of Operations Research 218(1):303–326