Computers & Operations Research 76 (2016) 60–72

Contents lists available at ScienceDirect

Computers & Operations Research

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

Heuristic decomposition approaches for an integrated task scheduling and personnel rostering problem

Pieter Smet [a,*], Andreas T. Ernst [b], Greet Vanden Berghe [a]

[a] KU Leuven, Department of Computer Science, CODeS & iMinds-ITEC, Gebroeders De Smetstraat 1, 9000 Gent, Belgium
[b] Monash University, School of Mathematical Sciences, Clayton, VIC 3800, Australia

ARTICLE INFO

Article history:
Received 26 January 2015
Received in revised form
12 April 2016
Accepted 26 May 2016
Available online 27 May 2016

Keywords:
Task scheduling
Personnel rostering
Timetabling
Column generation
Very large-scale neighbourhood search

ABSTRACT

Real world combinatorial optimisation problems do not often reduce to neatly delineated theoretical problems. Rather, they combine characteristics of various subproblems which then appear to be strongly intertwined. The present contribution introduces a challenging integration of task and personnel scheduling in which both tasks and shifts must be assigned to a set of multi-skilled employees. Three constructive heuristics, based on column generation and other decomposition schemes, are presented, as well as a very large-scale neighbourhood search algorithm to further decrease the schedule's cost. The performance of these algorithms is evaluated on a large set of diverse instances. Computational results illustrate the effectiveness of the proposed approaches, and provide insight into their behaviour. The initial benchmarks are published so as to encourage further research.

1. Introduction

After almost sixty years of research, personnel scheduling continues to introduce new challenges to academia. In these problems, shifts and days-off must be assigned to employees in order to meet the required staffing demand, while respecting legislative and organisational guidelines as well as personal requests [25]. Typically, demand is expressed in terms of shifts when the rosters are constructed. A later step in the scheduling process then assigns tasks taking into account employee availability based on the shift assignments [14]. This is the best possible practice in scenarios where tasks only become known after the rosters have been posted, such as when tasks are derived from bookings for a service. However, when tasks are known at the time of roster generation, assigning tasks and shifts simultaneously potentially leads to lower cost solutions compared to making the assignments sequentially [11]. This scenario is typical for regular timetabled services such as organising aircraft maintenance crews or scheduling railway operators.

The present paper introduces the integrated task scheduling and personnel rostering problem (TSPR). Tasks must be assigned to employees, while also determining shift and day-off assignments. A task is a time interval, defined by fixed start and end times, in which a qualified employee is required to perform an activity without interruption. A shift is a time interval, also defined by fixed start and end times, in which an employee can be assigned to tasks. The problem is solved over a scheduling period of multiple days, introducing additional complexity due to contractual restrictions concerning the assignment of shifts.

Decomposition of the problem is a natural approach for exploiting the relations between the personnel rostering and task scheduling problem. In the academic literature, both exact and heuristic cut generation algorithms have been proposed for problems similar to the TSPR [8,13]. Typically, these algorithms apply a natural decomposition of the problem in which task scheduling is solved as a subproblem which iteratively adds constraints to the personnel rostering problem. This paper introduces alternative decomposition schemes, as well as three very large neighbourhoods partly based on these decompositions. All approaches are evaluated on a large, varied dataset generated by a publicly available instance generator. The computational results are analysed to provide insight into the behaviour of different approaches, and are presented to the research community as initial benchmarks.

The remainder of the paper is organised as follows. Section 2 reviews the related literature. A formal definition of the TSPR is presented in Section 3. Section 4 introduces an alternative formulation and a column generation algorithm, which serve as the basis for one of the proposed heuristics. Three constructive

* Corresponding author.
E-mail addresses: pieter.smet@cs.kuleuven.be (P. Smet), andreas.ernst@monash.edu (A.T. Ernst), greet.vandenberghe@cs.kuleuven.be (G. Vanden Berghe).

http://dx.doi.org/10.1016/j.cor.2016.05.016
0305-0548/

heuristics are presented in Section 5, while Section 6 describes a very large-scale neighbourhood search algorithm. Details and results of the computational study are discussed in Section 7. Finally, Section 8 draws conclusions and identifies areas of future research.

2. Related literature

Few publications focus on the assignment of non-preemptive tasks and shifts. Dowling et al. [9] consider flexible shifts and tasks that are fixed in time. They address the problem in two separate, sequential phases, in which first the employees' shift roster is constructed and afterwards tasks are assigned within the shifts. The objective is to minimise over- and understaffing. Lapègue et al. [16] present a problem in which tasks are also fixed in time, without explicitly defining shifts. Rather, a set of guidelines is supplied to which a shift roster should adhere. Meisels and Schaerf et al. [18] discuss a class of general employee timetabling problems in which both tasks and shifts are fixed in time. In contrast to the TSPR, only one task can be assigned to each shift. Krishnamoorthy et al. [15] introduce a problem in which tasks, fixed in time, are assigned to multi-skilled employees whose working times are predetermined. Alternative algorithms for this problem are discussed by Baatar et al. [2], Smet et al. [24]. The objective is to minimise the total number of employees. Elahipanah et al. [10] discuss a task scheduling problem with both preemptive and non-preemptive tasks (see also Prot et al. [21]). The shift roster is considered part of the input, however, flexibility is achieved by allowing additional shifts to be assigned or existing shifts to be modified. Various costs resulting from understaffing, overtime, temporary shifts and task transitions are minimised.

The largest body of related research concerns preemptive tasks. When preemptive tasks are considered fixed in time, they are typically modelled as varying staffing requirements in time intervals. The problem presented by Detienne et al. [8] requires predefined work patterns and activities to be assigned to employees. The costs associated with the assigned work patterns should be minimised. Guyon et al. [13] extend this work by introducing time windows for the tasks. Côté et al. [7] use implicit models with context-free grammars to model complex rules regarding shift design. Musliu et al. [19] study the minimum shift design problem. The goal is to produce an efficient shift structure and a minimal workforce to undertake all the work without, however, explicitly assigning tasks within the shifts.

Robinson et al. [22] address a problem in which the preemptive tasks are defined by a release date and a deadline, such that their time of execution must also be determined. A tabu search algorithm is used to construct a days-off roster, after which, for each day, a network flow model generates task and shift assignments that minimise personnel costs. Brucker and Qu [3] extend this model with qualification requirements. Brucker et al. [4] study the complexity of various personnel scheduling models, including a project centered planning model that integrates preemptive task scheduling and work pattern assignment to employees. The objective is to minimise the completion time of the project.

Table 1 presents an overview of related work and situates the present contribution. The combination of fixed shifts and multiple non-preemptive tasks in a setting with multiple qualifications has received little or no attention in the academic literature, despite clearly having many practical applications. The intricate coupling between task and shift assignments results in an integrated problem which was previously deemed computationally impractical for realistically sized instances [11].

3. Problem definition

Let T be the set of tasks and E the set of employees. Typically, for each task t, only a subset of employees $E_t \subseteq E$ satisfy the qualification requirements. For each employee e, let $G_e = (V, A)$ be an interval graph with V the set of nodes and A the set of arcs. Each node in V corresponds to a task for which e is qualified. Two nodes are connected when their respective time intervals overlap. The set of maximal cliques in the G_e is defined as C_e. Clearly, each clique in C_e represents a subset of tasks that overlap in time, and of which at most one can be assigned to employee e. A polynomial time algorithm for finding all maximal cliques in G_e is presented by Krishnamoorthy et al. [15].

Let S be the set of shifts and D the set of days in the scheduling period. Let d_t denote the day on which task t must be scheduled. For each task t, the set $S_t \subseteq S$ consists of the shifts in which task t can be assigned based on the task's time interval. Breaks are not explicitly assigned, however, they can be taken into account provided they are fixed in time. All tasks overlapping with the fixed break should then be removed from S_t. For modelling purposes, the set S also contains a dummy shift s_0, representing a day-off. Shifts spanning two days, such as night shifts, are not considered in the model, that is, each shift starts and ends on the same day. The model makes no further assumptions concerning the structure of the shifts in S.

The objective is to minimise the schedule's cost, defined as a weighted sum of soft constraint violations. In practice, it is often impossible to respect all contractual constraints, as they are often imposed by authorities with conflicting priorities. Therefore, the established methodology for dealing with such constraints is to consider them as soft and penalise violations in the objective

Table 1
Overview of related work.

Reference	Non-preemptive tasks	Fixed tasks	Fixed shifts	Qualifications	Objective
TSPR	Yes	Yes	Yes	Yes	Weighted sum of constraint violations
Krishnamoorthy et al. [15]	Yes	Yes	Yes	Yes	Number of employees
Meisels and Schaerf [18]	Yes	Yes	Yes	Yes	Feasibility
Dowling et al. [9]	Yes	Yes	No	Yes	Deviation from coverage requirements
Lapègue et al. [16]	Yes	Yes	No	Yes	Deviation from targeted workload
Detienne et al. [8]	No	Yes	Yes	Yes	Scheduling costs
Côtè et al. [7]	No	Yes	No	Yes	Scheduling costs
Musliu et al. [19]	No	Yes	No	No	Number of different shifts
Brucker et al. [4]	No	No	Yes	Yes	Project completion time
Guyon et al. [13]	No	No	Yes	Yes	Scheduling costs
Brucker and Qu [3]	No	No	No	Yes	Scheduling costs
Robinson et al. [22]	No	No	No	No	Scheduling costs
Elahipanah et al. [10]	Both	No	No	Yes	Understaffing, overtime, temporary shifts and transition costs