European Journal of Operational Research 000 (2016) 1–12

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R

Column generation based approaches for a tour scheduling problem with a multi-skill heterogeneous workforce

Matthieu Gérard[a,b,c], François Clautiaux[d,e,*], Ruslan Sadykov[d,e]

[a] Company ASYS Paris, France
[b] Inria Lille - Nord Europe, France
[c] Université de Lille 1, France
[d] Institut de Mathématiques de Bordeaux (UMR CNRS 5251), Université de Bordeaux, 351, Cours de la Liberation - F 33 405 Talence, France
[e] Inria Bordeaux - Sud Ouest, France

A R T I C L E I N F O

Article history:
Received 6 March 2015
Accepted 20 January 2016
Available online xxx

Keywords:
Employee scheduling
Integer programming
Branch-and-price
Heuristics

A B S T R A C T

In this paper, we address a multi-activity tour scheduling problem with time varying demand. The objective is to compute a team schedule for a fixed roster of employees in order to minimize the over-coverage and the under-coverage of different parallel activity demands along a planning horizon of one week. Numerous complicating constraints are present in our problem: all employees are different and can perform several different activities during the same day-shift, lunch breaks and pauses are flexible, demand is given for 15 minutes periods. Employees have feasibility and legality rules to be satisfied, but the objective function does not account for any quality measure associated with each individual's schedule. More precisely, the problem mixes simultaneously days-off scheduling, shift scheduling, shift assignment, activity assignment, pause and lunch break assignment.

To solve this problem, we developed four methods: a compact Mixed Integer Linear Programming model, a branch-and-price like approach with a nested dynamic program to solve heuristically the subproblems, a diving heuristic and a greedy heuristic based on our subproblem solver. The computational results, based on both real cases and instances derived from real cases, demonstrate that our methods are able to provide good quality solutions in a short computing time. Our algorithms are now embedded in a commercial software, which is already in use in a mini-mart company.

1. Introduction

Employee scheduling is an important issue in retail (see Kabak, Ülengin, Aktas, Önsel, & Topcu, 2008), as personnel wages account for a large part of their operational costs. This problem raises considerable computational difficulties, especially when certain factors are considered, such as employee availability, fairness, strict labor rules, highly variable work demand, mixed full and part-time contracts, etc. Since the seminal work of Dantzig (1954), a large quantity of research papers have developed models and methods to assist managers and planners in their employee scheduling tasks (more than 300 papers published between 2004 and 2012 were surveyed in Van Den Bergh, Beliën, De Bruecker, Demeulemeester, & De Boeck (2012)). For a comprehensive literature review of classical studies on this problem, we refer to (Ernst, Jiang, Krishnamoorthy, & Sier, 2004).

In this paper, we study a real-life multi-activity tour scheduling problem with highly heterogeneous employees and flexible working hours. Given a fixed set of employees, the objective is to construct their work schedule or planning that minimizes the distance to the ideal coverage of the demand. Numerous complicating factors described in the literature are taken into account and, to the best of our knowledge, this paper is one of the first attempts (in parallel with Restrepo, Gendron, & Rousseau, 2015) to combine days-off scheduling, shift scheduling, shift assignment, activity assignment, pause and lunch break assignment.

Several features of our problem are still considered as major issues in the recent literature (Van Den Bergh et al., 2012): individual constraints and flexibility of employees, integrated days-off, shift scheduling and assignment (Lequy, Bouchard, Desaulniers, Soumis, & Tachefine, 2012a) and multi-activity

* Corresponding author at: Institut de Mathematiques de Bordeaux (UMR CNRS 5251), Universite de Bordeaux, 351, Cours de la Liberation - F 33 405 Talence, France. Tel.: +33 540002137; fax: +33 540002123.

E-mail address: francoisclautiaux@gmail.com, francois.clautiaux@math.u-bordeau.fr (F. Clautiaux).

http://dx.doi.org/10.1016/j.ejor.2016.01.036
0377-2217/

assignment (Lequy, Desaulniers, & Solomon, 2012b; Quimper & Rousseau, 2010; Restrepo, Lozano, & Medaglia, 2012). Although the *lunch break assignment* between two timeslots is taken into account in most research papers, *pause assignment* during activities themselves remains a gap in the academic literature (see Thompson & Pullman, 2007). To our knowledge, only (Widl & Musliu, 2014) deals with both types of breaks at the same time.

Although integer linear programming (ILP) models exist for this family of problems, they cannot be used directly to solve large scale problems with many constraints. Therefore, several works propose heuristics based on those ILP models to reduce their computational burden. Heuristic methods can be obtained by applying a hierarchical decomposition (see e.g. Hojati & Patil, 2011). First, good shifts are computed, and then employees are assigned to the shifts in a second phase. Unfortunately, this technique cannot be applied directly to our problem, where each employee can change activity during his shift and has his very specific features such as availabilities, skills and pre-assignments. When the time horizon is large, and the problem can be solved for a smaller time horizon (typically one week) without risking infeasibilities for the planning, an interesting approach (Stolletz & Zamorano, 2014) is to use a rolling horizon heuristic, where the problems related to smaller time horizons are solved in an iterative manner. In our problem, the total number of worked hours for each employee is fixed, which may lead such method to unfeasible schedules.

Many algorithms for solving such employee scheduling problems are based on the column-generation approach (see for example Eitzen, Panton, & Mills, 2004). Recent papers address shift or tour scheduling problems with branch-and-price methods. Boyer, Gendron, and Rousseau (2014); Côté, Gendron, and Rousseau (2013) and Restrepo et al. (2015) use branch-and-price to solve very general multi-activity shift scheduling problems. Their approaches rely on the description of shifts using a context-free grammar. Another recent work on the subject was realized by Brunner and Stolletz (2014). They use an ad-hoc branch-and-price method to solve a tour scheduling problem. The main ingredients of their approach are the use of variables related to day-shifts, which are recombined in the master problem, and stabilization strategies to reduce the number of column generation iterations. Another recent work (Dohn & Mason, 2013) uses branch-and-price in the context of employee-scheduling. They use a nested dynamic programming approach, which is well-suited to the structure of their problem.

Our approach is also based on a branch-and-price algorithm. However, the problem settings do not allow us to use directly the algorithms from Boyer et al. (2014); Côté et al. (2013); Restrepo et al. (2015). In our problem, each employee is different, the time horizon is much larger than the ones in Boyer et al. (2014); Côté et al. (2013), and many constraints restrict the construction of the shifts. This leads to a prohibitively large pricing problem solution time. Since our aim is to handle real-life instances, we had to use a heuristic version of the branch-and-price, where some constraints are treated heuristically in the subproblem. The hierarchical structure of our shifts called for an ad-hoc specific nested dynamic program (like Dohn & Mason, 2013), which proves to be much more efficient than a straightforward dynamic programming approach.

An important practical requirement is to find a good solution in a short amount of time (a few seconds for 100 employees). To respect this time limit, we designed a greedy algorithm based on our dynamic program. Also, a diving heuristic is proposed for cases when we have several minutes of computational time. Our algorithms have been implemented and are now embedded in a commercial software. They are able to find feasible solutions with good quality in a small or reasonable time for all test cases that were provided by our industrial partner. Our algorithms are now in use in a mini-mart company.

In Section 2, we describe formally our problem. Our column generation framework is presented in Section 3, followed by the nested dynamic program used to solve the pricing problem in Section 4. Our heuristic algorithms based on column generation are presented in Section 5, while computational experiments on real and generated instances are reported in Section 6.

2. Problem description

The problem consists in scheduling a fixed workforce to maximize the fit to a given time-varying demand. The planning horizon consists of $\mathcal{D}$ consecutive days. Each day is divided into the same number of successive time periods of equal length (15 minutes in this paper). Set $\mathcal{T}$ represents the different time periods in the discrete planning horizon. The set of heterogeneous employees is denoted by $\mathcal{E}$.

The whole set of activities that employees can carry out is divided into two distinct groups: *production activities* $\mathcal{A}$, related to work demands, and *pause activities* $\mathcal{P}$, related to non-productive activities. In our retail context, a production activity can represent, for example, the welcome desk, a cash desks line or a meat counter. Each employee $e \in \mathcal{E}$ has a set of production activities $\mathcal{A}(e, t)$ that he/she can perform at time period *t*. Set $\mathcal{P}(e, t)$ contains a pause if employee can take it at time period *t*; this set is empty otherwise. The beginning and the length of a pause are strictly constrained by the personalized pause policy of the company agreement. An employee *e* is unavailable at time period *t* if $\mathcal{A}(e, t) \cup \mathcal{P}(e, t) = \emptyset$. In this case, the planning computed for employee *e* cannot contain any activity at time *t*. Note that if an employee is unavailable the entire day, then a day-off has to be scheduled. Some employees may be pre-assigned to activities for certain time periods. In this case, finding a schedule that respects this pre-assigned tasks is a part of the problem.

The work demand $\mathrm{DE}_{a,t}$ represents the ideal number of employees needed to realize production activity *a* in the best possible conditions during time period *t* (see the representation given in Fig. 1). Satisfying exactly the demand is not mandatory : in most cases it is not possible. In this case, either an under-coverage, or an over-coverage is produced. Furthermore, if over-coverage (respectively under-coverage) exceeds the given threshold $\mathrm{OV}_{a,t}$ (respectively $\mathrm{UN}_{a,t}$), then it becomes critical and indicates that too many (respectively too few) employees have been assigned to activity *a* during time period *t*.

Our objective is to construct a feasible team schedule that minimizes the sum of the over-coverage and under-coverage costs for the whole planning horizon and all production activities.

2.1. A hierarchical structure of a team schedule

A feasible solution follows a hierarchical structure (see Fig. 2). For each level of the hierarchy, there is an associated set of constraints. This flexible structure does not rely on the use of a pre-computed day-shift or individual planning library, since the number of possibilities is far too large.

- A *team schedule* consists of a set of $|\mathcal{E}|$ valid employee plannings.
- An *individual planning* for employee *e* is a set of successive *day-shifts* and *days-off* over a week. Two consecutive day-shifts are separated by a *rest break*.
- A *day-off* represents a special day when employee *e* does not participate in any activity. Deciding whether or not an employee takes a day-off is part of the optimization process (but some days-off are mandatory if the employee is unavailable).