A Hybrid LS-CP Solver for the Shifts and Breaks Design Problem

Luca Di Gaspero · Johannes Gärtner · Nysret Musliu · Andrea Schaerf · Werner Schafhauser · Wolfgang Slany

1 Introduction

The problem of designing workforce shifts and breaks patterns is a relevant employee scheduling problem that arises in many contexts, especially in service industries. The issue is to find a minimum number of shifts, the number of workers assigned to them, and a suitable number of breaks so that the deviation from predetermined workforce requirements is minimized.

We tackle this problem by means of a hybrid strategy in the spirit of Large Neighborhood Search, which blends a Local Search based procedure for determining the shifts, with a Constraint Programming model for assigning breaks. This is a preliminary work and experimentation is currently underway on a set of benchmark instances employed in the literature.

2 Problem definition

Formally, we are given a set D of days, which are subdivided into a set of equally long timeslots. The *planning horizon* is therefore a set $\{\tau_1, \tau_2, \tau_3, \ldots, \tau_h\}$ of consecutive *timeslots* at a given time granularity, each belonging to a single day d. Moreover, for each timeslot τ, we are given a *staffing requirement* r_τ, which indicates the number of employees that should be working during timeslot τ.

L. Di Gaspero, A. Schaerf
DIEGM, University of Udine
E-mail: l.digaspero@uniud.it, schaerf@uniud.it

J. Gärtner
Ximes Inc., Austria
E-mail: gaertner@ximes.com

N. Musliu, W. Schafhauser
DBAI, Technische Universität Wien, Austria
E-mail: musliu@dbai.tuwien.ac.at, schafha@dbai.tuwien.ac.at

W. Slany
IST, Technische Universität Graz, Austria
E-mail: wolfgang.slany@tugraz.at

The problem consists in designing the shifts and break patterns, i.e., determining the starting time σ_i and the length λ_i of each shift s_i and, for each day d, the number of employees assigned w_{id} and the start α_{idw} and length β_{ide} of breaks for each employee e. An employee is considered to be working during the timeslots comprised in the shift but not in any of his/her breaks in that shift. More formally, an employee e works on timeslot $\tau \in d$ if $\tau \in [\sigma_i, \sigma_i + \lambda_i]$ and $\tau \notin [\alpha_{idw}, \alpha_{idw} + \beta_{idw}]$.

3 Local Search and Constraint Programming models

In the context of this problem it is useful to define an `Interval` as a structure of two variables `start` and `length`, which entirely determines a shift, a working period or a break.

Local Search deals with a search space composed of a set of shifts $\mathcal{S}$, each of them is characterized by the following decision variables:

- the interval spanned by the shift;
- the number of employee assigned on each day of the planning horizon;
- for each each day the number of breaks each employee has to take.

Notice that Local Search works on a partial representation of the solution, since the breaks are only specified in their number and not in the intervals they span. To complete this representation to a full solution we resort to a Constraint Programming model (described below) whose purpose is to determine the interval variables of each break.

The Neighborhood relations considered are similar to those employed in [1], slightly modified to deal with the addition of the number of breaks, plus some new move dealing directly with the break component. In detail we make use of the following moves:

- *Change Staff*: the staff of a shift in a given day is increased or decreased by one employee.
- *Resize Shift*: the length of a shift is increased or decreased by one timeslot, either on the left-hand side or on the right-hand side.
- *Insert Shift*: insert a new shift in the solution belonging to a given shift type.
- *Merge Shift*: two shifts are merged together and the employees assigned to them are added; the interval of the outcoming shift as well as the number of breaks for each day are the average of those of the two shifts merged.
- *Change Breaks*: the number of breaks of a shift in a given day is increased or decreased by one.

The cost function is the weighted sum of the deviation (excess and shortage, with different weights) from the working requirements at each timeslot plus another weighted component that accounts for the number of shifts employed in the solution. Notice that for an accurate computation of the deviation from the requirements a full solution is needed, therefore the cost function has to be computed only after a full solution has been determined by the CP model.

4 Conclusions and Future work

The proposed idea is still at an early development stage and the solver experimentation is currently underway on a set of benchmark instances available from the literature.

Preliminary results show that this approach can be feasible to find good quality solutions employing a reasonable number of shifts. However, at present we do not have a full understanding about the contribution of each neighborhood to solution quality. Moreover also the CP model could benefit of some improvement, for example by adding implied constraints which allow for a more accurate constraint propagation and performing a principled evaluation of different heuristics for variable and value selection.

References

1. Luca Di Gaspero, Johannes Gärtner, Guy Kortsarz, Nysret Musliu, Andrea Schaerf, and Wolfgang Slany. The minimum shift design problem. *Annals of Operations Research*, 155(1):79–105, 2007.