A column generation scheme for faculty time-tabling

Paolo Serafini, University of Udine, Italy

Abstract

In this paper we propose an integer linear programming approach to design faculty time-tables based on a column generation scheme. The problem we face consists of courses, classrooms and time slots. Courses must be assigned to both classrooms and time slots by respecting constraints of non simultaneous use of the same classrooms in the same time slot, non overlapping in time of certain groups of lectures, impossibility of simultaneity of two courses if taught by the same person. These are the main constraints we have to take care of. Besides the lecturers express preferences on the time slots.

The problem is usually modeled with binary variables expressing the fact that a certain course has been assigned to certain time slots and to certain classrooms, and constraining the variables accordingly. However, the integrality relaxation of this model does not provide a strong lower bound. Besides, in order to finely adjust the output of the ILP problem, it is difficult to interact in a flexible way with the ILP model in order to 'drive' it to a better solution. You may only fix a variable (thus freezing part of the time table) and, more important, it is not possible to relate the different time slots used by a particular course.

Here we propose an alternative model in which the variables correspond to time-table patterns for each course. Since the number of variables is exponential we must resort to a column generation approach. We will see that the subproblem to generate a column is very easy.

Let us now describe formally the problem. K is the set of classroom types. Classrooms of the same type are interchangeable. Thus it is simpler to first assign a course to a classroom type and later, via a simple assignment problem, assign the course to a specific classroom. Therefore in this phase we only consider classroom types. Let n_k be the number of classrooms of type $k \in K$. H is the set of time slots within a week. C is the set of courses. $C_q \subset C$, $q \in Q$, is a set of non overlapping courses. The reason for non overlapping may be the fact that the courses belong to the same year group, or they are taught by the same lecturer, or something else. Let $Q(c) := \{q \in Q : c \in C_q\}$. $P(c)$ is the set of time table patterns for the course c. Let us now define the following matrices:

$$a_{(kh)(jc)} = \begin{cases} 1 & \text{if course } c \text{ is assigned the time slot } h \text{ in a classroom of type } k \text{ for the pattern } j \in P(c) \\ 0 & \text{otherwise} \end{cases}$$

$$a'_{(h)(jc)} = \begin{cases} 1 & \text{if course } c \text{ is assigned the time slot } h \text{ for the pattern } j \in P(c) \\ 0 & \text{otherwise} \end{cases}$$

(clearly $a'_{(h)(jc)} = 0$ if and only if $a_{(kh)(jc)} = 0$ for each $k \in K$)

The variables are

$$x_{jc} = \begin{cases} 1 & \text{if pattern } j \text{ is used for course } c \\ 0 & \text{otherwise} \end{cases}$$

The constraints are as follows: to avoid simultaneous use of more than n_k classrooms of type k

$$\sum_{c \in C} \sum_{j \in P(c)} a_{(kh)(jc)}\, x_{jc} \le n_k \qquad k \in K, \quad h \in H \tag{1}$$

To impose non overlapping of courses in the same group q

$$\sum_{c \in C_q} \sum_{j \in P(c)} a'_{(h)(jc)}\, x_{jc} \leq 1 \qquad h \in H, q \in Q \tag{2}$$

To impose that a course is assigned a pattern

$$\sum_{j \in P(c)} x_{jc} = 1 \qquad c \in C \tag{3}$$

The objective function is the maximization of a preference

$$\sum_{c \in C} \sum_{j \in P(c)} r_{jc}\, x_{jc}$$

where

$$r_{jc} = \sum_{h \in H} a'_{(h)(jc)}\, s_{hc}$$

and s_{hc} is the preference of using the time slot h for the course c.

Let us define the dual variables w_{kh}, v_{hq}, u_c for the constraints (1), (2) and (3) respectively. The dual constraints are

$$\sum_{h \in H} \sum_{k \in K} a_{(kh)(jc)}\, w_{kh} + \sum_{h \in H} \sum_{q \in Q(c)} a'_{(h)(jc)}\, v_{hq} + u_c \geq \sum_{h \in H} a'_{(h)(jc)}\, s_{hc} \qquad j \in P(c), \quad c \in C$$

i.e.

$$\sum_{h \in H} (\sum_{k \in K} a_{(kh)(jc)}\, w_{kh} + \sum_{q \in Q(c)} a'_{(h)(jc)}\, v_{hq} - a'_{(h)(jc)}\, s_{hc}) + u_c \geq 0 \qquad j \in P(c), \quad c \in C$$

So in order to generate a time-table j for the course c, we have to minimize, with respect to a and a'

$$\sum_{h \in H} (\sum_{k \in K} a_{(kh)(jc)}\, w_{kh} + \sum_{q \in Q(c)} a'_{(h)(jc)}\, v_{hq} - a'_{(h)(jc)}\, s_{hc}) \tag{4}$$

Let us define $\hat{w}_{hc} = w_{kh}$ with k the classroom type required for the course c. If the course may be assigned to different classroom types then we may define $\hat{w}_{hc} = \min_k w_{kh}$ where the minimum is computed on the admissible k. Let us also define $\hat{v}_{hc} := \sum_{q \in Q(c)} v_{hq}$ Then minimizing (4) is equivalent to minimize, for each c

$$\sum_{h} (\,\hat{w}_{hc} + \hat{v}_{hc} - s_{hc})\, a'_{(h)(jc)} \tag{5}$$

Minimizing (5) is immediate. If we need to allocate d time slots for the course c we just select the d minimum values of $(\,\hat{w}_{hc} + \hat{v}_{hc} - s_{hc})$. Quite often we are not allowed to assign more than one time slot per day to a course. In this case the minimization of (5) is carried out by selecting the d minimum values of $(\,\hat{w}_{hc} + \hat{v}_{hc} - s_{hc})$ on different days.

Let M_c be the minimum obtained for the course c. If $M_c + u_c \geq 0$ the optimalty condition is satisfied for the course c whereas if $M_c + u_c < 0$ the time table obtained by minimizing (5) has to be inserted into the matrix.

The number of rows of the problem is given by $|H| \cdot |K| + |H| \cdot |Q| + |C|$, which can be large but not intractable (in our case it is around one thousand). The first computational experiments of the model show, as expected, a rather strong lower bound (the gap between the LP solution and the first incumbent is less than 1 %). The strategy to generate a 'good' solution (we don't pursue exact optimality since, after all, the objective function is artificial) is the following: if we end the column generation procedure with a fractional solution, we invoke a ILP routine on the generated columns to get a first incumbent. Then we start a branch-and-bound procedure by setting fractional variables to either 0 or 1. Since the lower bounds are good the search tree is rather small. However, we have to solve the problem of discarding generated columns which correspond to patterns already set to 0. This problem can be fixed if, in minimizing (5) for a certain subproblem, we compute the first α minima with $\alpha - 1$ the number of variables set to 0 for that particular subproblem. It can be shown that finding the first α minima can be done in time $O(\alpha \log d)$ (once the values $(\hat{w}_{hc} + \hat{v}_{hc} - s_{hc})$ have been sorted)

Let us note that we may introduce manually particular time-tables for a certain course and assign them a high coefficient in the objective function if we want to favour their appearance in the final schedule. We are currently experimenting this model.