DM872 - Course Timetabling

Thoso16

May 19, 2020

In the following, a model for course timetabling together with results for a specific data is presented. The model is a two-stage model where first events are given a time and then in stage 2, each event at the given time is assigned a room. First some notation must be introduced.

1 Notation

The notation is separated for each state of the model because some of the symbols change meaning in the second stage of the model. To start of, it is important to note that the timetabling is done over a period of 17 weeks, each consisting 5 days. This fact is used in the indexing of the mathematical formulation of the problem, but the formulation does not depend on it, meaning that one can change the number of weeks or days with out it affecting the model itself.

Stage 1

Sets

The most important sets are the following:

- E is the set of all events,
- T is the set of all available timeslots, where each $t \in T$ represents a period $(t, t+1)$, i.e. a period of one hour,
- B is the set of banned timeslots,
- R is the set of all rooms,
- $\mathcal{W}_e = \{W_1^e, W_2^e, \ldots, W_{17}^e\}$, where W_i^e is the set of events, $e \in E$, that must occur in week i,
- $\mathcal{W}_t = \{W_1^t, W_2^t, \ldots, W_{17}^t\}$, where W_i^t is the set of timeslots, $t \in T$, for week i,
- $\mathcal{W}_d = \{W_1^d, W_2^d, \ldots, W_{17}^d\}$, where $W_i^d = \{D_1, D_2, \ldots, D_5\}$ and $D_j = \{t \in T : t \in W_i^t, t \textit{ is a time for day } j\}$,
- $\mathcal{R} = \{R_t : t \in T\}$ where $R_t = \{r \in R : r \textit{ is available at time } t\}$,
- $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ where each $S_i \subset E$ for which only one of the events may happen each day.,
- $\mathcal{A} = \{(e_1, e_2) \in E \times E : e_1 \textit{ must happen before } e_2\}$
- $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ where each $C_i \subset E$ for which the events are not allowed to overlap,

- $\mathcal{D}$ is the set of different durations,
- E_d is the set of events with duration d,
- R_d^t is the set of rooms available in the period $(t, t+d-1)$, where d is the duration,
- $d(e)$ is the duration of $e \in E$,
- $d_0(t)$ is the first timeslot on the day that $t \in T$ belongs to.

Variables

The model consists of three variables. One main variable, x, which is binary, and two auxiliary variables w and y, which are non-negative integers. The main variable is two-dimensional and it is indexed by $I = \{(e,t) : e \in E, t \in T and\ e\ can\ start\ at\ t\}$. as indicated in the definition of I, the time component of x represents the starting time of the corresponding event. The excluded elements,$\{(e,t) in E \times T \setminus I\}$, are elements belonging to any of the following sets:

- $\bigcup_{i=1}^{17} \{(e,t) \in E \times T : e \in W_i^e, t \in \bigcup_{j=1, i \neq j}^{17} W_j^t\}$,
- $\{(e,t) \in E \times T : \forall t_b \in B\ s.t.\ t_b > t, t_b - t < d(e)\}$.

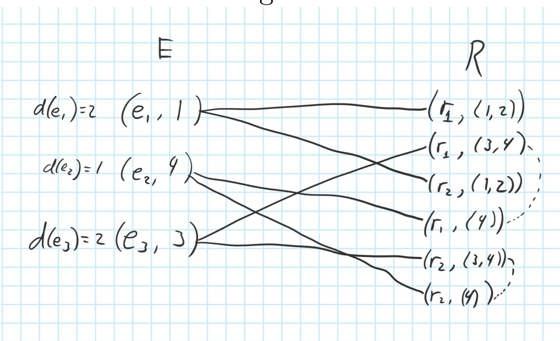
The auxiliary variables, w and y, are only used for optimizing the timetable with respect to the number of overlaps for student and the number times a teacher has a lecture on a day where a lecture has already been held, respectively. There w will be reffered to as the variable modeling the student conflicts and y will be reffered to as the variable that models teaching duties beyond a single duty per day. If we have $\mathcal{C}$ for the students then w is indexed by the set $I_s = \{(C,t) \in \mathcal{C} \times T : events\ in\ C\ belong\ to\ the\ same\ week\ as\ t\}$. If we instead have $\mathcal{C}$ for the teachers then y is indexed by $I_t = \bigcup_{i=1}^{17} \{(C,D) \in \mathcal{C} \times W_i^d : C \subset W_i^e\}$.

Stage 2

In the second stage, we only have three sets that we work with. The first set is denoted E and this consists of the solutions from the first stage, i.e. each element in E is a tuple consisting of an event and the time at which the event has been scheduled. The second set, R, is generated based on the first set such that for each $(e,t) \in E$ vertices for every room that can be used by that event at that event at that time. Thus, if the room, r, is not occupied in the period $(t, t+d(e)-1)$, where $d(e)$ is the duration of e, then the vertex $(r, (t, t+d(e)-1))$ is added to R. Some of the vertices in R are not allowed to be chosen at the same time, an example would be $(r_1, (1,2))$ and $(r_1, (2))$. The first one represents an occupation of room r_1 in the period $(1,3)$

where the room becomes available at time 3, but the other one represents that room r_1 is used in the period $(2,3)$ and therefore only one of these vertices may be chosen. We let A be one such set of vertices in R and let $\mathcal{A}$ be the collection of all such sets.

The second stage only has one binary variable $x_{e,r}$ indexed by all the arcs between the sets E and R. The following is a visualisation of the components of the second stage:



The solid arcs represent the valid arcs over which the variable is defined and the two set of vertices connected by a dotted line each represent a set in $\mathcal{A}$.

2 The model

Stage 1

Hard constraints

The first thing to ensure is that every event is scheduled and this is with the following constraint

$$\sum_{t \in W_i^t} x_{e,t} = 1, \forall i = 1, 2, \ldots, 17 \quad \forall e \in W_i^e. \tag{1}$$

Next there should not be scheduled more events at the same time than there are rooms available and since all events may be held in any room, this requirement is ensured by:

$$\sum_{e \in W_i^e} \sum_{l=\max(d_0(t), t-d(e)+1)}^{t} x_{e,l} \leq |R_t|, \forall i = 1, 2, \ldots, 17 \quad \forall t \in W_i^t. \tag{2}$$

Note that the constraints above, do not ensure that an event stays in the same room in its entire duration. To ensure this, the following constraint is added:

$$\sum_{e \in E_d} x_{e,t} \leq |R_d^t|, \forall t \in T, \quad \forall d \in \mathcal{D}. \tag{3}$$

To get the most out of every course no student should have more than one event per course per day:

$$\sum_{e \in S} \sum_{t \in D} x_{e,t} \leq 1, \forall S \in \mathcal{S} \quad \forall D \in W^d, \tag{4}$$

where for each S, W^d is the week corresponding the one where the events in S should be scheduled. For most courses there will be more than one type of event. In most cases there will be an intro class and a TE class and in every week, one would like the intro class to precede the TE class. Since the constraint (4) ensures that these events happen on different days, it suffices to ensure that the day of the intro class precedes the day of the TE class. This is done as follows:

$$\sum_{t \in D}(x_{e_1,t} - x_{e_2,t}) \geq 0, \forall (e_1, e_2) \in A \quad \forall D \in \{D_1, D_1 \cup D_2, D_1 \cup D_2 \cup D_3, D_1 \cup D_2 \cup D_3 \cup D_4\}, \tag{5}$$

where the D_i's belong to the W^d corresponding to the week for which e_1 and e_2 should be scheduled. Note that is suffices to only include for of the five days again due to (4). Another very important requirement is that there are no overlaps for the teachers. Let $\mathcal{C}$ contain the overlaps for the teachers, then overlaps can be avoided by including

$$\sum_{e \in C} \sum_{l=\max(d_0(t), t-d(e)+1}^{t} x_{e,l} \leq 1, \forall C \in \mathcal{C} \quad \forall t \in D, \tag{6}$$

where $D \in W^d$ for the week corresponding to the events in C. This has not been done due to lack of time, but one could also include the possibility of paired courses. The most straight forward way to include this would be to set the following constraint for each pair of events, (u, v), from the two courses:

$$x_{u,t} - x_{v,t} = 0, \forall t \in T.$$

A more efficient way would be to remove the events from one of the courses and thus only schedule that course, keeping in mind that the events from the other course should be scheduled simultaneously. However, to do this one must ensure that the information about teacher and student conflict is preserved. So if a student participated in the course that is removed, one should treat the problem as if the student now attends the other course.

Soft constraints

To ensure that w models the student overlaps, the following constraint is introduced with $\mathcal{C}$ being defined for the students:

$$\sum_{e \in C} \sum_{l=\max(d_0(t), t-d(e)+1}^{t} x_{e,l} \leq w_{C,t} + 1, \forall (C, t) \in I_s. \tag{7}$$

If at time t there is more than one ongoing even in C then $w_{C,t}$ is bounded from below by the number of overlaps at that time. Otherwise it is by definition bounded from below by 0 and thus when minimizing the sum over all w's one obtains the number of student overlaps. By the same idea, the constraint is set up for the variable y:

$$\sum_{e\in C}\sum_{t\in D} x_{e,t} \leq y_{C,D} + 1, \forall (C,D) \in I_t. \tag{8}$$

To avoid certain "bad" timeslots, one could simply add the following to the objective function:

$$\sum_{(e,t)\in I} p(t) * x_{e,t}, \tag{9}$$

where the function $p(t)$ assigns a cost to the timeslots that are less desirable depending on their desirability and a cost of 0 to all timeslots for which there are no preferences.

The full model for stage 1

The full model then becomes

$$\min c_1 * \sum_{i\in I_s} w_i + c_2 * \sum_{j\in I_t} y_j + c_3 * \sum_{(e,t)\in I} p(t) * x_{e,t}$$

$$\begin{aligned}
s.t. \sum_{t\in W_i^t} x_{e,t} &= 1, \forall i = 1,2,\ldots,17 \quad \forall e \in W_i^e,\\
\sum_{e\in W_i^e} \sum_{l=\max(d_0(t),t-d(e)+1)}^{t} x_{e,l} &\leq |R_t|, \forall i = 1,2,\ldots,17 \quad \forall t \in W_i^t,\\
\sum_{e\in S}\sum_{t\in D} x_{e,t} &\leq 1, \forall S \in \mathcal{S} \quad \forall D \in W^d,\\
\sum_{e\in S}\sum_{t\in D} x_{e,t} &\leq 1, \forall S \in \mathcal{S} \quad \forall D \in W^d,\\
\sum_{t\in D}(x_{e_1,t} - x_{e_2,t}) &\geq 0, \forall (e_1,e_2) \in A \quad \forall i = 1,2,3,4, \quad \forall D \in \bigcup_{j=1}^{i} D_j,\\
\sum_{e\in C} \sum_{l=\max(d_0(t),t-d(e)+1)}^{t} x_{e,l} &\leq 1, \forall C \in \mathcal{C} \quad \forall t \in D,\\
\sum_{e\in C} \sum_{l=\max(d_0(t),t-d(e)+1)}^{t} x_{e,l} &\leq w_{C,t} + 1, \forall (C,t) \in I_s,\\
\sum_{e\in C}\sum_{t\in D} x_{e,t} &\leq y_{C,D} + 1, \forall (C,D) \in I_t,
\end{aligned}$$

where c_1, c_2 and c_3 are chosen based on the importance of the soft constraints.

A remark about the implementation of the model is that it has not been implemented in its entirety all at once because there are exponentially many constraints for the overlaps for teachers and the rooms(both ensuring non-overlap and ensuring the same room is used for the entire duration of an event). Therefore these three types of constraints are added as needed be iteratively solving the model and adding the constraints for respectively teachers, times, and times and durations for which the constraints are violated. Note also that this helps avoid redundant constraints because some teachers may teach both the intro class and the TE class, but due to (4) these cannot overlap.

Stage 2

In stage two a linear program for assigning each element in E to an element in R is set up. There are a couple of ways to go about this set up and they differ by how one deals with the sets in $\mathcal{A}$. One way would be to introduce a constraint dealing with those sets and another way would be to for each $A \in \mathcal{A}$ merge the vertices and set all edges in the cut of the set to be connected to the merged vertex. Here the program is set up with the additional constraint, but if one merges the vertices in each $A \in \mathcal{A}$ then this constraint may simply be removed. The constraint is

$$\sum_{e \in E} \sum_{r \in A} x_{e,r} \leq 1, \forall A \in \mathcal{A}. \tag{10}$$

The last two constraints are similar to the ones in a one-sided matching problem. The first ensures that each event is assigned a room

$$\sum_{r \in R} x_{e,r} = 1, \forall e \in E, \tag{11}$$

and the last ensures that two or more events are not assigned the same room at the same time:

$$\sum_{e \in E} x_{e,r} \leq 1, \forall r \in R \setminus \bigcup_{i=1}^{|\mathcal{A}|} A. \tag{12}$$

Thus, the full model is

$$\min 0 \tag{13}$$

$$s.t. \sum_{r \in R} x_{e,r} = 1, \forall e \in E, \tag{14}$$

$$\sum_{e \in E} x_{e,r} \leq 1, \forall r \in R \setminus \bigcup_{i=1}^{|\mathcal{A}|} A, \tag{15}$$

$$\sum_{e \in E} \sum_{r \in A} x_{e,r} \leq 1, \forall A \in \mathcal{A}. \tag{16}$$

Since the function to minimize is constant, it is simply a feasible solution that is found.

3 Results

The most important thing is to check that the model indeed works as intended. To do so the different constraints are checked for the small data set. The constraint (1) is easily checked by comparing the sum, $\sum_{(e,t)\in I} x_{e,t}$ with the number of events and they both give 180. To test the room constraints, (2), all constraints except (1), which causes the matching problem to become infeasible. This is to be expected since there may have been scheduled too many events at the same time compared to the number of rooms available at that time. When doing this 13 events are scheduled on Friday at 4 pm in week 11 but only 12 rooms available. Adding the room constraints fixes this such that only 12 events are scheduled at that time. To check that no student has more than one event per day per course and that teacher conflicts are avoided we look at week 18. Before adding either of the two one obtains a table with all events but one being on Friday at 4-6 pm. The last one is the intro class for DM803 which is on Friday at 10 am to 12 pm. Adding the constraint (4), one obtains a table with all intro classes being on Monday at 8 am to 10 am except for the second intro class of DM872 and the second intro class of DM865. They are on Tuesday at 8 am to 10 am and Friday at 4 pm to 6 pm, respectively. All TE classes are on Friday at 4 pm to 6 pm. This fits with the expectations. Adding the constraints separates the events of DM872 and DM865, which is as expected since they are taught by the same teacher. The precedence constraints have also been tested. Finally the soft constraints are added to minimize teacher duties, student overlaps and the use of bad slots. The minimization of bad slots has only been done for events starting in the bad slots. This yields the following table for week 18:

Time	day 0	day 1	day 2	day 3	day 4
(8, 9)	[]	[]	[]	[]	[]
(9, 10)	[]	[DM872-I]	[DM865-I]	[DM873-T]	[DM803-T]
(10, 11)	[DM865-I]	[DM872-I]	[DM865-I]	[DM873-T]	[DM803-T]
(11, 12)	[DM865-I]	[]	[]	[DM872-I]	[DM872-T]
(12, 13)	[DM873-I]	[]	[]	[DM872-I]	[DM872-T]
(13, 14)	[DM873-I]	[]	[]	[]	[]
(14, 15)	[]	[]	[]	[]	[DM876-T]
(15, 16)	[DM803-I]	[DM876-I]	[]	[]	[DM876-T]
(16, 17)	[DM803-I]	[DM876-I]	[]	[]	[]
(17, 18)	[]	[]	[]	[]	[]
(18, 19)	[busy]	[busy]	[busy]	[busy]	[busy]
(19, 20)	[busy]	[busy]	[busy]	[busy]	[busy]

We see that the courses DM872 and DM865 are spread out as much as possible to avoid multiple lectures per day for one teacher. The reported result for this last run is that all student conflicts have been avoided and not event starts in a bad timeslot, but there are 7 lectures for which a teacher has already been teaching that day. In week 17 one teacher has 6 events, so that is an example of one teacher for which the number of teaching duties per day cannot be minimised to one for all days. The statistics of this run for the last iteration, i.e. when all necessary constraints have been added are the following. Before preprocessing there are 10820 rows, 17522 columns, 7581 of which are binary. After there are 10797 rows, 17475 columns, 7555 of which are binary. The GLPK solver has been used and I am not sure if the dual bound from the LP-relaxation is 0, 6.18 or 2.1 because I do not know what LP pertubation is.