The Examination Timetabling Problem

Project Work

Ainur Almukambetova

Vendel Péter Kupás

July 20, 2023

1 Problem

- data:
  - exams: $E = \{e_1, e_2, \dots, e_n\}$
  - students: $S = \{s_1, s_2, \dots, s_m\}$. Each student is enrolled in a non-empty subset of exams.
  - time slots: $Ts = \{t_1 \leq t_2 \leq \dots \leq t_T\}$
  - n_{e_1,e_2} = number of students enrolled both e_1 and e_2
    if $n_{e_1,e_2} > 0$ then e_1 and e_2 are conflicting.
- constraints:
  1. each exam is scheduled exactly once during the examination period
  2. two conflicting exams are not scheduled in the same time-slot
  3. given two exams $e_1, e_2 \in E$ scheduled at distance i of time-slots, with $1 \leq i \leq 5$, the relative penalty is
  $$rp_{e_1,e_2} = 2^{(5-i)} \frac{n_{e_1,e_2}}{|S|}$$
  4. we would like to minimize the sum of the penalties:

$$min \sum_{(e_1,e_2):\ e_1,e_2 \in E,\ e_1 \neq e_2} rp_{e_1,e_2}$$

2 basic Model

2.1 Data initialization

First we extracted the input data from the instances txt file and made the proper data structures:

- **exams**: List for the exams
- **time_slots**: List for the time slots from 1 to T.
- **enrollment_matrix**: a dictionary for the enrollments. The students are the keys and a list of exams belongs to a key.
- **students**: keys of the enrollment_matrix
- **conflicting exams**: A dictionary for the conflicting exams. For every pair of conflicting exams we indicated n_{e_1,e_2}, the number of conflicts. We store only one order of a pair. For example if exam e_1 and exam e_2 are conflicting, then (e_1, e_2) or (e_2, e_1) is a key in the dictionary, but not both of them.

2.2 Modelling constraints

We modeled the exercise as an integer (binary) programming problem.
First we introduced a binary variable $x_{t,e} \in \{0, 1\}$ for the all (t, e) time slot-exam pairs with the following property:

$$x_{t,e} = \begin{cases} 1, & \text{if the } e \in E \text{ exam is held in the } t \in Ts \text{ time-slot} \\ 0, & \text{otherwise} \end{cases}$$

These variables were enough to express the first two constraints and we needed to introduce new variables to express the penalties.

1. Each exam is scheduled exactly once during the examination period.

$$\sum_{k=1}^{T} x_{t_k,e} = 1, \quad \forall e \in E$$

2. Two conflicting exams are not scheduled in the same time-slot.

$$x_{t,e_1} + x_{t,e_2} \leq 1, \quad \forall t \in Ts, \ \forall (e_1, e_2) \in \text{conflicting\_exams}$$

If two exams are conflicting ($n_{e_1,e_2} > 0$), it is not possible that $x_{t,e_1} = x_{t,e_2} = 1$, but every other case (whether they are conflicting or not) satisfy the constraint.

3. To express the penalty function as a linear function we need to introduce new variables. For every conflicting pair of exams let $u_{e1,e2}$ be the penalty if the distance between them is not more then 5 time slot. Consider the next constraints for every conflicting pair $(e1, e2)$, for every $i \in \{1, 2, 3, 4, 5\}$ and for every $t \in \{t_1, \ldots, t_{T-i}$

$$u_{e1,e2} \geq 2^{5-i}(x_{t,e1} + x_{t+i,e2} - 1)\frac{n_{e1,e2}}{|S|}$$
$$u_{e1,e2} \geq 2^{5-i}(x_{t+i,e1} + x_{t,e2} - 1)\frac{n_{e1,e2}}{|S|}.$$

   If exam e_1 is held in time slot t and exam e_2 is held in time slot $t + i$ (or vica versa) then $u_{e1,e2}$ is not smaller than the penalty, otherwise we get a meaningless constraint for $u_{e1,e2}$. Variable $u_{e1,e2}$ was initialized with value 0 by Gorubi, so we got the penalty for the non conflicting pairs of exams.

   In the objective function we would like to minimize the total penalty, so u_{e_1,e_2} is as small as possible in an optimal solution. This implies that there exist an optimal solution where u_{e_1,e_2} is the penalty for the conflicting par of exams if and only if $x_{t,e_1} = x_{t+i,e_2} = 1$.

3 Advanced model

We worked with the same variables $(x_{t,e})$ and to formalize some constraint we needed to introduce new variables again.

3.1 Equity measures

- **Maximum distance** Let u be the number of time-slots between the two closest conflicting exams. Then we would like to maximize the value of u.

$$\max u$$

  So we need to express u somehow with the variables $x_{t,e}$.
  For every (e_1, e_2) conflicting pair and for every $t_1 \leq t_2$ time slots:

$$u \leq (x_{t_1,e_1} + x_{t_2,e_2} - 1)(t_2 - t_1 - T) + T$$

$$u \leq (x_{t_2,e_1} + x_{t_1,e_2} - 1)(t_2 - t_1 - T) + T$$

  Easy to check that with this expression we get that if exam e_1 and exam e_2 are conflicting and exam e_1 is held in time slot t_1 and exam e_2 is held in time slot t_2 (or vica versa) then $u \leq (t_2 - t_1)$, otherwise $u \leq T$ which is meaningless.

- **Number of students with back-to-back exams**

  Let u_s be the next binary variable for every student $s \in S$:

$$u_s = \begin{cases} 1, & \text{if student } s \text{ have two conflicting exams scheduled in consecutive time-slots} \\ 0, & \text{otherwise} \end{cases}$$

  Then we would like to minimize the number of students, where $u_s = 1$, that is

$$\min \sum_{s \in S} u_s$$

  So again for every student we need to express variable u_s with variables $x_{t,e}$. But it is simple. For every $t \in \{t_1, \dots, t_{T-1}\}$, for every (e_1, e_2) conflicting pair, if $s \in e_1$ and $s \in e_2$ then

$$u_s \geq x_{t,e_1} + x_{t+1,e_2} - 1$$
$$u_s \geq x_{t+1,e_1} + x_{t,e_2} - 1$$

3.2 Additional restrictions

- **At most 3 consecutive time slots can have conflicting exams.**

  This means that two consecutive time slots can not have conflicting exams. For every $t \in Ts$, (e_1, e_2) conflicting pair set the next constraint:

$$x_{t,e_1} + x_{t+1,e_2} \leq 1$$
$$x_{t+1,e_1} + x_{t,e_2} \leq 1$$

- **If two consecutive time slots contain conflicting exams, then no conflicting exam can be scheduled in the next 3 time slots**

  Let $u2_t$ and $u3_t$ be the next binary variables:

$$u2_t = \begin{cases} 1, & \text{if there is a conflicting pair in time periods } \{t, t+1\} \subset Ts \\ 0, & \text{otherwise} \end{cases}$$

$$u3_t = \begin{cases} 1, & \text{if there is a conflicting pair in time periods } \{t, t+1, t+2\} \subset Ts \\ 0, & \text{otherwise} \end{cases}$$

  Then the additional constraint is equivalent with this expression:

$$u2_t + u3_{t+2} \leq 1 \quad \forall t \in \{t_1, \dots, t_{T-4}\}$$

  If $u2_t = 1$ then $u3_{t+2}$ need to be equal with 0 and if $u2_t = 0$ then $u3_{t+2}$ can be either 0 or 1.

We need to express variables $u2_t$ and $u3_t$. But we can do this easily. For every $t_1, t_2 \in \{t, t+1\}$ and (e_1, e_2) conflicting pair

$$u2_t \geq x_{t_1,e_1} + x_{t_2,e_2} - 1$$

and for every $t_1, t_2 \in \{t, t+1, t+2\}$ and (e_1, e_2) conflicting pair

$$u3_t \geq x_{t_1,e_1} + x_{t_2,e_2} - 1.$$

- **Include a bonus profit each time no conflicting exams are scheduled for 6 consecutive time slots.**
  Let $u6_t$ be a binary variable with the next constraints:

  $$u6_t \geq x_{t_1,e_1} + x_{t_2,e_2} - 1,$$

  for every $t_1, t_2 \in \{t, t+1, \ldots, t+5\}$ and (e_1, e_2) conflicting pair. Then if there are conflicting exams in the set $\{t, t+1, \ldots, t+5\}$, then $u6_t = 1$, otherwise it can be either 0 or 1.

  We would like to minimize the

  $$\sum_{t:\ t \leq T-6} b \cdot u6_t$$

  expression, so it is the quantity which one we needed. ($b \geq 0$ is the value of the penalty if there are conflicting exams in the set $\{t, t+1, \ldots, t+5\}$)

- **Change the constraints that impose that no conflicting exams can be scheduled in the same time slot. Instead, impose that at most 3 conflicting pairs can be scheduled in the same time slot.**
  For every (e_1, e_2) conflicting pair instead of $(x_{t,e_1} + x_{t,e_2}) \leq 1$ we need to formalise the constraint above. We need to introduce the binary variables u_{t,e_1,e_2}, which is 1 if e_1 and e_2 is held in time slot t and 0 otherwise.

  $$u_{t,e_1,e_2} \geq x_{t,e_1} + x_{t,e_2} - 1$$

  Then the constraint for every $t \in Ts$:

  $$\sum_{(e_1,e_2)\text{ conflicting pair}} u_{t,e_1,e_2} \leq 3.$$

4 Experiments

We implemented all of the variants of the model in python. Unfortunately we couldn't try out the implementation on the instances due to the large size. We have the full licence on our computer offline, but our laptop wasn't enough strong to solve the problem with Gurobi in some minutes. So we wanted to use Colab to deal with this problem, but here the problem size is limited. The number of variables need to be under a limit. So we just applied our code to a small problem which was introduced in the project description. Here we got the proper results and we hope that our implementation is correct. These results can be seen in the colab project.