ORIE 4330 Term project – Spring 2021
Prelim Scheduling

October 18, 2021

1 Introduction

We will schedule be scheduling the Cornell Spring 2021 prelims(that is last years prelims so it is possible to work with realistic data that we had from last year).

Data

- Set of prelim exams, $I = \{1, \dots, M\}$. Each prelim $i \in I$ has
  - A unique exam id, which will be denoted by $i \in I$.
  - A class name, which is the name of the course associated with the prelim.
  - The academic organization(ie CS, ORIE, MATH), that the class belongs to.
  - Enrollment size $s_i \in S \in \mathbb{N}$, which is the number of students that have enrolled in the class.
  - The modality of the prelim, which can be true/false depending on if the prelim will be taken online or not.
  - They have a 1st preferred date, 2nd preferred date and 3rd preferred date. The departments would rather have the exam on those dates, in that order of preference, and it should be expressed in the objective function. Let us denote them as $p_{i,n}$, for $i \in I$ and $n = 1, 2, 3$ and $P_i = \{p_{i,1}, p_{i,2}, p_{i,3}\}$.
- A set of rooms N, For each room $r \in N$, there will be a capacity s_r of the room, which is the numbers of seats available in that room(it is actually a number small than the number of seats, since some seats have to be empty due to Covid restrictions). We also use the function $b(r)$ which will return the number of prelims that can be simultaneously assigned to room r. Initially, since each room can only have 1 exam, $b(r) = 1 \in N$ for all r. We will add an extra 'dummy' room, for scheduling the online exams, which will have $(b(r) = M$, since for the online exams we don't need physical space, and so all the online prelims can be scheduled in the same virtual dummy room. Later in the course, you may bneed to schedule exams in non-dummy rooms that have $b(r) > 1$. Every room is also associated with a building.

- It will also be useful to consider room buckets instead of the individual rooms. Every room bucket will have 2 attributes associated with it, its size and the number of rooms in the bucket. Let the set of all room buckets be B, and for $rb \in B$, denote by s_{rb} the size and by n_{rb} the number of rooms of the buckets. This would mean that in bucket rb, we have n_{rb} rooms of size s_{rb}. The room buckets will be given to you as an input.

- A set of days D, which is the days of the semester during which we will schedule prelims. Every day will have K slots for exams(the different times in each day).

- A coenrollment matrix C with size $I \times I$, whose entries (i, i') indicate the number of students enrolled both in i and i', along with a cutoff value κ to represent what value in the matrix should indicate a conflict.

- R will be the maximum number of rooms that an exam can be split into.

- Distance functions for both the distance between 2 rooms, as well as a room and an academic organization.

Our goal is to assign a date(that is a day/slot combination) to each prelim, as well as 1 or more rooms. That will be done in 2 stages. Stage 1 will take as input all of the above data, and assign a date to each prelim, and will also use the room buckets to ensure that we have enough rooms for that date. Then in stage 2, we take as input the exams of each date, and assign rooms to the exams on that date.

Bellow you can find the variables, constraints and objective function used in each stage.

Stage 1

Variables:

- Binary variables $y(i, d, k)$ to indicate if prelim i is assigned to day d and slot k. Note that we allow d to only take value that fall in P_i, the preferred dates of exam i.

- Integer variables $x(i, rb, d, k)$ to indicate to how many rooms of room bucket rb prelim i is assigned to, on day d and slot k. Sane restrictions as above apply to d, k.

- Integer variables $z(i)$ that indicate the number of rooms assigned to prelim i.

Constraints:

- Constraint to ensure that the variables z correctly count the number of rooms assigned to prelim i:

$$z(i) = \sum_{\substack{rb \in B \\ d \in P_i \\ k \in K}} x(i, rb, d, k), \forall i \in I$$

- Maximum number of rooms constraints:

$$z(i) \leq R, \forall i \in I$$

- Each prelim i is assigned to a unique slot:

$$\sum_{\substack{d \in P_i \\ k \in K}} y(i,d,k) = 1, \forall i \in I$$

- Respect the room bucket size for each date:

$$\sum_{i \in I : d \in P_i} x(i,rb,d,k) \leq n_{rb}, \forall rb \in B, d \in D, k \in K$$

- Each room should only be used by a prelim on the date that it is assigned:

$$x(i,rb,d,k) \leq Ry(i,d,k) \forall i \in I, d \in P_i, k \in K, rb \in B$$

- Each exam must be given enough seats:

$$\sum_{\substack{rb \in B \\ d \in P_i \\ k \in K}} s_{rb} x(i,rb,d,k) \geq s_i, \forall i \in I$$

- Coenrollment conflict constraints: For each pair of prelims i and i' for which $C(i,i') > \kappa$ and $P_i \bigcap P_{i'} \neq \emptyset$, we create a binary variable $c(i,i')$. These variables are meant to indicate if exams that have coenrollment conflicts are scheduled on the same date, and we will use them to penilize the objective function. Let us define $C = \{(i,i') \in I \times I | C(i,i') > \kappa \text{ and } P_i \cap P_{i'} \neq \emptyset\}$. Then we have the following constraint:

$$y(i,d,k) + y(i',d,k) - 1 \leq c(i,i'), \forall i,i' \in C, d \in P_i \cap P_{i'}, k \in K$$

  Note, that since the matrix C is symmetric, we will only create the variables $c(i,i')$ for the entries of the matrix in the upper triangular part.

Objective function:

Before we define the objective function, we will define some weights.

- The weight associated with the number of rooms used wr.
- wd_2 the weight assigned to exams that got their second preferred date.
- wd_3 the weight assigned to exams that got their third preferred date.
- $wc_{i,i'} = C(i,i') - \kappa$, the weight assigned to classes that have coenrollment conflicts.

The objective function has terms in order to minimize the following:

- The total number of rooms used:

$$\sum_{i \in I} wrz(i)$$

- The number of classes that got their second date preference:

$$\sum_{\substack{i \in I \\ k \in K}} wd_2 y(i, p_{i,2}, k)$$

- The number of classes that got their third date preference:

$$\sum_{\substack{i \in I \\ k \in K}} wd_3 y(i, p_{i,3}, k)$$

- The number of coenrollment conflicts:

$$\sum_{i,i' \in C} wc_{i,i'} c(i, i')$$

Notice we don't penalize classes that got their first date preference.

Integer program:

Putting everything together we get the following model for Stage 1:

$$\begin{aligned}
\min \quad & \sum_{i \in I} wr z(i) + \sum_{\substack{i \in I \\ k \in K}} wd_2 y(i, p_{i,2}, k) + \sum_{\substack{i \in I \\ k \in K}} wd_3 y(i, p_{i,3}, k) + \sum_{i,i' \in C} wc_{i,i'} c(i, i') \\
\text{subject to:} \quad & z(i) = \sum_{\substack{rb \in B \\ d \in P_i \\ k \in K}} x(i, rb, d, k), && \forall i \in I \\
& z(i) \leq R, && \forall i \in I \\
& \sum_{\substack{d \in P_i \\ k \in K}} y(i, d, k) = 1, && \forall i \in I \\
& \sum_{i \in I : d \in P_i} x(i, rb, d, k) \leq n_{rb}, && \forall rb \in B, d \in D, k \in K \\
& x(i, rb, d, k) \leq R y(i, d, k) && \forall i \in I, d \in P_i, k \in K, rb \in B \\
& \sum_{\substack{rb \in B \\ d \in P_i \\ k \in K}} s_{rb} x(i, rb, d, k) \geq s_i, && \forall i \in I \\
& y(i, d, k) + y(i', d, k) - 1 \leq c(i, i'), && \forall i, i' \in C, d \in P_i \cap P_{i'}, k \in K \\
& y(i, d, k) \in \{0, 1\}, && \forall i \in I, d \in P_i, k \in K \\
& x(i, rb, d, k) \in \mathbb{N}, && \forall i \in I, rb \in B, d \in P_i, k \in K \\
& z(i) \in \mathbb{N}, && i \in I \\
& c(i, i') \in \{0, 1\} && \forall i, i' \in C
\end{aligned}$$

Stage 2

Note that the model of stage 2 is meant to assign rooms to exams that are in a fixed day/slot, so the indices for day/slot will not be used, as the model is run on each day/slot combinations.

Variables:

- Binary variables $x(i, r)$ to indicate if prelim i is assigned to room r (notice we now use actual rooms not room buckets).
- Integer variables $z(i)$ that indicate the number of rooms assigned to prelim i.
- Binary variables $p(r, r')$ to indicate if rooms r, r' are assigned to the same prelim.

Constraints:

- Constraint to ensure that the variables z correctly count the number of rooms assigned to prelim i:
$$z(i) = \sum_{r \in N} x(i, r), \forall i \in I$$
- Maximum number of rooms constraints:
$$z(i) \leq R, \forall i \in I$$
- Each room is used at most once for the current day/slot combination:
$$\sum_{i \in I} x(i, r) \leq 1, \forall r \in N$$
- A constraint to correctly determine the value of $p(r, r')$:
$$p(r, r') \geq x(i, r) + x(i, r') - 1, \forall i \in I, r \in N, r' \in N$$
- Each exam must be given enough seats:
$$\sum_{r \in N} s_r x(i, r) \geq s_i, \forall i \in I$$

Objective function:

Our objective consists of the following terms:

- The total number of rooms used, weighted by w_r:
$$\sum_{i \in I} w_r z(i)$$

- The total sum of distances of the rooms assigned to a prelim to the academic organization of the class, weighted by w_{ac}:

$$\sum_{i \in I, r \in N} w_{ac} d(r, \text{acadorg}_i) x(i, r)$$

- The sum of squared distances of rooms that are assigned to the same prelim:

$$\sum_{r, r' \in N} d(r, r')^2 p(r, r')$$

Integer program:

Putting everything together we get the following model:

$$\begin{aligned}
\min \quad & \sum_{i \in I} wr z(i) + \sum_{i \in I, r \in N} w_{ac} d(r, \text{acadorg}_i) x(i, r) + \sum_{r, r' \in N} d(r, r')^2 p(r, r') & \\
\text{subject to:} \quad & z(i) = \sum_{r \in N} x(i, r), & \forall i \in I \\
& z(i) \leq R, & \forall i \in I \\
& \sum_{i \in I} x(i, r) \leq 1, & \forall r \in N \\
& p(r, r') \geq x(i, r) + x(i, r') - 1, & \forall i \in I, r \in N, r' \in N \\
& \sum_{r \in N} s_r x(i, r) \geq s_i, & \forall i \in I \\
& x(i, r) \in \{0, 1\}, & \forall i \in I, r \in N \\
& z(i) \in \mathbb{N}, & \forall i \in I \\
& p(r, r') \in \{0, 1\} & \forall r, r' \in N
\end{aligned}$$