Introduction

Course schedules are very important for a student's college experience. The right course schedule ensures that a student is set to graduate on time and that the workload for that semester is manageable. Creating a class schedule for students involves ensuring that they are enrolled in their required courses, that times don't conflict, and that they meet credit requirements. In addition, factors such as class difficulty and professor difficulty also play into creating a schedule.

In this project we show how class schedules can be created using an integer linear programming model. Our model will output feasible schedules that meet requirements as well as an optimal schedule that meets the requirements of students while also trying to avoid undesirable morning classes, difficult courses, and professors that are very difficult.

Our working model uses data from the IIT Course Status Report, as well as input from the user regarding what classes they must take, classes they desire to take, and how much they care about undesirable time slots to create an optimal schedule for the user.

Describe the Problem

Data

This is the data that we will use in our model:

- IIT Course Status Report
    - Provides final class schedules for each semester.
    - We use the following columns from the website: Course Registration Number, Course Code, Course Subject, Course Number, Course Title, Credits, Days, Time, Instructors
- Course Quality: Randomly generated integer in range 1-5 where 5 is a course of highest quality
- Professor Difficulty: Randomly generated integer in range 1-5 where 5 is a very difficult professor
- Note: The Course Quality and Professor Difficulty columns of our data was inspired by Rate My Professors. To further extend this project, one could webscrape the website to get the data. Another idea is to use course evaluation data, if it becomes available.

Assumptions

Next, let us state assumptions that are used in our model:

- The user only inputs courses that are in the IIT Course Status Report (we cannot create a requirement for a course that doesn't exist).

- The IIT Course Status Report Data will not change. The courses sections and corresponding professor and time-slot remain constant.

Approach

Variables

For each course in the IIT Course Status Report, there are one or more course sections. For example, a course can be CS100 which has 3 sections. If a student is required to take this course, then we set a constraint that they have to register for one of these sections.

If a student desires to take the course, then we have a constraint that they can be registered for at most one of the course's sections.

Intuitively, it can be observed that our decision variables are all of these course sections.

Constraints

The constraints for required and desired courses were discussed, but we have other constraints as well.

In our program, the user will input the minimum and maximum amount of credits they want to take for the semester. We allowed this to be inputted instead of setting it to the standard 12 credits minimum, 18 credits maximum as it may be the case that students have to take fewer than 12 credits minimum, or they want to take more than that amount of credits in a semester; same goes for the maximum amount of credits.

We set a constraint that for the course sections being selected, the sum of the number of credits has to be greater than or equal to the minimum number of credits and less than or equal to the maximum number of credits.

We also have time slot constraints to ensure that the courses do not conflict with each other. For each possible time slot (e.g. 3:15 - 4:30 PM, 1:50 - 3:05 PM), a student can be enrolled in at most one course section that has that time slot.

Cost Function

In addition to asking the user what their required and desired courses are, we also ask them the following two questions:

1. Hypothetically, there's a 1/5 quality professor at a non-8:35 AM time slot. How good does the professor have to be (1-5) to justify moving to 8:35 AM?
2. For every 1 point increase in difficulty, how many points better should a professor be to justify it?

The answer to the first question becomes the cost of a decrease in course quality, and the answer to the second question multiplied by the first becomes the penalty for an early class time slot. This can be done as these answers become ratios to the penalty for professor quality.

Let **a** be the answer to the first question, and **b** be the answer to the second question. we want to minimize sum([b * (5-course quality) + professor difficulty+ a*(b-1)]*course section variable), where course section variable is a binary variable that is either 0 or 1. 0 means it is not included in the schedule we will output and 1 means it will be included in that schedule.

Course section variable can equivalently be an integer variable with bounds 0 and 1.