DISCRETE OPTIMIZATION AND DECISION MAKING - FINAL PROJECT

EXAMINATION TIMETABLING PROBLEM

Academic Year 2022/2023

Nicole Olivetto - VR481171

1) THE EXAMINATION TIMETABLING PROBLEM

Let us consider a set E of exams, to be scheduled during an examination period at the end of the semester, and a set S of students. Each student is enrolled in a non-empty subset of exams. The examination period is divided into T ordered time-slots.
Given two exams $e1,e2 \in E$, let $ne1,e2$ be the number of students enrolled in both. Two exams $e1,e2 \in E$ are called conflicting if they have at least one student enrolled in both, i.e., if $ne1,e2 > 0$.
Rules and regulations impose that conflicting exams cannot take place in the same time-slot. Moreover, to promote the creation of timetables more sustainable for the students, a penalty is assigned for each pair of conflicting exams scheduled up to a distance of 5 time-slots. More precisely, given two exams $e1,e2 \in E$ scheduled at distance i of time-slots, with $1 \leq i \leq 5$, the relative penalty is $2^{(5-i)} \cdot ne1,e2/|S|$.
|S| The ETP aims at assigning exams to time-slots ensuring that:
each exam is scheduled exactly once during the examination period; two conflicting exams are not scheduled in the same time-slot;
the total penalty resulting from the created timetable is minimized.

We're assuming that in each time-slot there are enough resources (number of rooms, capacities) to accommodate all the exams and all the enrolled students

2) MAIN ELEMENTS

We consider the following elements:

- **Set of Exams (E):** A set of exams to be scheduled during an examination period.
- **Set of Students (S):** A set of students, where each student is enrolled in one or more exams.
- **Enrollments:** We maintain a dictionary to store the enrollments of each student, mapping students to the exams they are enrolled in.
- **Common Students:** We create a dictionary to track the number of common students between pairs of exams, which helps identify conflicting exams.
- **Number of Time Slots (T):** The examination period is divided into a set of ordered time slots.

3) DECISION VARIABLES

Decision variables are a fundamental component of any Integer Linear Programming (ILP) formulation, and they play a crucial role in representing the decisions or choices that need to be made to solve a particular optimization problem. In the context of the Exam Timetabling Problem (ETP), decision variables are used to represent and control the scheduling of exams within specific time slots.

In this case I used two sets of decision variables. The first set of decision variables, denoted as $x_{e,t}$, represents the *scheduling of exams*. These binary variables indicate whether each exam e is scheduled at a specific time slot t. If $x_{e,t} = 1$, it indicates that exam e is scheduled at time slot t, while $x_{e,t} = 0$ means that it is not scheduled at that time slot T.

The second set of decision variables, $conflict_{e_1,e_2,t}$, represents *conflicts between pairs of exams* e_1 and e_2 scheduled at the same time slot t.

If $conflict_{e_1,e_2,t} = 1$, it indicates that exams e_1 and e_2 are conflicting at time slot t, while $conflict_{e_1,e_2,t} = 0$ implies that there is no conflict between them at that time slot.

4) CONSTRAINTS

1. Exam Scheduling Constraint: Ensures that each exam is scheduled exactly once during the examination period. It sums up the binary variables $x_{e,t}$ over all time slots t and sets the sum equal to 1.

$$\forall e \in E : \quad \sum_{t=1}^{T} x_{et} = 1$$

- e represents each exam in the set of exams E.
- t represents each time slot from 1 to T, where T is the total number of time slots available for scheduling exams.
- x_{et} is the binary decision variable that indicates whether exam e is scheduled in time slot t.

So, the constraint ensures that for each exam e, there is exactly one time slot t where that exam is scheduled.

2. Conflicting Exams Constraint: Conflicting exams cannot be scheduled in the same time slot. For each pair of conflicting exams (e_1, e_2), if they share common students, they cannot be scheduled in the same time slot. For each pair of exams $e1$ and $e2$ with common students, and for every time slot t, this constraint ensures that at most one of the exams in the pair is scheduled at time slot t.

$$x_{e_1,t} + x_{e_2,t} \leq 1 \quad \text{for all } e1, e2 \in E \text{ and } t \in T$$

Specifically:
- $x_{e_1,t}$ and $x_{e_2,t}$ are binary decision variables indicating whether exam e_1 and exam e_2 are scheduled in time slot t, respectively.
- The inequality $x_{e_1,t} + x_{e_2,t} \leq 1$ ensures that at most one of the exams e_1 and e_2 can be scheduled in time slot t.

4) OBJECTIVE FUNCTION

The objective function implemented in this optimization model aims to minimize the overall penalty incurred due to scheduling conflicts between exams. This penalty is calculated based on the presence of common students enrolled in pairs of exams and the distance between the time slots assigned to these exams. The formula used to compute the penalty is $2^{(5-i)} \cdot \frac{n_{e_1,e_2}}{|S|}$, where i represents the absolute difference between the time slots assigned to the exams, n_{e_1,e_2} denotes the number of common students enrolled in exams e_1 and e_2, and $|S|$ indicates the total number of students.

$$\sum_{e_1 \in E} \sum_{e_2 \in E} \left(2^{(5-|t_1-t_2|)} \cdot \frac{n_{e_1,e_2}}{|S|} \cdot x_{e_1 t_1} \cdot x_{e_2 t_2} \right)$$

Specifically:

- $\sum_{e_1 \in E} \sum_{e_2 \in E}$ iterates over all pairs of exams e_1 and e_2 in the set E (the set of all exams).
- $2^{(5-|t_1-t_2|)}$ calculates the penalty based on the absolute difference between the time slots assigned to exams t_1 and t_2. The penalty decreases exponentially as the difference between the time slots increases, with a base of 2 raised to the power of $5 - |t_1 - t_2|$.
- $\frac{n_{e_1,e_2}}{|S|}$ computes the fraction of common students between exams e_1 and e_2 over the total number of students $|S|$. This fraction represents the proportion of students affected by the scheduling conflict between the two exams.
- $x_{e_1 t_1} \cdot x_{e_2 t_2}$ represent whether exams e_1 and e_2 are scheduled at time slots t_1 and t_2, respectively. Multiplying them ensures that the penalty is only considered when both exams are scheduled at their respective time slots.

5) EQUITY MEASURE

The measure "Number of students with back-to-back exams" can be very useful in evaluating the fairness and equity of an exam timetable. It quantifies the number of students who are scheduled to sit for two conflicting exams consecutively, with no break in between. A lower count in this measure indicates a more equitable timetable. Minimizing this metric enhances the overall well-being and academic experience of students, allowing them adequate time for preparation and recovery between exams. Therefore, prioritizing a timetable with a reduced number of students facing back-to-back exams aligns with the goal of promoting a supportive and balanced learning environment.

To calculate the number of students with back-to-back exams, we need to iterate over each student and check if they have conflicting exams scheduled in consecutive time-slots. The formula can be expressed as:

$$\text{Number of students with back-to-back exams} = \sum_{\text{student}} \sum_{t=1}^{T-1} \text{conflict}(t, t+1)$$

Where:

- student iterates over all students.

- T is the total number of time-slots.

- conflict$(t, t+1)$ is a binary variable that equals 1 if there is at least one pair of conflicting exams scheduled in time-slots t and $t+1$ for the current student, and 0 otherwise.

Instead of introducing additional constraints to handle back-to-back exams, I tried to use a different approach. The objective function is modified to directly address the issue of consecutive exams for students. It constructs a summation expression that evaluates the occurrence of back-to-back exams for students across all student enrollments and exam pairs. Here's a more detailed explaination:

- Iteration over Student Enrollments: the iteration `for student, exams in student_enrollments.items()` loops through each student in the `student_enrollments` dictionary along with the list of exams they are enrolled in.
- Exam Pair Selection: inside this iteration, another nested iteration `for exam1 in exams for exam2 in exams` is performed. This nested loop selects each pair of exams from the list of exams for the current student. The condition `(exam1 != exam2)` ensures that the selected pair of exams are different.
- Back-to-Back Exam Condition: the if condition `if (exam1, exam2) in common_students_dict` checks if the pair of exams share common students. This condition ensures that we only consider pairs of exams that have common students.
- Time Slot Limitation: the condition `(t < n_timeslots - 1)` ensures that the time slot index `t` does not exceed the total number of time slots minus one.
- Binary Variable Selection: within the summation expression, it selects the binary variables `x[exam1, t]` and `x[exam2, t + 1]` that represent whether exams `exam1` and `exam2` are scheduled in consecutive time slots, respectively.

6)ADDITIONAL RESTRICTIONS

The additional restriction that I tried to add to the base model are:

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

- Include a bonus profit each time no conflicting exams are scheduled for 6 consec- utive time slots.
- 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.

To implement them in the base model I added some new constraints.

To make sure that at most 3 consecutive time slots can have conflicting exams I added the following constraint:

$$\sum_{t'=t}^{t+2} \text{conflict}[e1,e2,t'] \leq 3$$

This formula represents the sum of conflicts between exams $e1$ and $e2$ over a range of three consecutive time slots, from t to $t+2$, ensuring that this sum does not exceed 3.

To add the second additional restriction, I implemented a code that iterates through each pair of exams (e1, e2) and each time slot (t), checking if conflicts occur in consecutive time slots (t and t+1). If conflicts occur in consecutive time slots, we add constraints to ensure that no conflicts involving any pair of exams can occur in the subsequent three time slots (t+2, t+3, and t+4). In other words, let conflict$[e1,e2,t]$ be a binary decision variable representing whether exams $e1$ and $e2$ conflict at time slot t. The constraint can be represented as:

If conflict$[e1,e2,t]$ + conflict$[e1,e2,t+1] = 2$ for any $e1,e2,t$, then:

$$\sum_{\substack{t'=t+2 \\ e1' \neq e2'}}^{t+4} \text{conflict}[e1', e2', t'] = 0$$

The third additional restriction says to include a bonus profit each time no conflicting exams are scheduled for 6 consecutive time slots. This bonus profit aims to incentivize the creation of timetables that offer students extended periods without exam conflicts, enhancing their overall experience and reducing stress.
I tried to implement a code that iterates through each pair of exams $e1$ and $e2$ in the exam set (E), ensuring they are different exams and checking if they are in the common students

dictionary. It then evaluates whether there are no conflicts between these exams in any six consecutive time slots. If this condition is met, a bonus profit of 50 is added to the total. This process is repeated for all pairs of exams, accumulating the bonus profit for each instance where the condition is satisfied.

The last additional restriction limits the number of conflicting pairs in the same time slot to at most 3 and can be expressed as:

$$\sum_{e_1 \in E} \sum_{e_2 \in E} \text{conflict}(e_1, e_2, t) \leq 3 \quad \text{for all } t \in T$$

- e_1 and e_2 represent two distinct exams from the set of all exams E.
- conflict(e_1, e_2, t) is a binary variable that indicates whether exams e_1 and e_2 are scheduled in the same time slot t. It takes the value of 1 if there is a conflict and 0 otherwise.
- the summation $\sum_{e_1 \in E} \sum_{e_2 \in E}$ iterates over all pairs of distinct exams.

I tried to add this in my code but it was giving me some errors so this part of the code is a comment.

7)RESULTS

Base model

In this section, the outcomes from applying the basic model, with a solver time limit set to 1000 seconds, are presented alongside the benchmarks (best available solutions for each instance). Notably, the model performs exceptionally well with the test instance, as evidenced by the objective function value of 3.375 observed in both columns.
Regarding the other instances I got feasible solutions within the 1000 seconds for all of them, but some of the results are pretty far from the benchmark.

	Results	Benchmark
Test	3,375	3,375

	Results	Benchmark
Instance01	184,959	157,033
Instance02	54,330	34,709
Instance03	63,064	32,627
Instance04	15,861	7,717
Instance05	37,365	12,901
Instance06	6,659	3,045
Instance07	18,763	10,050
Instance08	39,876	24,769
Instance09	20,811	9,818
Instance10	8,355	3,707
Instance11	10,429	4,395

Back-to-back Minimization

The first column indicates the number of students who experienced BTB exams, while the second column represents the corresponding penalty of this new exam schedule. The objective this time wasn't to minimize the penalty, so that's why it's higher than in the previous chart.

	Students with btb exams	Penalty
Test	0	3,375
Instance01	497	179,991
Instance02	227	52,414
Instance03	365	73,047
Instance04	435	13,172
Instance05	770	34,991
Instance06	1925	7,535
Instance07	185	18,969
Instance08	823	39,829
Instance09	608	24,732
Instance10	1538	10,094
Instance11	1396	12,149

Upon analyzing the data, we find that different instances exhibit varying degrees of impact. For instance, in "Instance01," a high penalty of 179,991 suggests a significant prevalence of back-to-back (BTB) exams among students. Conversely, "Instance06" presents an interesting scenario with a large number of students experiencing BTB exams (1925), yet a lower penalty (7,535). This could indicate a more balanced distribution of BTB exams among students in this instance or fewer instances of consecutive BTB exams per student. In essence, while both instances witness high BTB occurrences, the penalty reflects how evenly or unevenly these exams are distributed among students.