Report of The Examination Timetabling Problem

Mehrdad Ghadiri

February 2024

1 Problem Description and Formulation

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 $e_1, e_2 \in E$, let n_{e_1,e_2} be the number of students enrolled in both. Two exams $e_1, e_2 \in E$ are called conflicting if they have at least one student enrolled in both, i.e., if $n_{e_1,e_2} > 0$.

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 $e_1, e_2 \in E$ scheduled at a distance i of time-slots, with $1 \leq i \leq 5$, the relative penalty is

$$\frac{2^{(5-i)} \cdot n_{e_1,e_2}}{|S|}$$

The ETP aims at assigning exams to time-slots ensuring that:

1. Each exam is scheduled exactly once during the examination period.
2. Two conflicting exams are not scheduled in the same time-slot.
3. The total penalty resulting from the created timetable is minimized.

Lists and Matrices

- **ExamsList:** $[0, 1, 2, \ldots, E]$.

  ExamsList represents a list of exams involved in the optimization problem. Each element $i \in ExamsList$ corresponds to a unique exam. The entire list defines the list of exams under consideration in the optimization problem.

- **TimeList:** $[0, 1, 2, \ldots, T]$.

  TimeList represents the available time slots for scheduling exams during the examination period. Each element in this list corresponds to a unique time slot, and the entire list defines the set of time slots that can be utilized for scheduling exams.

- **Distance:** $[1, 2, 3, 4, 5]$.

  The list of distances plays a role in the formulation of the objective function and expresses the distance between time slots where conflicting exams are scheduled.

- **UC_Matrix:** The Unweighted Conflict Matrix, denoted is a binary matrix of dimensions $E \times E$, where E is the coming from the ExamsList. Elements of the UC_Matrix have binary value, where 0 indicates no conflicting exam between exam i and exam j, and 1 represents the existence of a conflicting exam.

- **WC_Matrix:** The Weighted Conflict Matrix, denoted as is a matrix of dimensions $E \times E$, where E is the coming from the ExamsList. A value of 0 indicates no conflicting exam, while other values show the existence of conflicting exams and the corresponding number of enrolled students.

Indices

- **Exam ID i:** $0 \leq i \leq E$, $i \in N_0$
- **Exam ID j:** $0 \leq j \leq E$, $j \in N_0$

  In the context of exam scheduling, the exam IDs (i and j) are identifiers of the exams in ExamsList.

- **Time ID k:** $0 \leq k \leq T$, $k \in N_0$

  In the context of Exam scheduling, Each exam is placed in one TimeSlot of the TimeList. The Identifier of a TimeSlot is k.

- **Distance ID d:** $1 \leq d \leq 5$, $d \in N$

  Distance ID denoted by d is used for calculating the penalty. It represents the time slot difference between the two conflicting Exams.

Variables Definition

- **Decision Variable**

  $$X_{i,k} \in \{0, 1\}$$

  The binary decision variable Xi, k is equal to 1 when the exam i placed in time slot k. Otherwise, the value is 0.

- **Auxiliary Variable**

$$Y_{i,j,d} \in \{0, 1\}$$

The binary Auxiliary variable Yi, j, k is equal to 1 when the Conflicting Exams i and j are scheduled in d Time slots differences with Each other. Otherwise, the value is 0.

Constraints Definition

- **Constraint 1:**

$$\sum_{i=0}^{E}\sum_{j=0}^{E}\sum_{k=0}^{T} UC\_Matrix(i, j) \cdot (X_{i,k} + X_{j,k}) \leq 1$$

This constraint ensures that the exams which are in conflict, represented by the pair (i,j), are not scheduled in the same time slot k.

- **Constraint 2:**

$$\sum_{k=0}^{T} X_{i,k} = 1 \quad \forall i \in ExamsList, \forall k \in TimeList$$

This constraint ensures that each exam in the Exams' List is scheduled in the time slots exactly once.

- **Constraint 3:**

$$\sum_{i=0}^{E}\sum_{j=0}^{E}\sum_{k=0}^{T}\sum_{d=1}^{5} (X_{i,k} + X_{j,k+d}) \leq Y_{i,j,d} + 1 \quad \forall i, j, k, d$$

where

$$(k + d) \in TimeList \quad and \quad UC\_Matrix[i, j] \neq 0,$$

This Constraint relates the Decision Variables $X_{i,k}$, $X_{j,k+d}$ and Auxiliary Variable $Y_{i,j,d}$ to each other and imposing a condition that ensures any conflicting exams are scheduled with a time gap of d between them.

Objective Function

$$Minimize \sum_{i=0}^{E}\sum_{j=0}^{E}\sum_{d=1}^{5} Y_{i,j,d} \cdot WC\_Matrix(i, j) \cdot \frac{2^{(5-d)}}{|S|}$$

where $|S|$ shows the Number of Students. The objective is to minimize the total penalty associated with conflicting exams.

2 General Overview of the Code

- **Input the Data**

  There are 11 Instances and Each of them have been Evaluated Separately. Each instance has 3 different files showing the

  1- The number of enrolled students per exam (Sheet1)

  2- The length of the examination period (Sheet2)

  3- The exams in which each student is enrolled (Sheet3)

- **Modifying the Exams and Time IDs**

  The Exams and ExamNumber columns in Sheet 1 and 3 of the Excel file correspond to the IDs of exams, starting from 1 for each instance. However, due to Python's indexing starting from 0, I have adjusted all Exam ID values by shifting them back by one. This means that the Exam IDs now start from 0 instead of 1, aiming to simplify the code and align with Python's indexing convention.

  Also for the reasons mentioned before, In Sheet 2 of the Excel file, the data represents the number of time slots, I have assigned IDs to each time slot, starting from zero.

- **Creation of List and Matrices**

  Following the adjustment of Exam and Time Slot IDs, lists for Exams, Time Slots, and Distances have been created. Additionally, two matrices have been generated for analysis: the Binary Unweighted Conflicted Exams Matrix, indicating the presence or absence of conflicts between exams, and the Weighted Conflicted Exams Matrix, showing the number of students attending each pair of conflicted exams.

- **Model and Environmet**

  After creation of the Necessary Lists and Matrices, The environment for the model was then set up:

  1) `env.setParam("Threads", 1)`
  This Parameter controls the number of threads for parallel algorithms.

  2)`env.setParam("Presolve", 1)`
  This Parameter controls the presolve level (conservative: 1).

  3)`env.setParam("MIPGap", 1e-4)`
  This Parameter sets the MIP gap tolerance.

  4)`env.setParam('Method', 0)`
  This is an Algorithm for the initial root relaxation of a MIP model (0=primal simplex).

  5)`env.setParam("TimeLimit", 1200)`
  This Parameter sets a 20-minute time limit.

6)`env.setParam("PreSparsify", 1)`
This Parameter reduces memory usage.

- **Optimization and Output**

  Once the Model is created the Variables, Constraints and Objective Function are added to the model and We run it. The results will be saved into 2 different text Files containing the Log File and Solution of the Optimization Problem. The log File records the messages and information during the Optimization process. On the other hand, the Solution file Contain the Final result of the Optimization process representing whether the Solution is Feasible, The value of the Objective Function and Exams Time Table.

3 Results

Table 1 presents optimization results, where Total Penalty 1 and Total Penalty 2 correspond to Time 1 and Time 2, respectively. Also, The Benchmark column allows us to evaluate the optimization results.

The blank spaces (-) in Table 1, under the Total Penalty 1 and 2 columns, indicate that a feasible solution was not achieved by the corresponding Time 1 and Time 2 in the Optimization Process.

Instance	**Time1**	**Total Penalty1**	**Time2**	**Total Penalty2**	**Benchmark**
Test	1 (s)	3.375	1 (s)	3.375	3.375
Instance 1	600	178.070	1200	160.792	157.033
Instance 2	1200	-	2400	52.104	34.709
Instance 3	1200	-	2400	54.565	32.627
Instance 4	1200	-	2400	12.462	7.717
Instance 5	1200	-	2400	29.726	12.901
Instance 6	3600	-	7200	-	3.045
Instance 7	600	18.899	1200	18.899	10.050
Instance 8	600	37.50	1200	37.50	24.769
Instance 9	1200	-	2400	19.720	9.818
Instance 10	4200	-	8400	-	3.707
Instance 11	4800	-	9600	-	4.395

Table 1: Optimization Results of Different Instances