Course project:
The Examination Timetabling Problem

R. Rizzi, R. Zanotti

Discrete Optimization and Decision Making - 2022/2023

Timetabling problems

- 4 main dimensions:
  - a finite set of time-slots
  - a finite set of resources
  - a finite set of meetings
  - a finite set of constraints
- The aim is to **assign time-slots and resources to meetings** to **satisfy the constraints as much as possible**
- Arise in every organizations of people, machines, activities and in the most various application domains:
  - transportation/logistics
  - machine/job scheduling
  - educational/academic institutions
  - personnel shifts

Educational timetabling

- important and time-consuming tasks which occur periodically (i.e. annually, quarterly) in all academic institutions:
    - school timetabling
    - course timetabling
    - examination timetabling
- the complexity can be exacerbated by several factors:
    - dimensions of the desired timetable
    - very limited resources (time-slots, number and capacity of the rooms, ...)
    - general or institution-specific requirements
        - *hard* constraints: have to be satisfied in each timetable
        - *soft* constraints: not compulsory but preferable
    - general or institution-specific goals

Examination Timetabling Problem (ETP)

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 $e_1, e_2 \in E$, 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 distance i of time-slots, with $1 \leq i \leq 5$, the relative penalty is $2^{(5-i)} \cdot \frac{n_{e_1,e_2}}{|S|}$.

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.

Assumption: in each time-slot there are enough resources (number of rooms, capacities) to accommodate all the exams and all the enrolled students

A clarifying example

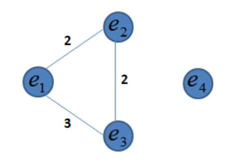
Input data

Exams: $\{e_1, \ldots, e_4\}$
Students: $\{s_1, \ldots, s_8\}$
Available time-slots: $\{t_1, \ldots, t_6\}$

Enrollments:

s_1: e_1 e_2 e_3
s_2: e_1 e_3
s_3: e_4
s_4: e_3
s_5: e_1 e_3
s_6: e_4
s_7: e_2 e_3
s_8: e_1 e_2

Conflict graph:



Sample solutions

An infeasible solution:

t_1	t_2	t_3	t_4	t_5	t_6
e_1 e_2		e_3		e_4	

A feasible solution:

t_1	t_2	t_3	t_4	t_5	t_6
e_1	e_2	e_3	e_4		

with obj $= (2 * 2^{5-1} + 3 * 2^{5-2} + 2 * 2^{5-1})/8 =$
$= (2 * 16 + 3 * 8 + 2 * 16)/8 = (32 + 24 + 32)/8 = \mathbf{11}$

An optimal solution:

t_1	t_2	t_3	t_4	t_5	t_6
e_1 e_4		e_2			e_3

with obj $= (2 * 2^{5-2} + 3 * 2^{5-5} + 2 * 2^{5-3})/8 =$
$= (2 * 8 + 3 * 1 + 2 * 4)/8 = (16 + 3 + 8)/8 = \mathbf{3.375}$

Advanced model 1/2

Equity Measures

In addition to the traditional objective of minimizing the total penalty, there are a number of other equity measures that can be considered when creating a timetable. Some of these measures include:

- **Maximum distance**: This measure is the maximum number of time-slots between any two conflicting exams. A timetable with a high maximum distance is more equitable for students, as it means that they will have more time between exams.
- **Number of students with back-to-back exams**: This measure is the number of students who have two conflicting exams scheduled in consecutive time-slots. A timetable with a low number of students with back-to-back exams is more equitable for students, as it means that they will have more time to rest between exams.

You may think of your own equity measure and formulate it. If you choose this route, justify the measure in the project report and presentation and provide an analysis of the impact.

Advanced model 2/2

Additional restrictions

- 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 consecutive 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.

Benchmark instances for the base problem: properties

11 instances for assessing your results with respect to established benchmarks

Name	$\vert E\vert$	$\vert S\vert$	enr	$\vert T\vert$	$density$	$benchmark$
instance01	139	611	5751	13	0.14	**157.033**
instance02	181	941	6034	21	0.29	**34.709**
instance03	190	1125	8109	24	0.27	**32.627**
instance04	261	4360	14901	23	0.18	**7.717**
instance05	461	5349	25113	20	0.06	**12.901**
instance06	622	21266	58979	35	0.13	**3.045**
instance07	81	2823	10632	18	0.42	**10.050**
instance08	184	2750	11793	10	0.08	**24.769**
instance09	381	2726	10918	18	0.06	**9.818**
instance10	543	18419	55522	32	0.14	**3.707**
instance11	682	16925	56877	35	0.13	**4.395**

where

- enr: total number of enrolments
- $density$: ratio between the number of pairs of conflicting exams and the total number of exams pairs (arc density of the conflicting graph)
- $benchmark$: obj value (total penalty) of the best solution available, not necessarily optimal [NB: rounded to the third decimal digit]

Benchmark instances: format

Each instance *instanceXX* is defined by 3 plain text files, with the same name:

- *instanceXX.exm*: defines the total number of students enrolled per exam.
  Format: 1 line per exam. Each line has the format

  `INT1 INT2`

  where `INT1` is the exam ID and `INT2` is the number of enrolled students in `INT1`.
- *instanceXX.slo*: defines the length of the examination period.
  Format: a single value in the format

  `INT`

  where `INT` is the number of available time-slots t_{max}. Hence, time-slots will have numeric IDs $\{1, 2, \ldots, t_{max}\}$.
- *instanceXX.stu*: defines the exams in which each student is enrolled.
  Format: 1 line for each enrollment. Each line has the format

  `sINT1 INT2`

  where `INT1` is the student ID and `INT2` is the ID of the exam in which student `INT1` is enrolled.

Project: tasks and deadlines

Required tasks:

1. Provide an Integer Linear Programming formulation for the basic ETP.
2. Implement the formulation in Python, using Gurobi.
3. Solve the benchmark instances through the implemented formulation with a short time limit (e.g., 10 minutes) and analyze the solutions.
4. Incorporate in your project the following requests:
    - Formulate and implement one or more of the equity measures and analyze their impact (compared to the standard obj function).
    - Formulate and include in the original model the additional restrictions.
5. Complete and deliver the following:
    - Project code
    - Results
    - Report summarizing the work
6. Prepare a presentation of the project to be discussed with all group members.

Deadlines:

- Material delivery deadline: At least one week before the agreed date for the project discussion.