Course assignment:
Examination Timetabling

R. Tadei, D. Manerba
Dept. of Control and Computer Engineering
Politecnico of Torino, Italy

01OUV - Optimization Methods and Algorithms - 2017/2018

Timetabling

- A *timetabling problem* is a setting with four parameters:
  - 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 the meetings so as to satisfy the constraints as much as possible.
- Timetabling problems arise in every organizations of people, machines, activities and in the most various applicative domains:
  - transportation/logistics
  - machine/job scheduling
  - educational/academic institutions
  - personnel shifts

Timetabling in education

- important and time-consuming tasks which occur periodically (i.e. annually, quarterly) in all academic institutions:
  - school timetabling
  - course timetabling
  - examination timetabling
- the complexity of such tasks may be exacerbated by several factors:
  - dimensions of the desired timetable
  - very limited resources (time, 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

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 subset of exams, at least one. The examination period is divided in t_{max} ordered time-slots.

By law, *conflicting* exams (i.e. having enrolled students in common) cannot take place during the same time-slot. Moreover, to incentive universities at creating timetables more sustainable for the students, the *Ministry of Education* assigns a penalty for each couple of conflicting exams scheduled up to 5 time-slots apart. More precisely, given the number $n_{e,e'}$ of students enrolled in both conflicting exams e and e', which are scheduled i time-slots apart, the penalty is calculated as $2^{(5-i)} n_{e,e'}/|S|$.

The Examination Timetabling Problem (ETP) aims at assigning each exam to a specific time-slot ensuring that:

- each exam is scheduled once during the period,
- two conflicting exams are not scheduled in the same time-slot.

The objective is to minimize the total penalty resulting from the created timetable.

Assumptions:

- during each time-slot there is always a number of available rooms greater than the total number of exams;
- rooms have enough capacity with respect to the number of enrolled students.

A clarifying example

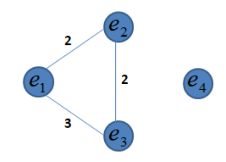
Input data

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

A feasible (optimal) solution:

t_1	t_2	t_3	t_4	t_5	t_6
e_1 e_4		e_2			e_3

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

Benchmark instances: properties

⟶ **7 public instances** will be shared in order to let you test and assess the results of your algorithms with respect to the benchmarks.
⟶ **X private instances** will be used to evaluate how the same algorithms are robust when solving instances unknown during the development.

	Visibility	$\lvert E\rvert$	$\lvert S\rvert$	enr	t_{max}	$density$	$benchmark$
instance01	Public	139	611	5751	13	0.14	**157.033**
instance02	Public	181	941	6034	21	0.29	**34.709**
instance03	Public	190	1125	8109	24	0.27	**32.627**
instance04	Public	261	4360	14901	23	0.18	**7.717**
instance05	Public	461	5349	25113	20	0.06	**12.901**
instance06	Public	622	21266	58979	35	0.13	**3.045**
instance07	Public	81	2823	10632	18	0.42	**10.050**
instance08 ... instanceXX	Private	-	-	-	-	-	-

- enr: total 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 in the literature, not necessarily optimal

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.

Assignment: tasks and deadlines organization

Required tasks:

- provide a Linear Programming formulation for the ETP (deadline 1)
- propose a solution approach for the ETP by exploiting one (or more) heuristic and meta-heuristic algorithms presented during the course
- develop/implement such a solution algorithm through a know programming language (preferably C/C++ or Java)
- solve the benchmark instances through the implemented algorithm
- deliver the project code, the results, and a report of the work (deadline 2)
- present the solution method adopted and the results obtained (to be defined within Jan 9–19, 2018, during the last lessons of the course)

Deadlines:

- deadline 1 (problem formulation): 26/11/2017, 11:59 p.m.
- deadline 2 (project code, results, and report): 08/01/2018, 11:59 p.m.

Assignment: groups and evaluation

Groups:

- 4/5 students per group (one leader must be chosen for corresponding)
- groups composition, up to you, must be decided and submitted before 13/10/2017 (no more than 25 groups per course!)
- students in charge of the group formation:
  - OMA A-L: **maher.karzoun@gmail.com** - goo.gl/SVWQqE
  - OMA M-Z: **santolini.lorenzo@gmail.com** - goo.gl/NZHqbt

Evaluation:

- one single grade [0,10] per group (no individual grades). The best group of each course will get 2 more points.
- based on:
  - correctness/completeness of your formulation
  - soundness/rationality/efficiency of your solution algorithm
  - goodness of your solutions (closeness to the benchmark) on public and private instances
  - quality/clarity of the presentation.

Support:

- during the course, 3 lessons (1h:30m each, approximately two weeks apart) will be dedicated to assistance in assignment development.
- a FAQ list concerning the project will be available. New entries appear as soon as new questions (of common interest) are posed by the students.

Assignment: given materials

- **instances.zip**: set of the public instances (format already explained) plus a test instance (with solutions) corresponding to the example shown before.
- **ETPchecker.exe**: a feasibility checker and obj calculator.
  It is a black-box tool able to read an ETP instance and a relative specified solution. If the provided solution is feasible, it returns the value of the objective value (total penalty) of that solution, otherwise it returns an error relative to some infeasible characteristic of the solution.

  How to use it from a command line:

  ```
  $ ETPchecker.exe instance_name -check solution_file
  ```

  PS: only 'logical' feasibility is checked! no guarantees if a non-correctly formatted solution is provided.
- **benchmarks.xlsx**: an Excel file containing the properties and the benchmark values of the public instances. Given the solution values generated by your algorithm, it automatically calculates the percentage gap with respect to the benchmarks.

Assignment: deliveries specifications (1)

Model formulation:

- upload a pdf file (max 2 pages) named **ETPmodel_OMAXX_groupYY**, where 'XX' is 'AL' or 'MZ' (depending on your OMA module) and 'YY' is the number of your group, in *'Elaborati'* section of the course's site
- the file must contain:
  - the definition of the notation used for the parameters and for the variables (notation already introduced here must be maintained)
  - the LP formulation and a brief explanation of its objective function and constraints.

Assignment: deliveries specifications (2)

Project code, results and report:

- upload a .zip archive file named **ETPproject_OMAXX_groupYY** in *'Elaborati'* section of the course's site
- the archive must contain:
  - 7 text files containing your best solutions of the 7 given public instances (specifications follow)
  - an .xlcx file named **benchmarks_OMAXX_groupYY**: it is the given *benchmarks.xlxs* file duly completed with your results
  - an .exe file named **ETPsolver_OMAXX_groupYY** (specifications follow)
  - a folder containing the complete development of the software (source code, project files, ...)
  - a .pdf file named **ETPreport_OMAXX_groupYY** containing the report of the work. Max. 3 pages, focused on algorithms and results.

Assignment: solver and solutions specifications

The solver:

- a command line application that, inputed with an instance name and a time limit t_{lim} (in seconds), solves the ETP problem on the specified instance within the time limit and generates a solution text file
- the software must be called through the following syntax:
  `$ ETPsolver_OMAXX_groupYY.exe instancename -t timelimit`
- must be working stand-alone, i.e. without the need of installing any additional software
- must write/overwrite a text file any time a new best solution is found!

A solution:

- a text file named **instancename_OMAXX_groupYY.sol** containing, for each exam, the assigned time-slot
- format: 1 line per exam. Each line has the format

  `INT1 INT2`

  where `INT1` is the exam ID and `INT2` is the ID of the assigned time-slot (IDs must correspond to those read in the instance files)
- must be feasible!

Assignment: deliveries specifications (3)

Final presentation:

- in English, to the whole class, using your laptop linked to the projector
- 15 minutes max. per group (including set-up time and 2-3 minutes for possible questions)
- at least half+1 elements of the group must present (4-student group: 3; 5-student group: 4)
- no need to present the problem, the instances, or whatever is common to all groups
- focus on your solution ideas and algorithms, on their tuning, on results and performances
- include some conclusive considerations
- upload the presented slides as a .pdf file named **ETPslides_OMAXX_groupYY** in *'Elaborati'* section of the course's site after the presentation