A variant of the high-school timetabling problem and a software solution for it based on integer linear programming

Iulian Ober

University of Toulouse, France

August 24, 2016

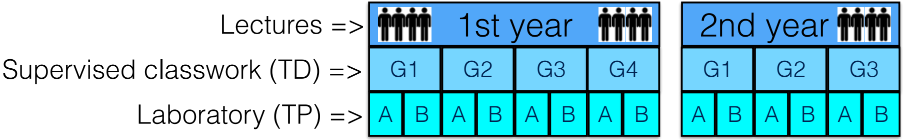
Outline

Outline

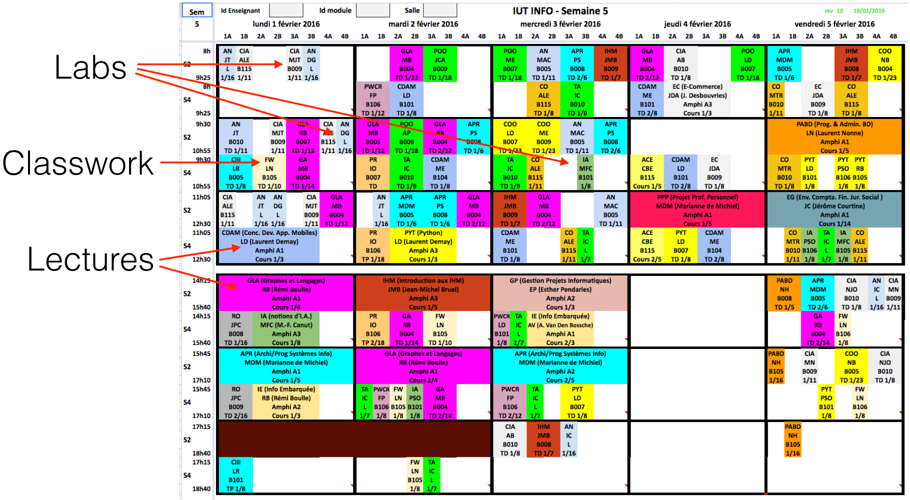
Timetable at the Toulouse 2 University Institute of Technology (IUT)

A particular type of High-School Timetabling problem.

Hierarchical division of student groups:



An example


Labs
Classwork
Lectures
IUT INFO - Semaine 5
rev 10 19/01/2016
lundi 1 février 2016
mardi 2 février 2016
mercredi 3 février 2016
jeudi 4 février 2016
vendredi 5 février 2016

Features

- predefined course-group-teacher assignment
- different room pools (per type):
  - laboratories (6), classrooms (5), multimedia rooms (2), amphitheaters (1)
  - rooms in a pool are interchangeable
  - subject to availability constraints
- dense program:
  - e.g., 33h over 4.5 days for G1A
  - potentially long days: start at 8 AM, end at 6:40 PM
- many research professors and adjunct professors (main job in industry) $\Rightarrow$ tight availability constraints
- Potentially overlapping time slots, e.g., 8-10 AM $\parallel$ (8-9:30 AM, 9h30-11 AM) (may be modeled with disjoint time slots by subdividing them)

Features

- predefined course-group-teacher assignment
- different room pools (per type):
  - laboratories (6), classrooms (5), multimedia rooms (2), amphitheaters (1)
  - rooms in a pool are interchangeable
  - subject to availability constraints
- dense program:
  - e.g., 33h over 4.5 days for G1A
  - potentially long days: start at 8 AM, end at 6:40 PM
- many research professors and adjunct professors (main job in industry)
  $\Rightarrow$ tight availability constraints
- Potentially overlapping time slots, e.g., 8-10 AM $\|$ (8-9:30 AM, 9h30-11 AM)
  (may be modeled with disjoint time slots by subdividing them)

Features

- predefined course-group-teacher assignment

- different room pools (per type):
  - laboratories (6), classrooms (5), multimedia rooms (2), amphitheaters (1)
  - rooms in a pool are interchangeable
  - subject to availability constraints

- dense program:
  - e.g., 33h over 4.5 days for G1A
  - potentially long days: start at 8 AM, end at 6:40 PM

- many research professors and adjunct professors (main job in industry)
  $\Rightarrow$ tight availability constraints

- Potentially overlapping time slots, e.g., 8-10 AM $\|$ (8-9:30 AM, 9h30-11 AM)
  (may be modeled with disjoint time slots by subdividing them)

Features

- predefined course-group-teacher assignment
- different room pools (per type):
  - laboratories (6), classrooms (5), multimedia rooms (2), amphitheaters (1)
  - rooms in a pool are interchangeable
  - subject to availability constraints
- dense program:
  - e.g., 33h over 4.5 days for G1A
  - potentially long days: start at 8 AM, end at 6:40 PM
- many research professors and adjunct professors (main job in industry) $\Rightarrow$ tight availability constraints
- Potentially overlapping time slots, e.g., 8-10 AM $\|$ (8-9:30 AM, 9h30-11 AM) (may be modeled with disjoint time slots by subdividing them)

Motivation for this work

First of all, practical !

- need of convenient tooling
- keep control over the resolution engine, in order to:
  - fine-tune the solution
  - add new types of constraints as they come up

⇒ **XLSScheduler**: an automatic generator

- based on open off-the-shelf ILP solvers (Cbc, Gurobi)
- interacting via easy-to-use data format (Excel), batch style
- reasonably generic (reusable, extensible. . .)

Motivation for this work

First of all, practical !

- need of convenient tooling
- keep control over the resolution engine, in order to:
  - fine-tune the solution
  - add new types of constraints as they come up

⇒ **XLSScheduler**: an automatic generator

- based on open off-the-shelf ILP solvers (Cbc, Gurobi)
- interacting via easy-to-use data format (Excel), batch style
- reasonably generic (reusable, extensible. . .)

Outline

Problem formalization

Time slots:

- $Slots$ set (e.g., $\{Mo0800\_0930, Mo0800\_1000, Mo0930\_1100, ...\}$)
- $SlotTypes$ set, $slType : Slots \to SlotTypes$ (e.g., use duration as type)
- $rank : Slots \to \mathbb{N}$ defines partial order between slots. E.g.:
  - $rank(Mo1415\_1545) = 4$
  - $rank(Mo1545\_1715) = 5$
  - $rank(Mo1715\_1845) = 6$
  - $rank(Mo1545\_1745) = 5$
  - $rank(Tu0800\_0930) = 8$
- $overlap \subseteq Slot \times Slot$. $(s_1, s_2) \in overlap$ (denoted $s_1 \parallel s_2$) iff slots s_1 and s_2 overlap chronologically. E.g.:
  - $Mo1545\_1745 \parallel Mo1545\_1715$
  - $Mo1545\_1745 \parallel Mo1715\_1845$

Problem formalization

Time slots:

- $Slots$ set (e.g., $\{Mo0800\_0930, Mo0800\_1000, Mo0930\_1100, ...\}$)
- $SlotTypes$ set, $slType : Slots \rightarrow SlotTypes$ (e.g., use duration as type)
- $rank : Slots \rightarrow \mathbb{N}$ defines partial order between slots. E.g.:
  - $rank(Mo1415\_1545) = 4$
  - $rank(Mo1545\_1715) = 5$
  - $rank(Mo1715\_1845) = 6$
  - $rank(Mo1545\_1745) = 5$
  - $rank(Tu0800\_0930) = 8$
- $overlap \subseteq Slot \times Slot$. $(s_1, s_2) \in overlap$ (denoted $s_1 \parallel s_2$) iff slots s_1 and s_2 overlap chronologically. E.g.:
  - $Mo1545\_1745 \parallel Mo1545\_1715$
  - $Mo1545\_1745 \parallel Mo1715\_1845$

Problem formalization

Time slots:

- $Slots$ set (e.g., $\{Mo0800\_0930, Mo0800\_1000, Mo0930\_1100, ...\}$)
- $SlotTypes$ set, $slType : Slots \rightarrow SlotTypes$ (e.g., use duration as type)
- $rank : Slots \rightarrow \mathbb{N}$ defines partial order between slots. E.g.:
  - $rank(Mo1415\_1545) = 4$
  - $rank(Mo1545\_1715) = 5$
  - $rank(Mo1715\_1845) = 6$
  - $rank(Mo1545\_1745) = 5$
  - $rank(Tu0800\_0930) = 8$
- $overlap \subseteq Slot \times Slot$. $(s_1, s_2) \in overlap$ (denoted $s_1 \parallel s_2$) iff slots s_1 and s_2 overlap chronologically. E.g.:
  - $Mo1545\_1745 \parallel Mo1545\_1715$
  - $Mo1545\_1745 \parallel Mo1715\_1845$

Problem formalization

Teachers:

- $Teachers$ set
- $tAvail : Teachers \times Slots \rightarrow \{0, 1\}$ defines teacher availability per time-slot.
- $tPref : Teachers \times Slots \rightarrow \{0, 1\}$ defines teacher preferences per time-slot.

Slots / Instructor availability (=0,5 or 1) and preferences (=1)	BC	BE	BEV	BMF	BQ	BW	CH	DC	DDM	DE	DEF	DNB	DNH	FE	FQ	GN	GNP	GQ	GT	HQ	JD	JP	KD	KEB	KNC	KQD	KT	LC	ME	MH	MO
Mo 9-11 AM	0,0	0,5	1,0	0,5	0,0	0,5	0,5	1,0	1,0	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	0,0
Mo 8-9:30	0,0	0,5	1,0	0,5	0,0	0,5	0,5	1,0	1,0	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	0,0
Mo 9:30-11	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
Mo 11-12:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
Mo 14-15:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,0	1,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Mo 15:30-17	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,0	1,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Mo 17-18:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,0	1,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Tu 8-10 AM	0,0	0,5	1,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Tu 8-9:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Tu 9:30-11	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Tu 11-12:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Tu 14-15:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
Tu 15:30-17	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
Tu 17-18:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
We 8-10 AM	0,0	0,5	0,0	0,5	0,5	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
We 8-9:30	0,0	0,5	0,0	0,5	0,5	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
We 9:30-11	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
We 11-12:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	0,0
We 14-15:30	1,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	1,0
We 15:30-17	1,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	1,0
We 17-18:30	1,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
Th 8-9:30	0,5	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	0,0
Th 9:30-11	0,5	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	0,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
Th 11-12:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	0,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
Th 14-15:30	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
Th 15:30-17	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
Th 17-18:30	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
Fr 8-9:30	1,0	0,5	0,0	0,5	0,0	0,5	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	1,0	0,0
Fr 9:30-11	1,0	0,5	0,0	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	1,0	1,0
Fr 11-12:30	1,0	0,5	0,0	0,5	0,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	1,0	1,0
Fr 14-15:30	1,0	0,5	0,0	0,5	0,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	1,0	1,0
Fr 15:30-17	1,0	0,5	0,0	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	0,0	0,5	0,5	0,5	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	1,0	1,0
Fr 17-18:30	1,0	0,5	0,0	0,5	0,0	0,5	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	0,0	0,5	0,5	0,5	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	1,0	0,5

Problem formalization

Student groups:

- $Groups$ set
- $subgroup \subseteq Groups \times Groups$. $(g_1, g_2) \in subgroup$ denoted $g_1 \prec g_2$

Problem formalization

Rooms:

- $RoomCategories$ set
- $rAvail : RoomCategories \times Slots \to \mathbb{N}$ defines how many rooms of a particular category are available at each time-slot

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \rightarrow Teachers$
- $courseSlotType : Courses \rightarrow SlotTypes$
- $courseRoomCat : Courses \rightarrow RoomCategories$
- $courseRoomNb : Courses \rightarrow \mathbb{N}$
- $courseGroupNb : Courses \times Group \rightarrow \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \rightarrow \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \to Teachers$
- $courseSlotType : Courses \to SlotTypes$
- $courseRoomCat : Courses \to RoomCategories$
- $courseRoomNb : Courses \to \mathbb{N}$
- $courseGroupNb : Courses \times Group \to \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \to \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \rightarrow Teachers$
- $courseSlotType : Courses \rightarrow SlotTypes$
- $courseRoomCat : Courses \rightarrow RoomCategories$
- $courseRoomNb : Courses \rightarrow \mathbb{N}$
- $courseGroupNb : Courses \times Group \rightarrow \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \rightarrow \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \rightarrow Teachers$
- $courseSlotType : Courses \rightarrow SlotTypes$
- $courseRoomCat : Courses \rightarrow RoomCategories$
- $courseRoomNb : Courses \rightarrow \mathbb{N}$
- $courseGroupNb : Courses \times Group \rightarrow \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \rightarrow \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \rightarrow Teachers$
- $courseSlotType : Courses \rightarrow SlotTypes$
- $courseRoomCat : Courses \rightarrow RoomCategories$
- $courseRoomNb : Courses \rightarrow \mathbb{N}$
- $courseGroupNb : Courses \times Group \rightarrow \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \rightarrow \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \rightarrow Teachers$
- $courseSlotType : Courses \rightarrow SlotTypes$
- $courseRoomCat : Courses \rightarrow RoomCategories$
- $courseRoomNb : Courses \rightarrow \mathbb{N}$
- $courseGroupNb : Courses \times Group \rightarrow \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \rightarrow \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \to Teachers$
- $courseSlotType : Courses \to SlotTypes$
- $courseRoomCat : Courses \to RoomCategories$
- $courseRoomNb : Courses \to \mathbb{N}$
- $courseGroupNb : Courses \times Group \to \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \to \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \to Teachers$
- $courseSlotType : Courses \to SlotTypes$
- $courseRoomCat : Courses \to RoomCategories$
- $courseRoomNb : Courses \to \mathbb{N}$
- $courseGroupNb : Courses \times Group \to \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \to \{0, 1\}$

Problem formalization

Courses:

- $Courses$ set
- $courseTeacher : Courses \rightarrow Teachers$
- $courseSlotType : Courses \rightarrow SlotTypes$
- $courseRoomCat : Courses \rightarrow RoomCategories$
- $courseRoomNb : Courses \rightarrow \mathbb{N}$
- $courseGroupNb : Courses \times Group \rightarrow \mathbb{N}$. $courseGroupNb(c, g)$ defines how many sessions there are in the course c for the group g.
- $consecCourses \subseteq Courses$ (courses whose instances must be consecutive)
- $precedes \subseteq Course \times Course$. $(c_1, c_2) \in precedes$ (denoted $c_1 \ll c_2$) means all time-slots allocated to c_1 must chronologically precede the slots of c_2.
- $courseAvail, coursePref : Courses \times Slots \rightarrow \{0, 1\}$

Model variables

$$\mathbf{x}_{s,c,g} \in \{0,1\} \text{ for } s \in Slots,\ c \in Courses,\ g \in Groups$$

$\mathbf{x}_{s,c,g} = 1$ iff group g has a course c in time-slot s

Hard constraints

All the courses are allocated a slot:

$$\forall c \in Courses, \forall g \in Groups : \sum_{s \in Slots} \mathbf{x}_{s,c,g} = courseGroupNb(c,g)$$

Hard constraints

No group has two courses in parallel (nor a course in parallel with another course of one of its super-groups):

$$\forall s \in Slots, \forall g \in Groups :$$

$$\sum_{c \in Courses} \left(\mathbf{x}_{s,c,g} + \sum_{\substack{g' \in Groups \\ g \prec g'}} \mathbf{x}_{s,c,g'} \right) \leq 1$$

Taking into account slot overlapping:

$$\forall s, s' \in Slots \text{ such that } s \parallel s', \forall g \in Groups :$$

$$\sum_{c \in Courses} \left(\mathbf{x}_{s,c,g} + \mathbf{x}_{s',c,g} + \sum_{\substack{g' \in Groups \\ g \prec g'}} (\mathbf{x}_{s,c,g'} + \mathbf{x}_{s',c,g'}) \right) \leq 1$$

Hard constraints

No group has two courses in parallel (nor a course in parallel with another course of one of its super-groups):

$$\forall s \in Slots, \forall g \in Groups :$$

$$\sum_{c \in Courses} \left(\mathbf{x}_{s,c,g} + \sum_{\substack{g' \in Groups \\ g \prec g'}} \mathbf{x}_{s,c,g'} \right) \leq 1$$

Taking into account slot overlapping:

$$\forall s, s' \in Slots \text{ such that } s \parallel s', \forall g \in Groups :$$

$$\sum_{c \in Courses} \left(\mathbf{x}_{s,c,g} + \mathbf{x}_{s',c,g} + \sum_{\substack{g' \in Groups \\ g \prec g'}} (\mathbf{x}_{s,c,g'} + \mathbf{x}_{s',c,g'}) \right) \leq 1$$

Hard constraints

There are enough rooms of each category for each slot:

$$\forall s \in Slots, \forall t \in RoomCategories:$$

$$\sum_{\substack{c \in Courses \\ courseRoomCat(c)=t}} \sum_{g \in Groups} courseRoomNb(c) \cdot \mathbf{x}_{s,c,g} \leq rAvail(t,s)$$

No precise room assignment $\Rightarrow$ smaller model.

Extra constraint for taking into account slot overlapping

Hard constraints

There are enough rooms of each category for each slot:

$$\forall s \in Slots, \forall t \in RoomCategories:$$

$$\sum_{\substack{c \in Courses \\ courseRoomCat(c)=t}} \sum_{g \in Groups} courseRoomNb(c) \cdot \mathbf{x}_{s,c,g} \leq rAvail(t, s)$$

No precise room assignment $\Rightarrow$ smaller model.

Extra constraint for taking into account slot overlapping

Hard constraints

There are enough rooms of each category for each slot:

$$\forall s \in Slots, \forall t \in RoomCategories:$$

$$\sum_{\substack{c \in Courses \\ courseRoomCat(c)=t}} \sum_{g \in Groups} courseRoomNb(c) \cdot \mathbf{x}_{s,c,g} \leq rAvail(t,s)$$

No precise room assignment $\Rightarrow$ smaller model.

Extra constraint for taking into account slot overlapping

Hard constraints

Consecutiveness:

$$\forall c \in consecCourses, \forall g \in Groups,$$

$$\forall s, s' \in Slots \text{ such that } rank(s') - rank(s) \geq courseGroupNb(c, g) :$$

$$\mathbf{x}_{s,c,g} + \mathbf{x}_{s',c,g} \leq 1$$

Precedence:

$$\forall c, c' \in Courses \text{ such that } c \ll c',$$

$$\forall g, g' \in Groups \text{ such that } courseGroupNb(c, g) \cdot courseGroupNb(c', g') > 0,$$

$$\forall s, s' \in Slots \text{ such that } rank(s) < rank(s') :$$

$$\mathbf{x}_{s,c',g'} + \mathbf{x}_{s',c,g} \leq 1$$

Hard constraints

Consecutiveness:

$$\forall c \in consecCourses, \forall g \in Groups,$$

$$\forall s, s' \in Slots \text{ such that } rank(s') - rank(s) \geq courseGroupNb(c, g) :$$

$$\mathbf{x}_{s,c,g} + \mathbf{x}_{s',c,g} \leq 1$$

Precedence:

$$\forall c, c' \in Courses \text{ such that } c \ll c',$$

$$\forall g, g' \in Groups \text{ such that } courseGroupNb(c, g) \cdot courseGroupNb(c', g') > 0,$$

$$\forall s, s' \in Slots \text{ such that } rank(s) < rank(s') :$$

$$\mathbf{x}_{s,c',g'} + \mathbf{x}_{s',c,g} \leq 1$$

Hard constraints

Other constraints:

- The slots allocated to a $Course$ have the right $SlotType$
- The teachers are available at the allocated slots
- A teacher does not have several courses in parallel
- Course availability per slot ($courseAvail$) is observed

Soft constraints and objective

Soft constraint: minimize use of unpreferred slots.

Objective = weighted sum of:

- cost incurred by the use of unpreferred slots of teachers

$$UT = \sum_{\substack{s \in Slots \\ c \in Courses, t = courseTeacher(c) \\ g \in Groups}} (tAvail(t,s) - tPref(t,s)) \cdot \mathbf{x}_{s,c,g}$$

- cost incurred by the use of unpreferred slots for courses

$$UC = \sum_{\substack{s \in Slots \\ c \in Courses \\ g \in Groups}} (courseAvail(c,s) - coursePref(c,s)) \cdot \mathbf{x}_{s,c,g}$$

Soft constraints and objective

Soft constraint: minimize use of unpreferred slots.

Objective = weighted sum of:

- cost incurred by the use of unpreferred slots of teachers

$$UT = \sum_{\substack{s \in Slots \\ c \in Courses, t = courseTeacher(c) \\ g \in Groups}} (tAvail(t, s) - tPref(t, s)) \cdot \mathbf{x}_{s,c,g}$$

- cost incurred by the use of unpreferred slots for courses

$$UC = \sum_{\substack{s \in Slots \\ c \in Courses \\ g \in Groups}} (courseAvail(c, s) - coursePref(c, s)) \cdot \mathbf{x}_{s,c,g}$$

Soft constraints and objective

Soft constraint: minimize use of unpreferred slots.

Objective = weighted sum of:

- cost incurred by the use of unpreferred slots of teachers

$$UT = \sum_{\substack{s \in Slots \\ c \in Courses, t = courseTeacher(c) \\ g \in Groups}} (tAvail(t,s) - tPref(t,s)) \cdot \mathbf{x}_{s,c,g}$$

- cost incurred by the use of unpreferred slots for courses

$$UC = \sum_{\substack{s \in Slots \\ c \in Courses \\ g \in Groups}} (courseAvail(c,s) - coursePref(c,s)) \cdot \mathbf{x}_{s,c,g}$$

Other soft constraints: minimizing "long days"

Avoid days that start at 8AM to 6:45PM

- $Days$ is a set of the days of week
- $first : Days \to Slots$ maps each day to its first slot
- $last : Days \to Slots$ maps each day to its last slot
- $\mathbf{L}_{d,t} \in \{0,1\}$ for $d \in Days$ and $t \in Teachers$: auxiliary variable ($\mathbf{L}_{d,t} = 1$ iff d is a "long day" for teacher t
- constraint linking $\mathbf{L}$ to $\mathbf{x}$:

$$\forall d \in Days, t \in Teachers :$$

$$\sum_{\substack{c \in Courses \\ courseTeacher(c)=t}} \sum_{g \in Groups} (\mathbf{x}_{first(d),c,g} + \mathbf{x}_{last(d),c,g}) - 2 \cdot \mathbf{L}_{d,t} \leq 1$$

- additional cost component weighted into the cost function:

$$LT = \sum_{\substack{t \in Teachers \\ d \in Days}} \mathbf{L}_{d,t}$$

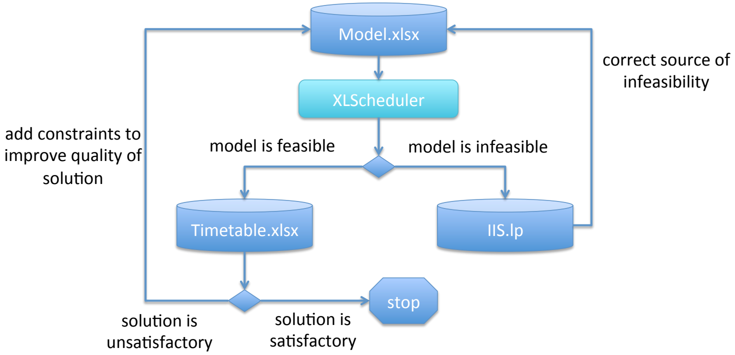
Other soft constraints

- Minimizing "long days" for students
- Clustering busy times
- ...

Outline

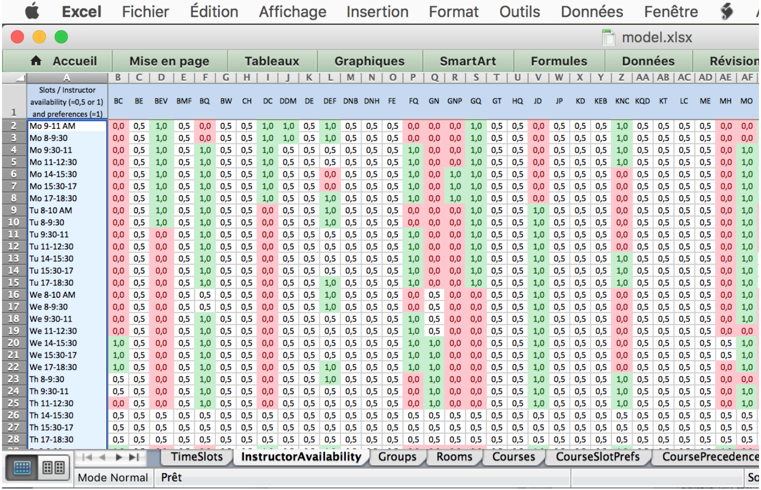
XLSScheduler tool

- Batch tool based on off-the-shelf ILP solvers (Gurobi or Cbc)
- Available as online service (Cbc-based, upon request)
- Workflow:



XLSScheduler tool

Input/Output format: convenient XLSX file format



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1	Slots / Instructor availability (=0,5 or 1) and preferences (=1)	BC	BE	BEV	BMF	BQ	BW	CH	DC	DDM	DE	DEF	DNB	DNH	FE	FQ	GN	GNP	GQ	GT	HQ	JD	JP	KD	KEB	KNC	KQD	KT	LC	ME	MH	MO
2	Mo 9-11 AM	0,0	0,5	1,0	0,5	0,0	0,5	0,5	1,0	1,0	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	0,0
3	Mo 8-9:30	0,0	0,5	1,0	0,5	0,0	0,5	0,5	1,0	1,0	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	0,0
4	Mo 9:30-11	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
5	Mo 11-12:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
6	Mo 14-15:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,0	1,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
7	Mo 15:30-17	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	1,0	0,0	1,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
8	Mo 17-18:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,0	1,0	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
9	Tu 8-10 AM	0,0	0,5	1,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
10	Tu 8-9:30	0,0	0,5	1,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
11	Tu 9:30-11	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
12	Tu 11-12:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
13	Tu 14-15:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
14	Tu 15:30-17	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
15	Tu 17-18:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,0	0,0	1,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
16	We 8-10 AM	0,0	0,5	0,0	0,5	0,5	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
17	We 8-9:30	0,0	0,5	0,0	0,5	0,5	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
18	We 9:30-11	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
19	We 11-12:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	0,5	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	0,0
20	We 14-15:30	1,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	1,0
21	We 15:30-17	1,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	1,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	1,0
22	We 17-18:30	1,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,0	1,0
23	Th 8-9:30	0,5	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	1,0	0,5	0,5	0,5	0,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	0,0
24	Th 9:30-11	0,5	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	0,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
25	Th 11-12:30	0,0	0,5	0,0	0,5	1,0	0,5	0,5	0,0	0,5	0,5	0,5	0,5	0,5	0,5	0,0	1,0	0,0	0,0	0,5	0,5	1,0	0,5	0,5	0,5	1,0	0,5	0,5	0,5	0,5	0,0	1,0
26	Th 14-15:30	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
27	Th 15:30-17	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
28	Th 17-18:30	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5

Experimental results

- 2 versions of the tool:
  - Python / Gurobi (Python API + solver)
    http://www.gurobi.com
  - Python / PuLP / Cbc
    https://projects.coin-or.org/Cbc
- high speed (within seconds) and high quality for real-life workloads
- both solvers perform very well, but slow model creation with PuLP

Outline

Conclusion

- practical approach for a relatively classical HSTT problem
- generic, extensible solution
- quick and good quality results
- possibility to troubleshoot and refine results

https://www.irit.fr/~Iulian.Ober/XLSScheduler

That's all folks!

Thank you!

Questions?