ICAI Time Schedule Optimization

The project's objective is to optimize the schedule for the Mathematical Engineering and Artificial Intelligence degree of ICAI. Currently, this degree is comprised of 3 different years, with 5 groups in total. The first and second degree years have 2 groups while the third year has only one.

To achieve this, students must attend the exact number of hours of each subject they are enrolled in and fulfill their curriculum (SHR). None of the subjects can be taught for a different number of hours per week than the stipulated ones.

Teachers need to have the necessary capacities to impart their knowledge to students (T2S). Not all teachers can impart all the classes. For example, David Alfaya can only teach Algebra and Discrete Mathematics but not Physics or Computer Vision.

Classes must be taught by a certain number of teachers per group (CTC). For example, an Algebra class needs only one teacher for a group while a dynamic systems laboratory or optimization class need more than one teacher.

There are also some physical restrictions. For example, teachers cannot clone themselves (NTC), so one teacher cannot teach more than one class at the same time. The same applies to students, who cannot attend two classes at the same time (NSD).

Although the university is available from 8 to 12 and 15 to 21, depending on what year of study the students are in, they will only be able to have class at certain hours (EOO). First and second year students have class from 8 to 12 and third-year students from 16 to 20. So, even though first year students could have classes during the evening, the principal has to organize the schedule so that all their subjects are imparted during the morning hours.

Plus, there are some rules that makes both the teachers and the students lifes better when it comes to scheduling classes. During a day, a group cannot have more than two hours of a certain subject (NSA). Furthermore, if a group has two classes of a subject on a day, these classes have to go together (NHS). They cannot be split by other subjects. For example, on a Monday, third-year students may have two hours of Optimization, and so these classes should be consecutive. It is not possible to have Optimization at 16:00, Computer Vision at 17:00 and Optimization again at 18:00; instead, it would be Optimization from 16:00 to 18:00 and then Computer Vision.

The space in the university is limited. There are not enough physical laboratories in order to impart all of the practical classes of certain subjects at the same time, so there has to be some account of how many laboratories each group needs for each subject (NCO). For example, third year students need 3 robotic labs for Computer Vision.

Professors were asked about their availability to impart classes at each time slot of the week. The schedule should have teachers giving their classes at the days and hours they can. If it's impossible for a teacher to impart his classes at a certain hour and day, this parameter will be a 0, and if he is accessible, it will be a 1 (CTA).

Although they were available, they may have personal preferences (CTP). If for a certain reason, a teacher prefers not to give a class at a certain hour of a day, we'll take that into account, and we will try to assign the teacher another hour or day. This will be included in the objective function as a soft constraint.

Hard Constraints

- CTC (complete teacher capacity):
  $\sum_t x = \mathrm{TN}_{gs} \cdot \gamma_{gdhs} \quad \forall g, d, h, s$

This constraint assures that each class is given by the exact number of teachers. The gamma auxiliary variable is needed because we can either give the class or not, so the sum will add up to either TN or 0, and nothing in between.

- NSD & EOO (non-subject duplication and early or overnight):
  $\sum_{s,t} x \cdot \frac{1}{\mathrm{TN}_{gs}} = \mathrm{A}_{gh} \quad \forall g, d, h$

This constraint assures that each group is given only one subject at a specific day and hour, in the groups early or overnight schedule. If A_{gh} (availability) is 0, the group will not be imparted any class at that slot. It is divided by TN to eliminate the repetitions created by the number of teachers. It is important to consider that when programming the constraint, it has to be controlled that TN is not 0 and skip the constraint when it happens.

- SHR & T2S & CTA (subject hours requirement, teacher to speciality and checking teacher availability):
  $\sum_{d,h,t} x \cdot \mathrm{TS}_{st} \cdot \mathrm{TA}_{tdh} = \mathrm{HS}_s \cdot \mathrm{TN}_{gs} \quad \forall g, s$

This constraint controls that the teacher, in an available hour and day, gives a subject which he has knowledge about. TS controls the specialty, and TA their availability. The sum has to be equal to the H (hour requirements for the subject), multiplied by the TN, which eliminates repetitions of the number of teachers.

- NTC (no teacher clonation):
  $\sum_{g,s} x \leq 1 \quad \forall d, h, t$

This constraint controls that a teacher is giving at most one subject to a single group in a given day and hour.

- TCP (teacher consistency principle):
  $\sum_{d,h} x = \mathrm{H}_s \cdot \zeta_{gst} \quad \forall g, s, t$

This constraint's controls that a teacher gives all the classes of a subject to a group, or that he doesn’t teach them at all. This is controlled by an auxiliar variable zeta.

- NCO (no class overflow):
  $\sum_{g,s,t} x \cdot \frac{\mathrm{Q}_{cgs}}{\mathrm{TN}_{gs}} \leq \mathrm{L}_c \quad \forall c, d, h$

This constraint controls that there is always less classes occupied of a certain type than L (classes available of each type of class). It is multiplied by Q, which is the number of classes that a group needs, divided by TN in order to avoid the repetition of teachers.

- NSA (no subject abuse):
  $\sum_{h,t} x \leq 2 \cdot \mathrm{TN}_{gs} \quad \forall g, d, s$

This constraint controls that a subject is not given more than two hours a day for every group.

- NHS (no holes in subject):
  $\sum_t x_{h+1} - \sum_t x_h \leq \delta'_{gdhs} \cdot \mathrm{TN}_{gs} \quad \forall g, d, h < 10, s$
  $\sum_t x_{h+1} - \sum_t x_h \geq -\delta''_{gdhs} \cdot \mathrm{TN}_{gs} \quad \forall g, d, h < 10, s$
  $\sum_h \delta'_{gdhs} + \delta''_{gdhs} \leq 2 \quad \forall g, d, s$

This constraint will control that during a day, if a subject is given twice for a group, this classes will be imparted in continued hours. It has to be counted how many positive and negative edges there are for a subject and group in a day. In order to achieve this, two types of auxiliary variables are created, one to count the positive edges (δ') and another one for the negative edges (δ'').The sum of these variables has to be less than or equal to 2.

Objective Function (Soft Constraints)

We will use the concept of soft constraints, which are constraints that we are allowed to break, but we will be penalized for doing so. The objective function is the following:

- CTP (checking teacher preference):

$$min_x \sum_{g,d,h,s,t} x_{gdhst} \cdot \mathrm{TP}_{dht}$$

This penalizes the model for every teacher that is teaching in a time slot that he doesn't prefer.

Model

The model chosen is ConcreteModel from the Pyomo Python library. This is the summary of the model:

```
Number of objectives: 1
Number of constraints: 21785
Number of variables: 204930
Number of binary variables: 204930
Number of integer variables: 204930
Number of continuous variables: 0
Number of nonzeros: 1531240
Sense: minimize
```

This are the coefficient statistics of the model, that give us an idea about the numerical stability of the model:

```
Coefficient statistics:
  Matrix range      [3e-01, 4e+00]
  Objective range   [1e+00, 1e+00]
  Bounds range      [1e+00, 1e+00]
  RHS range         [1e+00, 8e+00]
```

The model takes **4.3** seconds to load.

Results

We first tried with glpk, but it was too slow, so we ended up using Gurobi as our preferred solver. This is are the results:

```
Solved with primal simplex

Use crossover to convert LP symmetric solution to basic solution...

Root relaxation: objective 0.000000e+00, 1230 iterations, 0.09 seconds (0.09 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.00000    0    4          -    0.00000      -     -    0s
H    0     0                       3.0000000    0.00000   100%     -    0s
H    0     0                       2.0000000    0.00000   100%     -    1s
H    0     0                       0.0000000    0.00000  0.00%     -    1s

Cutting planes:
  Gomory: 2
  Cover: 8
  Clique: 4
  MIR: 1

Explored 1 nodes (2751 simplex iterations) in 1.14 seconds (1.25 work units)
Thread count was 20 (of 20 available processors)

Solution count 3: 0 2 3

Optimal solution found (tolerance 1.00e-04)
Best objective 0.000000000000e+00, best bound 0.000000000000e+00, gap 0.0000%
```

We can see that the model is able to find the optimal solutio with a 0.0% gap between primal and dual bounds.
The model takes **4.95** seconds to solve.

↓↓↓↓↓ IMPORTANT ↓↓↓↓↓

> This is a link with an example schedule for IMAT:
> IMAT Schedule

> This is the link to the repository with all the data and code used to create the schedule:
> Repository

↑↑↑↑↑ IMPORTANT ↑↑↑↑↑