Algorithmic Methods for Mathematical Models

– COURSE PROJECT –

In a scientific conference, the program chair (PC) is in charge of scheduling the different talks. Scheduling a talk means assigning a room and a time slot to it. The PC is particularly concerned about the way talks on related topics are scheduled.

More precisely, we have a set of n talks $N = \{1, 2, \dots, n\}$ and two symmetric Boolean matrices P_{ij} and S_{ij} such that respectively $P_{ij} = 1$ if and only if talks i and j are *primarily related*, and $S_{ij} = 1$ if and only if talks i and j are *secondarily related*. The input guarantees that any two talks cannot be both primarily and secondarily related; that is, for any $1 \leq i \leq j \leq n$, it cannot be the case that $P_{ij} = 1$ and $S_{ij} = 1$. For convenience, it is ensured that the matrices also satisfy that $P_{ii} = S_{ii} = 0$ for all $1 \leq i \leq n$.

For our purposes, the conference consists of a sequence of t time slots $T = \{1, 2, \dots, t\}$ and a sequence of r lecture rooms $R = \{1, 2, \dots, r\}$ that can be used in parallel during all time slots. A conference program is an assignment of a lecture room and a time slot to each talk so that two different talks are not assigned the same room at the same time. The number of talks n, the number of time slots t and the number of lecture rooms r satisfy that $0 < n \leq t \times r$.

A conference program has to satisfy some additional constraints in order to be valid. Namely, any two primarily related talks cannot be scheduled at the same time. On the other hand, it is not allowed that such talks are scheduled too far apart either: the time distance between two primarily related talks should be at most d, where this parameter is part of the input and satisfies $0 < d < t$. E.g., if $d = 1$ and talks 3 and 4 are primarily related, then they cannot be scheduled at time slots 2 and 5 respectively, as their time distance would be $5 - 2 = 3 > 1$.

Finally, we measure the cost of a valid program by the number of different pairs of talks that are secondarily related and scheduled at the same time. The goal of this project is to find a valid conference program that minimizes this quantity.

1. **Work to be done:**
   (a) State the problem formally. Specify the inputs and the outputs, as well as any auxiliary sets of indices that you may need, and the objective function.
   (b) Build an integer linear programming model for the problem and implement it in OPL.
   (c) Because of the complexity of the problem, heuristic algorithms can also be applied. Here we will consider the following:
      i. a greedy constructive algorithm,
      ii. a greedy constructive + a local search procedure,
      iii. GRASP as a meta-heuristic algorithm. You can reuse the local search procedure that you developed in the previous step.

      Design the three algorithms and implement them in the programming language you prefer.
   (d) Tuning of parameters and instance generation:

      Let us define the *load* of an instance as $\frac{n}{t \times r}$.
      i. Implement an instance generator that generates random instances for given values of n, t, r.
      ii. Tune the α parameter of the GRASP constructive phase with a set of randomly generated instances of a large enough size.

iii. Study the time that CPLEX needs to solve a set of mid-size instances: Generate instances with the same size but different load and solve them. The minimum time to solve these instances has to be at least 10 min. Select the load l that takes the maximum time to solve.

iv. Generate problem instances with increasingly larger size but with the same load l found in the previous step. Solving each instance with CPLEX should take from 1 to 30 min.

(e) Compare the performance of solving the model with CPLEX with applying the heuristic algorithms, both in terms of computation time and of quality of the solutions as a function of the size of the instances.

(f) Prepare a report and a presentation of your work on the project

2. **Report:**

Prepare a report (8-10 pages) in PDF format including:

- The formal problem statement.
- The integer linear programming model, with a definition and a short description of the variables, the objective function and the constraints. Do not include OPL code in the document, but rather their mathematical formulation.
- For the meta-heuristics, the pseudo-code of your constructive, local search, and GRASP algorithms, including equations for describing the greedy cost function(s) and the RCL.
- Tables or graphs with the tuning of parameters and instance generation.
- Tables or graphs with the comparative results.

Together with the report, you should also give all sources (OPL code, programs of the meta-heuristics, instance generator, etc.) and instructions on how to use them, so that results can be easily reproduced.

3. **Presentation:**

You are expected to make a presentation of your work (7-10 minutes long) at the end of the course.

The slots of Friday 28/05/20 and Monday 31/05/20 will we devoted to these presentations.

Due to constraints related to the pandemic, the presentations on Friday will be face-to-face, while the presentations on Monday will be online[1], at the Google Meet room of the subject:

`https://meet.google.com/zpm-pgbw-yxa`

The schedule will be announced in its due time.

The slides of the presentation in PDF format should be delivered together with the report by **Thursday 27/05/20**.

The idea is that the presentation can contain figures, plots, equations, algorithms, etc. with a very short text that basically helps to understand them. It is expected that you give full explanation of those contents during your presentation. On the other hand, the report should contain that explanation in a well-organized manner as a text.

[1]Depending on the evolution of the pandemic, it could be that eventually both days are face-to-face. If that were the case, you would be notified in advance via Racó.