Algorithmic Methods for Mathematical Models

– COURSE PROJECT –

The Dean of the School of Computer Science of Barceburg has decided that the curriculum of the degree of Computer Science needs to be updated, so as to keep pace with the advances in technology. To that end, it has been decided that a committee should be organized, which should make an initial proposal to be later discussed by the School Board. The participants in this committee have to be chosen from the faculty. The faculty has N members, who are indexed by consecutive numbers: $1, 2, \ldots, N$. Each of these members belongs to one of the D departments of the university (which again will be indexed with numbers $1, 2, \ldots, D$). For each $1 \leq i \leq N$, the department of i will be denoted by d_i. The commission has to be formed in such a way that, for $1 \leq p \leq D$, department p has exactly a number n_p of participants in the commission. Moreover, it is known that some members of the faculty do not work well together. For this reason, an $N \times N$ symmetric matrix m has been computed, where the coefficient m_{ij} is a real number between 0 and 1 meaning the compatibility between i and j (the higher the number, the higher the compatibility). As would be expected, the diagonal of the compatibility matrix consists of 1's. Since having two incompatible participants (that is, with 0 compatibility) in the commission would render it inoperative, this situation should be forbidden. In addition, if there are two participants in the commission i and j such that $0 < m_{ij} < 0.15$, then there must be a third participant in the commission k such that $m_{ik} > 0.85$ and $m_{kj} > 0.85$; that is, if i and j are not totally incompatible but still get on poorly, then there must be some k who gets along well with both and can intermediate in case of conflict. Finally, it is desired that the average compatibility among all pairs of participants in the commission is maximized. The goal is, given an instance of input data, to determine how the committee should be organized.

For example, let us assume that the involved departments are the "*Software Department*" (department 1) and the "*Hardware Department*" (department 2), hence $D = 2$. The commission should be formed by 3 participants from the "*Software Department*" and 3 participants from the "*Hardware Department*", so $n_1 = n_2 = 3$. The faculty consists of $N = 8$ members: $1 \ldots 4$ belong to the "Software Department", and $5 \ldots 8$ belong to the "*Hardware Department*". Hence $d_1 = d_2 = d_3 = d_4 = 1$, and $d_5 = d_6 = d_7 = d_8 = 2$. Finally, the compatibility matrix is:

$$\begin{pmatrix}
1.00 & 0.50 & 0.75 & 0.90 & 0.15 & 0.40 & 1.00 & 0.90 \\
0.50 & 1.00 & 0.00 & 0.00 & 0.60 & 0.80 & 1.00 & 0.00 \\
0.75 & 0.00 & 1.00 & 0.25 & 0.55 & 0.75 & 1.00 & 0.60 \\
0.90 & 0.00 & 0.25 & 1.00 & 0.40 & 0.20 & 1.00 & 0.10 \\
0.15 & 0.60 & 0.55 & 0.40 & 1.00 & 0.15 & 1.00 & 0.15 \\
0.40 & 0.80 & 0.75 & 0.20 & 0.15 & 1.00 & 1.00 & 0.20 \\
1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \\
0.90 & 0.00 & 0.60 & 0.10 & 0.15 & 0.20 & 1.00 & 1.00
\end{pmatrix}$$

Now, a commission consisting of $\{1, 2, 3, 5, 6, 7\}$ would not be valid, because although there are exactly 3 members of each department as needed, for example there is an incompatibility between 2 and 3. On the other hand, a commission consisting of $\{1, 3, 4, 5, 6, 8\}$ satisfies all requirements, and the average compatibility among all pairs of participants is $(0.75 + 0.90 + 0.15 + 0.40 + 0.90 + 0.25 + 0.55 + 0.75 + 0.60 + 0.40 + 0.20 + 0.10 + 0.15 + 0.15 + 0.20)/15 = 0.43$. Still, it is not the optimal solution for this instance: the best commission consists of $\{1, 3, 4, 6, 7, 8\}$, with average compatibility $(0.75+0.90+0.40+1.00+0.90+0.25+0.75+1.00+0.60+0.20+1.00+0.10+1.00+0.20+1.00)/15 = 0.67$.

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.* Given an instance of input to the problem, the value of N is the *size* of the instance.

      i. Implement an instance generator that produces random instances for a given size.

      ii. Tune the α parameter of the GRASP constructive phase with a set of randomly generated instances of large enough size.

      iii. Generate problem instances with increasingly larger size. Solving each instance with CPLEX should take from 1 to 30 min.

   (e) Compare the performance of CPLEX with 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 with the comparative results.

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

3. **Presentation:**

   You are expected to make a presentation of your work at the end of the course (at most 10 minutes long; overtime presentations will be terminated). All group members must participate in the presentation and know all the work in the project. The slots of 12/12/24, 16/12/24 and 19/12/24 will be devoted to these presentations. The schedule will be announced in its due time.

   The slides of the presentation in PDF format should be delivered with the report by **10/12/24**.

   The slides can contain plots, equations, algorithms, ... with a very short text that helps to understand them. You are expected to give a full explanation of those contents in the presentation. On the other hand, the report should contain that explanation in a well-organized way as a text.