ORIE 51350 - Project - Andrea Lodi

We are given a set of n machines ($j = 1, \dots, n$) and a set of m jobs ($i = 1, \dots, m$). Each job is characterized by a processing time $p_i \in \mathcal{R}_+$ and by a memory requirement $r_i \in \mathcal{R}_+$. The processing time and memory requirement of a job do not depend on the machine in which the job is executed. All machines are identical and characterized by a time capacity P and a memory capacity R. We assume that $p_i \leq P$ and $r_i \leq R$ for all $i = 1, \dots, m$.

We want to execute all jobs on the machines in such a way that the number of used machines is minimized.

Questions:

1. Propose a "natural" (descriptive) Integer Linear Programming (ILP) formulation and implement it.
2. Propose an extended formulation of the problem with a variable per feasible job assignments to machines. Implementing such a formulation as well, assuming to be able to enumerate all the feasible job assignments to machines.
3. Solve the ILP formulations of questions 1 and 2 above on the three provided datasets (10 instances each). Collect and report data on

   - average quality of the LP relaxations,
   - average number of branch-and-bound nodes, and
   - average computing times and number of problems solved in either 120 CPU seconds or 300 CPU seconds.

   Based on the above data, compare the two formulations.
4. Assume now that instead of minimizing the number of used machines we want to minimize the completion time of the machine, among those used, that finishes the latest. The completion time for a machine j is given by the sum of the processing times of the jobs executed by machine j. For this variant of the problem,

   (a) Write a "natural" (descriptive) ILP formulation and implement it.

   (b) Solve the ILP formulation on the provided single dataset (10 instances), collect and report data as for point 3 above.
5. **Bonus question**: for the initial problem, the data package provides a single instance in which enumerating all feasible job assignments to machines is not practical. Discuss how to solve the problem and experiment with the associated data.

Requirements and Deliverables:

- All implementations can either be in Julia (using JuMP and Gurobi) or Python (using the `gurobipy` package for example).
- Implementation: provide the Julia (or Python) script(s) or Jupyter notebook(s) containing your code to points 1, 2 and 4a.
- Writeup: a LaTeX PDF file for the problem formulations of the points 1, 2 and 4a above and answers to the questions 3 and 4b.
- Please submit a single zip file containing the implementation and writeup files.