Algorithmic Methods for Mathematical Models

– COURSE PROJECT –

A large software company needs to hire **exactly** wr programmers for a concrete project. For that purpose, it contacts a set $\mathcal{P}$ of programmer providers. Each provider $p \in \mathcal{P}$ offers the following information:

- $available\_workers_p$: number of workers it provides, which is always an even number.
- $cost\_contract_p$: cost to be paid if some of the workers provided by p are hired.
- $country_p$: country of the provider.
- $cost\_worker_p$: cost of each worker.

We want to decide how many workers to choose from each provider in order to minimize the cost, but taking some constraints into account: we cannot hire workers from two providers from the same country and, for each provider, we can only hire zero, all or half of the workers it offers.

However, every provider p offers an additional batch of workers of the same size ($available\_workers_p$) to be hired. This batch can only be used if all the original workers offered by p are hired. The main difference is that, from this second batch of workers, we can choose any number, not necessarily half or all of them.

Finally, we want to take into account the taxes to be paid in each of the countries we hire workers from. These taxes have to be added to the cost caused by hiring workers. All countries have the same tax policy: for the first 5 workers hired, the tax is $cost\_1$ per worker. For the next 5 workers hired, it is $cost\_2$ per worker. For the remaining ones, it is $cost\_3$ per worker. It will always hold that $cost\_1 < cost\_2 < cost_3$. For example, if $cost\_1 = 2$, $cost_2 = 10$, $cost_3 = 20$ and the company hires 8 workers from a certain provider, it will pay to the corresponding country $5 \cdot 2 + 3 \cdot 10$. If it hires 12, it will pay $5 \cdot 2 + 5 \cdot 10 + 2 \cdot 20$. And if it only hires 3 it will pay $3 \cdot 2$. Note that the number of workers to be considered is not the total, but only the ones assigned to the provider.

1. **Work to be done:**
   - Formally state the problem
   - Devise an integer linear programming model for the optimization problem and implement it in OPL
   - Because of the complexity of the optimization problem, heuristic algorithms are needed. We are considering both constructive and local search procedures, as well as GRASP as a meta-heuristic algorithm. Implement them in the programming language you prefer.
   - Compare the performance of solving the model and the heuristics in terms of computation time and quality of the solutions. To that end, generate increasingly larger problem instances until solving takes around 1 hour.
   - Compare the performance of the two heuristics in terms of solving time and quality of the solution for even larger problem instances.

2. **Report**

   Prepare a report (8-10 pages) including:

   - Problem statement.
   - Integer lineal model, including the definition of the sets and parameters, the model itself and a short description of the objective function and every constraint. Do not include OPL code, but rather its mathematical formulation.
   - For the heuristics, the pseudo-code of constructive, local search, and GRASP algorithms, including equations for describing the greedy cost function(s) and the RCL.
   - Comparative results.
   - Together with the report, you should also provide all sources and instructions on how to use them, so that results can be easily reproduced. If you implemented an instance generator, please provide it as well.