Algorithmic Methods for Mathematical Models

– COURSE PROJECT –

A bus company has a set $\mathcal{S}$ of bus services to operate. For that purpose it can use a set $\mathcal{D}$ of drivers and a set $\mathcal{B}$ of buses. The goal is to assign one bus and one driver to each service, while satisfying some constraints. As expected, a driver (and also a bus) can operate multiple services.

For each bus service we know its starting time, its duration in minutes and kilometers, and the number of passengers to be transported in the service. Each bus service starts in the headquarters of the company and finishes there as well. For each bus in $b \in \mathcal{B}$ we know its capacity cap_b (i.e., number of passengers it can transport), and the cost in euros per minute ($euros\_min_b$) and per kilometer ($euros\_km_b$) for using that bus. Finally, for each driver $d \ \in \mathcal{D}$ we know the maximum number of minutes max_d she can work.

We need to help the bus company to decide which bus and driver is assigned to each service. However, not any assignment is valid, since services should be operated by buses with enough capacity; the same bus or driver cannot serve two services that overlap in time; and we should respect the maximum number of working minutes for each driver (e.g. if a driver can work at most 6 minutes, it cannot operate 3 services with durations 4, 2 and 1 minutes). Additionally, we can use at most $maxBuses$ buses.

Among all possible solutions, we want the one with minimum cost. Apart from the cost of using the buses, the company pays each driver CBM euros for the first BM minutes she works, and pays CEM euros for the remaining minutes (if any). For example, if $BM = 200$, $CBM = 0.5$ and $CEM = 0.8$ and a driver works 300 minutes, she will be paid $200 * 0.5 + 100 * 0.8$ euros. If she works 150 minutes, she will be paid $150 * 0.5$ euros. We can assume that always $CEM > CBM$.

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.