Algorithmic Methods for Mathematical Models (AMMM)

Lab Session 2 – Mixed Integer Linear Programs

In this second session we are going to continue with our example of assigning tasks to computers in a datacenter.

As we saw when solving the *P1* model, tasks could be served in several CPUs provided that the entire processing load was served. However, this is not a realistic scenario, since it would involve many memory transfers among CPUs.

Hence, in the new problem *P2*, we want to ensure that a task is executed in just one CPU. To do this, integer, specifically binary, variables need to be introduced, and thus we are dealing with an Integer Linear Program (ILP), or more exactly with a Mixed Integer Linear Program (MILP) since there will be some remaining continuous variables.

1. Problem statement

The *P2* problem can be formally stated as follows:

Given:

- The set T of tasks. For each task t the amount of resources requested r_t is specified.
- The set C of computers. For each computer c the available capacity r_c is specified.

Find the assignment of tasks to computers subject to the following constraints:

- Each task is assigned to exactly one computer.
- The capacity of each computer cannot be exceeded.

with the *objective* to minimize the highest loaded computer.

2. ILP formulation

The *P2* problem can be modeled as an Integer Linear Program. To this end, the following sets and parameters are defined:

T	Set of tasks, index t.
C	Set of computers, index c.
r_t	Resources requested by task t.
r_c	Available capacity of computer c.

The following decision variables are also defined:

x_{tc} binary. Equal to 1 if task t is served from computer c; 0 otherwise.
z positive real with the ratio of load of the highest loaded computer.

The MILP model for the *P2* problem is the same as *P1* except for variable definitions.

3. Tasks

~~In pairs~~, do the following tasks and prepare a lab report.

a) Implement the *P2* model in OPL and solve it using CPLEX. Use "*boolean*" to define binary variables in OPL.
b) Compare *P1* and *P2* in terms of the value of the optimal solution, solving algorithm, solving time, and number of variables and constraints.
c) Solve *P2* with the following data file, where a new task has been added. Analyze the obtained results and compare them when solving *P1* with the same data.

Table 1 New data file

```
nTasks=5;
nCPUs=3;

rt=[261.27 560.89 310.51 105.80 344.7];
rc=[505.67 503.68 701.78];
```

d) Modify the *P2* model to allow rejecting tasks, i.e. some tasks might not be processed. To this aim, consider a new parameter *K* defining the maximum number of tasks that can be rejected (*P2d*). Analyze the obtained results varying the value of *K*.
e) Modify the objective function so as to minimize the amount of not served load (*P2e*).
f) Compare all three models in terms of number of variables, constraints and execution time.