ALGORITHMIC METHODS FOR MATHEMATICAL MODELS - Lab 2

Goktug Cengiz

October 2019

1 Tasks

1. **Implement the model for P2 in OPL and solve it using CPLEX (use the keyword boolean to define binary variables in OPL). Compare the solutions for P1 and P2 in terms of the value of the optimal solutions, solving time and number of variables and constraints(you can find this information in the statistics tab of the ILOG Optimization Studio).**

   Implementation is done.

   dvar boolean x_tc[t in T, c in C];

Country List				
	Optimal Solution	Solving Time	N. variables	N.constraints
P1	0.723773179127243		T * C + 1	3
P2	0.799239077773661		T * C + 1	3

   Both problems have the same number of variables and constraints. It seems that in P1 the solution optimal solution is better.

2. **Solve P2 with the following data file, where a new task has been added. Give a short explanation as to why the problem is now unsatisfiable.**

   This problem is unsatisfiable, we cannot assign all the tasks to these CPUs, they do not have enough capacity.

3. **Use the model in the attached file P2a.mod to allow rejecting tasks, i.e., we now have the possibility of not processing a given number of tasks. To this aim, consider the new parameter K defining the maximum number of tasks that can be rejected. What is the load of the highest loaded computer for each possible value of K ?**

   Done!

   We have to change the constraint 1 and add a constraint 4:

   // Constraint 1 forall(t in T) sum(c in C) x_tc[t,c] <= 1;

   // Constraint 4 sum(c in C) sum(t in T) x_tc[t,c] >= nTasks-K;

   In the first constraint, we allow that the sum(c in C) x_tc[t,c] can be 0, in that case it will mean that the task t won't served. In the constraint added, we force that the number of tasks to be solved must be at lest the number of tasks minus the number of tasks that can be rejected.

   - With K = 0: results don't exist
   - With K = 1: Value of objective function : 0.641939069223973
   - With K = 2: Value of objective function : 0.516680838282536
   - With K = 3: Value of objective function : 0.372296161190117
   - With K = 4: Value of objective function : 0.150759497278349
   - With K = 5: Value of objective function : 0

   The results coincide with what we think, when there're much tasks that can be not solved, we improve the objective function by solving less tasks, until K = nTasks that means we can not to solve any task, in that case the optimal will be not solving any task.

4. **Run the third model contained in the attached file P2b.mod where the amount of non-served load is minimized and compare all three models in terms of the number of variables, constraints and execution time (taking K= 1 in the case of model P2a.mod).**

   (a) First we have to set K as a variable instead of a parameter:

   dvar int K;

   (b) Second we have to change the objective function and set the variable K as the variable we need to optimize:

   minimize K;

   For the problem in part c the solution obtained is 1, and for the problem in P1 the result obtained is 0, which coincide with our expectations because in part c we can obtain result with assigning 1 task less than the original problem, and for the problem in P1 all the tasks can be assigned.

	Solving Time(seconds)	N. variables	N.constraints
a	0.45	T * C + 1	3
d	0.35	T * C + 2	4
e	0.32	T * C + 2	4