In Class Exercises (3/8)

Creative LP Modeling

Exercise 1: (12 minutes) Spend 8 minutes to brainstorm with your neighbor to complete as much of the following exercise as you can. After that, spend 4 minutes to discuss with another pair of students and complete any missing part.

Consider the production planning LP on the board, with decision variables X, Y, and Z, corresponding to the amount produced of three types of products.

a) Change the objective to **maximize the minimum** of the amount produced of each product. (Hint: $\min(X, Y, Z)$ is not a linear objective, but you can still model it as an LP by adding an auxilliary decision variable U, modifying the objective, and adding a few more constraints.)

b) Change the objective to **minimize the variation** among the three variables, that is, minimize the difference between the maximum of X, Y, Z and the minimum of X, Y, Z, subject to an additional constraint that the profit is at least 400. (Hint: you need to define two auxiliary variables, U and L.)

c) Suppose that the three products have sizes 4, 5, and 7 respectively. Write a constraint requiring the **average size** of products produced to be no more than 6.

d) (**"Soft Constraint"**) Suppose that in additional to a capacity of 60 for material 1, it is possible to purchase additional material 1 at a cost of 3/unit. The overal profit is then the profit from sales ($20X + 10Y + 30Z$) minus the cost of additional purchase. How would you modify the objective and the material 1 constraint to model this?

Exercise 2: (5 minutes) Consider the transportation setting from last class (transporting goods from plants A, B to demand centers 1, 2 and 3). Pick one of the above modeling tricks, and **think of a business scenario** in which one would be interested in formulating a LP applying the trick.

Index Notation

Exercise 3: (8 minutes) Data: Let $I = \{A, B\}$, $J = \{1, 2, 3\}$. For every plant $i \in I$ and demand center $j \in J$, let c_{ij} be the unit transportation cost from plant i to demand center j. Let q_j be the capacity at plant i. Let d_j be the demand requirement at center j.

Decision variables: let x_{ij} denote the amount transported from plant i to plant j. (This must be non-negative.)

Formulate the LP from last class (see board) in index notation. Here are the steps.

a) Write the objective in terms of c_{A1}, c_{A2}, etc, instead of the numbers. (Don't use summation notation yet.)

b) Write the above expression using summation notation. (You will need a double summation, over I and over J.)

c) Write the LHS of each constraint using summation notation. The constraints are: Capacity A, Capacity B, Demand 1, Demand 2, and Demand 3.

d) Abbreviate the above using the "for all ..." notation after the equation, to eliminate repetitions. Then complete the formulation by writing out the objective.

Implementing Large LPs in Gurobi

The following code implements the LP from last class using index notation (exercise 3). Write down any questions you have about the code.

```
[5]: # Explicitly constructing a simple production planning LP
     import gurobipy as grb
     import pandas as pd

     # Data
     I=['A','B'] # plants
     J=[1,2,3] # demand centers
     q=pd.Series([20,15],index=I)
     d=pd.Series([10,10,10],index=J)
     c=pd.DataFrame([[3,7,5],[5,3,3]],index=I,columns=J)

     # Build model
     mod=grb.Model()
     x={}
     for i in I:
         for j in J:
             x[i,j]=mod.addVar(lb=0,name='x[{0},{1}]'.format(i,j))
     for i in I:
         mod.addConstr(grb.quicksum(x[i,j] for j in J)<=q.loc[i],name='Capacity {0}'.format(i))
     for j in J:
         mod.addConstr(grb.quicksum(x[i,j] for i in I)>=d.loc[j],name='Demand {0}'.format(j))
     mod.setObjective(grb.quicksum(c.loc[i,j]*x[i,j] for i in I for j in J),sense=grb.GRB.MINIMIZE)

     # Solve and print output
     mod.optimize()

     print('Optimal objective: {0:.2f}'.format(mod.ObjVal))
     print('Optimal solution')
     for var in mod.getVars():
         print('\t{0}= {1:.2f}'.format(var.VarName,var.x))
     print('\nShadow prices')
     for constr in mod.getConstrs():
         print('\t{0}: {1:.2f}'.format(constr.ConstrName,constr.PI))
```

```
Optimal objective: 100.00
Optimal solution
        x[A,1]= 10.00
        x[A,2]= 0.00
        x[A,3]= 5.00
        x[B,1]= 0.00
        x[B,2]= 10.00
        x[B,3]= 5.00

Shadow prices
        Capacity A: 0.00
        Capacity B: -2.00
        Demand 1: 3.00
        Demand 2: 5.00
        Demand 3: 5.00
```