LP Modeling (3/8)

Learning Objectives:

- Formulating linear constraints to encapsulate complex relationships. (Analyze)
- Using index notation to express large linear programs. (Analyze)
- Translating a LP formulation that uses index notation to Gurobi code. (Code)

Clarifying Shadow Prices

You do not need to calculate shadow prices by hand. They are automatically computed by the LP solver. The formula from last class are to help you understand what the number represents.

For each shadow price, the **Allowable Range** (obtained in Gurobi by `c.SARHSLow` and `c.SARHSUp`, where `c` is the constraint object), tells you what is the minimum and maximum Right Hand Side (RHS) of the constraint in which the shadow price is valid. Within this range, if the shadow price is p, a δ change in the RHS of the constraint results in a $p\delta$ change in the value of the objective. For example, if the constraint is

$$\text{Material 1:} \qquad 4X + Y <= 60$$

And the shadow price is 3.33 and the allowable range is $[48, 120]$, then a 10 unit increase in results in an increase in the optimal objective value by 33.3 (holding all other constraints fixed). A 10 unit decrease results in a decrease of 33.3. We do not know what happens with a 70 unit increase, because $60 + 70 = 130$ lies outside the allowable range.

The allowable range is calculated by the LP solver based on what the RHS of the constraint must be before the set of binding constraints changes. When a previously non-binding constraint becomes binding, or when a previously binding constraint becomes non-binding, the shadow price changes.

Creative LP Modeling

Suppose we have the production planning LP with 3 decision variables X, Y, Z, corresponding to the amount produced of each of the three types of products.

$$\begin{aligned}
\text{maximize} \quad & 20X + 10Y + 30Z \\
\text{subject to:} \quad & \\
\text{(Material 1)} \quad & 4X + Y + 2Z \leq 60 \\
\text{(Material 2)} \quad & 2Y + Z \leq 48 \\
\text{(Labor)} \quad & X + Y + Z \leq 30 \\
\text{(Non-negativity)} \quad & X, Y, Z \geq 0
\end{aligned}$$

The above LP maximizes profit (20,10,and 30 are the profit from selling each of unit of X, Y, and Z), subject to not using more material than available. Product X requires 4 units of material 1, and 1 hour of labor, etc. In this section, we illustrate modeling tricks to modify the above LP to model other scenarios, using only linear constraints and objectives.

Maximizing the minimum of certain decision variables

Suppose X, Y, Z are three ingredients of a meal, and we need each meal to be balanced, meaning that each meal contains 1 unit of each product. Suppose we want to maximize the number of balanced meals produced, instead of maximizing the profit as before. In other words, we want to maximize the non-linear function $\min(X, Y, Z)$, subject to the material and labor constraints as before. To do this, we define a new decision variable L denoting the number of

balanced meals produced. (L stands for "lower-bound," as it must lower bound the three other decision variables.) This can be achieved as follows.

$$\begin{aligned}
\text{maximize} \quad & L \\
\text{subject to:} \quad & \\
\text{(Lower bound)} \quad & X, Y, Z \geq L \\
\text{(Material 1)} \quad & 4X + Y + 2Z \leq 60 \\
\text{(Material 2)} \quad & 2Y + Z \leq 48 \\
\text{(Labor)} \quad & X + Y + Z \leq 30 \\
\text{(Non-negativity)} \quad & X, Y, Z \geq 0
\end{aligned}$$

Here, the "Lower bound" constraint is 3 constraints: $X \geq L$, $Y \geq L$, $Z \geq L$. These three constraints make sure that the number of balanced meals is not more than the number of each of the three necessary ingredients. This recipe essentially changes the previous LP to maximize the non-linear objective $\min(X, Y, Z)$, but do so while staying within what's allowed in a LP. The same idea can be used to minimize the maximum of a set of variables.

Minimizing the variation between certain decision variables

Suppose we want to make X, Y, Z as equal as possible, subject to making more profit than 400. In other words, we want to express as a LP the non-linear program:

$$\begin{aligned}
\text{minimize} \quad & \max(X, Y, Z) - \min(X, Y, Z) \\
\text{subject to:} \quad & \\
\text{(Profit bound)} \quad & 20X + 10Y + 30Z \geq 400 \\
\text{(Material 1)} \quad & 4X + Y + 2Z \leq 60 \\
\text{(Material 2)} \quad & 2Y + Z \leq 48 \\
\text{(Labor)} \quad & X + Y + Z \leq 30 \\
\text{(Non-negativity)} \quad & X, Y, Z \geq 0
\end{aligned}$$

To transform this into a linear program, we use a similar trick as above, with two auxilliary variables U and L, which stands for an "upper bound" and "lower bound" of the two variables. The objective becomes to minimize the difference between the upper and lower bounds, subject to these being valid bounds, as well as the other constraints above.

$$\begin{aligned}
\text{minimize} \quad & U - L \\
\text{subject to:} \quad & \\
\text{(Upper bound)} \quad & X, Y, Z \leq U \\
\text{(Lower bound)} \quad & X, Y, Z \geq L \\
\text{(Profit bound)} \quad & 20X + 10Y + 30Z \geq 400 \\
\text{(Material 1)} \quad & 4X + Y + 2Z \leq 60 \\
\text{(Material 2)} \quad & 2Y + Z \leq 48 \\
\text{(Labor)} \quad & X + Y + Z \leq 30 \\
\text{(Non-negativity)} \quad & X, Y, Z \geq 0
\end{aligned}$$

Constraints on the ratios of decision variables

Suppose that the three products have environmental hazard ratings of 4, 5 and 7 respectively, but a government regulation says that in order to qualify for a certain tax break, we need the

average environmental hazard rating of products produced to be at most 6. In other words, we need

$$\frac{4X + 5Y + 7Z}{X + Y + Z} \leq 6.$$

This can be transformed to a linear constraint by multiplying out the denominator, so

$$4X + 5Y + 7Z \leq 6(X + Y + Z).$$

This trick can be applied whenever we guarantee that the denominator is non-negative, which is true in this case since $X, Y, Z \geq 0$ by assumption. After simplifying the above by bringing the variables to the left hand side (LHS), the complete LP is given below.

$$\begin{aligned}
\text{maximize} \quad & 20X + 10Y + 30Z \\
\text{subject to:} \quad & \\
\text{(Ratio constraint)} \quad & -2X - Y + Z \leq 0 \\
\text{(Material 1)} \quad & 4X + Y + 2Z \leq 60 \\
\text{(Material 2)} \quad & 2Y + Z \leq 48 \\
\text{(Labor)} \quad & X + Y + Z \leq 30 \\
\text{(Non-negativity)} \quad & X, Y, Z \geq 0
\end{aligned}$$

"Soft constraints"

Suppose that is possible to purchase additional material 1 and material 2 at a cost of \$3/unit and \$5/unit, and additional labor at a cost of \$8/hour. Let A_1, A_2 and A_3 be the additional amount purchased of the three resources, the following linear program modify the three constraints to allow for additional resources added, while penalizing the objective by subtracting the cost of the additional purchases. This essentially makes the constraint not a "hard" one that must be satisfied, but allows violations of the constraint at a certain cost.

$$\begin{aligned}
\text{maximize} \quad & 20X + 10Y + 30Z - 3A_1 - 5A_2 - 8A_3 \\
\text{subject to:} \quad & \\
\text{(Material 1)} \quad & 4X + Y + 2Z \leq 60 + A_1 \\
\text{(Material 2)} \quad & 2Y + Z \leq 48 + A_2 \\
\text{(Labor)} \quad & X + Y + Z \leq 30 + A_3 \\
\text{(Non-negativity)} \quad & X, Y, Z, A_1, A_2, A_3 \geq 0
\end{aligned}$$

Index Notation

In this section, we formulate the transportation LP from last class using index notation: There are 2 production plants, A and B, with capacities 20 and 15 respectively. There are 3 demand centers, 1, 2, 3, with demand of 10 each. The cost of transporting each unit of good from each plant to each demand center is shown below.

	1	2	3
A	3	7	5
B	5	3	3

The following LP minimizes total transportation cost subject to satisfying demand at all three centers and not exceeding the capacity of each plant. In this LP, x_{A1} is the amount to be shipped from plant A to region 1, x_{A2} is from plant B to region 2, etc.

$$
\begin{aligned}
\text{minimize:} \quad & 3x_{A1} + 7x_{A2} + 5x_{A3} + 5x_{B1} + 3x_{B2} + 3x_{B3} \\
\text{subject to:} & \\
\text{(Capacity A)} \quad & x_{A1} + x_{A2} + x_{A3} \leq 20 \\
\text{(Capacity B)} \quad & x_{B1} + x_{B2} + x_{B3} \leq 15 \\
\text{(Demand 1)} \quad & x_{A1} + x_{B1} \geq 10 \\
\text{(Demand 2)} \quad & x_{A2} + x_{B2} \geq 10 \\
\text{(Demand 3)} \quad & x_{A3} + x_{B3} \geq 10 \\
\text{(Non-negativity)} \quad & x_{A1}, x_{A2}, x_{A3}, x_{B1}, x_{B2}, x_{B3} \geq 0
\end{aligned}
$$

Step 1: Represent all data and decision variables in index notation

Indices: $I = \{A, B\}$ is the set of plants, $J = \{1, 2, 3\}$ is the set of demand centers.

Data: 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_i 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.)

As an illustration, the above LP is now,

$$
\begin{aligned}
\text{minimize:} \quad & c_{A1}x_{A1} + c_{A2}x_{A2} + c_{A3}x_{A3} + c_{B1}x_{B1} + c_{B2}x_{B2} + c_{B3}x_{B3} \\
\text{subject to:} & \\
\text{(Capacity A)} \quad & x_{A1} + x_{A2} + x_{A3} \leq q_A \\
\text{(Capacity B)} \quad & x_{B1} + x_{B2} + x_{B3} \leq q_B \\
\text{(Demand 1)} \quad & x_{A1} + x_{B1} \geq d_1 \\
\text{(Demand 2)} \quad & x_{A2} + x_{B2} \geq d_2 \\
\text{(Demand 3)} \quad & x_{A3} + x_{B3} \geq d_3 \\
\text{(Non-negativity)} \quad & x_{A1}, x_{A2}, x_{A3}, x_{B1}, x_{B2}, x_{B3} \geq 0
\end{aligned}
$$

Step 2: Verbally summarize the objective and constraints in terms of indices

Example of verbal summary:

- Objective: summing the product of unit transportation cost (c_{ij}) and amount transported (x_{ij})
- There is a capacity constraint **for each plant in** I. For each plant, the left hand side (LHS) is the total shipment out of this plant. This is no more than the right hand side (RHS), which is the capacity of the plant
- There is a demand constraint **for each demand center in** J. For each demand center, the LHS is total shipment to this demand center. This is at least the RHS, which is the demand requirement at the center.

Step 3: Translate the verbal description into summation notation

When writing summation notation, we have to note two things:

- **What is inside the summation.** (i.e., for the objective, it is the cost multiplied by the amount transported from each demand center to each plant.)

- **What index is being summed over, and what index is fixed** (i.e., for the objective, we are summing over both index sets: all plants and all demand centers. For the capacity constraint for a plant, we are only summing over the demand centers, but the plant is fixed.)

The following is the mathematical translation of the verbal summary:

- Objective:
$$\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}.$$
(Summing cost times amount transported across all plants $i \in I$ and all demand centers $j \in J$.) The notation $\in$ denotes "in" the set.
- For each plant $i \in I$, the capacity constraint is
$$\sum_{j \in J} x_{ij} \leq q_i.$$
The LHS is the shipment out of the plant i across all demand centers $j \in J$. The RHS is the capacity of the plant.
- For each demand center $j \in J$, the demand constraint is
$$\sum_{i \in I} x_{ij} \geq d_j.$$
The LHS is the shipment into the demand center j across all plants $i \in I$. The RHS is the demand requirement of the center.

Altogether, the LP formulation in index notation is as follows.
Decision variables: Amount shipped x_{ij} from each plants $i \in I$ to each demand center $j \in J$.

$$\begin{aligned}
\text{Minimize} \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & & \\
\text{s.t.} \quad & & & \\
\text{(Capacity)} \quad & \sum_{j \in J} x_{ij} \leq q_i & & \text{for each plant } i \in I \\
\text{(Demand)} \quad & \sum_{i \in I} x_{ij} \geq d_j & & \text{for each demand center } j \in J \\
\text{(Non-negativity)} \quad & x_{ij} \geq 0 & & \text{for each } i \in I \text{ and } j \in J
\end{aligned}$$

The "for every" on the extreme right shows that the given constraint is repeated. In other words, there is one capacity constraint for each element of I. So if there are 5 plants, there are 5 capacity constraints.

Implementing in Gurobi

The code to implement the above as a Gurobi Model is given at the end of this note.

Before building the model, we store the data in pandas Series and DataFrames. See end of reference document for Python constructs from Exam 1 for explanation of the `pd.Series` and `pd.DataFrame` commands for creating Series and DataFrames, and the `.loc` indexing command for Series and DataFrame objects.

The main action happens between the comment `# Build model` and `# Solve and print output`. Notice that there is a one to one correspondence between the code and the above LP formulation.

For example, the definition of the decision variables as well as the non-negativity constraints correspond to the code that adds a decision variable for every i and j, setting a lower bound of zero (`lb=0`).

```
x={}
for i in I:
    for j in J:
        x[i,j]=mod.addVar(lb=0)
```

The objective

$$\text{Minimize} \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

corresponds to the code

```
mod.setObjective(sum(c.loc[i,j]*x[i,j] for i in I for j in J),sense=grb.GRB.MINIMIZE)
```

The capacity constraints

$$\sum_{j \in J} x_{ij} \leq q_i \qquad \text{for each plant } i \in I$$

correspond to the code that adds a constraint to the model for every i,

```
for i in I:
    mod.addConstr(sum(x[i,j] for j in J)<=q.loc[i])
```

Finally, the demand constraints

$$\sum_{i \in I} x_{ij} \geq d_j \qquad \text{for each demand center } j \in J$$

correspond to

```
for j in J:
    mod.addConstr(sum(x[i,j] for i in I)>=d.loc[j])
```

After building the model, we run `mod.optimize()` to calculate the optimal objective, and print the results.

- `mod.ObjVal` is the value of the optimal objective.
- `mod.getVars()` returns a list of variable objects.
- For each variable object `var`, `var.VarName` gives the name specified when defining the object using `mod.addVar`. `var.x` gives the value of the variable.
- `mod.getConstrs()` returns a list of constraint objects.
- For each constraint object `constr`, `constr.ConstrName` gives the name of the constraint, and `constr.PI` gives the shadow price.

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

    # Solve and print output
    mod.setParam('OutputFlag',False)
    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
```