Week 7: Introduction to Linear Optimization

Session 13: Introduction to Second Half of the Course

Decision: What you control.
Objective: A metric for quantifying how good is a decision.
Constraints: What decisions are acceptable vs. unacceptable?

Four Steps of Optimization Modeling

1. English description: write a succinct verbal description of the decision, objective and constraints.
2. Concrete formulation: translate the above into a linear optimization formulation, illustrating with made-up numbers from a toy example.
3. Abstract formulation: identify patterns in the above and rewrite the formulation into one that can be scaled up to arbitrary data, by defining data variables and using index and summation notations.
4. Reusable software: write Python code to take in any input data of a certain format and output the optimal decision.

Illustration of Where we are Headed

Amazon.com is expanding its business by launching a physical store in West LA. As the manager, you need to select which bestsellers to carry at the store's grand opening. The following table provides the list of Top 10 Bestsellers in Literature & Fiction, along with their genres. Note that some bestsellers belong to more than one genre.

Rank \ Genre	Literary	Sci-Fi	Romance	Thriller
1	√			
2		√		√
3			√	√
4	√		√	
5	√			
6			√	
7		√		
8				√
9	√	√		
10			√	

Help the company decide which bestsellers to carry, so as to minimize the number of bestsellers carried, while ensuring that there are at least two bestsellers in each genre.

The above inputs are only for illustrative purposes. In the end, you would create a tool that the company can use to solve the above problem for arbitrary input data.

Step 1. English Description (Weeks 8-9)

Decision: Which bestsellers to carry.
Objective: Minimize the total number of bestsellers carried.
Constraints: For each of the four genres, we need to carry at least two books of that genre. In other words, for each genre,

$$\text{\# of books carried of this genre} \geq 2$$

Step 2. Concrete Formulation (Weeks 8-9)

Decision variables: Let x_i denote whether to carry i, where $i \in \{1, 2, \cdots, 10\}$. (Binary)
Objective:

$$\text{Minimize: } x_1 + x_2 + \cdots + x_{10}.$$

Constraints:

$$\begin{aligned} \text{(Literary)} \quad & x_1 + x_4 + x_5 + x_9 \geq 2 \\ \text{(Sci-Fi)} \quad & x_2 + x_7 + x_9 \geq 2 \\ \text{(Romance)} \quad & x_3 + x_4 + x_6 + x_{10} \geq 2 \\ \text{(Thriller)} \quad & x_2 + x_3 + x_8 \geq 2 \end{aligned}$$

Step 3. Abstract Formulation (Weeks 10-11)

Data:

- B: the set of books.
- G: the set of genres.
- B_g: the set of books of genre g.
- q_g: how many books we need of genre g.

Decision Variables: Let x_b deibite whether to carry book b. (Binary)
Objective and constraints:

$$\begin{aligned} \text{Minimize:} \quad & \sum_{b \in B} x_b \\ \text{subject to:} \quad & \\ \text{(Enough books in genre)} \quad & \sum_{b \in B_g} x_b \geq q_g \quad \text{for each genre } g \in G. \end{aligned}$$

```
[2]: # Corresponding Python code
     B=range(1,11)
     G=['Literary','Sci-Fi','Romance','Thriller']
     booksInGenre={'Literary':[1,4,5,9],'Sci-Fi':[2,7,9],'Romance':[3,4,6,10],'Thriller':[2,3,8]}
     q={'Literary':2,'Sci-Fi':2,'Romance':2,'Thriller':2}
     from gurobipy import Model,GRB
     mod=Model()
     x=mod.addVars(B,vtype=GRB.BINARY)
     mod.setObjective(sum(x[b] for b in B))
     for g in G:
         mod.addConstr(sum(x[b] for b in booksInGenre[g])>=q[g])
     mod.setParam('OutputFlag',False)
     mod.optimize()
     print('Minimum # of books:',mod.objval)
     print('Books to include: ',[b for b in B if x[b].x==1])
```

```
Minimum # of books: 4.0
Books to include:  [2, 3, 4, 9]
```

Step 4. Reusable Software (Week 12)

See the two inputs files attached on Blackboard (`07-books-input-1.xlsx` and `07-books-input-2.xlsx`) and the corresponding output files (`07-books-output-1.xlsx` and `07-books-output-2.xlsx`) generated by a Python script that you will be able to write in Week 12.

In-class Exercise

Think of a decision you are interested in optimizing, either from your personal life or from an industry you are interested in. Describe the decision, the objective and the constraints. On a piece of paper or using Excel, sketch out what the input data might look like, as well as the desired output data encoding the optimal decision.

Formulating a Linear Optimization Model

A small factory can make two products, X and Y. The following table summarizes the required inputs to produce each product and the profit of each.

	Product X	Product Y
Steel	4 kg	1 kg
Plastic	0 kg	2 kg
Labor	1 hour	1 hour

Suppose that each unit of X makes a profit of 100 dollars and each unit of Y a profit of 200 dollars. Moreover, the daily supply of steel is 60kg, of plastic is 48 kg and of labor is 30 hours. How should the factory optimize its production plan to maximize profit?

Linear Optimization Formulation

The following is an example of a **Linear Program (LP)**, which is a linear optimization formulation in which all the decision variables are continuous.

Decision Variables:

- X: the amount of product X to produce per day. (Continuous)
- Y: the amount of product Y to produce per day. (Continuous)

Objective:

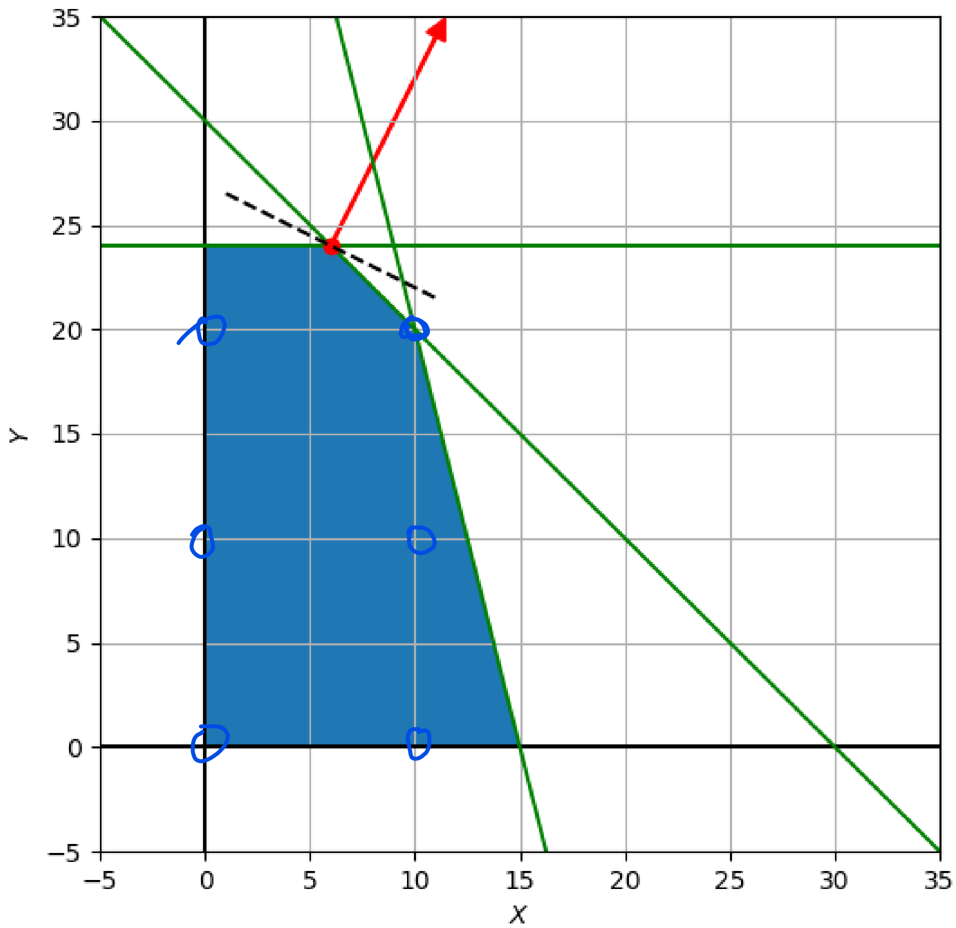
$$\text{Maximize:} \qquad 100X + 200Y$$

Constraints:

$$\begin{aligned} \text{(Steel)} \quad & 4X + Y \leq 60 \\ \text{(Plastic)} \quad & 2Y \leq 48 \\ \text{(Labor)} \quad & X + Y \leq 30 \\ \text{(Non-negativity)} \quad & X, Y \geq 0 \end{aligned}$$

The explanations of constraints, such as (Steel) or (Plastic), are optional.

Geometric Illustration



As can be seen, the optimal solution is $(X, Y) = (6, 24)$, with profit $(100)(6) + (200)(24) = 5400$.

In-class Exercise

Suppose that both X and Y now have to be integer multiples of 10. Mark in the above graph all of the feasible points (X, Y). Identify the optimal (X, Y) under this new business constraint and calculate the optimal profit. (10,20) = 5000

Installing Gurobi

In order to solve linear optimization problems in Python, you need to install a solver, which is a separate piece of software and not part of Python. The best solver is called Gurobi, and is free for academic use. (Once you learn how to use Gurobi, it is straightforward to learn other solvers, as the overall idea is the same.)

Step 1. Request a free academic license on Gurobi.com

You should use your `.edu` email address to register on Gurobi.com as an academic user, using this link: https://www.gurobi.com/downloads/free-academic-license/

After registering and logging in, visit the above link again and request a free individual-level academic license.

Step 2a. Installing Gurobi via conda

In Anaconda Prompt (Windows) or Terminal (Mac), type:

```
conda install -c http://conda.anaconda.org/gurobi gurobi
```

Step 3. Download the License File while on USC Campus or on USC VPN

While you are connected using USC campus Wi-Fi, or when you are at home but connected to USC VPN, log in to the Gurobi website and navigate to your license and follow the instructions there to run the given `grbgetkey` command in Anaconda prompt (in Windows) or a Terminal (in Mac or Linux) followed by your license code. An example is as follows, but you need to replace the long string with your personal license key.

```
grbgetkey ae36ac20-16e6-acd2-f242-4da6e765fa0a
```

(Instructions for setting up USC VPN if you can't come to campus to do this step: https://itservices.usc.edu/vpn/)

Step 4. Test your Installation

Run the code cell from the illustrative example at the beginning of this session, to see if it obtains the desired output.