Contents

Week 7: Introduction to Linear Optimization . . . 1
Session 13: Introduction to Second Half of the Course . . . 1
7.1 Overview . . . 1
7.2 Formulating a Linear Optimization Model . . . 3
7.3 Recognizing Linear Formulations . . . 5
7.4 Installing Gurobi . . . 5
Exercise 7.1: Testing your Gurobi Installation . . . 6
Session 14: Typesetting and Solving a Linear Optimization Model . . . 7
7.5 Solving a Linear Optimization Model using Gurobi . . . 7
Exercise 7.2: Numerically Solving the GTC Production Planning LP . . . 8
7.6 Typesetting a Linear Optimization Model using LaTex . . . 8
Exercise 7.3: Typesetting the GTC Production Planning LP . . . 10
Exercise 7.4: Production Planning . . . 10
Exercise 7.5: Portfolio Planning . . . 11

Week 7: Introduction to Linear Optimization

This week overviews the second half of the course and introduces linear optimization. By the end of the week, you will be able to formulate and solve simple linear optimization problems.

Session 13: Introduction to Second Half of the Course

7.1 Overview

Key Terms

Decision: What you control.

Objective: A metric for quantifying how good is a decision.

Constraints: What decisions are acceptable vs. unacceptable?

The 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	✓			

Rank \ Genre	Literary	Sci-Fi	Romance	Thriller
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 denote whether to carry book b. (Binary)

Objective and constraints:

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

```
[7]: # 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: Thinking in Terms of Optimization

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, sketch out what the input data might look like, as well as the desired format of the output data.

7.2 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:

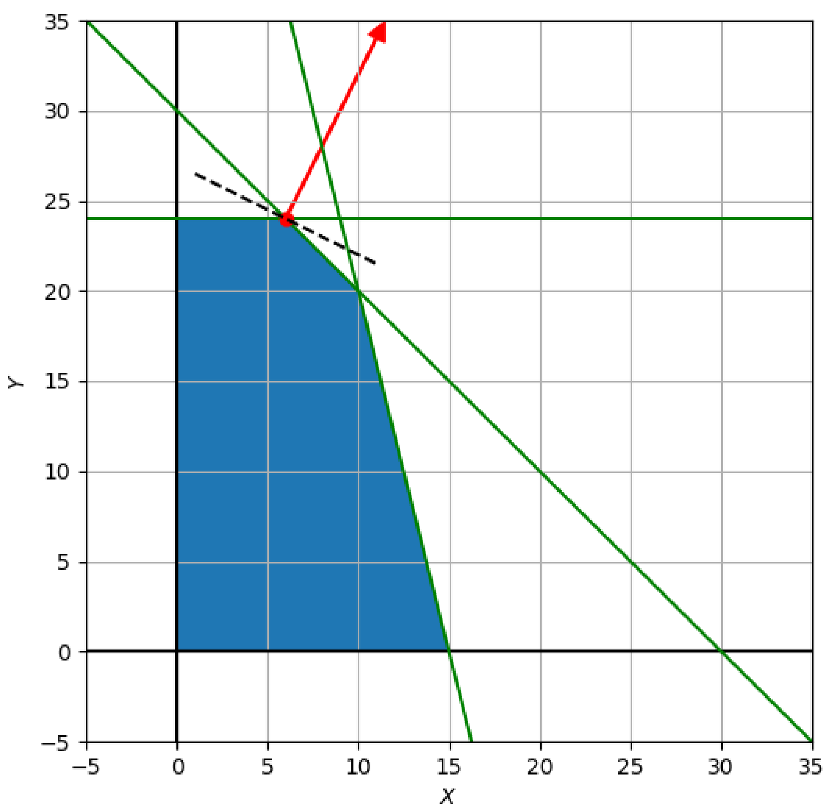
$$\text{Maximize:} \quad 100X + 200Y$$

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: Batch Production

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.

7.3 Recognizing Linear Formulations

In a linear formuation of an optimization problem, the objective must be a linear function of the decision variables. For each constraint, both sides must be a linear function of the decision variables, and the sign must be one of $\geq$, $\leq$ or $=$ ($<$ and $>$ are not allowed).

In-Class Exercise: Is this a Linear Constraint?

In each question, assume that X, Y and Z are input data and U, V and W are decision variables. Determine if each is a linear constraint.

a) $XY \geq ZU$.

b) $XY(U+V) \leq \frac{UX^2+2^Z W}{Y+Z}$.

c) $U(V+X) = 1+V$.

d) $U^2 = V^2$.

e) $UV + WX \leq XZ$.

f) $UX - VZ \leq XYW$.

g) $U > 2$.

h) $3 \leq \frac{U}{V}$.

i) $1 = 0$.

7.4 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. Installing Gurobi via conda

If you did not install Gurobi at the beginning of the semester, then type the following command in Anaconda Prompt (Windows) or Terminal (Mac):

```
conda activate py3k
conda install -c http://conda.anaconda.org/gurobi gurobi
```

If you named your environment something different than `py3k` at the beginning of the semester, then replace `py3k` with the name of your environment.

Step 2. Request a free academic license on Gurobi.com

You should use your `.edu` email address to register on Gurobi.com as an academic user, following the instructions here: https://www.gurobi.com/features/academic-named-user-license/

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 click "Licenses" and generate a "Named-User Academic" license. 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/)

Exercise 7.1: Testing your Gurobi Installation

Run the following code cell from the illustrative example at the beginning of Week 7, to see if it obtains the desired output.

```
[2]: # Code to test your Gurobi installation
     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]
```

Session 14: Typesetting and Solving a Linear Optimization Model

7.5 Solving a Linear Optimization Model using Gurobi

Example from Last Session

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. The following linear program (LP) helps the factory to decide how much of each product to produce so as to maximize profit.

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:

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

[3]:
```
# Solving numerically using Gurobi
from gurobipy import Model, GRB
mod=Model()
X=mod.addVar()
Y=mod.addVar()
mod.setObjective(100*X+200*Y,sense=GRB.MAXIMIZE)
mod.addConstr(4*X+Y<=60)
mod.addConstr(2*Y<=48)
mod.addConstr(X+Y<=30)
mod.optimize()
print(f'\nOptimal daily profit:',mod.objVal)
print(f'Optimal daily production: X={X.x} Y={Y.x}')
```

```
Gurobi Optimizer version 10.0.2 build v10.0.2rc0 (linux64)

CPU model: Intel(R) Xeon(R) CPU E5-1603 v4 @ 2.80GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Optimize a model with 3 rows, 2 columns and 5 nonzeros
Model fingerprint: 0x5d10e453
Coefficient statistics:
  Matrix range     [1e+00, 4e+00]
  Objective range  [1e+02, 2e+02]
```

```
  Bounds range     [0e+00, 0e+00]
  RHS range        [3e+01, 6e+01]
Presolve removed 1 rows and 0 columns
Presolve time: 0.01s
Presolved: 2 rows, 2 columns, 4 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.2000000e+04   8.250000e+00   0.000000e+00      0s
       2    5.4000000e+03   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.01 seconds (0.00 work units)
Optimal objective  5.400000000e+03

Optimal daily profit: 5400.0
Optimal daily production: X=6.0 Y=24.0
```

In-Class Exercise: GTC Production Planning

Formulate a linear program to solve the following problem: The Gemstone Tool Company (GTC) produces wrenches and pliers. Each product is made of steel, and requires using a Molding Machine and an Assembly Machine. The daily availability of each resource, as well as the resources required to produce one units of each tool, are shown below.

	Wrench (1 unit)	Plier (1 unit)	Daily Availability
Steel	1.5 lbs	1.0 lbs	27,000 lbs
Molding Machine	1.0 hours	1.0 hours	21,000 hours
Assembly Machine	0.3 hours	0.5 hours	9,000 hours

There is demand for 16,000 wrenches and 15,000 pliers per day, and the amount produced cannot exceed the demand. Each wrench earns a profit of .10 dollars and each plier earns a profit of .13 dollars. GTC would likes to decide the amount of wrenches and pliers to produce in order to maximize its profit. For simplicity, assume that the amount of each product produced in a day can be fractional.

Exercise 7.2: Numerically Solving the GTC Production Planning LP

Numerically solve the linear program from the in-class exercise using Gurobi. You may follow the "template" code for LP from last session, which is given immediately before the in-class exercise. Ignore the log that Gurobi prints out.

```
Optimal profit: 2505.0
Optimal production plan W=7500.0 P=13500.0
```

7.6 Typesetting a Linear Optimization Model using LaTex

The following example illustrates how to nicely display a linear optimization formulation using LaTex, which is the most widely used method of typesetting mathematics.

```
**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:**
$$\begin{aligned}
\text{(Steel)} && 4X+Y & \le 60 \\
\text{(Plastic)} && 2Y & \le 48 \\
\text{(Labor)} && X+Y & \le 30 \\
\text{(Non-negativity)}&& X,Y & \ge 0 \\
\end{aligned}$$
```

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:

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

Notice that the variables X and Y are in a special font denoting mathematical variables. Moreover, notice that the linear program above is centered and aligned, both at the signs as well as at the constraint labels.

Explanation of above

To render an expression using LaTex, the expression must be enclosed with dollar signs. For example, the expression `$X>0$` is rendered $X > 0$. A single dollar sign is for mathematical expressions within the same line, and double dollar signs are for a standalone mathematical expressions in its own line. Hence, `$$X > 0.$$` is rendered as

$$X > 0.$$

To make the linear program aligned, we not only use the double dollar signs, but also use the `\begin{aligned} \end{aligned}` commands. (Double click the linear program above to see the code.) Within this block of LaTex script,

- `\text{ }` is for displaying the enclosed string as plain text, without the mathematical rendering.
- `\quad` is for creating a horizontal space. `\qquad` is the same as two `\quad`'s.
- `&` is for alignment. The convention is right align before the first `&`, then left align between the first and second `&` of each line, then right align again between the second and third `&` and so on. Hence, to make it right aligned both before and after the alignment character, we use a double `&&` after the `\text{}`. If this is confusing, you can simply copy the above convention (`&&` after the explanation of constraint, and `&` before the sign).
- \\ is for creating a new line. Notice that we end the line early using \\ for "subject to" and "maximize".
- `\le` (less than or equal to) is for $\le$, and `\ge` (greater than or equal to) is for $\ge$. This looks better than $<=$ and $>=$.
- `_` can be used for subscripts, such as a_1. If multiple characters need to be subscripted, you need `_{...}`, such as `a_{12}`: a_{12}.

Unfortunately, it is difficult to debug LaTex within Jupyter notebook. Hence, if there is any error at all, then the LaTex will not render and you will see your original script. When this happens, try to render one line at a time and see which line is causing the error. Common errors include:

- Not having matching braces, { }. Similarly, not having matching dollar signs to begin and end the block of math, or not matching begin{aligned} with end{aligned}.

- Having blank lines in the aligned environment. (Latex in Markdown cells is very fragile)
- Too few or two many alignment characters &, or not ending the line with \\. The last line in the aligned environment does not need a \\.
- Using & outside of the aligned environment.

Exercise 7.3: Typesetting the GTC Production Planning LP

Typeset the GTC Production Planning LP from the In-Class Exercise in a Markdown cell using Latex.

Exercise 7.4: Production Planning

The Magnetron Company manufactures and markets microwave ovens. Currently, the company produces two models: full-size and compact. Production is limited by the amount of labor available in the general assembly and electronic assembly departments, as well as by the demand for each model. Each full-size oven requires 2 hours of general assembly and 2 hours of electronic assembly, whereas each compact oven requires 1 hour of general assembly and 3 hours of electronic assembly. In the current production period, there are 500 hours of general assembly labor available and 800 hours of electronic assembly labor available.

In additional, the company estimates that it can sell at most 220 full-size ovens and 180 compact ovens in the current production period. The earnings contribution per oven is 120 dollars for a full-size oven and 130 dollars for a compact oven. The company would like to find an earnings-maximizing production plan for the current production period.

a) Succintly describe the decision, objective and constraints in English.

Decision:

Objective:

Constraints:

b) Translate the above English description into a concrete formulation of a linear optimization problem.

Decision variables:

Objective:

Constraints:

c) Solve your formulation numerically using Gurobi.

```
Optimal solution
Maximum earnings 40500.0
F=175.0 C=150.0
```

Exercise 7.5: Portfolio Planning

An investor would like to construct an optimal portfolio consisting of five possible funds. (A portfolio consists of a certain amount of money in each fund.) The five funds and their respective fund categories, risk levels, and percentage annual returns are shown below.

Fund	Category	Risk Level	Percentage Annual Return
A	Money Market	1	4.50%
B	Money Market	2	5.62 %
C	Bond	2	6.80%
D	Bond	3	10.15%
E	Aggressive Growth	5	20.60%

The risk level of each fund is rated on a scale of 1 to 5, where 1 is very conservative and 5 is very risky. The investor would like to maximize the total monetary amount earned subject to the following restrictions:

1. The average risk level of the entire investment should not exceed 2.5. (The average here is weighted by the amount of money in each fund. For example, if the entire investment consists of 7500 in C and 1000 in D, then the average risk level is $(7500 \times 2 + 1000 \times 3)/(7500 + 1000) \approx 2.12$.)
2. At least 30% of the money invetsed should be placed in money market funds.
3. At most 2,000 dollars should be invested in the aggressive growth fund.
4. The total amount of initial investment should be between 5,000 and 10,000 dollars (inclusive).

a) Succintly describe the decision, objective and constraints in English.

Decision:

Objective:

Constraints:

b) Translate the above English description into a concrete formulation of a linear optimization problem. **Note that you must transform all non-linear constraints into a linear form (see Section 7.3).**

Decision Variables:

Objective:

Constraints:

c) Solve your formulation numerically using Gurobi.

```
Optimal solution:
Maximum monetary amount earned: 96975.0
A=4500.0 B=0.0 C=0.0 D=3500.0 E=2000.0
```