Mod ule– Linear Programming

Modified/Inspired from Stanford's CS161 by Nayyar Zaidi

Today

- Four things on Agenda:
    - Optimization Problems
    - Linear Programming
    - Gaussian Elimination
    - Simplex Algorithm

Optimization

- Optimization problems are ubiquitous in our daily lives, and each stems from our need to make the right decisions. We already have seen several examples of optimization problems such as knap-sack, minimum spanning trees etc.
- Optimization problems are ubiquitous in our daily lives.
  - Which route to take when commuting to Deakin?
  - How many hours to allocate for SIT320 studies?
  - If you are in retail, you need to decide which products to display? Or, how many of each product should be ordered?
  - Which Advertisement should be shown to a browser (based on cookies) to maximize the click-through rate?
- You can see that in optimization problems, we are required to make a bunch of decisions and we hope those decisions are right or optimal.
- The decisions in an optimization problem often are represented in a mathematical model by the symbols $\mathbf{X_1, X_2, \ldots, X_n}$. also referred as the decision variables (or simply the variables) in the model.

Constraints

- In many optimization problems, we have restrictions, or constraints, that are likely to be placed on the alternatives available to the decision maker.
- For example, when determining the number of products to manufacture, a production manager probably is faced with a limited amount of raw materials and a limited amount of labor.

$$\text{less than or equal to constraint - } f(X_1, X_2, \ldots, X_n) \leq b$$
$$\text{greater than or equal to constraint - } f(X_1, X_2, \ldots, X_n) \geq b$$
$$\text{equal to constraint - } f(X_1, X_2, \ldots, X_n) = b$$

Objective Function

- In any optimization problem, there is some goal or objective that the decision maker considers when deciding which course of action is best.
- The objective in an optimization problem is represented mathematically by an objective function in the general format:

$$\max \textit{ or } \min f(X_1, X_2, \ldots, X_n)$$

Mathematical Formulation

- The goal in optimization is to find the values of the decision variables that maximize (or minimize) the objective function without violating any of the constraints.

$$
\begin{aligned}
\text{max or min} \quad & f(X_1, X_2, \ldots, X_n) \\
\text{subject to} \quad & f_1(X_1, X_2, \ldots, X_n) \le b_1 \\
& f_2(X_1, X_2, \ldots, X_n) \ge b_2 \\
& \vdots \\
& f_m(X_1, X_2, \ldots, X_n) = b_m
\end{aligned}
$$

Linear Programming

- We will discuss a technique called **linear programming** (LP), which involves creating and solving optimization problems with linear objective functions and linear constraints ONLY.

$$
\begin{aligned}
\text{max or min} \quad & c_1X_1 + c_2X_2 + \ldots + c_nX_n \\
\text{subject to} \quad & a_{11}X_1 + a_{12}X_2 + \ldots + a_{1n}X_n \le b_1 \\
& a_{21}X_1 + a_{22}X_2 + \ldots + a_{2n}X_n \ge b_2 \\
& \vdots \\
& a_{m1}X_1 + a_{m2}X_2 + \ldots + a_{mn}X_n = b_m
\end{aligned}
$$

AS $\rightarrow X_1 \quad X_2 \leftarrow$ HL

Tub: Fg, Pumps, TS

Blue Ridge Hot Tubs manufactures and sells two models of hot tubs: the Aqua-Spa and the Hydro-Lux. Howie Jones, the owner and manager of the company, needs to decide how many of each type of hot tub to produce during his next production cycle. Howie buys prefabricated fiberglass hot tub shells from a local supplier and adds the pump and tubing to the shells to create his hot tubs. (This supplier has the capacity to deliver as many hot tub shells as Howie needs.) Howie installs the same type of pump into both hot tubs. He will have only 200 pumps available during his next production cycle. From a manufacturing standpoint, the main difference be- tween the two models of hot tubs is the amount of tubing and labor required. Each Aqua-Spa requires 9 hours of labor and 12 feet of tubing. Each Hydro-Lux requires 6 hours of labor and 16 feet of tubing. Howie expects to have 1,566 production labor hours and 2,880 feet of tubing available during the next production cycle. Howie earns a profit of $350 on each Aqua-Spa he sells and $300 on each Hydro-Lux he sells. He is confident that he can sell all the hot tubs he produces. The question is, how many Aqua-Spas and Hydro-Luxes should Howie produce if he wants to maximize his profits during the next production cycle?

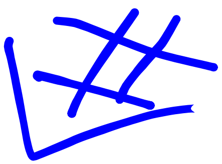
90%

$$\boxed{\begin{aligned} X_1 + X_2 &\le 200 \\ 9X_1 + 6X_2 &\le 1556 \\ 12X_1 + 16X_2 &\le 2880 \\ X_1, X_2 &\ge 0 \end{aligned}}$$

$$\boxed{350X_1 + 300X_2}$$

LP

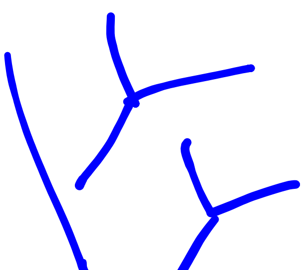
Simplex



X_1^2

$\sqrt{X_1}$

X_1X_2



$$A = LU$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ a_{1} & 0 & 0 & 0 \\ a_{21} & a_{12} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

$$\begin{array}{cc} L & U \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_{11} & 1 & 0 & 0 \\ a_{21} & a_{22} & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix} & \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \end{array}$$

$$y_1 = 1$$

$$a_{11} y_1 + y_2 = 2$$

$$a_{21} y_1 + a_{22} y_3 + y_3 = 3$$

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & 0 & b_{33} & b_{34} \\ 0 & 0 & 0 & b_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$b_{44} x_4 = 4$$

$$Ax = b$$

$$x = A^{-1} b$$

$$O(n^3)$$

$$LUx = b$$

$$Ly = b$$

$$Ux = y$$

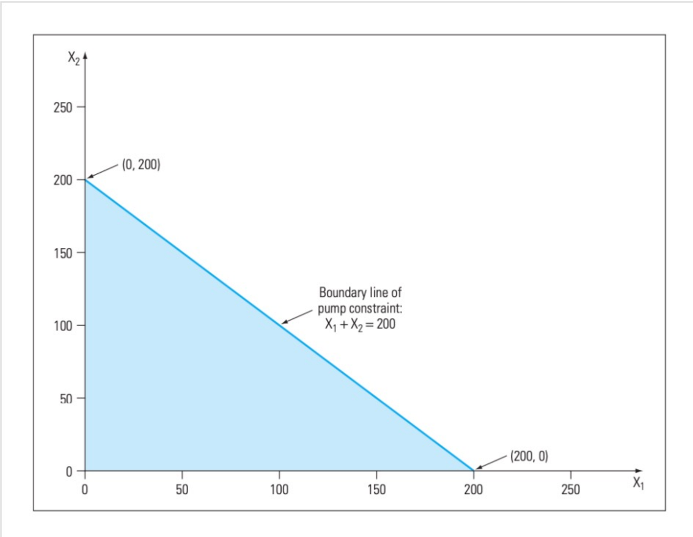
What is a Graphical Approach?

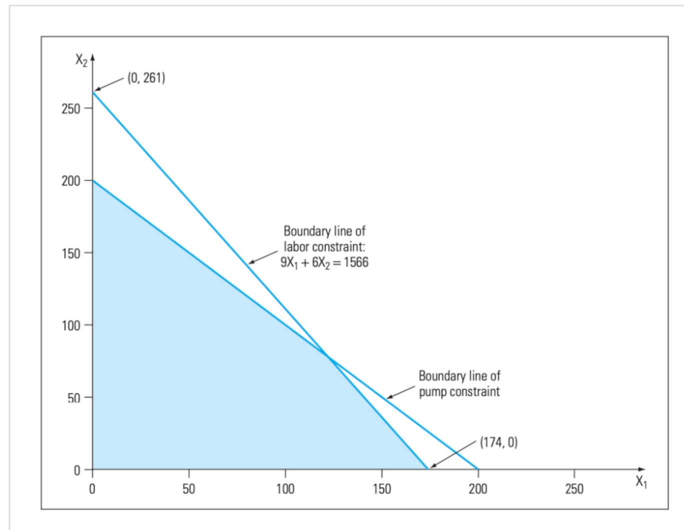
- Not use in practice
  - Only to gain an intuition
  - Of course, you can not plot for more than two decision variables

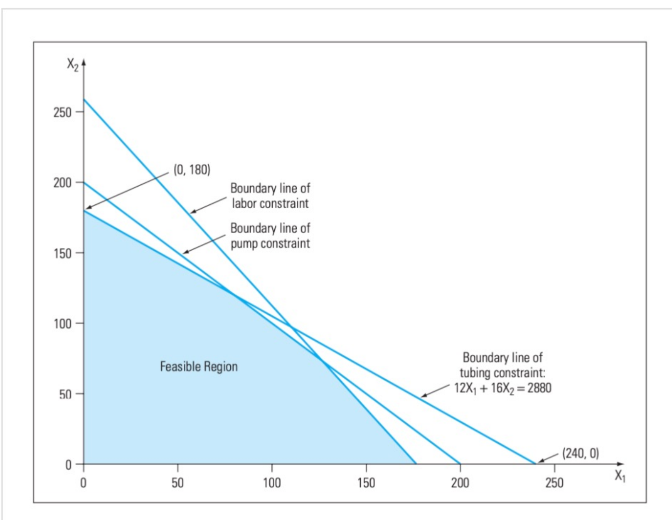
- **If** you have only two decision variables, you are welcome to try this approach, however, other options such as Simplex is recommended.

Example

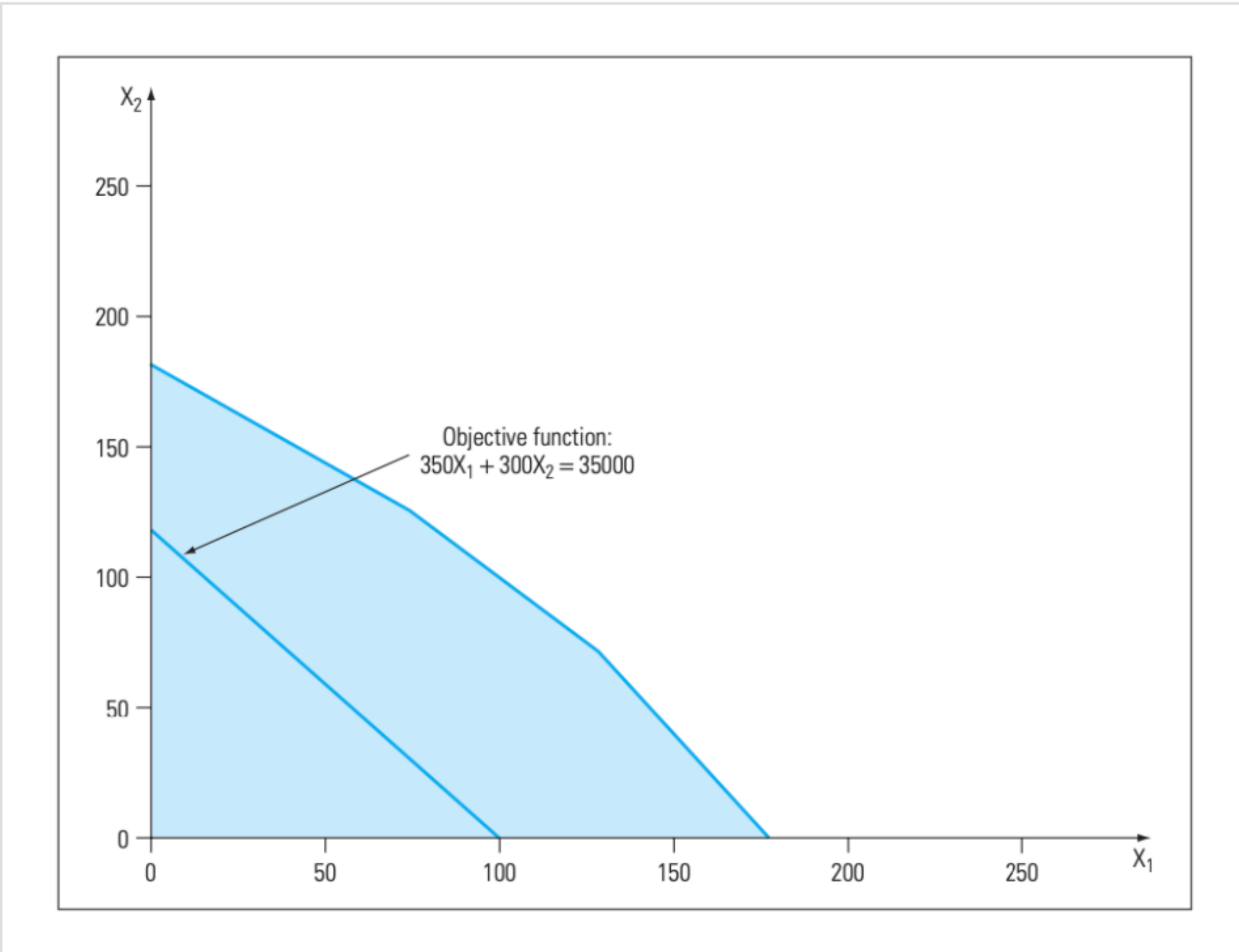
$$
\begin{aligned}
\text{max} \quad & 350X_1 + 300X_2 \\
\text{subject to} \quad & X_1 + X_2 \leq 200 \\
& 9X_1 + 6X_2 \leq 1556 \\
& 12X_1 + 16X_2 \leq 2880 \\
& X_1 \geq 0 \\
& X_2 \geq 0
\end{aligned}
$$







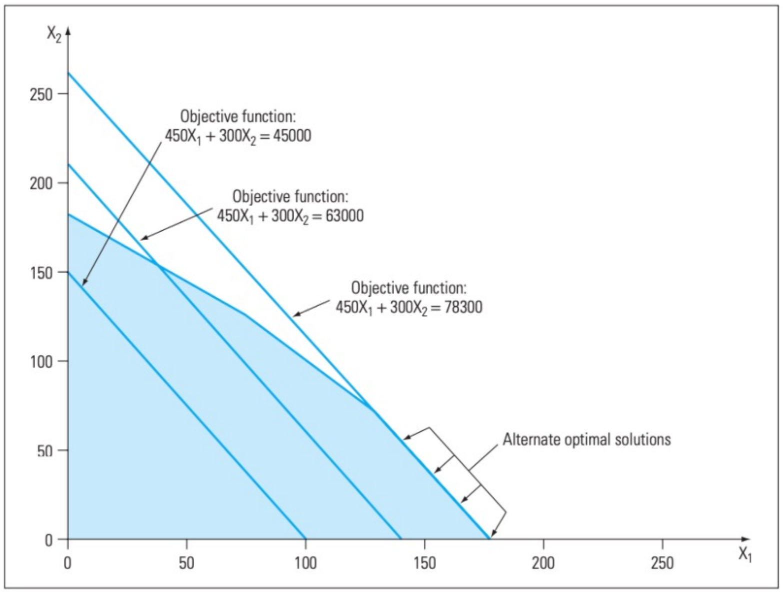
$$
\begin{aligned}
\text{max} \quad & 350X_1 + 300X_2 \\
\text{subject to} \quad & X_1 + X_2 \leq 200 \\
& 9X_1 + 6X_2 \leq 1556 \\
& 12X_1 + 16X_2 \leq 2880 \\
& X_1 \geq 0 \\
& X_2 \geq 0
\end{aligned}
$$



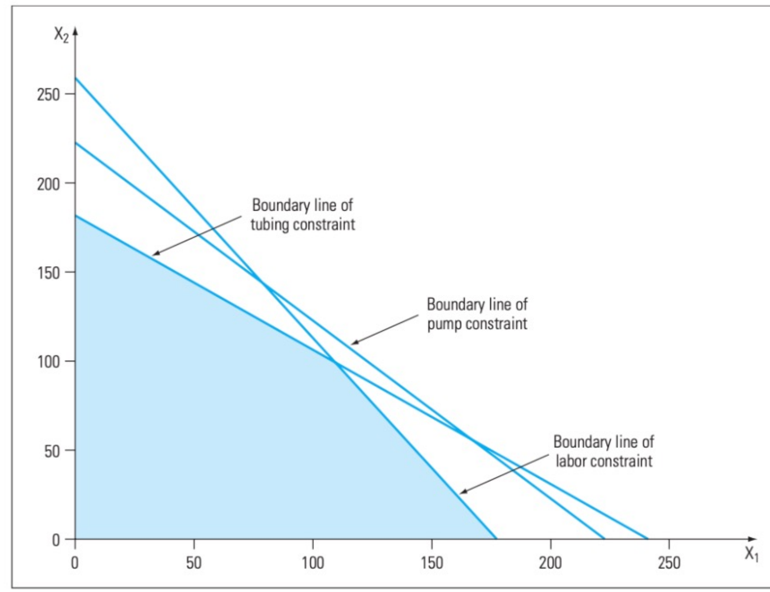
Issues

- Alternative Optimal Solutions
  - More than one optimal solution

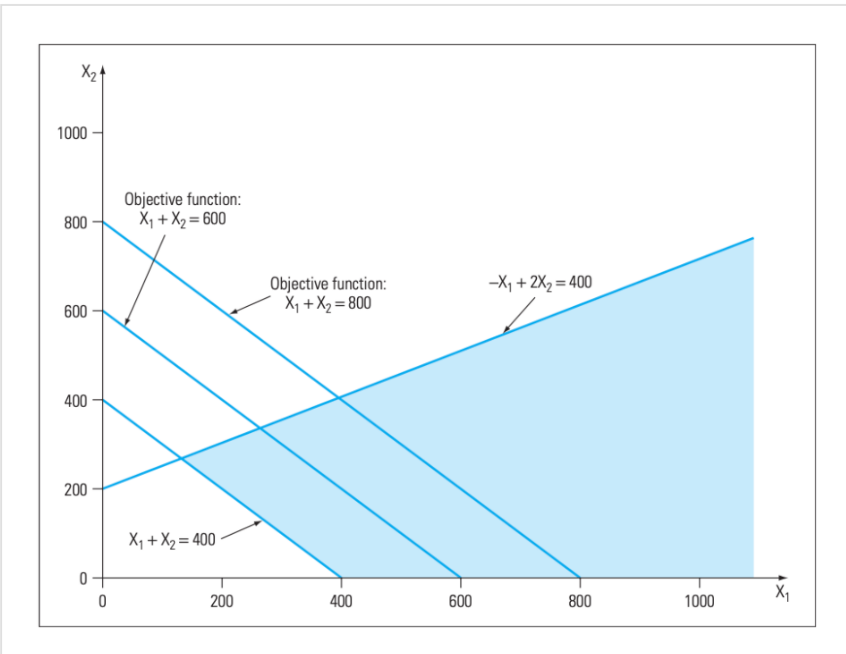
- Redundant constraints!
- Unbounded Solutions
- Infeasability



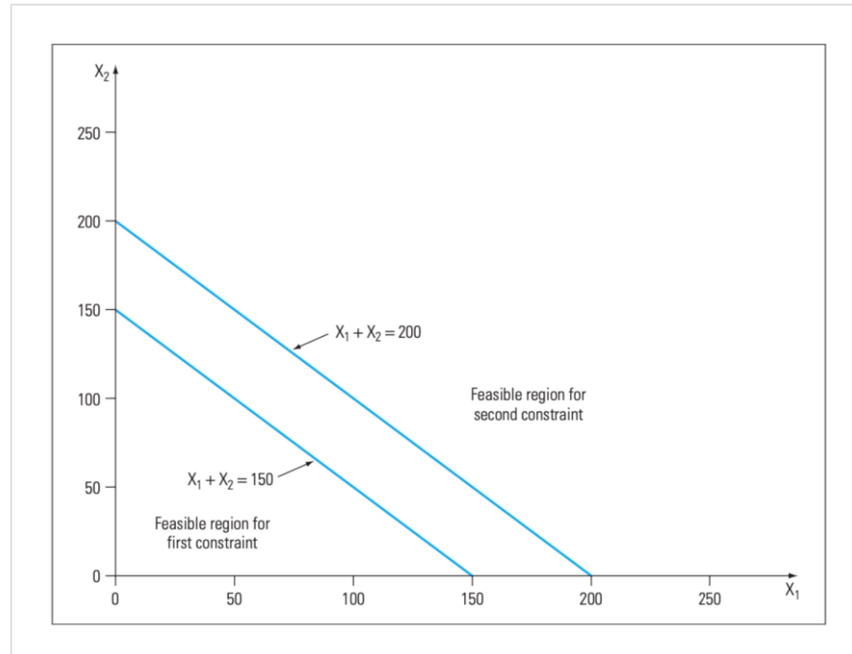
X_2
X_1
Objective function: $450X_1 + 300X_2 = 45000$
Objective function: $450X_1 + 300X_2 = 63000$
Objective function: $450X_1 + 300X_2 = 78300$
Alternate optimal solutions



X_2
X_1
Boundary line of tubing constraint
Boundary line of pump constraint
Boundary line of labor constraint



X_2
1000
800
600
400
200
0
Objective function:
$X_1 + X_2 = 600$
Objective function:
$X_1 + X_2 = 800$
$-X_1 + 2X_2 = 400$
$X_1 + X_2 = 400$
0
200
400
600
800
1000
X_1



X_2
250
200
150
100
50
0
$X_1 + X_2 = 200$
Feasible region for
second constraint
$X_1 + X_2 = 150$
Feasible region for
first constraint
0
50
100
150
200
250
X_1

Standard Form

- Maximize an objective function
  - With all constraints less than or equal to

- E.g.,

$$\text{maximize} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \ \text{ for } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \ \text{ for } j = 1, 2, \ldots, n$$

(Not a) Standard Form

$$\begin{aligned} \text{maximize} \quad & 2X_1 - 3X_2 \\ \text{subject to} \quad & X_1 + X_2 = 7 \\ & X_1 - 2X_2 \leq 4 \\ & X_1 \geq 0 \end{aligned}$$

- Standard Form:

$$\begin{aligned} \text{maximize} \quad & 2X_1 - 3X_2 + 3X_3 \\ \text{subject to} \quad & -X_1 - X_2 + X_3 \leq -7 \\ & X_1 + X_2 - X_3 \leq 7 \\ & X_1 - 2X_2 + 2X_3 \leq 4 \\ & X_1, X_2, X_3 \geq 0 \end{aligned}$$

Slack Form

- To efficiently solve a linear program with the Simplex algorithm, we would like to convert our problem into a form where:.
  - Non-negative constraints are the only inequality constraints
  - All other constraints are equalities
- Solution:
  - We should convert our problem into standard form and then convert that standard form into a slack form.
  - This will be followed by the application of Simplex

Slack Form

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i$$

$$\begin{aligned} \text{maximize} \quad & 2X_1 - 3X_2 + 3X_3 \\ \text{subject to} \quad & -X_1 - X_2 + X_3 \leq -7 \\ & X_1 + X_2 - X_3 \leq 7 \\ & X_1 - 2X_2 + 2X_3 \leq 4 \\ & X_1, X_2, X_3 \geq 0 \end{aligned}$$

$$s = b_i - \sum_{j=1}^{n} a_{ij}x_j$$

$$s \geq 0$$

$$\begin{aligned} \text{maximize} \quad & 2X_1 - 3X_2 + 3X_3 \\ \text{subject to} \quad & -7 + X_1 + X_2 - X_3 = X_4 \\ & 7 - X_1 - X_2 + X_3 = X_5 \\ & 4 - X_1 + 2X_2 - 2X_3 = X_6 \\ & X_1, X_2, X_3, X_4, X_5, X_6 \geq 0 \end{aligned}$$

Slack Form

- Leads to a system of Linear Equations
  - Which can be represented Succinctly

$$Ax = b$$

$$x = A^{-1}b$$

$$\begin{aligned} 2X_1 - 3X_2 + 3X_3 &= z \\ -7 + X_1 + X_2 - X_3 &= X_4 \\ 7 - X_1 - X_2 + X_3 &= X_5 \\ 4 - X_1 + 2X_2 - 2X_3 &= X_6 \\ X_1, X_2, X_3, X_4, X_5, X_6 &\geq 0 \end{aligned}$$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2, \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n. \end{aligned}$$

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

LU Decomposition

- Inverse of a matrix can only be computed for a full-Rank matrix.
- In practice, we can have situation where matrix has a rank less than **n**, therefore, relying on the inverse of matrix for obtaining the solution can be quite risky. Not to mention that inverse has a complexity of $\mathbf{O(n^3)}$
  - LU Decomposition is an alternative

$$A = LU$$

- L is a lower-triangle matrix
- U is an upper-triangle matrix

LU Decomposition

- Algorithm: Solving system of linear equations
  - $Ax = b$
  - If $A = LU$, then
    - $LUx = b$
  - Let $Ux = y$ (output y is actually a vector)
  - Then $Ly = b$
    - Solve of unknown y using Forward Substitution
  - $Ux = y$
    - Solve of unknown x using Bacward Substitution

Algorithm for LU Decomposition

- Gauss Elimination process
  - Check the video in CloudDeakin

Gauss Elimination

$$A = \begin{bmatrix} 2 & 3 & 1 & 5 \\ 6 & 13 & 5 & 19 \\ 2 & 19 & 10 & 23 \\ 4 & 10 & 11 & 31 \end{bmatrix} \simeq \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ a_{11} & 1 & 0 & 0 \\ a_{21} & a_{22} & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & 0 & b_{33} & b_{34} \\ 0 & 0 & 0 & b_{44} \end{bmatrix}}_{U}$$

$R_2 - 3R_1$

6 9 3 15

$R_3 - R_1$

$R_4 - 2R_1$

4 6 2 10

$$A = \begin{bmatrix} 2 & 3 & 1 & 5 \\ 0 & 4 & 2 & 4 \\ 0 & 16 & 9 & 18 \\ 0 & 4 & 9 & 21 \end{bmatrix}$$

$R_3 - 4R_2$

16 8 16

$R_4 - R_2$

$R_4 - 7R_3$

7 14

$$A = \begin{bmatrix} 2 & 3 & 1 & 5 \\ 0 & 4 & 2 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 7 & 17 \end{bmatrix}$$

$$\cancel{A} = U = \begin{bmatrix} 2 & 3 & 1 & 5 \\ 0 & 4 & 2 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 2 & 1 & 7 & 1 \end{bmatrix}$$

LU Decomposition Algorithm

LU-DECOMPOSITION(A)

```
n = A.rows
let L and U be new n × n matrices
initialize U with 0s below the diagonal
initialize L with 1s on the diagonal and 0s above the diagonal
for k = 1 to n
    u_kk = a_kk
    for i = k + 1 to n
        l_ik = a_ik / u_kk        // l_ik holds v_i
        u_ki = a_ki               // u_ki holds w_i^T
    for i = k + 1 to n
        for j = k + 1 to n
            a_ij = a_ij − l_ik u_kj
return L and U
```

Simplex

- System expressed in the following form:

$$
\begin{aligned}
3X_1 + X_2 + 2X_3 &= z \\
X_1 + X_2 + 3X_3 + X_4 &= 30 \\
2X_1 + 2X_2 + 5X_3 + X_5 &= 24 \\
4X_1 + X_2 + 2X_3 + X_6 &= 36 \\
X_1, X_2, X_3, X_4, X_5, X_6 &\geq 0
\end{aligned}
$$

Slack Form

- Leads to a system of Linear Equations
  - Which can be represented Succinctly

- Redundant constraints!
- Unbounded Solutions
- Infeasability