Linear Programming

- **LP** is a general method to solve optimization problem with linear objective function and linear constraints.

- **Diet Problem Example:** Feeding an army.

- **4 menu choices:** Fish, Pizza, Hamburger, Burrito.

- **Each choice has benefits (nutrients) and a cost (calories).**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>D</th>
<th>Calories</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>200</td>
<td>90</td>
<td>100</td>
<td>600</td>
</tr>
<tr>
<td>P</td>
<td>75</td>
<td>80</td>
<td>250</td>
<td>800</td>
</tr>
<tr>
<td>H</td>
<td>275</td>
<td>80</td>
<td>510</td>
<td>550</td>
</tr>
<tr>
<td>B</td>
<td>500</td>
<td>95</td>
<td>200</td>
<td>450</td>
</tr>
<tr>
<td>RDA</td>
<td>2000</td>
<td>300</td>
<td>475</td>
<td></td>
</tr>
</tbody>
</table>

- **Determine least-caloric diet that satisfies RDA of nutrients.**
LP Formulation

• Introduce variables $x_F, x_P, x_H, x_B$ for the amount (units) consumed of each food.

• **Obj. Function** is total calories.

\[
\text{minimize} \quad 600x_F + 800x_P + 550x_H + 450x_B
\]

• Each of the nutrient constraint can be expressed as a **linear function**:

\[
200x_F + 75x_P + 275x_H + 500x_B \geq 2000
\]
\[
90x_F + 80x_P + 80x_H + 95x_B \geq 300
\]
\[
100x_F + 250x_P + 510x_H + 200x_B \geq 475
\]

• Finally, all food quantities consumed should be non-negative:

\[
x_F \geq 0; \quad x_P \geq 0; \quad x_H \geq 0; \quad x_B \geq 0
\]

• This is an example of **Linear Prog.**
Manufacturing Example

- A plant has 3 types of machines, $A, B, C$.

- Choice of 4 products: 1, 2, 3, 4.

- Resource consumption (how much machine time a product uses) and profit per unit.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>1.5</td>
<td>3.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4.2</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>1</td>
<td>3</td>
<td>6.5</td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
<td>2</td>
<td>1.5</td>
<td>3.8</td>
</tr>
<tr>
<td>Avail.</td>
<td>20/wk</td>
<td>30/wk</td>
<td>15/wk</td>
<td></td>
</tr>
</tbody>
</table>

- Machine shop operates 60 hrs/week.

- Determine most profitable product mix.
LP Formulation

- Introduce **variables** $x_1, x_2, x_3, x_4$ for the units of different products manufactured.

- **Obj. Function** is total profit.

  \[
  \text{maximize} \quad 3.5x_1 + 4.2x_2 + 6.5x_3 + 3.8x_4
  \]

- Each of the machines provides a constraint, expressed as a **linear function**:

  \[
  2x_1 + 2x_2 + 0.5x_3 + 1.5x_4 \leq 1200 \\
  0.5x_1 + 2x_2 + x_3 + 2x_4 \leq 1800 \\
  1.5x_1 + x_2 + 3x_3 + 1.5x_4 \leq 900
  \]

- Finally, all product quantities should be **non-negative**:

  \[
  x_1, \ x_2, \ x_3, \ x_4 \geq 0
  \]
Transportation Example

- $m$ supply sites, and $n$ demand sites.
- Supply site $i$ produces $a_i$ units; demand site $j$ requires $b_j$ units.
- $c_{ij}$ cost of shipping one unit from $i$ to $j$.

- Determine optimal shipping cost.
LP Formulation

- Let \( x_{ij} \) be the amount shipped from \( i \) to \( j \). Then, the LP is

\[
\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}
\]

- subject to

\[
\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, 2, \ldots, m
\]

\[
\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n
\]

\[
\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j
\]

\[
x_{ij} \geq 0
\]
General Form of LP

- \(n\) unknowns \(x_1, \ldots, x_n\), and \(m\) constraints.

\[
\max \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n
\]

- subject to

\[
\begin{align*}
a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n & \leq b_1 \\
a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n & \leq b_2 \\
\vdots
\end{align*}
\]

\[
a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n & \leq b_m
\]

\[
x_1, x_2, \ldots, x_n \geq 0
\]

- A company making \(n\) products, each requiring some amount each of \(m\) resources. Item \(i\) has profit \(c_i\).

- \(b_j\) is total resource of type \(j\). \(a_{ij}\) is the amount of resource \(j\) needed by item \(i\).
Matrix Form

- maximize $cx$
- subject to $Ax \leq b$
- $x \geq 0$

- The input data are:
  - the $m \times n$ coefficient matrix $A$,
  - the $m$-vector of right hand sides $b$, and
  - the $n$-vector of objective costs $c$. 
Slack Variables

- **An inequality** \( a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i \) can be converted to an **equation** by introducing a **slack variable** \( s_i \geq 0 \):

  \[
a_{i1}x_1 + \cdots + a_{in}x_n + s_i = b_i
  \]

- Similarly, an inequality like \( \sum_{j=1}^{n} a_{ij}x_j \geq b_i \) can be written as

  \[
a_{i1}x_1 + \cdots + a_{in}x_n - s_i = b_i
  \]

- The **Standard form Linear Program** is

  \[
  \text{maximize } \quad cx \\
  \text{subject to } \quad Ax = b \\
  \quad \quad \quad \quad \quad \quad \quad x \geq 0
  \]

- Slack variable coefficients in the obj function are typically zero.
Geometrical Method

- 2-variable LP can be solved graphically.
- Useful teaching tool: Helps build intuition about the problem.
- Not a practical method for practical linear programming problems.
- We plot constraints on a rectangular coordinate system.
- A constraint is drawn as a line. The inequality constraint restricts the feasible region to one side of the line.
- The non-negative constraints $x_1, x_2 \geq 0$ restrict the feasible region to the first quadrant.
Example

• We will consider the following LP:

\[
\text{max } z = 2x_1 + 3x_2
\]

• subject to

\[
\begin{align*}
x_1 - 2x_2 & \leq 4 \\
2x_1 + x_2 & \leq 18 \\
x_2 & \leq 10 \\
x_1, x_2 & \geq 0
\end{align*}
\]
Illustration

- Building the feasible region, one constraint at a time.
Optimal Solution

- **Examine Level curves:** \( z = 1, 2 \) etc.
- **E.g.** \( z = 24 \) defines the line \( 2x_1 + 3x_2 = 24 \). Similarly, for \( z = 30 \).
- **Parallel lines**—they have same slope.
- **Line with highest \( z \) value that intersects feasible region.**
Algebra of LP

• **Important points:**
  Feasible region is convex polyhedron, optimal solution is a corner.

• **Simplex Method** is a famous method for solving LP, invented by George Dantzig.

• Introduce algebra and geometry to explain how it works.
• A hyperplane in $\mathbb{R}^n$ is the set of points $x = \{x_1, x_2, \ldots, x_n\}$ that satisfy $ax = b$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.

• Examples: Line $a_1x_1 + a_2x_2 = b$ in $\mathbb{R}^2$.

• $H^+ = \{x : ax \geq b\}$ is the positive halfspace.

• Vector $a$ is gradient of the linear function $f(x) = ax$, and so is normal to $H$.

• A polyhedral set is the intersection of halfspaces.
Convexity

- A set $S$ is **convex** if for any two points $x, y \in S$, the line segment $xy$ also lies in $S$.

- That is, $\alpha x + (1 - \alpha)y \in S$ if $x, y \in S$, for all $\alpha \in [0, 1]$.

- **Examples:** A hyperplane is convex. A halfspace is convex. Intersection of convex sets is convex.

- $\alpha x + (1 - \alpha)y$, where $\alpha \in [0, 1]$ is called **convex combination** of $x, y$.

- $\alpha x + (1 - \alpha)y$, for $\alpha \in (-\infty, \infty)$ is called **linear combination** of $x, y$. 
Extreme Points

- A point $x$ is **extreme point** of $S$ if it is not a strict convex combination of two other points of $S$.

- **Corners** of a polyhedron are extreme. All boundary points of a **sphere** are extreme.

- Let $S = \{x : Ax = b, x \geq 0\}$, where $A$ is $m \times n$ matrix of rank $m \leq n$.

- A point $x$ is an extreme point of $S$ iff $x$ is the intersection of $n$ linearly independent hyperplanes.
Basic Feasible Solution

• LP \( Ax = b \). Assume \( \text{rank}(A) = m \leq n \).

• Partition cols as \( A = (B : N) \), where
  \( B \) is \( m \times m \) basis matrix
  \( N \) is \( m \times (n - m) \) non-basis matrix.

• The LP system becomes
  \[
  Bx_B + Nx_N = b
  \]

• Because \( B \) is non-singular, it is invertible:
  \[
  x_B = B^{-1}b - B^{-1}Nx_N
  \]

• Set \( x_N = 0 \). We get \( x_B = B^{-1}b \).

• \( x_B \) is vector of basic variables,
  \( x_N \) vector of non-basic vectors.

• If \( x_B \geq 0 \), then the solution \( x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} \) is called a basic feasible solution.
Example of BFS

- Consider linear system:
  \[2x_1 + x_2 + x_3 + x_4 = 15\]
  \[x_1 + 3x_2 + x_3 - x_5 = 12\]

- \[A = \begin{pmatrix} 2 & 1 & 1 & 1 & 0 \\ 1 & 3 & 1 & 0 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 15 \\ 12 \end{pmatrix}.\]

- Consider basis matrix of 1st and 3rd cols:
  \[B = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}.\]

- Inverse matrix is \[B^{-1} = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}.\]

- BFS is \[x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, \text{ where } x_N = 0, \text{ and} \]
  \[x_B = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = B^{-1}b \]
  \[= \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 15 \\ 12 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}.\]
**Theorem:** x is an extreme point of $S = \{x : Ax = b, x \geq 0\}$ iff x is a BFS.

- **Each hyperplane $\leftrightarrow$ variable, which is 0 on the hyperplane.**
- **Each corner has 2 variables zero, and 3 non-zero.**

**Linear System**

\[
\begin{align*}
  x_1 - 2x_2 + x_3 &= 4 \\
  2x_1 + x_2 + x_4 &= 18 \\
  x_2 + x_5 &= 10 \\
  x_1, x_2, x_3, x_4, x_5 &\geq 0
\end{align*}
\]
Simplex Example

• Consider LP  \( \max \ z = 2x_1 + 3x_2 \)

• subject to

\[
\begin{align*}
x_1 - 2x_2 + x_3 &= 4 \\
2x_1 + x_2 + x_4 &= 18 \\
x_2 + x_5 &= 10 \\
x_1, x_2, x_3, x_4, x_5 &\geq 0
\end{align*}
\]

• Initial basic feasible solution:  
  \((x_3, x_4, x_5) = (4, 18, 10), \text{ and } (x_1, x_2) = (0, 0).\)

• Objective function value is zero.

• **Entering variable.** Pick a \(x_i\) with \(c_i > 0\) (max) in the objective, and make it basic.  
  (E.g. \(x_2\))

• **Departing variable.** Increase \(x_1\) from zero, until some basic variable becomes zero  
  (and non-basic).

• **Repeat until all vars in objective function have negative coefficients.**
Simplex Example

• Consider LP \( \max z = 2x_1 + 3x_2 \)

• subject to

\[
\begin{align*}
    x_1 - 2x_2 + x_3 & = 4 \\
    2x_1 + x_2 + x_4 & = 18 \\
    x_2 + x_5 & = 10 \\
    x_1, x_2, x_3, x_4, x_5 & \geq 0
\end{align*}
\]

• Initial BFS form:

\[
\begin{align*}
    x_3 & = 4 - x_1 + 2x_2 \\
    x_4 & = 18 - 2x_1 - x_2 \\
    x_5 & = 10 - x_2
\end{align*}
\]

• Set \( x_1, x_2 = 0, x_3 = 4, x_4 = 18, x_5 = 10 \). \( z = 0 \).

• Entering variable is \( x_2 \).

• Max value of \( x_2 \) is \( \min\{\infty, 18, 10\} \)

• Departing variable is \( x_5 \).
Simplex Example

• Re-arrange the linear system:

\[
\begin{align*}
x_2 &= 10 - x_5 \\
x_3 &= 24 - x_1 - 2x_5 \\
x_4 &= 8 - 2x_1 + x_5
\end{align*}
\]

• Set \( x_1 = x_5 = 0 \), and \( x_2 = 10, x_3 = 24, x_4 = 8 \).

• Obj. function becomes

\[
z = 30 + 2x_1 - 3x_5 = 30.
\]

• Entering variable is \( x_1 \).

• Max value of \( x_1 \) is \( \min\{\infty, 24, 4\} \).

• Departing variable is \( x_4 \).
Simplex Example

• Re-arrange the linear system:

  \[ x_1 = 4 - \frac{1}{2}x_4 + \frac{1}{2}x_5 \]
  \[ x_2 = 10 - x_5 \]
  \[ x_3 = 20 + \frac{1}{2}x_4 - \frac{5}{2}x_2 \]

• Set \( x_4 = x_5 = 0 \), and \( x_1 = 4, x_2 = 10, x_3 = 20 \).

• Obj. function becomes

  \[ z = 38 - x_4 - 2x_5 = 38 \]

• This is optimal because all coeffs are negative.
Complexity of LP

• Simplex invented by G. Dantzig in 1947.
• One of the most popular and widely used optimization method. Many commercial software packages.
• Extremely fast in practice, solves LP with $10^5$ constraints and variables in secs or mins.
• Worst-case complexity exponential! Klee-Minty (1972)
• Interior point method of Karmakar (1984) first method to be theoretically in $P$ and comparable in practice.
• Open Problem: Is there a pivoting rule for which simplex is polynomial?
LP Duality

- **Duality** turns out to be a deep property of linear programming.
- Many **famous theorems** in other areas are an easy consequence of **LP Duality**.
- In order to motivate its power, we consider an **investment game**.
- **Investment Game** played during **Mon-Fri** period.
- **Rules:** If invest $x$ one morning, and $2x$ next morning, then receive $4x$ on third morning.
- Amount received on day $D$ can be **invested same day for next round**.
- Inability to pony up $2x$ on second day forfeits original $x$.
- **Determine optimal investment strategy.**
Strategy 1

- Starting fortune is $1.
- Invest $1/3 on Mon, and $2/3 on Tues.
- Invest remaining $8/9 on Thurs.
- Collect $16/9 on Friday.
- Total gain is $7/9.
- Can we do better?
### Strategy 2

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1/2</td>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/4</td>
<td>1/2</td>
<td></td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>1/2</td>
<td></td>
<td>(2)</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

- **Total gain is $1.**
- **Can we do still better?**
- **This is a maximization LP, and we want an upper bound on the solution.**
- **LP duality aims to answer this question.**
Illustrating Example

- Consider LP $\text{max } z = 4x_1 + x_2 + 5x_3 + 3x_4$
- subject to
  \[
  x_1 - x_2 - x_3 + 3x_4 \leq 1 \\
  5x_1 + x_2 + 3x_3 + 8x_4 \leq 55 \\
  -x_1 + 2x_2 + 3x_3 - 5x_4 \leq 3 \\
  x_i \geq 0
  \]
- Suppose instead of computing optimal $z^*$, we just want a quick estimate of the max possible (upper bound).
- We can get lower bounds by trying feasible solutions:
  - $(0, 0, 1, 0) \rightarrow z^* = 5$
  - $(2, 1, 1, 1/3) \rightarrow z^* = 15$
  - $(3, 0, 2, 0) \rightarrow z^* = 22$
- However, this method still gives no estimate of an upper bound.
Illustrating Example

• LP max \[ z = 4x_1 + x_2 + 5x_3 + 3x_4 \]

• subject to

\[
\begin{align*}
x_1 - x_2 - x_3 + 3x_4 & \leq 1 \\
5x_1 + x_2 + 3x_3 + 8x_4 & \leq 55 \\
-x_1 + 2x_2 + 3x_3 - 5x_4 & \leq 3 \\
x_i & \geq 0
\end{align*}
\]

• Multiply 2nd constraint by \( \frac{5}{3} \):

\[
\frac{25}{3} x_1 + \frac{5}{3} x_2 + \frac{15}{3} x_3 + \frac{40}{3} x_4 \leq \frac{275}{3}
\]

• This implies \( z^* \leq \frac{275}{3} \), because term by term this constraint dominates obj. function \( z \).

• One more. Add constraints (2) and (3),

\[
4x_1 + 3x_2 + 6x_3 + 3x_4 \leq 58
\]

• This implies \( z^* \leq 58 \).
General Strategy

• Form **linear combinations** of constraints.

• **Multiply constraint** \((i)\) by \(y_i\), then add
  \[
  y_1(x_1 - x_2 - x_3 + 3x_4) + \\
y_2(5x_1 + x_2 + 3x_3 + 8x_4) + \\
y_3(-x_1 + 2x_2 + 3x_3 - 5x_4) \leq y_1 + 55y_2 + 3y_3.
  \]

• **Rewrite**
  \[
  (y_1 + 5y_2 - y_3)x_1 + (-y_1 + y_2 + 2y_3)x_2 + \\
  (-y_1 + 3y_2 + 3y_3)x_3 + (3y_1 + 8y_2 - 5y_3)x_4 \leq y_1 + 55y_2 + 3y_3.
  \]

• **For this to dominate** \(z\) **term by term**
  \[
  y_1 + 5y_2 - y_3 \geq 4 \\
  -y_1 + y_2 + 2y_3 \geq 1 \\
  -y_1 + 3y_2 + 3y_3 \geq 5 \\
  3y_1 + 8y_2 - 5y_3 \geq 3 \\
  y_i \geq 0
  \]

• **We then have** \(z \leq y_1 + 55y_2 + 3y_3\).
Our goal is to find smallest $z$ subject to constraints on $y_i$.

We have a linear program:

**minimize** $u = y_1 + 55y_2 + 3y_3$

**subject to**

\[
\begin{align*}
y_1 + 5y_2 - y_3 & \geq 4 \\
-y_1 + y_2 + 2y_3 & \geq 1 \\
-y_1 + 3y_2 + 3y_3 & \geq 5 \\
3y_1 + 8y_2 - 5y_3 & \geq 3 \\
y_i & \geq 0
\end{align*}
\]

This LP is called the **Dual** of the original LP.
Duality Theorems

Weak Duality: If $z$ and $u$ are feasible solutions of the primal and dual LPs, then

$$z \leq u$$

- Follows by definition of dual LP. Each feasible solution of dual is an upper bound on primal.

Strong Duality: If primal LP has finite optimal solution $z^*$, then the dual also has a finite optimal solution $u^*$ and

$$z^* = u^*$$
Back to Investment Game

- Variables $x_i$ to denote investment amounts.

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$2x_1$</td>
<td>$(4x_1)$</td>
<td>$2x_2$</td>
<td>$(4x_2)$</td>
</tr>
<tr>
<td>$x_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The initial amount is $1$.
- The final amount is $1 + x_1 + x_2 + x_3$.
- Formulate as LP: \( \text{max} \ 1 + x_1 + x_2 + x_3 \)

\[
\begin{align*}
1 - x_1 & \geq 0 \\
1 - 3x_1 - x_2 & \geq 0 \\
1 + x_1 - 3x_2 - x_3 & \geq 0 \\
1 + x_1 + x_2 - 3x_3 & \geq 0 \\
x_i & \geq 0
\end{align*}
\]
Back to Investment Game

• **Investment game LP:** \( \text{max } 1 + x_1 + x_2 + x_3 \)

• subject to

\[
\begin{align*}
1 - x_1 & \geq 0 \\
1 - 3x_1 - x_2 & \geq 0 \\
1 + x_1 - 3x_2 - x_3 & \geq 0 \\
1 + x_1 + x_2 - 3x_3 & \geq 0 \\
x_i & \geq 0
\end{align*}
\]

• **Linear combination** \( 2 \times (2) + (3) + (4): \)

\[
4x_1 + 4x_2 + 4x_3 \leq 4, \text{ which implies}
\]

\[
1 + (x_1 + x_2 + x_3) \leq 2
\]

• **Thus, strategy 2 is optimal.**
Integer Programming

- maximize $cx$
- subject to $Ax \leq b$
- $x$ integer 
  (only diff from LP)

- IP feasible space is discrete, and smaller than LP feasible space, which is continuous.
• One might think that restricting variables makes problem easier to solve.

• In fact, IP is NP-Complete.

• Reduction from Boolean 3-SAT.

• $\Phi = (x_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_2 \lor x_3 \lor \overline{x}_4) \land \cdots$

• Boolean Satisfiability of $\Phi$ can be encoded as an IP:

• minimize $x_1$

• subject to

\[
x_1 + (1 - x_3) + x_4 \geq 1
\]

\[
(1 - x_2) + x_3 + (1 - x_4) \geq 1
\]

\[
\vdots
\]

\[
x_i \in \{0, 1\}
\]
Formulating IP Problems

- **Knapsack Problem**: \( n \) items, weights \( a_i \), capacity \( b \).
  - maximize \( \sum_{i=1}^{n} c_i x_i \)
  - subject to \( \sum_{i=1}^{n} a_j x_j \leq b \)
    \[ x_i \text{ binary} \]

- **Vertex Cover**: \( G = (V, E) \)
  - minimize \( \sum_{i=1}^{n} x_i \)
  - subject to \( x_i + x_j \geq 1 \), for each edge \( (i, j) \in E \)
    \[ x_i \text{ binary} \]
Network Problems using IP

• Min Cost Network Flows.

• **Given is a network** \( G = (V, E) \), supply \( b_i \) for each node \( i \), capacity \( c_{ij} \) for each edge, and lower and upper bounds on flow for each edge.

• **minimize** \[ \sum c_{ij}x_{ij} \]

• **subject to**

\[
\sum_{j: (i, j) \in E} x_{ij} - \sum_{j: (j, i) \in E} x_{ji} = b_i
\]

\[l_{ij} \leq x_{ij} \leq u_{ij}, \text{ for all } (i, j) \in E\]

• No need to restrict \( x_{ij} \) to integers.
Network Problems using IP

- Shortest Path: 1 unit flow from $s$ to $t$.
- minimize $\sum c_{ij}x_{ij}$
- subject to

\[
\sum_{j: (s, j) \in E} x_{sj} = 1
\]

\[
\sum_{j: (j, t) \in E} x_{jt} = -1
\]

\[
\sum x_{ij} - \sum x_{ji} = 0 \text{ if } i \neq s, t
\]

$x_{ij} \in \{0, 1\}$ for all $(i, j) \in E
Network Problems using IP

• Assignment Problem.

• minimize \( \sum c_{ij}x_{ij} \)

• subject to

\[
\sum_{j:(i,j) \in E} x_{ij} = 1 \quad \text{for all } i \in X
\]

\[
\sum_{j:(j,i) \in E} x_{ji} = 1 \quad \text{for all } i \in Y
\]

\[x_{ij} \in \{0, 1\} \quad \text{for all } (i,j) \in E\]
LP Relaxation of IP

- An IP formulation with integrality constraint removed is called the LP relaxation.

- If IP is $\max cx$, with optimal solution $z_I$, and $z_L$ is the optimal solution of relaxed LP, then

  $$z_I \leq z_L$$

- It follows because IP feasible space is a subset of LP feasible space.

- When can one guarantee that $z_L = z_I$?

- Specifically, shortest path and assignment formulations require integer variables. But they have combinatorial poly-time algorithms.
Total Unimodularity

- Consider an IP: \( \max cx \) s.t. \( Ax \leq b \), and \( x \) integer.

- The constraint matrix \( A \) is called Totally Unimodular (TUM) if \( \det(B) \) is in \( \{0, 1, -1\} \) for all \( 2 \times 2 \) submatrices \( B \) of \( A \).

- E.g. if each col of \( A \) has exactly one 1, one -1, and all other zero entries, then \( A \) is TUM.

- If \( A \) is the incidence matrix of a graph, then \( A \) is TUM.

- Thus, our network problems (shortest path, maxflow, assignment) can be solved through LP relaxation.