Lecture 16

November 17, 2025

Discrete Optimization

Today, we consider optimization problems with discrete variables:

$$\begin{aligned} \min\ & c^T x + d^T y \\ & Ax + By = b \\ & x, y \geq 0 \\ & x \text{ integer} \end{aligned}$$

This is called a **mixed integer programming** (MIP) problem

Without continuous variables y, it is called an **integer program** (IP)

If instead of $x \in \mathbb{Z}^n$ we have $x \in \{0,1\}^n$: **binary optimization** problem

Very powerful modeling paradigm

Example: Knapsack

- n items
- Item j has weight w_j and reward r_j
- Have a bound K on the weight that can be carried in the knapsack
- Want to select items to maximize the total value

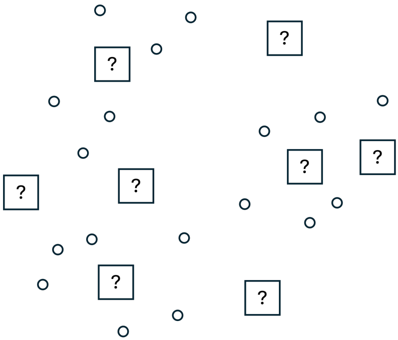
Example: Knapsack

- n items
- Item j has weight w_j and reward r_j
- Have a bound K on the weight that can be carried in the knapsack
- Want to select items to maximize the total value

$$\begin{aligned} \text{maximize} \; & \sum_{j=1}^{n} r_j x_j \\ \text{subject to} \; & \sum_{j=1}^{n} w_j x_j \leq K \\ & x_j \in \{0,1\}, \quad j = 1, \ldots, n. \end{aligned}$$

Example: Facility Location

- n potential locations to open facilities
- Cost c_j for opening a facility at location j
- m clients who need service
- Cost d_{ij} for serving client i from facility j
- Smallest cost for opening facilities while serving all clients



Example: Facility Location

- n potential locations to open facilities
- Cost c_j for opening a facility at location j
- m clients who need service
- Cost d_{ij} for serving client i from facility j
- Smallest cost for opening facilities while serving all clients

$$
\begin{aligned}
\min \; & \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad \forall\, i \\
& x_{ij} \leq y_j, \qquad \forall i,\ \forall j \\
& x_{ij}, y_j \in \{0, 1\}
\end{aligned}
$$

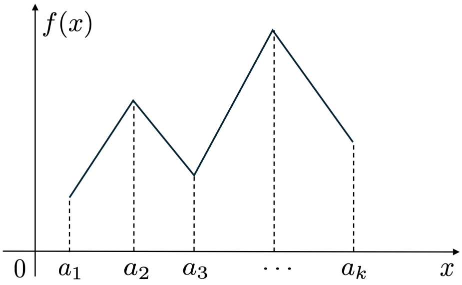
Example: Facility Location

- n potential locations to open facilities
- Cost c_j for opening a facility at location j
- m clients who need service
- Cost d_{ij} for serving client i from facility j
- Smallest cost for opening facilities while serving all clients

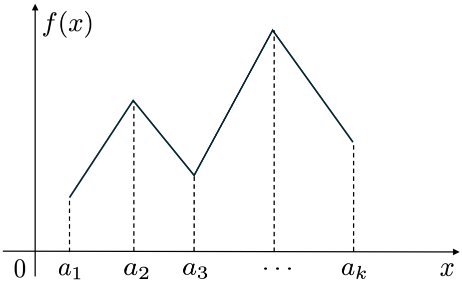
$$
\begin{aligned}
\min\ & \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad \forall\, i \\
& x_{ij} \le y_j, \qquad \forall i,\ \forall j \\
& x_{ij}, y_j \in \{0, 1\}
\end{aligned}
$$

$$
\begin{aligned}
\min\ & \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad \forall\, i \\
& \sum_{i=1}^{m} x_{ij} \le m y_j, \quad \forall j \\
& x_{ij}, y_j \in \{0, 1\}.
\end{aligned}
$$

Example: Piecewise Linear Cost



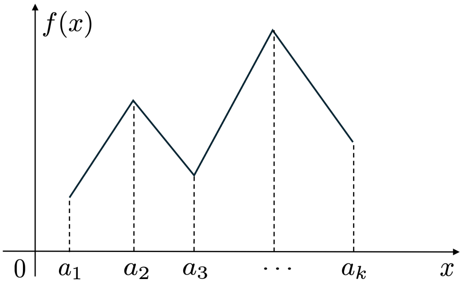
Example: Piecewise Linear Cost



- Idea: $x = \sum_{i=1}^{k} \lambda_i a_i$
- Cost: $\sum_{i=1}^{k} \lambda_i f(a_i)$
- How to impose adjacency?

$$x = \lambda_i a_i + \lambda_{i+1} a_{i+1}$$

Example: Piecewise Linear Cost



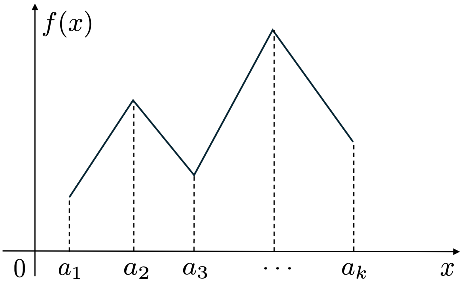
- Idea: $x = \sum_{i=1}^{k} \lambda_i a_i$
- Cost: $\sum_{i=1}^{k} \lambda_i f(a_i)$
- How to impose adjacency?

$$x = \lambda_i a_i + \lambda_{i+1} a_{i+1}$$

- New binary variables $\{y_i\}_{i=1}^{k}$ to impose:

$$y_j = 1 \;\Rightarrow\; \lambda_i = 0 \text{ for } i \notin \{j, j+1\}$$

Example: Piecewise Linear Cost



- Idea: $x = \sum_{i=1}^{k} \lambda_i a_i$
- Cost: $\sum_{i=1}^{k} \lambda_i f(a_i)$
- How to impose adjacency?

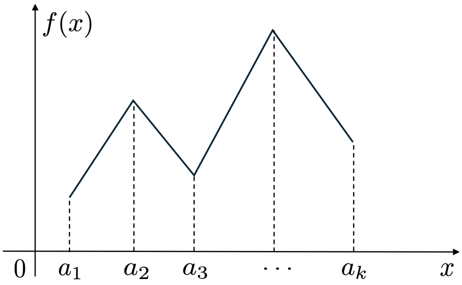
$$x = \lambda_i a_i + \lambda_{i+1} a_{i+1}$$

- New binary variables $\{y_i\}_{i=1}^{k}$ to impose:

$$y_j = 1 \;\Rightarrow\; \lambda_i = 0 \text{ for } i \notin \{j, j+1\}$$

$$\begin{aligned}
\textstyle\sum_{i=1}^{k} \lambda_i &= 1, \\
\lambda_1 &\leq y_1, \\
\lambda_i &\leq y_{i-1} + y_i, \;\; i = 2, \ldots, k-1, \\
\lambda_k &\leq y_{k-1}, \\
\textstyle\sum_{i=1}^{k-1} y_i &= 1, \\
\lambda_i &\geq 0, \\
y_i &\in \{0, 1\}, \;\; \forall i.
\end{aligned}$$

Example: Piecewise Linear Cost



Nowadays, easy to model. In Gurobipy:

- $y = 1$ implies $\ell(x) \geq 0$:

  addConstr((y==1) >> ($\ell(x)$>=0))

- $y = f(x)$:

  addGenConstrPWL(x,y,xpnts,ypnts)

- Idea: $x = \sum_{i=1}^{k} \lambda_i a_i$
- Cost: $\sum_{i=1}^{k} \lambda_i f(a_i)$
- How to impose adjacency?

$$x = \lambda_i a_i + \lambda_{i+1} a_{i+1}$$

- New binary variables $\{y_i\}_{i=1}^{k}$ to impose:

$$y_j = 1 \;\Rightarrow\; \lambda_i = 0 \text{ for } i \notin \{j, j+1\}$$

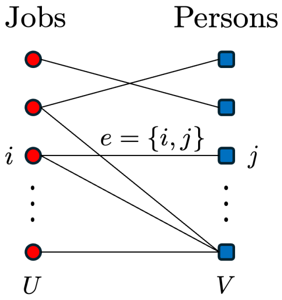
$$\begin{array}{rcl}
\sum_{i=1}^{k} \lambda_i & = & 1, \\
\lambda_1 & \leq & y_1, \\
\lambda_i & \leq & y_{i-1} + y_i, \;\; i = 2, \ldots, k-1, \\
\lambda_k & \leq & y_{k-1}, \\
\sum_{i=1}^{k-1} y_i & = & 1, \\
\lambda_i & \geq & 0, \\
y_i & \in & \{0, 1\}, \;\; \forall i.
\end{array}$$

Example: Matching Problems

- Set U of jobs/tasks to complete; set V of persons available to work
- Each task assigned to at most one person; a person can only complete some tasks
- Reward w_{ij} if task $i \in U$ completed by person $j \in V$

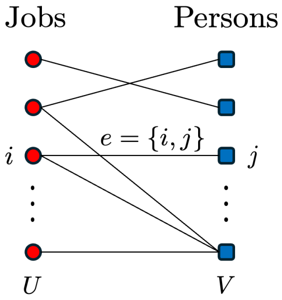
Example: Matching Problems

- Set U of jobs/tasks to complete; set V of persons available to work
- Each task assigned to at most one person; a person can only complete some tasks
- Reward w_{ij} if task $i \in U$ completed by person $j \in V$
- Graph representation $G = (\mathcal{N}, \mathcal{E})$
- $e \equiv \{i, j\} \in \mathcal{E}$ indicates $j \in V$ can complete task $i \in U$



Example: Matching Problems

- Set U of jobs/tasks to complete; set V of persons available to work
- Each task assigned to at most one person; a person can only complete some tasks
- Reward w_{ij} if task $i \in U$ completed by person $j \in V$
- Graph representation $G = (\mathcal{N}, \mathcal{E})$
- $e \equiv \{i, j\} \in \mathcal{E}$ indicates $j \in V$ can complete task $i \in U$



$x_e \in \{0, 1\}$: whether edge selected

$$\text{maximize} \sum_{e \in E} w_e x_e$$

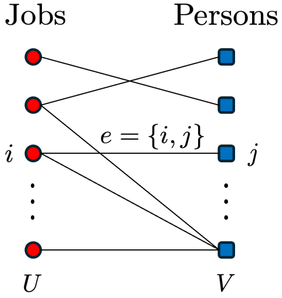
$$\sum_{e \in \delta(i)} x_e \leq 1, \quad \forall\, i \in N,$$

$$x_e \in \{0, 1\},$$

$\delta(i) := \{e \in E : e = \{i, j\} \text{ for } j \in N\}$: all edges incident on i

Example: Matching Problems

- Set U of jobs/tasks to complete; set V of persons available to work
- Each task assigned to at most one person; a person can only complete some tasks
- Reward w_{ij} if task $i \in U$ completed by person $j \in V$
- Graph representation $G = (\mathcal{N}, \mathcal{E})$
- $e \equiv \{i, j\} \in \mathcal{E}$ indicates $j \in V$ can complete task $i \in U$



$x_e \in \{0, 1\}$: whether edge selected

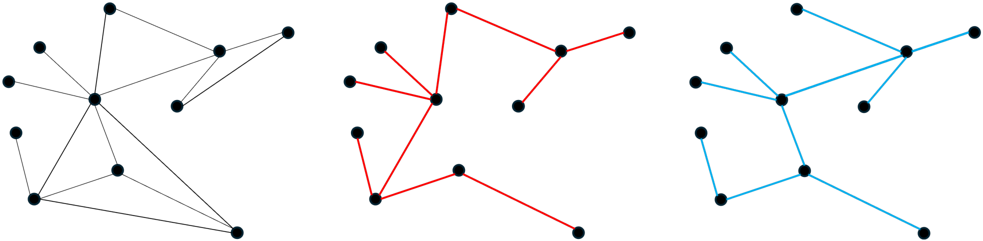
$$
\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} w_e x_e \\
& \sum_{e \in \delta(i)} x_e \le 1, \quad \forall\, i \in N, \\
& x_e \in \{0, 1\},
\end{aligned}
$$

$\delta(i) := \{e \in E : e = \{i, j\} \text{ for } j \in N\}$: all edges incident on i

Many variations: minimize cost, require jobs completed, perfect matching, ...

Example: Minimum Spanning Tree

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find **minimum spanning tree** (MST)
  (subset of edges that connect all nodes in $\mathcal{N}$ at minimum cost)



Example: Minimum Spanning Tree

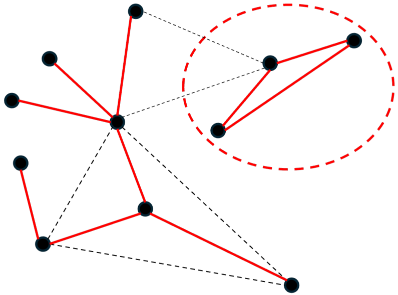
- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find **minimum spanning tree** (MST)
  (subset of edges that connect all nodes in $\mathcal{N}$ at minimum cost)

$$
\begin{aligned}
\min \quad & \sum_{e \in \mathcal{E}} c_e x_e \\
& x_e \in \{0, 1\} \\
\textbf{(Connectivity)} \quad & \sum_{e \in \mathcal{E}} x_e = n - 1 \\
\textbf{(Cutset)} \quad & \sum_{e \in \delta(S)} x_e \geq 1, \;\; S \subset \mathcal{N}, S \neq \emptyset
\end{aligned}
$$

Example: Minimum Spanning Tree

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find **minimum spanning tree** (MST)
  (subset of edges that connect all nodes in $\mathcal{N}$ at minimum cost)

$$\begin{aligned} \min \ & \sum_{e \in \mathcal{E}} c_e x_e \\ & x_e \in \{0, 1\} \\ \textbf{(Connectivity)} \ & \sum_{e \in \mathcal{E}} x_e = n - 1 \\ \textbf{(Cutset)} \ & \sum_{e \in \delta(S)} x_e \geq 1, \ S \subset \mathcal{N}, S \neq \emptyset \end{aligned}$$



Example: Minimum Spanning Tree

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find **minimum spanning tree** (MST)
  (subset of edges that connect all nodes in $\mathcal{N}$ at minimum cost)

$$\min \sum_{e \in \mathcal{E}} c_e x_e$$

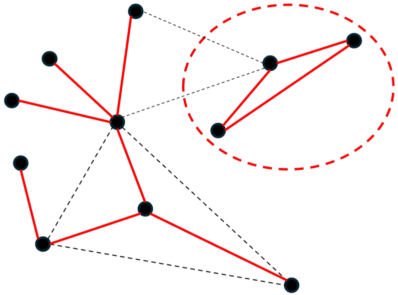
$$x_e \in \{0, 1\}$$

(Connectivity) $$\sum_{e \in \mathcal{E}} x_e = n - 1$$

(Cutset) $$\sum_{e \in \delta(S)} x_e \geq 1, \; S \subset \mathcal{N}, S \neq \emptyset$$

... or ...

(Subtour-elimination) $$\sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \; S \subset \mathcal{N}, S \neq \emptyset$$



Example: Minimum Spanning Tree

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find **minimum spanning tree** (MST)
  (subset of edges that connect all nodes in $\mathcal{N}$ at minimum cost)

$$\min \sum_{e \in \mathcal{E}} c_e x_e$$

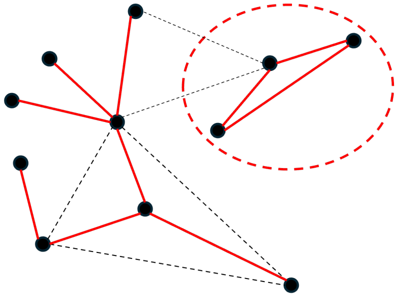
$$x_e \in \{0, 1\}$$

$$\textbf{(Connectivity)} \quad \sum_{e \in \mathcal{E}} x_e = n - 1$$

$$\textbf{(Cutset)} \quad \sum_{e \in \delta(S)} x_e \geq 1, \ S \subset \mathcal{N}, S \neq \emptyset$$

… or …

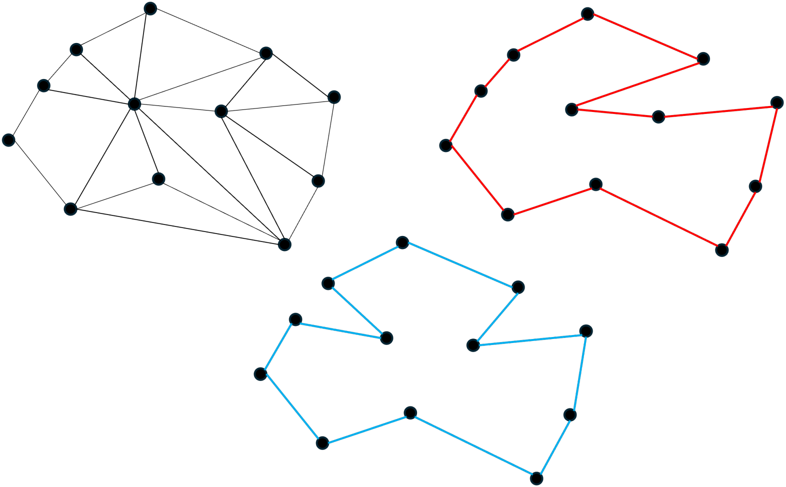
$$\textbf{(Subtour-elimination)} \quad \sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \ S \subset \mathcal{N}, S \neq \emptyset$$



Both exponentially-sized formulations! *Any preference between them?*

Example: Traveling Salesperson Problem (TSP)

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find a **tour** *(cycle that visits each node exactly once)* with minimum cost



Example: Traveling Salesperson Problem (TSP)

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find a **tour** *(cycle that visits each node exactly once)* with minimum cost

$$
\begin{aligned}
\min \quad & \sum_{e \in \mathcal{E}} c_e x_e \\
& x_e \in \{0, 1\} \\
\textbf{(Connectivity)} \quad & \sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N \\
\textbf{(Cutset)} \quad & \sum_{e \in \delta(S)} x_e \geq 2, \forall S \subset N, S \neq \emptyset
\end{aligned}
$$

Example: Traveling Salesperson Problem (TSP)

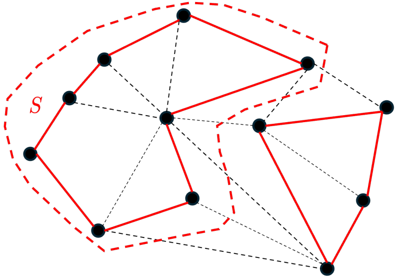
- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find a **tour** *(cycle that visits each node exactly once)* with minimum cost

$$\min \sum_{e \in \mathcal{E}} c_e x_e$$

$$x_e \in \{0, 1\}$$

$$\textbf{(Connectivity)} \quad \sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N$$

$$\textbf{(Cutset)} \quad \sum_{e \in \delta(S)} x_e \geq 2, \forall S \subset N, S \neq \emptyset$$



Example: Traveling Salesperson Problem (TSP)

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find a **tour** *(cycle that visits each node exactly once)* with minimum cost

$$\min \sum_{e \in \mathcal{E}} c_e x_e$$

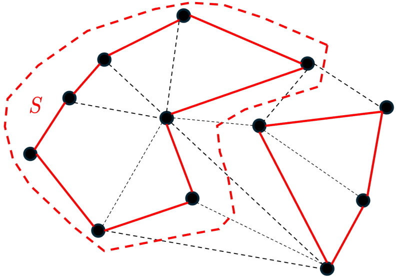
$$x_e \in \{0, 1\}$$

(Connectivity) $\displaystyle\sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N$

(Cutset) $\displaystyle\sum_{e \in \delta(S)} x_e \geq 2, \forall S \subset N, S \neq \emptyset$

... or ...

(Subtour-elimination) $\displaystyle\sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \forall S \subset N, S \neq \emptyset$



Example: Traveling Salesperson Problem (TSP)

- Given an undirected graph $G = (\mathcal{N}, \mathcal{E})$; $|\mathcal{N}| = n$, $|\mathcal{E}| = m$
- Edge $e \in \mathcal{E}$ has associated cost c_e
- Find a **tour** *(cycle that visits each node exactly once)* with minimum cost

$$\min \sum_{e \in \mathcal{E}} c_e x_e$$

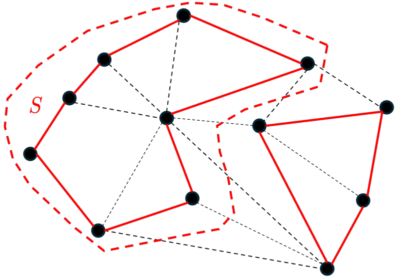
$$x_e \in \{0, 1\}$$

(Connectivity) $$\sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N$$

(Cutset) $$\sum_{e \in \delta(S)} x_e \geq 2, \forall S \subset N, S \neq \emptyset$$

... or ...

(Subtour-elimination) $$\sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \forall S \subset N, S \neq \emptyset$$



Again exponentially-sized formulations! *Any preference between them?*

Bad News First

Bad News First

Example. The optimal solution in an IP **might not exist**:

$$\sup_{x,y} \; x + \sqrt{2}y$$
$$x + \sqrt{2}y \le \frac{1}{2}$$
$$x, y \in \mathbb{Z}.$$

Bad News First

Example. The optimal solution in an IP **might not exist**:

$$\sup_{x,y} \; x + \sqrt{2}y$$
$$x + \sqrt{2}y \leq \frac{1}{2}$$
$$x, y \in \mathbb{Z}.$$

Example. Consider the following pair of problems:

$$(\mathcal{P}) \;\; \min_{x \geq 0} \; x \qquad\qquad (\mathcal{D}) \;\; \max_{p} \; p$$
$$2x = 1 \qquad\qquad 2p \leq 1$$

Bad News First

Example. The optimal solution in an IP **might not exist**:

$$\begin{aligned} \sup_{x,y}\ & x + \sqrt{2}y \\ & x + \sqrt{2}y \leq \frac{1}{2} \\ & x, y \in \mathbb{Z}. \end{aligned}$$

Example. Consider the following pair of problems:

$$(\mathcal{P})\ \min_{x \geq 0}\ x \qquad\qquad (\mathcal{D})\ \max_{p}\ p$$

$$2x = 1 \qquad\qquad 2p \leq 1$$

- $x, p \in \mathbb{R} \Rightarrow$ this is a primal-dual pair; optimal value $\frac{1}{2}$ by strong duality

Bad News First

Example. The optimal solution in an IP **might not exist**:

$$\begin{aligned} \sup_{x,y}\ & x + \sqrt{2}y \\ & x + \sqrt{2}y \leq \frac{1}{2} \\ & x, y \in \mathbb{Z}. \end{aligned}$$

Example. Consider the following pair of problems:

$$(\mathcal{P})\ \min_{x \geq 0}\ x \qquad\qquad (\mathcal{D})\ \max_{p}\ p$$

$$2x = 1 \qquad\qquad 2p \leq 1$$

- $x, p \in \mathbb{R} \Rightarrow$ this is a primal-dual pair; optimal value $\frac{1}{2}$ by strong duality
- $x, p \in \mathbb{Z} \Rightarrow (\mathcal{P})$ infeasible, $(\mathcal{D})$ has optimal value 0.

  Strong duality does not hold in IPs

Bad News First

Unfortunately, (M)IPs are **significantly harder** than LPs

Theorem

Given a matrix $A \in \mathbb{Q}^{m \times n}$ *and a vector* $b \in \mathbb{Q}^m$*, the problem: "does* $Ax \leq b$ *have an integral solution* x*" is* **NP-complete**.

- IP "feasibility problem" is already in the hardest class of problems in NP

Bad News First

Unfortunately, (M)IPs are **significantly harder** than LPs

Theorem

Given a matrix $A \in \mathbb{Q}^{m \times n}$ and a vector $b \in \mathbb{Q}^m$, the problem: "does $Ax \leq b$ have an integral solution x" is **NP-complete**.

- IP "feasibility problem" is already in the hardest class of problems in NP
- Despite this, substantial theory and scalable algorithms exist for IPs
- We will focus on optimization problems with **rational entries**: $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n$ (in fact, often **integer**)
- We assume that the **feasible set is bounded**

Lower Bounds Again

Same question as in LP: *how can we find a good lower bound?*

If we relaxed integrality requirements, we would get at LP!

Lower Bounds Again

Same question as in LP: *how can we find a good lower bound?*

If we relaxed integrality requirements, we would get at LP!

Definition (LP relaxation)

The **linear programming relaxation** for the integer program

$$
\begin{aligned}
\min\ & c^\mathsf{T}x + d^\mathsf{T}y \\
& Ax + By = b \\
& x, y \geq 0 \\
& x \in \{0,1\}^{n_1}, y \in \mathbb{Z}^{n_2},
\end{aligned}
$$

is obtained by replacing $x \in \{0,1\}^{n_1}$ with $x \in [0,1]^{n_1}$ and $y \in \mathbb{Z}^{n_2}$ with $y \in \mathbb{R}^{n_2}$.

Lower Bounds Again

Same question as in LP: *how can we find a good lower bound?*

If we relaxed integrality requirements, we would get at LP!

Definition (LP relaxation)

The **linear programming relaxation** for the integer program

$$\begin{aligned} \min\ & c^\mathsf{T}x + d^\mathsf{T}y \\ & Ax + By = b \\ & x, y \geq 0 \\ & x \in \{0,1\}^{n_1}, y \in \mathbb{Z}^{n_2}, \end{aligned}$$

is obtained by replacing $x \in \{0,1\}^{n_1}$ with $x \in [0,1]^{n_1}$ and $y \in \mathbb{Z}^{n_2}$ with $y \in \mathbb{R}^{n_2}$.

Observation

1) OPT(LP relaxation) $\leq$ OPT(IP optimal value).

2) If LP relaxation's optimal solution is feasible for the IP, it is optimal for the IP.

Key Q: How good is this bound?

LP Relaxation for Facility Location IP

Recall the **two** formulations of the Facility Location Problem

(FL)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m$$

$$x_{ij} \leq y_j, \quad i = 1, \ldots, m, \ \ j = 1, \ldots, n$$

$$x_{ij}, y_j \in \{0, 1\}$$

(AFL)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} \leq m y_j, \quad j = 1, \ldots, n$$

$$x_{ij}, y_j \in \{0, 1\}.$$

LP Relaxation for Facility Location IP

Recall the **two** formulations of the Facility Location Problem

(FL)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \dots, m$$

$$x_{ij} \le y_j, \quad i = 1, \dots, m, \;\; j = 1, \dots, n$$

$$x_{ij}, y_j \in \{0, 1\}$$

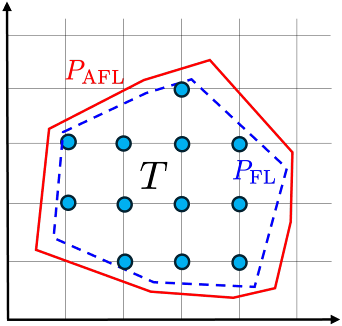
(AFL)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \dots, m$$

$$\sum_{i=1}^{m} x_{ij} \le m y_j, \quad j = 1, \dots, n$$

$$x_{ij}, y_j \in \{0, 1\}.$$

- $P_{\mathbf{FL}}, P_{\mathbf{AFL}}$: feasible sets for LP relaxations
- $P_{\mathrm{FL}} \subseteq P_{\mathrm{AFL}}$ and can have **strict** inclusion



LP Relaxation for Facility Location IP

Recall the **two** formulations of the Facility Location Problem

(FL)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m$$

$$x_{ij} \leq y_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

$$x_{ij}, y_j \in \{0, 1\}$$

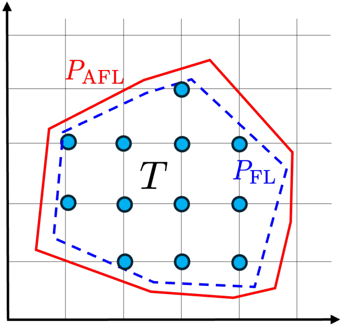
(AFL)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} \leq m y_j, \quad j = 1, \ldots, n$$

$$x_{ij}, y_j \in \{0, 1\}.$$

- $P_{\mathbf{FL}}, P_{\mathbf{AFL}}$: feasible sets for LP relaxations
- $P_{\mathrm{FL}} \subseteq P_{\mathrm{AFL}}$ and can have **strict** inclusion
- **(FL) provides better lower bound than (AFL)**
- **Same IP feasible set, smaller LP feasible set!**



LP Relaxation for Minimum Spanning Tree Problem

(Cutset MST)

$$
\begin{aligned}
&\sum_{e \in \mathcal{E}} x_e = n - 1, \\
&\sum_{e \in \delta(S)} x_e \geq 1, \quad S \subset \mathcal{N}, S \neq \emptyset \\
&x_e \in \{0, 1\}
\end{aligned}
$$

(Subtour-elimination MST)

$$
\begin{aligned}
&\sum_{e \in \mathcal{E}} x_e = n - 1, \\
&\sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \quad S \subset \mathcal{N}, S \neq \emptyset, \\
&x_e \in \{0, 1\}.
\end{aligned}
$$

- $P_{\mathbf{cut}}, P_{\mathbf{sub}}$: feasible sets for LP relaxations

LP Relaxation for Minimum Spanning Tree Problem

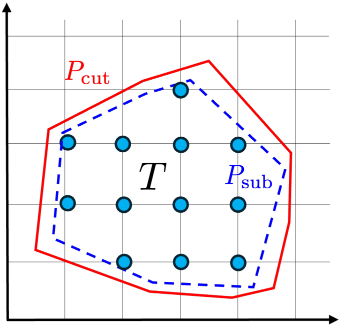
(Cutset MST)

$$\sum_{e \in \mathcal{E}} x_e = n - 1,$$
$$\sum_{e \in \delta(S)} x_e \geq 1, \quad S \subset \mathcal{N}, S \neq \emptyset$$
$$x_e \in \{0, 1\}$$

(Subtour-elimination MST)

$$\sum_{e \in \mathcal{E}} x_e = n - 1,$$
$$\sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \quad S \subset \mathcal{N}, S \neq \emptyset,$$
$$x_e \in \{0, 1\}.$$

- $P_{\textbf{cut}}, P_{\textbf{sub}}$: feasible sets for LP relaxations
- $P_{\text{sub}} \subseteq P_{\text{cut}}$ and can have **strict** inclusion
  (Proof in the notes)
- **(SUB) provides better lower bound than (CUT)**
- **Same IP feasible set, smaller LP feasible set!**



LP Relaxation for Traveling Salesperson Problem (TSP)

(Cutset TSP)

$$\sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N$$
$$\sum_{e \in \delta(S)} x_e \geq 2, \forall S \subset N, S \neq \emptyset$$

(Subtour-elimination TSP)

$$\sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N$$
$$\sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \forall S \subset N, S \neq \emptyset.$$

- $P_{\mathbf{cut}}, P_{\mathbf{sub}}$: feasible sets for LP relaxations

LP Relaxation for Traveling Salesperson Problem (TSP)

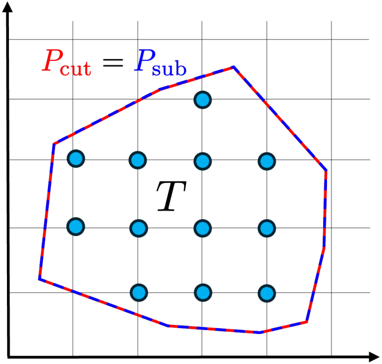
(Cutset TSP)

$$\sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N$$
$$\sum_{e \in \delta(S)} x_e \geq 2, \forall S \subset N, S \neq \emptyset$$

(Subtour-elimination TSP)

$$\sum_{e \in \delta(\{i\})} x_e = 2, \forall i \in N$$
$$\sum_{e \in \mathcal{E}(S)} x_e \leq |S| - 1, \forall S \subset N, S \neq \emptyset.$$

- $P_{\mathbf{cut}}, P_{\mathbf{sub}}$: feasible sets for LP relaxations
- $P_{\text{sub}} = P_{\text{cut}}$

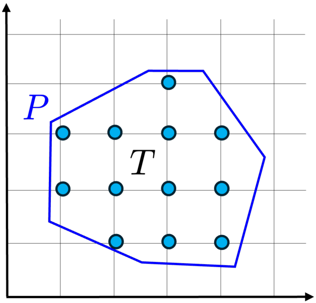


Strength of IP Formulation

- Different formulations of the same IP can result in **different LP relaxations**

- *What is an "ideal" formulation?*

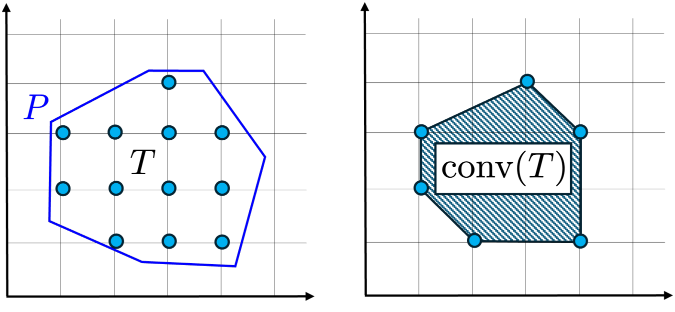
Strength of IP Formulation



(a)

- Consider an IP with bounded feasible set
- T : all feasible points to the IP
- P : feasible set for LP relaxation to IP

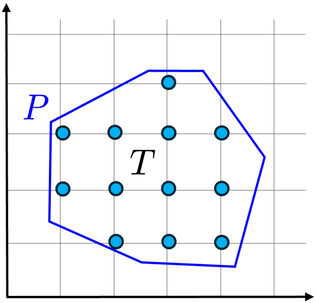
Strength of IP Formulation



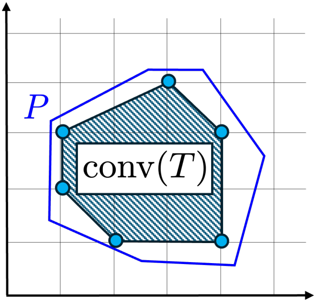
(a) (b)

- Consider an IP with bounded feasible set
- T : all feasible points to the IP
- P : feasible set for LP relaxation to IP
- $\mathrm{conv}(T)$: the convex hull of T
  - a polyhedron because we assumed bounded feasible set

Strength of IP Formulation



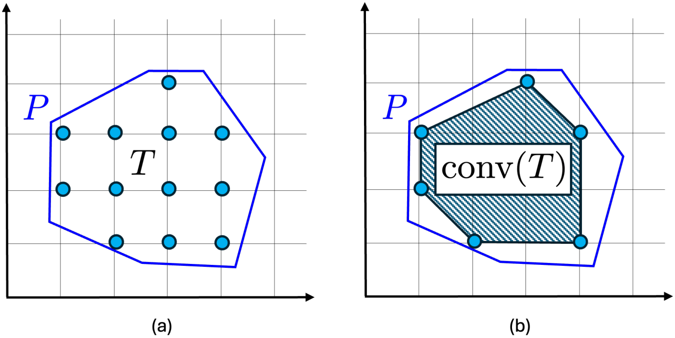
(a)



(b)

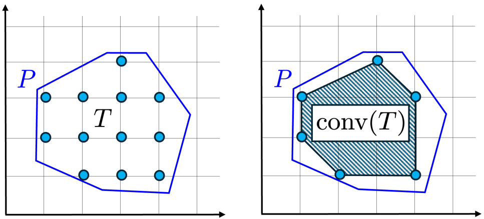
- Consider an IP with bounded feasible set
- T : all feasible points to the IP
- P : feasible set for LP relaxation to IP
- $\mathrm{conv}(T)$: the convex hull of T
  - a polyhedron because we assumed bounded feasible set
- We always have: $T \subseteq \mathrm{conv}(T) \subseteq P$.

Strength of IP Formulation



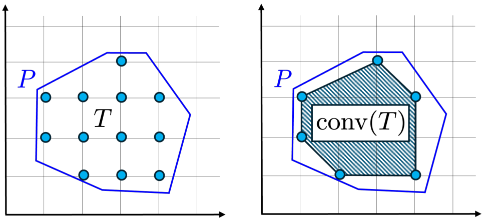
- Consider an IP with bounded feasible set
- T : all feasible points to the IP
- P : feasible set for LP relaxation to IP
- $\mathrm{conv}(T)$: the convex hull of T
  - a polyhedron because we assumed bounded feasible set
- We always have: $T \subseteq \mathrm{conv}(T) \subseteq P$.
- **Ideal LP relaxation** would have $P = \mathrm{conv}(T)$

Strength of IP Formulation



- **Quality of IP formulation** : how closely its LP relaxation approximates $\mathrm{conv}(T)$
- For an IP and two equivalent formulations A, B: **A is stronger than B** if $P_A \subset P_B$
- **Constraints** play a more subtle role in IPs than in LPs
  - Adding valid constraints for T that cut off fractional points from P is very useful!
  - More constraints not necessarily worse in IP!

Strength of IP Formulation



- **Quality of IP formulation** : how closely its LP relaxation approximates $\mathrm{conv}(T)$
- For an IP and two equivalent formulations A, B: **A is stronger than B** if $P_A \subset P_B$
- **Constraints** play a more subtle role in IPs than in LPs
  - Adding valid constraints for T that cut off fractional points from P is very useful!
  - More constraints not necessarily worse in IP!

1. Discuss a few **ideal formulations** : $P = \mathrm{conv}(T)$
2. Discuss how to **improve** formulations by adding **cuts**
3. Discuss **algorithms/solution approaches**

Ideal Formulations

Setup:

- $P = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ polyhedral set, with $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$
- **Goal:** conditions on A so that P **is integral**, i.e., $P = \mathrm{conv}\, x \in P : x \in \mathbb{Z}^n$

Can anyone recall Cramer's rule?

Ideal Formulations

Setup:

- $P = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ polyhedral set, with $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$
- **Goal:** conditions on A so that P **is integral**, i.e., $P = \operatorname{conv} x \in P : x \in \mathbb{Z}^n$

Can anyone recall Cramer's rule?

Proposition (Cramer's Rule)

Let $A \in \mathbb{R}^{n \times n}$ be a nonsingular matrix. For $b \in \mathbb{R}^n$,

$$Ax = b \quad \Longrightarrow \quad x = A^{-1}b \quad \Longrightarrow \quad x_i = \frac{\det(A^i)}{\det(A)}, \ \forall i,$$

where A^i is the matrix with columns $A^i_j = A_j$ for all $j \in \{1, \dots, n\} \setminus \{i\}$ and $A^i_i = b$.

Ideal Formulations

Setup:

- $P = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ polyhedral set, with $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$
- **Goal:** conditions on A so that P **is integral**, i.e., $P = \operatorname{conv} x \in P : x \in \mathbb{Z}^n$

Can anyone recall Cramer's rule?

Proposition (Cramer's Rule)

Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix. For $b \in \mathbb{R}^n$,

$$Ax = b \quad \Longrightarrow \quad x = A^{-1}b \quad \Longrightarrow \quad x_i = \frac{\det(A^i)}{\det(A)},\ \forall i,$$

where A^i is the matrix with columns $A^i_j = A_j$ for all $j \in \{1,\dots,n\} \setminus \{i\}$ and $A^i_i = b$.

If $\det(A) \in \{1, -1\}$, that would be nice!

(Total) Unimodularity

Definition

1. $A \in \mathbb{Z}^{m \times n}$ of full row rank is **unimodular** if the $\det(A_B) \in \{1, -1\}$ for every basis B.
2. $A \in \mathbb{Z}^{m \times n}$ is **totally unimodular** if the determinant of each square submatrix of A is 0, 1, or -1.

- **Unimodularity** allows handling standard form $\{x \in \mathbb{Z}^n_+ \mid Ax = b\}$
- **Total Unimodularity (TU)** allows handling inequality form $\{x \in \mathbb{Z}^n_+ \mid Ax \leq b\}$

(Total) Unimodularity

Definition

1. $A \in \mathbb{Z}^{m \times n}$ of full row rank is **unimodular** if the $\det(A_B) \in \{1, -1\}$ for every basis B.
2. $A \in \mathbb{Z}^{m \times n}$ is **totally unimodular** if the determinant of each square submatrix of A is 0, 1, or -1.

- **Unimodularity** allows handling standard form $\{x \in \mathbb{Z}^n_+ \mid Ax = b\}$
- **Total Unimodularity (TU)** allows handling inequality form $\{x \in \mathbb{Z}^n_+ \mid Ax \leq b\}$
- **Note:** a TU matrix must belong to $\{0, 1, -1\}^{m \times n}$, but not a unimodular matrix:

$$\text{e.g.} \quad A = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$$

(Total) Unimodularity

Definition

1. $A \in \mathbb{Z}^{m\times n}$ of full row rank is **unimodular** if the $\det(A_B) \in \{1, -1\}$ for every basis B.
2. $A \in \mathbb{Z}^{m\times n}$ is **totally unimodular** if the determinant of each square submatrix of A is 0, 1, or -1.

- **Unimodularity** allows handling standard form $\{x \in \mathbb{Z}^n_+ \mid Ax = b\}$
- **Total Unimodularity (TU)** allows handling inequality form $\{x \in \mathbb{Z}^n_+ \mid Ax \leq b\}$
- **Note:** a TU matrix must belong to $\{0, 1, -1\}^{m\times n}$, but not a unimodular matrix:

$$\text{e.g.}\quad A = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$$

- Will provide easier ways to test for U and TU, but first let's see why we care...

(Total) Unimodularity Yields Integral LP Relaxations

Theorem

1. *The matrix* $A \in \mathbb{Z}^{m \times n}$ *of full row rank* **is unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.
2. *The matrix* A **is totally unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.

(Total) Unimodularity Yields Integral LP Relaxations

Theorem

1. *The matrix* $A \in \mathbb{Z}^{m \times n}$ *of full row rank* **is unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.
2. *The matrix* A **is totally unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.

Proof. (a) "$\Rightarrow$" Because A unimodular, for any $b \in \mathbb{Z}^m$ with $P(b) \neq \emptyset$, any basic feasible solution $x = (x_B, x_N) \in P(b)$ must satisfy $x_B = A_B^{-1} b \in \mathbb{Z}^{|B|}$.

(Total) Unimodularity Yields Integral LP Relaxations

Theorem

1. *The matrix* $A \in \mathbb{Z}^{m \times n}$ *of full row rank* **is unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.
2. *The matrix* A **is totally unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.

Proof. (a) "$\Rightarrow$" Because A unimodular, for any $b \in \mathbb{Z}^m$ with $P(b) \neq \emptyset$, any basic feasible solution $x = (x_B, x_N) \in P(b)$ must satisfy $x_B = A_B^{-1} b \in \mathbb{Z}^{|B|}$.

"$\Leftarrow$" We have that $P(b) \neq \emptyset$ is integral $b \in \mathbb{Z}^m$. Let B be any basis of A.

- Sufficient to prove that A_B^{-1} is integral; (A_B integral and $\det(A_B) \cdot \det(A_B^{-1}) = 1$ would imply that $\det(A_B) \in \{1, -1\}$ and thus A is unimodular)

(Total) Unimodularity Yields Integral LP Relaxations

Theorem

1. *The matrix* $A \in \mathbb{Z}^{m \times n}$ *of full row rank* **is unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.
2. *The matrix* A **is totally unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.

Proof. (a) "$\Rightarrow$" Because A unimodular, for any $b \in \mathbb{Z}^m$ with $P(b) \neq \emptyset$, any basic feasible solution $x = (x_B, x_N) \in P(b)$ must satisfy $x_B = A_B^{-1} b \in \mathbb{Z}^{|B|}$.

"$\Leftarrow$" We have that $P(b) \neq \emptyset$ is integral $b \in \mathbb{Z}^m$. Let B be any basis of A.

- Sufficient to prove that A_B^{-1} is integral; (A_B integral and $\det(A_B) \cdot \det(A_B^{-1}) = 1$ would imply that $\det(A_B) \in \{1, -1\}$ and thus A is unimodular)
- To prove A_B^{-1} integral, consider $b = A_B \cdot z + e_i$ where z is an integral vector
- Then $A_B^{-1} \cdot b = z + A_B^{-1} e_i$

(Total) Unimodularity Yields Integral LP Relaxations

Theorem

1. *The matrix* $A \in \mathbb{Z}^{m \times n}$ *of full row rank* **is unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.
2. *The matrix* A **is totally unimodular if and only if** *the polyhedron* $P(b) = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ **is integral** *for all* $b \in \mathbb{Z}^m$ *with* $P(b) \neq \emptyset$.

Proof. (a) "$\Rightarrow$" Because A unimodular, for any $b \in \mathbb{Z}^m$ with $P(b) \neq \emptyset$, any basic feasible solution $x = (x_B, x_N) \in P(b)$ must satisfy $x_B = A_B^{-1} b \in \mathbb{Z}^{|B|}$.

"$\Leftarrow$" We have that $P(b) \neq \emptyset$ is integral $b \in \mathbb{Z}^m$. Let B be any basis of A.

- Sufficient to prove that A_B^{-1} is integral; (A_B integral and $\det(A_B) \cdot \det(A_B^{-1}) = 1$ would imply that $\det(A_B) \in \{1, -1\}$ and thus A is unimodular)
- To prove A_B^{-1} integral, consider $b = A_B \cdot z + e_i$ where z is an integral vector
- Then $A_B^{-1} \cdot b = z + A_B^{-1} e_i$
- By choosing z large so $z + A_B^{-1} e_i \geq 0$, we obtain a b.f.s. for $P(b)$
- Because $P(b)$ integral, $A_B^{-1} e_i$ must be integral
- Repeat argument for all e_i to proves that A_B^{-1} is integral.

(b) Similar logic, omitted (see notes)

Checking for Total Unimodularity

Proposition

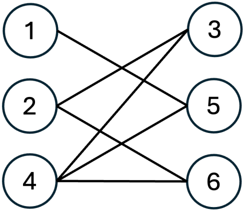
Consider a matrix $A \in \{0, 1, -1\}^{m \times n}$. The following are equivalent:

1. *A is totally unimodular.*
2. *A^T is totally unimodular.*
3. *$[A^T \; -A^T \; I \; -I]$ is totally unimodular.*
4. *$\{x \in \mathbb{R}^n_+ \mid Ax = b, 0 \leq x \leq u\}$ is integral for all integral b, u.*
5. *$\{x \mid a \leq Ax \leq b, \ell \leq x \leq u\}$ is integral for all integral a, b, ℓ, u.*
6. *Each collection of columns of A can be partitioned into two parts so that the sum of the columns in one part minus the sum of the columns in the other part is a vector with entries $0, +1$, and -1. (By part 2, a similar result also holds for the rows of A.)*
7. *Each nonsingular submatrix of A has a row with an odd number of non-zero components.*
8. *The sum of entries in any square submatrix with even row and column sums is divisible by four.*
9. *No square submatrix of A has determinant $+2$ or -2.*

#6 perhaps most useful in practice...

Examples of TU Matrices #1

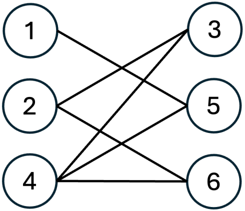
- $G = (\mathcal{N}, \mathcal{E})$ **undirected graph**
- $A \in \{0,1\}^{|\mathcal{N}| \times |\mathcal{E}|}$ is the node-edge incidence matrix of G
  $A_{i,e} = 1$ *if and only if* $i \in e$



	$\{1,5\}$	$\{2,3\}$	$\{2,6\}$	$\{4,3\}$	$\{4,5\}$	$\{4,6\}$
1	1	0	0	0	0	0
2	0	1	1	0	0	0
3	0	1	0	1	0	0
4	0	0	0	1	1	1
5	1	0	0	0	1	0
6	0	0	1	0	0	1

Examples of TU Matrices #1

- $G = (\mathcal{N}, \mathcal{E})$ **undirected graph**
- $A \in \{0,1\}^{|\mathcal{N}| \times |\mathcal{E}|}$ is the node-edge incidence matrix of G
  $A_{i,e} = 1$ *if and only if* $i \in e$



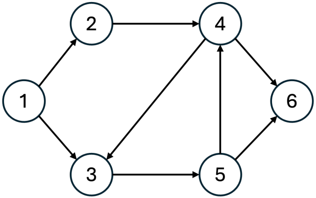
	$\{1,5\}$	$\{2,3\}$	$\{2,6\}$	$\{4,3\}$	$\{4,5\}$	$\{4,6\}$
1	1	0	0	0	0	0
2	0	1	1	0	0	0
3	0	1	0	1	0	0
4	0	0	0	1	1	1
5	1	0	0	0	1	0
6	0	0	1	0	0	1

- A **is TU if and only if** G **is bipartite**
- Bipartite matching problems have integral LP relaxations...

Examples of TU Matrices #2

- $D = (V, A)$ is a **directed graph**
- M is the $V \times A$ incidence matrix of D

$$M_{v,a} = \begin{cases} 1 & \text{if and only if } a = (\cdot, v) \text{ (arc } a \text{ enters node } v\text{)} \\ -1 & \text{if and only if } a = (v, \cdot) \text{ (arc } a \text{ leaves node } v\text{)} \\ 0 & \text{otherwise.} \end{cases}$$

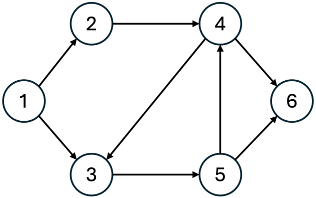


	$(1,2)$	$(1,3)$	$(2,4)$	$(4,3)$	$(3,5)$	$(5,4)$	$(4,6)$	$(5,6)$
1	-1	-1	0	0	0	0	0	0
2	1	0	-1	0	0	0	0	0
3	0	1	0	1	-1	0	0	0
4	0	0	1	-1	0	1	-1	0
5	0	0	0	0	1	-1	0	-1
6	0	0	0	0	0	0	1	1

Examples of TU Matrices #2

- $D = (V, A)$ is a **directed graph**
- M is the $V \times A$ incidence matrix of D

$$M_{v,a} = \begin{cases} 1 & \text{if and only if } a = (\cdot, v) \text{ (arc } a \text{ enters node } v\text{)} \\ -1 & \text{if and only if } a = (v, \cdot) \text{ (arc } a \text{ leaves node } v\text{)} \\ 0 & \text{otherwise.} \end{cases}$$

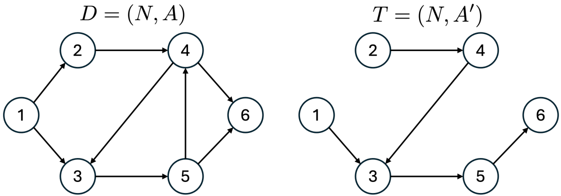


	$(1,2)$	$(1,3)$	$(2,4)$	$(4,3)$	$(3,5)$	$(5,4)$	$(4,6)$	$(5,6)$
1	-1	-1	0	0	0	0	0	0
2	1	0	-1	0	0	0	0	0
3	0	1	0	1	-1	0	0	0
4	0	0	1	-1	0	1	-1	0
5	0	0	0	0	1	-1	0	-1
6	0	0	0	0	0	0	1	1

- Then M **is TU**
- **Network flow problems** (e.g., Prosche Motors) with integral arc capacities and integral supply/demand have integral LP relaxations

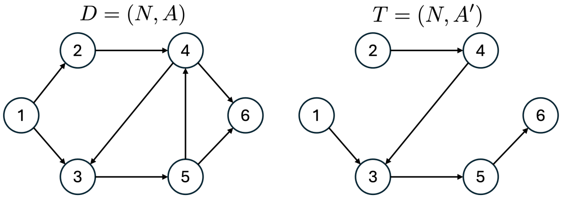
Examples of TU Matrices #3

- $D = (V, A)$ is a **directed graph**, $T = (V, A_0)$ is a directed tree on V



Examples of TU Matrices #3

- $D = (V, A)$ is a **directed graph**, $T = (V, A_0)$ is a directed tree on V

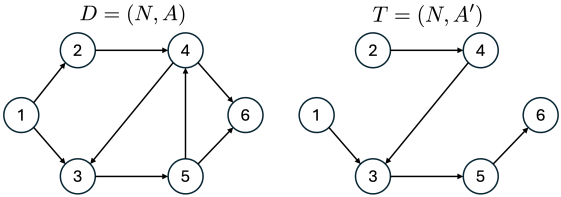


- M is the $A_0 \times A$ matrix defined as follows: for $a = (v, w) \in A$ and $a' \in A_0$,

$$M_{a',a} = \begin{cases} +1 & \text{if the unique } v - w \text{ path in } T \text{ passes through } a' \text{ forwardly} \\ -1 & \text{if the unique } v - w \text{ path in } T \text{ passes through } a' \text{ backwardly} \\ 0 & \text{if the unique } v - w \text{ path in } T \text{ does not pass through } a'. \end{cases}$$

Examples of TU Matrices #3

- $D = (V, A)$ is a **directed graph**, $T = (V, A_0)$ is a directed tree on V



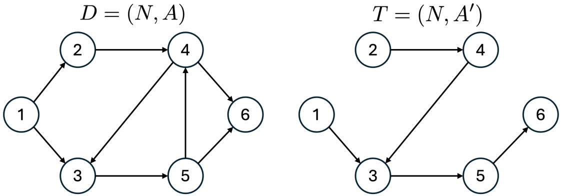
- M is the $A_0 \times A$ matrix defined as follows: for $a = (v, w) \in A$ and $a' \in A_0$,

$$M_{a',a} = \begin{cases} +1 & \text{if the unique } v - w \text{ path in } T \text{ passes through } a' \text{ forwardly} \\ -1 & \text{if the unique } v - w \text{ path in } T \text{ passes through } a' \text{ backwardly} \\ 0 & \text{if the unique } v - w \text{ path in } T \text{ does not pass through } a'. \end{cases}$$

	$(1,2)$	$(1,3)$	$(2,4)$	$(4,3)$	$(3,5)$	$(5,4)$	$(4,6)$	$(5,6)$
$(1,3)$	1	1	1	0	0	0	0	0
$(2,4)$	-1	0	0	0	0	0	0	0
$(4,3)$	-1	0	0	1	0	-1	1	0
$(3,5)$	0	0	0	0	1	-1	1	0
$(5,6)$	0	0	0	0	0	0	1	1

Examples of TU Matrices #3

- $D = (V, A)$ is a **directed graph**, $T = (V, A_0)$ is a directed tree on V



- M is the $A_0 \times A$ matrix defined as follows: for $a = (v, w) \in A$ and $a' \in A_0$,

$$M_{a',a} = \begin{cases} +1 & \text{if the unique } v - w \text{ path in } T \text{ passes through } a' \text{ forwardly} \\ -1 & \text{if the unique } v - w \text{ path in } T \text{ passes through } a' \text{ backwardly} \\ 0 & \text{if the unique } v - w \text{ path in } T \text{ does not pass through } a'. \end{cases}$$

- Then M **is TU**

- All previous examples were **special cases** of this

- Paul Seymour: **all TU matrices** generated from network matrices and **two** other matrices