MAT3007 Optimization
Lecture 11 Integer Optimization

Yuang Chen

School of Data Scienc
The Chinese University of Hong Kong, Shenzhen

June 30, 2025

Midterm Results

- Average: 55.6
- Medium: 59.5
- Percentile Table

Percentile	Score
Max	95
20%	74
40%	63.8
50%	59.5
60%	53.2
80%	37.6
Min	2

- Grading: 20% Homework, 40% Midterm, 40% Final
- I will give as many A, A-, and B+ grades as the university policy allows (the max possible percentage).

Second Half of Semester

What are in final exam?

- Modeling (MILP)
- Branch-and-Bound Algorithm
- Convexity
- Optimality Conditions
- KKT Conditions
- Algorithms for Unconstrained Optimization

IP

Nonlinear/General

Final exam: July 24 1:30 - 4:30 Pm

Outline

Outline

Integer Linear Program

ILP

An integer linear program (IP) is a linear program with the additional constraint that all variables must be integers:

$$
\begin{aligned}
\text{minimize}_{\boldsymbol{x}} \quad & \boldsymbol{c}^T \boldsymbol{x} \\
\text{subject to} \quad & A\boldsymbol{x} = \boldsymbol{b} \\
& \boldsymbol{x} \geq 0 \\
& \boldsymbol{x} \in \mathbb{Z}^n
\end{aligned}
$$

Z^+: set of nonnegative integers

$\to$ discrete variable

Here we use $\mathbb{Z}$ to denote the set of integers.

- One may also encounter mixed integer programs (MIP), in which one set of variables must be integer and the rest are allowed to be continuous

Mixed Integer Linear Program (MILP)

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & c^\mathsf{T} x \\
\text{s.t.} \quad & a_j^\mathsf{T} x \le b_j, \quad j = 1, 2, .., m \\
& x \in \underbrace{\mathbb{R}^{n-p}}_{\text{continuous}} \times \underbrace{\mathbb{Z}^p}_{\text{integer}}
\end{aligned}
$$

A Special Case: Binary IP

A special and important class of integer program is those where the integer variables are required to be binary, that is, they are required to take values of 0 or 1.

$$
\begin{aligned}
\min \quad & f(x) = c^T x \\
\text{s.t.} \quad & g_j(x) \leq 0, \quad j = 1, 2, \ldots, m \\
& x \in \mathbb{R}^{n-p} \times \{0, 1\}^p
\end{aligned}
$$

$a_j^T x - b_j$

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

The variables in a binary IP are also called indicator variables. They are powerful in modeling.

- Indivisible decisions
- Yes/No choices
- Logical conditions
- Nonconvex functions and sets

define $x_A = \begin{cases} 1 & \text{if event A happens} \\ 0 & \text{otherwise} \end{cases}$

Local vs Global Optimal

- Local optimal solutions may be very far from a global optimal solution in integer optimization.
- Example

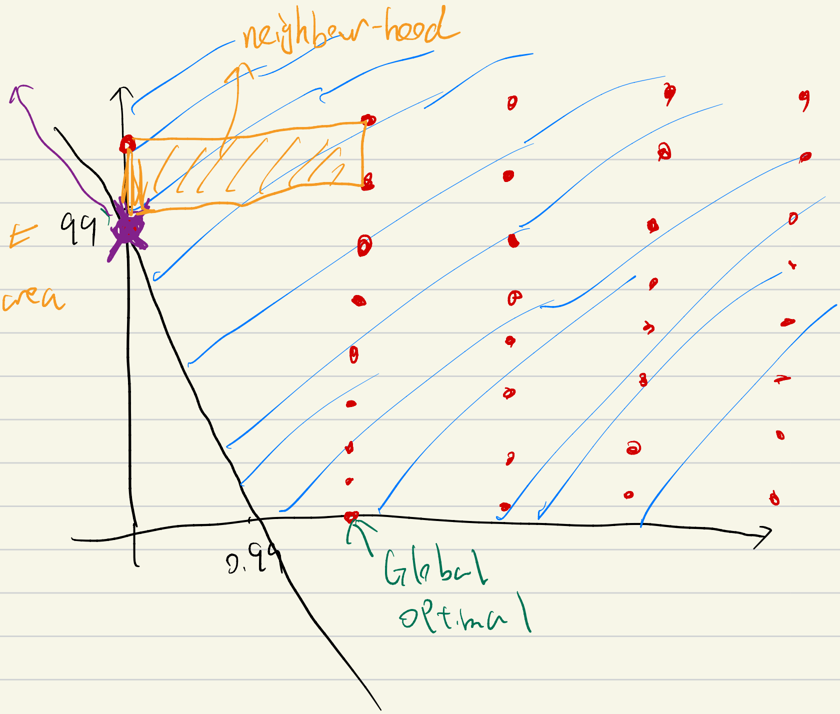
feasible region for IP is not convex set

$$\begin{aligned} \min \quad & x + y \\ \text{s.t.} \quad & 100x + y \geq 99 \qquad x, y \geq 0 \\ & x, y \in \mathbb{Z}^+ \end{aligned}$$

- The solution (1,0) is a global optimal solution with objective value 1.
- The solution (0,99) is a "local" optimal solution with objective 99. It has equal or better objective than all "neighboring" feasible integer solutions.

local optimal

(3,99) gives smallest x+y ≤ 99 in the orange area



Example: Knapsack Problem

John is planning for a trip. There are n items he would like to bring with him.

- The ith item has value v_i
- The weight of ith item is a_i
- His bag has a maximum allowable weight C
- He wants to bring as much value as possible

$$x_i = \begin{cases} 1 & \text{if bring item } i \\ 0 & \text{o.w.} \end{cases}$$

Decision variables:

- x_i: whether to bring ith item or not. $x_i \in \{0, 1\}$

Optimization problem:

$$\begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \sum_{i=1}^{n} v_i x_i \\ \text{subject to} & \sum_{i=1}^{n} a_i x_i \leq C \\ & x_i \in \{0, 1\} \quad \forall i = 1, \ldots n \end{array}$$

It is a binary optimization problem

Greedy Method: Enumeration

- Small discrete optimization problems can be solved by enumerating all possibilities
- Example:

$$
\begin{aligned}
\text{maximize}_{\boldsymbol{x}} \quad & \textstyle\sum_{i=1}^n v_i x_i \\
\text{subject to} \quad & \textstyle\sum_{i=1}^n a_i x_i \le C \\
& x_i \in \{0,1\} \quad \forall i = 1, \ldots, n
\end{aligned}
$$

i	1	2	3	4
	1	0	0	0
	1	1	0	0
	1	1	1	0
	0	1	0	0
		⋮		

- There are 2^n possible values of the binary variables
- If n is small, we can check each to see if it is feasible, and then choose a feasible solution with maximum objective value

Enumeration Result

```python
# Solving an example discrete optimization problem by enumeration
import itertools
from random import randint
import time

# generate random problem instance
n = 25
r = [randint(0,9) for j in range(n)]
c = [randint(0,9) for j in range(n)]
B = sum(c)/2.0

# all possible solutions
solutions = list(itertools.product([0, 1], repeat=n))

xbest = []
vbest = -1

# check each solution
start_time = time.time()
for x in solutions:
    # check feasibility
    val = sum( [c[j]*x[j] for j in range(n)])
    if val <= B:
        obj = sum( [r[j]*x[j] for j in range(n)])
        if obj > vbest:
            vbest = obj
            xbest = x

print 'n:', n
print 'Time:', (time.time() - start_time)
```

n	Time (secs)
5	0.0012
10	0.0049
15	0.1943
20	7.4745
25	311.9747

With n=50 estimated time > 330 years!!

Computation Complexity for Optimization Problems

- Linear programming and many important classes of convex optimization problems (e.g. conic programming and semidefinite programming) are known to be in **P**.
- Most integer optimization problems are **NP-hard**.
- It is widely believed that there is little hope of finding a polynomial time algorithm for discrete optimization problems.

Outline

If-Then Condition I

1
- If A is chosen, then B is chosen.

$$x_A \leq x_B$$

2
- If A is chosen, then B is not chosen.

$$x_A + x_B \leq 1$$

3
- If A is chosen, then ~~both~~ B or C are chosen.

$$x_A \leq x_B + x_C$$

4
- If B or C is chosen, then A is chosen.

$$x_A \geq x_B, \quad x_A \geq x_C$$

$$X_A = \begin{cases} 1 & \text{if } A \text{ is chosen} \\ 0 & \text{o.w.} \end{cases}$$

$$X_B = \begin{cases} 1 & \text{if } B \text{ is chosen} \\ 0 & \text{o.w.} \end{cases}$$

1. $X_A \leq X_B$

A is chosen $\Rightarrow X_A = 1 \Rightarrow 1 \leq X_B \Rightarrow X_B = 1$

$\Downarrow$

B is chosen

2. $X_A + X_B \leq 1$

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

$X_A = 1 \Rightarrow 1 + X_B \leq 1 \Rightarrow X_B \leq 0 \Rightarrow X_B = 0$

3. $X_A \leq X_B + X_C$

$X_A = 1 \Rightarrow X_B + X_C \geq 1 \Rightarrow X_B = 1$ or $X_C = 1$ or both

4. $X_A \geq X_B, \quad X_A \geq X_C$

$$\begin{cases} X_B = 1 \Rightarrow X_A \geq 1 \Rightarrow X_A = 1 \\ X_C = 1 \Rightarrow X_A \geq 1 \Rightarrow X_A = 1 \end{cases}$$

If A is chosen, then B is not chosen.

$$X_A + X_B \leq 1$$

- $\frac{X_A}{3} + X_B \leq \frac{2}{3}$ ✗ (Wrong)
- $X_A + X_B \leq \frac{4}{3}$ (Right)

$X_A = 1 \Rightarrow \frac{1}{3} + X_B \leq \frac{2}{3} \Rightarrow X_B \leq \frac{1}{3} \Rightarrow X_B = 0$

$X_A = 1 \Rightarrow 1 + X_B \leq \frac{4}{3} \Rightarrow X_B \leq \frac{1}{3} \Rightarrow X_B = 0$

$$\frac{x_A}{3} + x_B \le \frac{2}{3}$$

$$x_A = 0 \Rightarrow x_B \le \frac{2}{3} \Rightarrow x_B = 0$$

This is additional constraint!

If-Then Condition II

- If A is chosen, then both B and C are chosen.

$$x_A \le x_B, \quad x_A \le x_C$$

$x_A = 1 \Rightarrow \begin{cases} x_B \ge 1 \\ x_C \ge 1 \end{cases} \Rightarrow \begin{cases} x_B = 1 \\ x_C = 1 \end{cases}$

- If B and C are chosen, then A is chosen.

$$x_A \ge x_B + x_C - 1$$

$x_B = x_C = 1$
$\Downarrow$
$x_A \ge 1 \Rightarrow x_A = 1$

- A is chosen if and only if B and C are chosen.

$$\left.\begin{array}{l} x_A \le x_B, \quad x_A \le x_C \\ x_A \ge x_B + x_C - 1 \end{array}\right\} \text{linear}$$

$$\Downarrow$$

$$x_A = \underbrace{x_B \cdot x_C}_{\text{AND}} \rightarrow \text{nonlinear}$$

If-Then Condition III

1 • If A is chosen, then $x \leq 8$.

$$x \leq 8 + M(1 - y)$$

2 • If A is chosen, then $x \geq 8$.

$$x \geq 8 - M(1 - y)$$

3 • If $x < 8$, then A is chosen.

$$x \geq 8 - My$$

4 • If $x > 8$, then A is chosen.

$$x \leq 8 + My$$

$y = \begin{cases} 1 & \text{if A is chosen} \\ 0 & \text{o.w.} \end{cases}$

1. $x \le 8 + M(1-y)$

M: Big Number

$M = +\infty$

Big-'M' Notation

$y = 1 \Rightarrow x \le 8$

Try: $x \le 8y$ (wrong!)

$y = 0 \Rightarrow x \le 0$ Additional!

$y = 0 \Rightarrow x \le 8 + M \Rightarrow x \le +\infty$ 'redundant'

2. $x \ge 8 - M(1-y)$

$y = 1 \Rightarrow x \ge 8$

$y = 0 \Rightarrow x \ge 8 - M \Rightarrow x \ge -\infty$

3. If $x < 8$, then A is chosen.

Contra-Positive Statement

If A, then B

$\Updownarrow$

If not B, then not A

If A is not chosen, then $x \geq 8$.

$$x \geq 8 - My$$

$y=0 \Rightarrow x \geq 8$

$y=1 \Rightarrow x \geq 8-M \Rightarrow x \geq -\infty$

4. If $x > 8$, then A is chosen

$\Updownarrow$

If A is not chosen, then $x \leq 8$

$$x \leq 8 + My$$

$y=0 \Rightarrow x \leq 8$

$y=1 \Rightarrow x \leq 8+M \Rightarrow x \leq +\infty$

If-Then Condition IV

1. - If $x_1 > 4$, then $x_2 \leq 8$.

$$\begin{cases} x_1 \leq 4 + My \\ x_2 \leq 8 + M(1-y) \\ y \in \{0,1\} \end{cases}$$

2. - If $x_1 < 4$, then $x_2 \geq 8$.

$$\begin{cases} x_1 \geq 4 - My \\ x_2 \geq 8 - M(1-y) \\ y \in \{0,1\} \end{cases}$$

1. If $x_1 > 4$, then $x_2 \le 8$

$$\Updownarrow$$

$$\begin{cases} \text{If } x_1 > 4, \text{ then } A \text{ is chosen.} \\ \text{If } A \text{ is chosen, then } x_2 \le 8. \end{cases}$$

2. If $x_1 < 4$, then $x_2 \ge 8$

$$\Downarrow$$

$$\begin{cases} \text{If } x_1 < 4, \text{ then } A \text{ is chosen.} \\ \text{If } A \text{ is chosen, then } x_2 \ge 8. \end{cases}$$

General "If ... then ... " Conditions

If $f(x_1, x_2, ..., x_n) > a$, then $g(x_1, x_2, ..., x_n) \geq b$.

$$\begin{cases} f(x_1, x_2, ..., x_n) \leq a + My \\ g(x_1, x_2, ..., x_n) \geq b - M(1-y) \\ y \in \{0, 1\} \end{cases}$$

$\begin{cases} \text{If } f(x_1, \cdots, x_n) > a, \text{ then } A \text{ is chosen} \\ \text{If } A \text{ is chosen, then } g(x_1, \cdots, x_n) \geq b \end{cases}$

Outline

Disjunctive conditions: "Either ... Or ..." Conditions

Disjunctive constraints: **at least one** of a set of constraints is satisfied. It's possible the set of constraints are all satisfied.

$$\text{Either A or B} \iff \text{If not A, then B}$$
$$\text{Either A or B} \iff \text{If not B, then A}$$

- Either $x_1 \leq 4$ or $x_2 \leq 8$

$$\text{Either } x_1 \leq 4 \text{ or } x_2 \leq 8$$
$$\Updownarrow$$
$$\text{If } x_1 > 4\text{, then } x_2 \leq 8$$
$$\Updownarrow$$
$$\begin{cases} x_1 \leq 4 + My \\ x_2 \leq 8 + M(1-y) \\ y \in \{0,1\} \end{cases}$$

General Disjunctive Constraints

$$\text{Either } f(x_1, x_2, ..., x_n) \leq a \text{ or } g(x_1, x_2, ..., x_n) \leq b$$

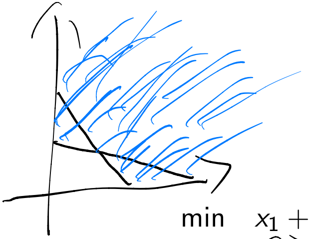
$$\Updownarrow$$

$$\text{If } f(x_1, x_2, ..., x_n) > a, \text{ then } g(x_1, x_2, ..., x_n) \leq b$$

$$\Updownarrow$$

$$\begin{cases} f(x_1, x_2, ..., x_n) \leq a + My \\ g(x_1, x_2, ..., x_n) \leq b + M(1 - y) \\ y \in \{0, 1\} \end{cases}$$

Example



non convex set

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & \text{either } 2x_1 + x_2 \geq 6 \text{ or } x_1 + 2x_2 \geq 7 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$$\begin{cases} 2x_1 + x_2 \geq 6 - My \\ x_1 + 2x_2 \geq 7 - M(1-y) \\ y \in \{0, 1\} \end{cases}$$

Alternative Way

Either $f(x_1, x_2, ..., x_n) \leq a$ or $g(x_1, x_2, ..., x_n) \leq b$

$y_1 = 1$ $\quad$ $y_2 = 1$

$$\begin{cases} \text{If } y_1 = 1, \text{then } f(x_1, x_2, ..., x_n) \leq a \\ \text{If } y_2 = 1, \text{then } g(x_1, x_2, ..., x_n) \leq b \\ y_1 + y_2 \geq 1 \\ y_1, y_2 \in \{0, 1\} \end{cases}$$

$y_1 = 1$ $\quad$ $y_2 = 1$

Either $f_1(x_1, x_2, ..., x_n) \leq a_1$ or $f_2(x_1, x_2, ..., x_n) \leq a_2$ or $f_3(x_1, x_2, ..., x_n) \leq a_3$

$y_3 = 1$

$$\begin{cases} \text{If } y_1 = 1, \text{then } f_1(x_1, x_2, ..., x_n) \leq a_1 \\ \text{If } y_2 = 1, \text{then } f_2(x_1, x_2, ..., x_n) \leq a_2 \\ \text{If } y_3 = 1, \text{then } f_3(x_1, x_2, ..., x_n) \leq a_3 \\ y_1 + y_2 + y_3 \geq 1 \\ y_1, y_2, y_3 \in \{0, 1\} \end{cases}$$

Common Mistake

If $y = 1$, then $f(x) \le a$

$$\not\Downarrow$$

$$f(x) \le ay \quad ✗$$

Correct one:

$$f(x) \le a + M(1-y) \quad ✓$$

Alternative Formulation for Disjunctive Constraints

Either $f_1(x_1, x_2, ..., x_n) \leq a_1$ or $f_2(x_1, x_2, ..., x_n) \leq a_2$ or $f_3(x_1, x_2, ..., x_n) \leq a_3$

$$\begin{cases} f_1(x_1, x_2, ..., x_n) \leq a_1 + M(1 - y_1) \\ f_2(x_1, x_2, ..., x_n) \leq a_2 + M(1 - y_2) \\ f_3(x_1, x_2, ..., x_n) \leq a_3 + M(1 - y_3) \\ y_1 + y_2 + y_3 \geq 1 \\ y_1, y_2, y_3 \in \{0, 1\} \end{cases}$$

Outline

Discrete variables

$x \in \{1.23, 2.56, 7.85, 9.11\}$

- A discrete variable has a domain of the form

$$y_1, y_2, \ldots, y_k$$

$$x \in \{a_1, a_2 \ldots, a_K\}$$

- If $a_1, a_2 \ldots, a_K$ are contiguous integers, we can model as

$$\underline{a_1 \leq x \leq a_k, x \in \mathbb{Z}}$$

- Otherwise we can model as

$$\rightarrow \quad x = \sum_{k=1}^{K} a_k y_k, \ \sum_{k=1}^{K} y_k = 1, y_k \in \{0, 1\}, k = 1, .., K$$

$x = a_1 y_1 + a_2 y_2 + \cdots + a_K y_K, \ y_1 + y_2 + \cdots + y_K = 1, \ y_k \in \{0,1\}$

Semicontinuous variables

- A semicontinuous variable has a domain of the form

$$x \in [l_1, u_1] \cup [l_2, u_2] \cup \cdots \cup [l_K, u_K]$$

  where $l_1 \leq u_1 \leq \cdots \leq l_K \leq u_k$
- One modeling approach for this system is

$$x = \sum_{k=1}^{K} z_k$$

$$l_k y_k \leq z_k \leq u_k y_k \quad k = 1, ..., K$$

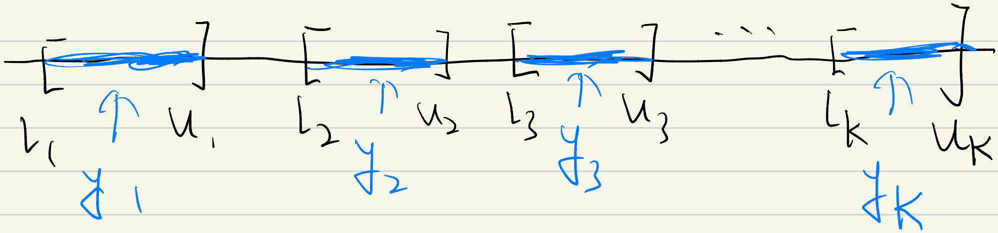
$$\sum_{k=1}^{K} y_k = 1$$

$$y_k \in \{0, 1\} \quad k = 1, ..., K$$

$x \in [1,2] \vee [3,4] \vee [5,6] \vee [7,8]$

$$\begin{cases} 1 \le x \le 2 \\ 3 \le x \le 4 \\ 5 \le x \le 6 \\ 7 \le x \le 8 \end{cases} \rightarrow \text{Wrong!}$$

infeasible



$$\begin{cases} \cancel{l_k y_k \le x \le u_k y_k}, \quad \forall k = 1, \dots, K \\ \sum_{k=1}^{K} y_k = 1 \\ y_k \in \{0, 1\}, \quad \forall k = 1, \dots, K \end{cases}$$

WRONG!

If $y_k = 0$, $\quad 0 \le x \le 0 \Rightarrow x = 0$ ✗

Correct:

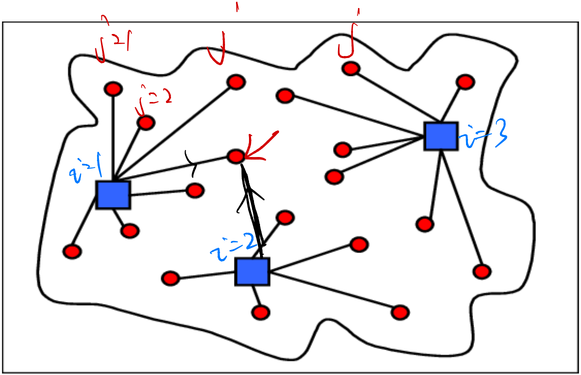
$$\begin{cases} l_k y_k \le z_k \le u_k y_k, \quad \forall k \\ x = \sum_{k=1}^{K} z_k \end{cases}$$

Outline

Example: Warehouse/Facility Location Problem

- There are n warehouses available for use
- Opening warehouse i has a fixed operating cost f_i
- There are m customers
- Customer j has demand d_j. It costs c_{ij} to ship one unit of product from warehouse i to customer j
- The objective is to satisfy all customers' demands, while minimizing the total costs (operating + shipment)

Facility Location Problem



Formulation

$y_i = \begin{cases} 1 & \text{if open warehouse } i \\ 0 & \text{o.w.} \end{cases}$

Decision variables:

- y_i: whether to open warehouse i or not ($y_i \in \{0, 1\}$)
- x_{ij}: how many units to ship from warehouse i to customer j

Objective function:

$$\text{minimize}_{\boldsymbol{x},\boldsymbol{y}} \quad \sum_{i=1}^{n} f_i y_i + \sum_{i,j} c_{ij} x_{ij}$$

Constraints:

- Need to satisfy every customer's demand: $\sum_{i=1}^{n} x_{ij} \geq d_j$ $\forall j$
- A location can provide supply only if it is opened :

$$x_{ij} \leq d_j y_i \quad \begin{cases} \text{if } y_i = 0, \text{ then } x_{ij} = 0 \\ \text{if } y_i = 1, \text{ then } x_{ij} \leq d_j \end{cases}$$

Formulation

Therefore we can formulate the warehouse/facility location problem as follows:

$$
\begin{aligned}
\text{minimize}_{\boldsymbol{x},\boldsymbol{y}} \quad & \textstyle\sum_{i=1}^{n} f_i y_i + \sum_{i,j} c_{ij} x_{ij} \\
\text{subject to} \quad & \textstyle\sum_{i=1}^{n} x_{ij} \geq d_j, && \forall j \\
& x_{ij} \leq d_j y_i, && \forall i, j \\
& x_{ij} \geq 0, && \forall i, j \\
& y_i \in \{0, 1\}, && \forall i
\end{aligned}
$$

It is a mixed integer linear program.

Outline

LP Relaxation of an IP

$x_0 = 3$ Satisfy constraint $x^* = 3.3$

Integer Program (IP):

$$
\begin{aligned}
\text{IP} = \min \quad & c^\top x \\
\text{s.t.} \quad & Ax = b \\
& x \geq 0 \\
& x \in \mathbb{Z}^n
\end{aligned}
\qquad X^{IP}
$$

LP Relaxation of IP:

$$
\begin{aligned}
\text{LP} \leq \min \quad & c^\top x \\
\text{s.t.} \quad & Ax = b \\
& x \geq 0
\end{aligned}
\qquad X^{LP}
$$

- LP relaxation removes the integer requirement.
- The feasible region of the LP relaxation is a superset of the feasible region of the IP.

$$X^{IP} \subseteq X^{LP}$$

LP Relaxation as a Bound for IP

Theorem (LP Relaxation as a Bound for IP)

1. *For a maximization problem, the optimal value of the LP relaxation provides an upper bound for the optimal value of the IP.*
2. *For a minimization problem, the optimal value of the LP relaxation provides a lower bound for the optimal value of the IP.*

The difference between the optimal value of the LP and the IP is called the **integrality gap**

- For maximization problems, the integrality gap is $v^{LP} - v^{IP}$
- For minimization problems, the integrality gap is $v^{IP} - v^{LP}$

For Max

V^{LP}

Integrality Gap

GAP

V^{IP}

V^{0}

For Min

V^{0}

V^{IP}

GAP

Integrality Gap

V^{LP}

Use LP Relaxation as a Bound

Consider a maximization problem, suppose we solved the LP relaxation and get the optimal value is v^{LP} (with some fractional solution).

Then we use some method to obtain a feasible integral solution (e.g., by rounding the solution to the LP relaxation) and find the objective value is v^0.

Then the difference between the obtained solution and the optimal integral solution (whose objective value is v^{IP}) satisfies:

$$0 \leq v^{IP} - v^0 \leq v^{LP} - v^0$$

That is, one can use the LP relaxation solution to construct a bound on how good a certain integral solution is.

Some Good Cases

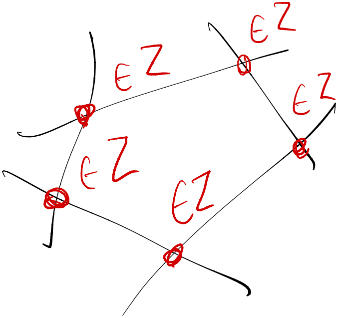
There are some cases where solving the LP relaxation can give one very good solution

Theorem

If the optimal solution to the LP relaxation is integral, then the solution must be optimal to the IP problem.

- Question: when?
- Answer: Remember for LP, there must exist an optimal solution that is a basic feasible solution. If we can guarantee that every BFS is integral, then the LP must have an integer optimal solution.

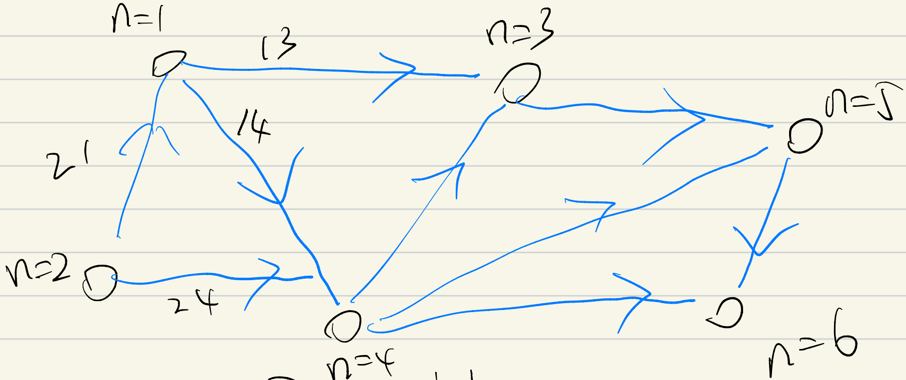
Outline

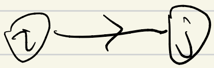


Network / Graph

$\mathcal{N}$

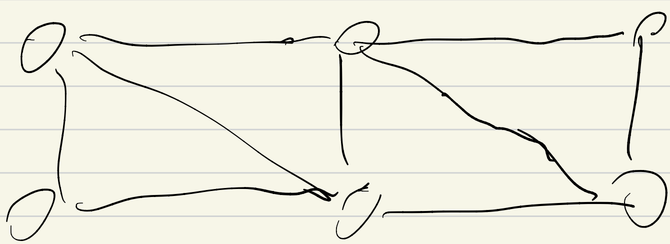
Directed Graph (Nodes, Arcs)





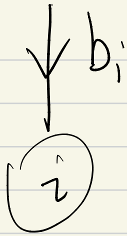
(i, j)

Undirected Graph (Nodes, Edges)



Minimum Cost Network Flow Model

- We are given a directed network $G = (N, A)$ with a set of nodes N and a set of arcs A.
- Each node $i \in N$ has an associated "supply" b_i. If $b_i > 0$ the node is a supply node, if $b_i < 0$ it is a demand node. We will assume that the network is balanced, i.e. $\sum_{i \in N} b_i = 0$.
- Each arc $(i, j) \in A$ has an associated cost c_{ij} and capacity u_{ij}.
- A flow on this network is a set of values on the arcs that obey capacities and satisfy flow conservation at the nodes.
- The goal is to find flows on each arc with minimum cost.



supply

b_i

$$b_i \begin{cases} >0 \rightarrow \text{supply} \\ <0 \rightarrow \text{demand} \\ =0 \rightarrow \text{transchipment} \end{cases}$$

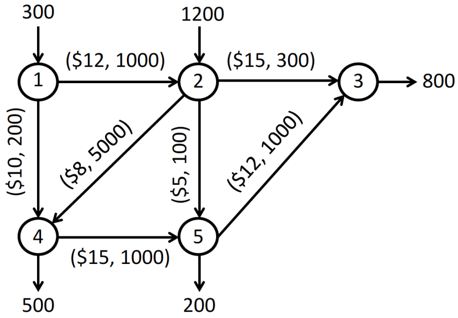
Minimum Cost Network Flow Problem

Let x_{ij} denote the flow from i to j

$$
\begin{aligned}
\min \quad & \sum_{(ij)\in A} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j:(ij)\in A} x_{ij} - \sum_{j:(ji)\in A} x_{ji} = b_i \quad \forall i \in N \\
& 0 \le x_{ij} \le u_{ij} \quad \forall (ij) \in A
\end{aligned}
$$

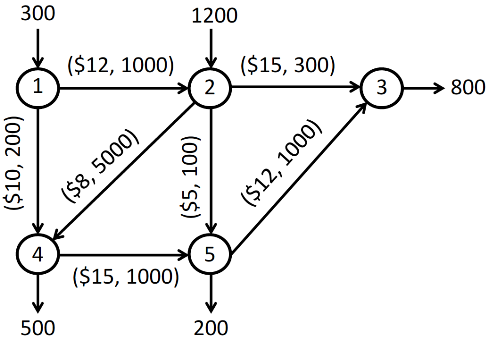
Example

- A company has two manufacturing plants and 3 distribution centers (nodes)
- The plants and DCs are connected by transportation channels (arcs)
- The production amount and demand at each node is shown
- The capacity and cost per unit for each channel (arc) is shown
- Find the cheapest way to move units from plants to DCs

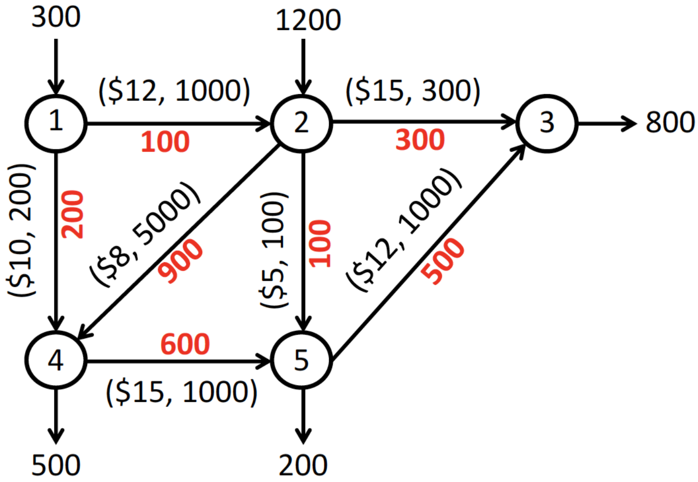


Example Formulation

$$
\begin{array}{ll}
\min & 12x_{12} + 10x_{14} + 15x_{23} + 8x_{24} \\
 & +5x_{25} + 15x_{45} + 12x_{53} \\
\text{s.t.} & x_{12} + x_{14} = 300 \\
 & x_{23} + x_{24} + x_{25} - x_{12} = 1200 \\
 & -x_{23} - x_{53} = -800 \\
 & x_{45} - x_{14} - x_{24} = -500 \\
 & x_{53} - x_{25} - x_{45} = -200 \\
 & 0 \le x_{12} \le 1000,\ \ 0 \le x_{14} \le 200, \\
 & 0 \le x_{23} \le 300, 0 \le x_{24} \le 5000 \\
 & 0 \le x_{25} \le 100,\ \ 0 \le x_{45} \le 1000 \\
 & 0 \le x_{53} \le 1000
\end{array}
$$



Example Solution



Integrality of Min Cost Network Flow

Theorem

For any minimum cost network flow problem (LP), if all supplies (b_i) and capacities (u_{ij}) are integers, then the problem has an optimal solution with integer flow on each arc.

Totally Unimodular Matrix

Definition

A matrix A is said to be *totally unimodular* (TU) if the determinant of each square submatrix of A is either 0, 1, or -1.

Examples:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Observation: If A is TU, then each element $a_{ij} \in \{-1, 0, 1\}$.

Theorem

In the network flow problem, the matrix of flow conservation constraints is totally unimodular.

One Sufficient Condition for TU

Theorem

Let A be an $m \times n$ matrix. Then the following conditions together are sufficient for A to be totally unimodular:

1. *Every column of A contains at most two non-zero entries;*
2. *Every entry in A is 0, $+1$, or -1;*
3. *The rows of A can be partitioned into two disjoint sets B and C such that*
   - (a) *If two non-zero entries in a column of A have the same sign, then the row of one entry is in B, and the row of the other in C;*
   - (b) *If two non-zero entries in a column of A have opposite signs, then the rows are both in B, or both in C.*

Example

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

- $B = \{1, 2\}$ and $C = \{3, 4\}$

$$A = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix}$$

- $B = \{1, 2, 3, 4\}$ and $C = \phi$

Cramer's Rule

Theorem

Let B be a nonsingular $m \times m$ matrix. Let x be a solution to $Bx = b$, then

$$x_j = \frac{\det(B^j)}{\det(B)} \quad \forall\, j = 1, \ldots, m$$

where B^j is B with the j-th column replaced by b.

Properties Totally Unimodular Matrix

Theorem

If the matrix A is TU and the vector b has integer entries, then the polyhedron

$$X = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\}$$

(if nonempty) has integral extreme points (i.e., each extreme point vector has integral entries).

Corollary

If the matrix A is TU and the vectors a, b, d, f have integer entries, then the following two polyhedra

$$\begin{aligned} X =& \{x \in \mathbb{R}^n : Ax \leq b\} \\ Y =& \{x \in \mathbb{R}^n : a \leq Ax \leq b, d \leq x \leq f\} \end{aligned}$$

(if nonempty) have integral extreme points.

Problems Equivalent to Min Cost Network Flow

- Transportation
- Transshipment
- Assignment $\longrightarrow$ Min Cost Network Flow
- Shortest Path
- Max Flow