Readings: - Chapter 1: Modeling

- Chapter 12: Nonlinear Programming

- Optimization problem:
  - Decision variables
  - objective functions
  - Constraints

<u>Example:</u> knapsack problem: Given a set of items, each with a mass and a value, determine the number (zero or one) of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

15 kg

bag

| 12kg, \$4 |, | 1kg, \$3 |, | 2k, \$6 |

items

n : items, x_i : whether or not choose item i,

v_i : value of item i, W = total allowed weight

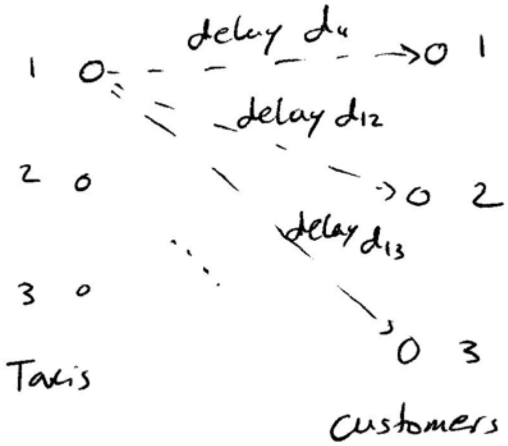
$$\max \sum_{i=1}^{n} v_i x_i \quad \Rightarrow \text{objective}$$

$$\left.\begin{array}{ll} \text{s.t.} & \sum_{i=1}^{n} w_i x_i \le W \\ & x_i \in \{0,1\} \quad i=1,\dots,n \\ & (x_i(x_i-1)=0) \end{array}\right\} \Rightarrow \text{Constraints.}$$

Decisions: $x_1, x_2, \dots, x_n$

Example: Assignment problem: A taxi firm has three taxis available and three customers wishing to be picked up as soon as possible.

Goal: speedy pickups.



$\rightarrow$ d_{ij} : given delay for taxi i to pick up customer j.

Decisions: x_{ij} : should Taxi i pick up customer j ? Yes (1) / No (0)

$i = 1, 2, 3$
$j = 1, 2, 3$

$$\Rightarrow \quad \sum_j x_{ij} = 1 \quad , \quad i = 1, 2, 3 \quad : \quad \text{Taxi constraint}$$

$$\sum_i x_{ij} = 1 \quad , \quad j = 1, 2, 3 \quad : \quad \text{Customer Constraint}$$

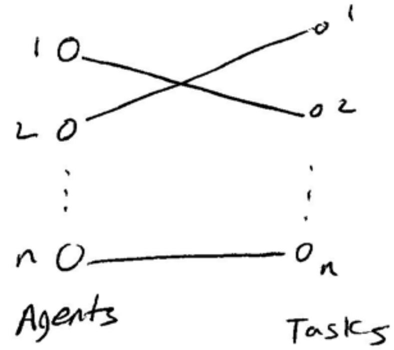
$$(\, x_{11} + x_{12} + x_{13} = 1 \, , \quad x_{11} + x_{21} + x_{31} = 1 \,)$$

objective: minimize $\sum_i \sum_j d_{ij} x_{ij}$

- other Constraints: $x_{ij} \in \{0, 1\}$

- General version: Assign tasks to agents

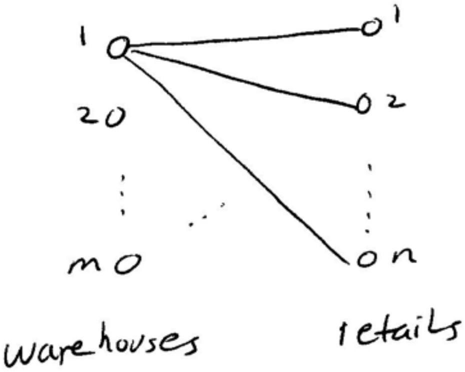
* There are a number of agents and a number of tasks. Any agent can be assigned to perform a task, incurring some cost. Assign exactly one agent to each task.



$$\Rightarrow \min \sum_{i,j} c_{ij} x_{ij}$$

c_{ij} : Cost for agent $\underline{i}$ to perform task $\underline{j}$.

<u>Example:</u> Transportation problem: A company has $\underline{m}$ warehouses and $\underline{n}$ retail outlets. A single product is to be shipped from the warehouses to the outlets. Each warehouse has a given level of supply. Each outlet has a given level of demand. There are transportation costs between every pair of warehouse and outlet. The costs are linear.



a_i : supply at warehouse $\underline{i}$

b_j : demand at outlet $\underline{j}$

c_{ij} : transportation cost

- Decision variables: x_{ij} : size of shipment from warehouse i to outlet j
  $i=1,\dots,m$
  $j=1,\dots,n$

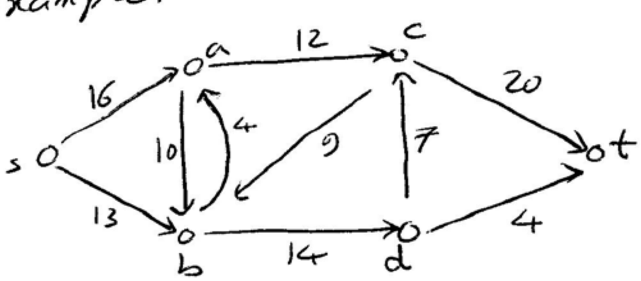
- objective : minimize $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$

- Constraints :
  - $\sum_{j=1}^{n} x_{ij} \le a_i$: constraint for warehouse i
  - $\sum_{i=1}^{n} x_{ij} = b_j$: constraint for outlet j
  - $x_{ij} = 0,1,2,\dots$: meaningful size

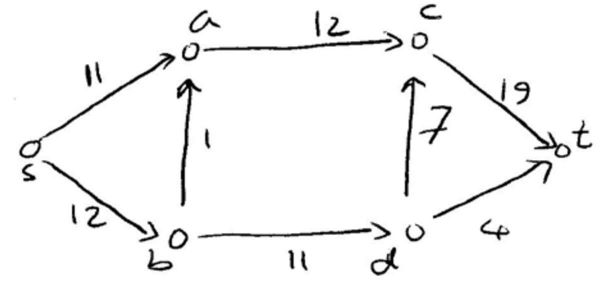
Example: <u>Flow Network</u>: Given a directed graph, where each edge (i,j) is associated with the capacity c_{ij}, there are two specials nodes: soure s and sink t $(s \neq t)$. Maximize the total amount of flow from s to t subject to two constraints:

- Flow on each edge (i,j) doesn't exceed c_{ij}.
- For every node $v \neq s,t$, incoming flow is equal to outgoing flow at node v.

Example:



solution $\Longrightarrow$



- Decisions: x_{ij} : Flow over edge $\overrightarrow{(i,j)}$ $\quad i \longrightarrow j$

- Constraints:
  - $x_{ij} \in [0, c_{ij}]$
  - $\sum\limits_{(i,v)} x_{iv} = \sum\limits_{(v,j)} x_{vj}$ Conservation of flow at node $\underline{\underline{v}}$

- objectives: $\min \sum\limits_{(s,i)} x_{si} - \sum\limits_{(j,s)} x_{js}$

(Flow: people, material, electricity, ...)