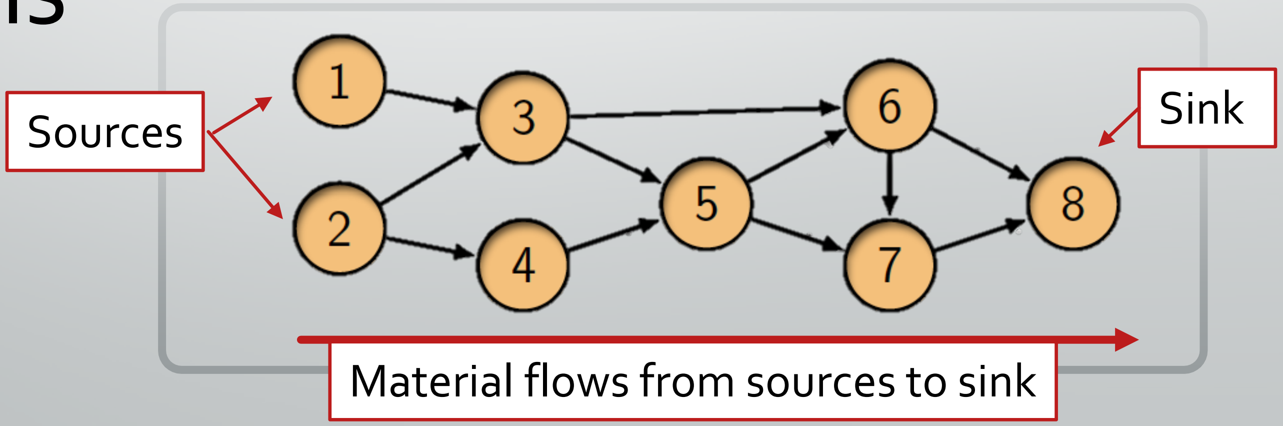
Lecture 5 – Network Flow Problems: Min-Cost Network Flow

Module 5 – Special Cases of Linear Programs

CS/ISyE/ECE 524

Min-Cost Network Flow (MCNF) Problems

- The transportation problem is an example of the **min-cost network flow problem**
  - Typically depicted as a *directed graph* where the arcs represent the **flow** of material from one or more **sources** to one or more **sinks**
- A node can be a source, a sink, or neither
  - A source i *supplies* material and has demand $b_i > 0$
  - A sink j *receives* material and has demand $b_j < 0$
  - Otherwise, the demand at node i is $b_i = 0$
- Each edge in the network has an associated flow cost c_{ij}
- Each edge (typically) has a minimum (ℓ_{ij}) and/or maximum (u_{ij}) capacity



The general MCNF model

Data

- N – the set of nodes
- E – the set of edges
- b_i – the supply/demand of node $i \in N$ (will be zero for most nodes)
- c_{ij} – cost of traveling on edge $(i,j) \in E$
- u_{ij} – upper bound on edge $(i,j) \in E$
- ℓ_{ij} – lower bound on edge $(i,j) \in E$

Objective

- Minimize costs:

$$\min_{x} \sum_{(i,j) \in E} c_{ij} x_{ij}$$

Question: How do we write something like this in matrix notation ($Ax = b$)?

Variables

x_{ij} = total flow on edge $(i,j) \in E$ (nonnegative)

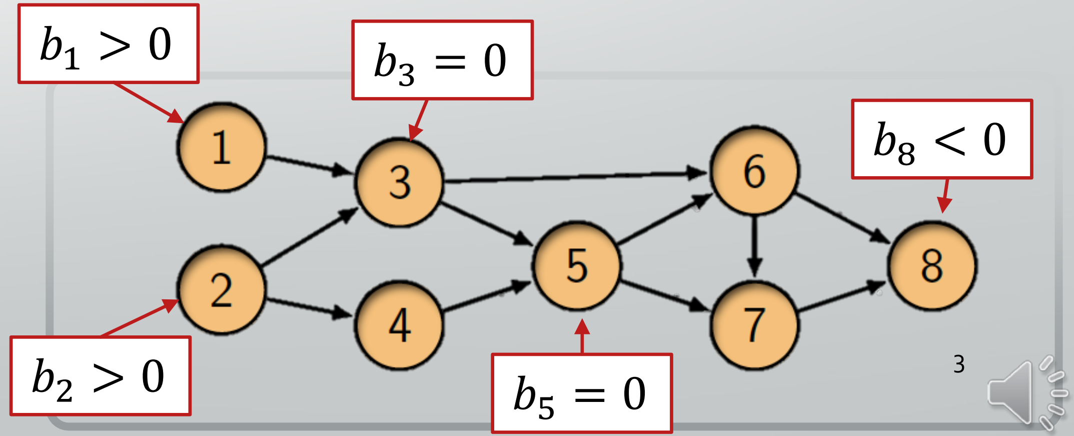
Constraints

- Capacity on each edge:

$$\ell_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in E$$

- Flow balance (what comes out = what goes in)

$$\sum_{j \in N} x_{kj} - \sum_{i \in N} x_{ik} = b_k \quad \forall k \in N$$



Toward a matrix representation of MCNF problems: the **incidence matrix**

- To write a MCNF problem in matrix form ($Ax = b$), we need to understand the structure of a flow balance constraint:
  - For a given node i, we have flow out – flow in = b_i
  - Coefficients on flow variables *entering* i are -1; coefficients on flow variables *leaving* i are $+1$
- Recall the Han Solo transportation problem
  - Flow balance constraint for node 1 (storage site 1) is:

$$x_{1J} + x_{1S} + x_{1P} - 0x_{2J} - 0x_{2S} - \ \ldots - 0x_{3P} = b_1 = 24$$

  - Flow balance constraint for node J (Jabba) is:

$$-x_{1J} + 0x_{1S} + 0x_{1P} - x_{2J} + \cdots - x_{3J} + 0x_{3S} + 0x_{3P} = b_J = -16$$

Arcs

Nodes

$$\underbrace{\begin{array}{c|c} & \begin{array}{ccccccccc} 1J & 1S & 1P & 2J & 2S & 2P & 3J & 3S & 3P \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ J \\ S \\ P \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix} \end{array}}_{\text{Incidence matrix } A}$$

The **incidence matrix** of a directed graph is a matrix indexed by nodes and edges that tells us for each node: 1. which edges exist at that node and 2. whether each edge enters (-1) or leaves (+1) the node.

Balanced problems

Fact: The incidence matrix has the property that the sum over the entries in each column is zero.

- In other words $1^T A = \mathbf{0}$ ($\mathbf{0}$ is a row vector containing n zeroes)
- Since $Ax = b$, this means $1^T Ax = 1^T b = 0$ (0 is a scalar here)
- Therefore, $\sum_{i=1}^{|N|} b_i = 0$ (sum of supply and demand is zero)

A **balanced** problem is one in which the total supply is equal to the total demand.

Note: an unbalanced problem is *always* infeasible! But these situations often arise in practice. We can still solve these problems:

- If supply $>$ demand, turn $Ax = b$ into $Ax \leq b$.
- If supply $<$ demand, turn $Ax = b$ into $Ax \geq b$.

Question: Are balanced MCNF problems always feasible?

Transshipment problems

The **transshipment problem** is a type of MCNF problem that is related to the transportation problem

The structure and goal of the transshipment problem are nearly the same as those of the transportation problem. There can be additional nodes between sources and destinations (think warehouses that can store goods).

- Sinks have demands
- Sources have supply limits
- Edges (might) have capacity limits
- Each edge has an associated cost
- For warehouses, inflow = outflow