Chap. 14: Network flow problems

Mathematical models for flows in networks

Models are linear, based on graphs

A directed graph: An ordered pair (N, A) where

- N is a finite set, called nodes/vertices/points
- A is a finite set of ordered pairs of nodes, called edges/arcs/lines.

Drawing a directed graph: (i,j) denotes the edge from node i to node j

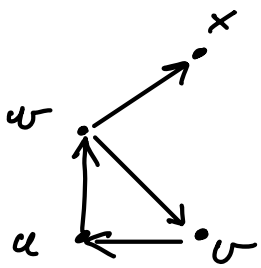


(arrow ending at j).

i initial node, j terminal node

end nodes of (i,j).

Example:



$N = \{u, v, w, x\}$

$A = \{(v,u), (u,w), (w,v), (w,x)\}$

A flow in a directed graph:

- Goods flow along the edges, in the direction the edges show.
- Each node has a certain supply/demand for goods, denoted b_i

$b_i > 0$: i supply node

$b_i < 0$: i demand node

$b_i = 0$: transshipment node.

We want $\sum_{i \in N} b_i = 0$ (everything produced in the graph is also consumed in the graph).

We let x_{ij} denote the flow along (i,j)

(often demanded ≥ 0, an upper capacity a_{ij}, $0 \leq x_{ij} \leq a_{ij}$

Shipping along (i,j) has an associated cost c_{ij} (cost of shipping one unit along (i,j))

Total cost of shipping in the graph: $\sum_{(i,j)\in A} c_{ij} x_{ij}$

Flow balance equations (for node k):

$$\text{outflow} - \text{inflow} = b_k$$

(in the book: written as $\text{inflow} - \text{outflow} = -b_k$)

$$\Updownarrow$$

$$\underbrace{\sum_{i:(i,k)\in A} x_{ik}}_{\text{inflow}} - \underbrace{\sum_{j:(k,j)\in A} x_{kj}}_{\text{outflow}} = -b_k$$

Minimum cost network problem

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i:(i,k)\in A} x_{ik} - \sum_{j:(k,j)\in A} x_{kj} = -b_k \quad \longrightarrow A$$

This is an LP problem, with a special structure. Matrix form:

$$\min \; c^T x$$

$$\text{s.t.} \quad Ax = -b \qquad (*)$$

$$x \geq 0$$

$\leftarrow$ primal problem

A: one row for each node ($|N|$ rows)

one column for each edge ($|A|$ columns)

In column (i,j) of A: entry j is 1, entry i is -1 } for each edge.

In a row: a 1 for each incoming edge

a -1 for each outgoing edge.

A is also called the node-edge incidence matrix

Dual problem

$$\max\ -b^T y \quad \text{s.t.}\ \ A^T y + z = c,\ \ z \geq 0 \qquad (**)$$

equality constraint in (*) ⇓ y_i unconstrained in (**)

y_i: one component for each node,

z: one component for each edge.

Component form of (**):

$$\max\ -\sum b_i y_i$$
$$\text{s.t.}\ \ y_j - y_i + z_{ij} = c_{ij} \quad (i,j) \in A$$
$$z_{ij} \geq 0$$
$$y_i \text{ unconstrained}$$

↓

$$-\min\ \sum b_i y_i$$
$$\text{s.t.}\ \ y_j - y_i \leq c_{ij} \quad (i,j) \in A$$

LP theory to find an optimal solution:

complementary slack: $x_{uv} z_{uv} = 0$ ($y_i w_i = 0$ automatic since $w_i = 0$)

Feasibility:

primal: $x \geq 0$ + flow balance ($Ax = -b$)

dual: $y_j - y_i \leq c_{ij}$ for all (i,j)

In particular, if $y_j < y_i + c_{ij}$ ($\Leftrightarrow z_{ij} > 0$), then x_{ij} must be 0

Terms from graph theory

A path: A sequence $v_1, \ldots, v_t$ of nodes where either (v_i, v_{i+1}) or (v_{i+1}, v_i) is in A (v_1, v_t are called the end nodes of the path)

A connected graph: A graph where any pair of nodes is connected by a path

Cycle: A path where the end nodes coincide.

A **subgraph** of $(\mathcal{N}, \mathcal{A})$: A pair $(\mathcal{N}', \mathcal{A}')$ with $\mathcal{N}' \subseteq \mathcal{N}$, $\mathcal{A}' \subseteq \mathcal{A}$, and where all edges in $\mathcal{A}'$ have end nodes in $\mathcal{N}'$.

A tree: A connected graph without cycles ($\Leftrightarrow$ #edges = #nodes − 1)

A spanning tree: A subgraph of the form $(\mathcal{N}, \mathcal{A}')$ which is a tree

We will explain that:

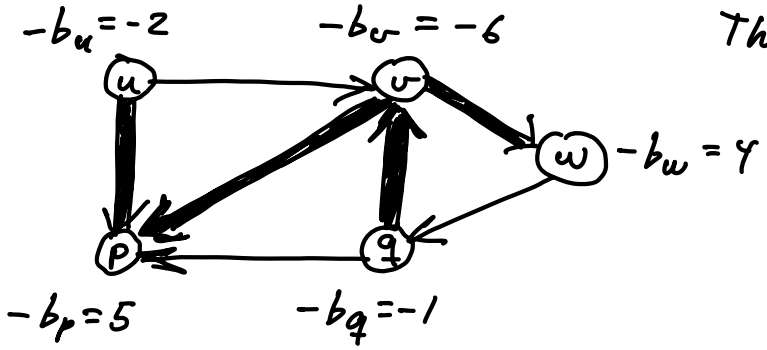
- Spanning trees $\leftrightarrow$ LP bases
- Calculation of LP bases $\leftrightarrow$ Simple operation on spanning trees.
- pivot $\leftrightarrow$ simple transformation from one spanning tree to another.

For a given spanning tree: we compute a **tree solution** as follows:

1/ Set $x_e = 0$ for all edges not in the spanning tree ($e = (i,j)$, nonbasic)

2/ starting at a **leaf node**, use the flow balance equations to compute each x_{ij}, where (i,j) is in the spanning tree (basic variables)

Example:



Thick edges: spanning tree

What is the corresponding tree solution?

1/ nonbasic variable: $x_{uv} = x_{qp} = x_{wq} = 0$

2/ The remaining 4 flows are computed starting at the 3 leaf nodes

a/ u is a leaf node: outflow − inflow $= b_u = 2$ ($= x_{up}$) $= 2$ $\Rightarrow$ $x_{up} = 2$

b) w is a leaf node: outflow − inflow $= b_w = -4$

$-x_{vw} = -4$ } $\Rightarrow x_{vw} = 4$

c) q is a leaf node: outflow − inflow $= b_q = 1$

$x_{qv} = 1$ } $\Rightarrow x_{qv} = 1$

3) Only x_{vp} left:

flow balance at p:

$$\begin{aligned} \text{outflow} - \text{inflow} &= b_p = -5 \\ -x_{up} - x_{vp} &= -5 \\ -2 - x_{vp} &= -5 \\ x_{vp} &= 3 \end{aligned}$$

flow balance at v:

$$\begin{aligned} \text{outflow} - \text{inflow} &= b_v = 6 \\ x_{vw} + x_{vp} - x_{qv} &= 6 \\ 4 + x_{vp} - 1 &= 6 \\ x_{vp} &= 3 \end{aligned}$$

The equations we solved:

$$\begin{aligned} x_{up} &= 2 \\ -x_{vw} &= -4 \\ x_{qv} &= 1 \\ \text{(p)}\quad -x_{up} - x_{vp} &= -5 \\ \text{(v)}\quad x_{vw} - x_{qv} + x_{vp} &= 6 \end{aligned}$$

We see that, in any case, we get a 4×4 invertible submatrix of A, which is lower triangular $\Rightarrow$ invertible.

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -1 \end{pmatrix}$$

rows: flow balance in $|V|-1$ nodes

columns: edges in spanning tree.

Remarks: 1. A connected graph always has a spanning tree.

2. A spanning tree always gives $|N|-1$ linearly independent rows in B, so that $\text{rank}(A) \geq |N|-1$.

Since all rows in A sum together to 0, also $\text{rank}(A) \leq |N|-1$.

It follows that $\text{rank}(A) = |N|-1$.

Conclusion: Given a spanning tree, the corresponding tree solution can be found as the basic solution when any node (called the root node) is taken out of the flow balance equations.

Basic variables: edges in the spanning tree.

Nonbasic variables: edges outside spanning tree.