Network flows

(Ahuja - Magnanti - Orlin: chapters 1, 2, 3 (essentially 3.5))

Review of basic concepts and used notation

Chapter 1 (until page 7, row +3)

- Introduction to network flows and their application in nowadays life
  - how to move some entity (electricity, a consumer product, a person, a vehicle...) from some points to other points in an underlying network in an "efficient" way;
  - typical of areas such as applied mathematics, computer science, engineering, management...
Basic problems:

- **Shortest path problem**: (assumed known!)

- **Maximum flow problem**: given arc capacities, how can we send as much flow (good) as possible from a source to a destination?

- **Minimum cost flow problem**: if we incur a cost per unit flow in a capacitated network, how can we send units of a good from some points to other points at a minimum cost?

*NB*: these are special LPs (and so, polynomially solvable); however, for efficiency reasons they are addressed directly via graph theory, and not via a LP perspective (although many concepts derive from LP theory!)
Minimum cost flow

- distribution of a product from plants to warehouses
- routing of vehicles along a street network

→ See Chapter 2 (from 2.1 to 2.3: To Read) for basic notation and definitions of graph theory

Let $G = (N, A)$ directed network
- $N$ set of $n$ nodes
- $A$ set of $m$ directed arcs
- $c_{ij}$ cost per unit flow on $(i, j)$, $\forall (i, j) \in A$
- $u_{ij}$ capacity of $(i, j)$, $\forall (i, j) \in A$
  ("maximum" amount of flow)
- $b(i) \in \mathbb{Z}$ supply/demand of mode $i$, $\forall i \in N$
Assumption (the opposite w.r.t. RO course):

- $b(i) > 0$ supply mode
- $b(i) < 0$ demand mode (with demand $-b(i)$)
- $b(i) = 0$ transshipment mode

Decision variables (flow variables):

- $x_{ij}$ flow to push along $(i,j)$, $\forall (i,j) \in A$

Mathematical model (LP):

**Min** $\sum c_{ij} x_{ij}$

$\sum x_{ij} - \sum x_{ji} = b(i) \quad \forall i \in N$

Forward Star of $i$

Backward Star of $i$

$0 \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in A$

$e_{ij}$ in A.M.O.
Necessary condition for a feasible solution: \( \sum_{i \in N} b(i) = 0 \)

Compact form:

\[
\begin{align*}
\text{Min} & \quad c \cdot x \\
N \cdot x & = b \\
0 & \leq x \leq u
\end{align*}
\]

\( N \ (n \times m) \) is the node-arc incidence matrix of \( G \): 

\[
N = \begin{bmatrix}
(i, j) & +1 \\
(i, j) & -1 \\
\vdots & 0 \quad \text{elsewhere}
\end{bmatrix}
\]

\( c \in \mathbb{R}^m \) cost vector

\( x \in \mathbb{R}^m \) variable vector

\( b \in \mathbb{R}^n \) balance vector

\( u \in \mathbb{R}^m \) capacity vector
Integrality assumption: $c$, $u$ and $b$ are integer-valued

Special cases

1. Shortest path from $s$ to $t\in G$
   - Send 1 unit of flow from $s$ to $t$ at a minimum cost

Special case of model (MCF) with

\[
\begin{align*}
    b(s) &= 1 \\
    b(t) &= -1 \\
    b(i) &= 0 \quad \forall i \neq s, t
\end{align*}
\]

2. Maximum flow from $s$ to $t\in G$

Complementary: no costs but arc capacities ($u_{ij}$)
Special case of model (MCF) with an extra arc from $\epsilon$ to $s$:

![Diagram showing an arc from $\epsilon$ to $s$]

\[ c_{\epsilon s} = -1 \]

and $b(i) = 0 \quad \forall i \in N$

Then: the minimum cost flow solution maximizes the flow along $(\epsilon, s)$, i.e., the flow amount sent from $s$ to $\epsilon$ along $G$!
Chapter 2

(Section 2.4 (only "Working with Reduced Costs" and "Working with Residual Networks"))

Many network flow algorithms work with "reduced costs" $c_{ij}^\pi$ instead of the actual costs $c_{ij}$:

given a number $\pi(i)$ associated with $i \in N$ (potential of $i$), the reduced cost of $(i, j)$ is

$$c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j)$$

Let $z(0)$ be the objective function w.r.t. $\sum c_{ij}$ and $z(\pi)$ w.r.t. $\sum c_{ij}^\pi$.

Then:

**Property:** $z(\pi) = z(0) - \sum_{i \in N} \pi(i) \cdot b(i) = z(0) - \pi b$.

Therefore, since $\pi b$ is a constant, the
minimum cost flow problems with costs \( \nu^{xy} \) and \( \nu^{xy'} \) have the same optimal solutions. Therefore we can use either \( \nu^{xy} \) or \( \nu^{xy'} \).

In particular, what is the effect of working with reduced costs on cycles and paths (important structures at algorithmic level)?

\[ \sum_{(i,j) \in \mathcal{W}} \nu^{ij} = \sum_{(i,j) \in \mathcal{W}} c_{ij} \]

\[ \sum_{(i,j) \in \mathcal{W}} c_{ij} = c_{12} - \pi(1) + \pi(2) + c_{23} - \pi(2) + \pi(3) + c_{34} - \pi(3) + \pi(4) + c_{41} - \pi(4) + \pi(1) = \]

The cost is the same!
A directed path from $s$ to $t$:

$\pi = s \rightarrow 4 \rightarrow 2 \rightarrow t$

$$\sum_{(i,j) \in P} c_{i,j} \pi_{i,j} = c_{s1} - \pi(s) + \pi(1) + c_{12} - \pi(1) + \pi(2) + c_{2t} - \pi(2) + \pi(t) = \sum_{(i,j) \in P} c_{i,j} - \pi(s) + \pi(t)$$

Difference depending on the potential of terminals $s$ and $t$

Another basic concept in designing network flow algorithms is the residual network (or graph): it is an auxiliary network which measures how we can "move flow" w.r.t. a feasible solution $x^*$.

Given a flow $x^*$ on $G$, replace each arc $(i,j) \in G$ by two arcs:

- $(i,j)$ with cost $c_{i,j}$ and residual capacity $x_{ij}^* = u_{ij} - x_{ij}^*$


\((d, i)\) with cost \(-c_{d,i}\) and residual capacity \(r_{d,i} = x_{d,i}\)\\

**Def**: the residual network w.r.t. \(x^0\), 
\(G(x^0) = (N, A(x^0))\), where \(A(x^0)\) contains the arcs with a positive residual capacity.

**NB**: we assume that \(G\) does not contain both \((i, d)\) and \((d, i)\), \(\forall i, d \in N\).

**NB**: in the maximum flow case, no cost is associated with the arcs \(wi\) in \(G(x^0)\).

**Example**

\[
\begin{align*}
G(x^0) &
\end{align*}
\]

\[
\begin{align*}
G &
\end{align*}
\]
The maximum flow: basic ideas

(Ahuja-Magnanti-Orlin: Chapter 6 (6.1, 6.3, 6.4 (until page 181, row 5), 6.5)

- already covered by the course RO

Let \( G = (N, A) \) be a directed network

- \( u_{ij} \in \mathbb{Z^+} \) capacity of \((i,j)\), \( \forall (i,j) \in A 

- \( s \in N \) source node

- \( t \in N \) destination node

We wish to push the maximum amount of flow from \( s \) to \( t \) by satisfying the arc capacities and the flow conservation constraints.

We can state a direct LP formulation (in place of the one introduced before, as special minimum cost flow):
A basic concept is a partition of \( N \) into \( S \) and \( \overline{S} \) given a cut, denoted by \([S, \overline{S}]\), an \( S \)-\( t \) cut is a cut, s.t. e \( \in S \) and \( e \notin \overline{S} \).

A flow

\[
\begin{align*}
    \max & \text{ value of the flow} \\
    \text{subject to} & \sum_{x \in \delta^+ (s)} x_i = -\sum_{x \in \delta^- (t)} x_i \\
    & \sum_{i=1}^{n} x_i = 1 \\
    & x_i \geq 0, \quad i=1,2,\ldots,n
\end{align*}
\]

\( f(S) = \sum_{e \in \delta (S)} f_e \)
Capacity of $[s, \bar{s}]$:

$$u[s, \bar{s}] = \sum u_{ij}$$

(1a)

forward of the cut

Minimum cut: $s-t$ cut with minimum capacity among all $s-t$ cuts.

Property: for any flow $x$ (of value $v$) and any $s-t$ cut $[s, \bar{s}]$:

$$v \leq u[s, \bar{s}]$$

Indeed, the max-flow min-cut theorem states that, for some flow $x^*$ and some $s-t$ cut $[s^*, \bar{s}^*]$, $v^* = u[s^*, \bar{s}^*]$.

When this happens $\Rightarrow x^*$ is a maximum flow and $[s^*, \bar{s}^*]$ is a minimum cut.