Flows in Networks

Definition: A *network* (G, s, t, c) is a directed graph $G(V, E)$ with two designated vertices s and t, the source and the sink, and a nonnegative capacity function on the edges, $c : E(G) \to R^+$.

Definition: A *flow* in a network (G, s, t, c) is a function $f : E \to R$ satisfying
1) $0 \le f(e) \le c(e)$ for each edge e,
2) for each vertex $v \ne s, t$ $\sum_{u \in V} f(uv) = \sum_{w \in V} f(vw)$ (i.e. the amount into v = amount out of v; conservation law or Kirchhoff's law).

Definition: The *value of the flow* f is $m(f) = \sum_{v \in V} f(vt)(- \sum_{w \in V} f(tw)$, i.e. the (net) amount flowing into t.

Definition: A *s, t-cut* C in a network is the set of edges between X and $V \setminus X$, where X is a subset of the vertices containing s but not containing t.

Definition: The *the capacity* of the s, t-cut C is $c(C) = \sum_{u \in X, v \in V \setminus X} c(uv)$.

Proposition: 1.) $m(f) = \sum_{v \in V} f(sv)(- \sum_{w \in V} f(ws)$ (the (net) amount out of s),
2.) $m(f) = \sum_{u \in X, v \in V \setminus X} f(uv) - \sum_{w \in V \setminus X, z \in X} f(wz)$ for any s, t-cut C (the (net) amount through C),
3.) $m(f) \le c(C)$ for any s, t-cut C.

Definition: An *augmenting path* is a (not necessarily directed) path from s to t, on which the edges in direction are not full (i.e. $f(e) < c(e)$ for these) and the opposite edges are not empty (i.e. $f(e) > 0$ for those).
We can increase the value of the flow on an augmenting path by
$\min\{\min\{c(e) - f(e) : e \text{ is an in direction edge}\},\ \min\{f(e) : e \text{ is an opposite edge}\}\}$.

Theorem: The following are equivalent:
1.) f is a maximum flow.
2.) There is no augmenting path for f.
c.) There is an s, t-cut C such that $m(f) = c(C)$.

Theorem (Ford-Fulkerson or maxflow-mincut thm): $\max\ m(f) = \min\ c(C)$.

An **algorithm** to find the maximum flow in a graph:
1. start form the all 0 flow.
2. use augmenting paths.
3. if there are no more augmenting paths, then the flow is maximum, and we can prove it by finding a minimum cut by taking X to be the vertices on the beginnings of all the increasing paths from s.

Theorem (Edmonds-Karp): if in each step we choose (one of) the shortest increasing path(s), then the algorithm terminates in a polynomial (of the number of vertices) number of steps.

Proposition (Integrality lemma): If in a network the capacity of each edge is an integer then there is a maximum flow whose value on each edge is an integer.

Generalisations of flows

1. More sources $(s_1, \ldots, s_k)$, more sinks $(t_1, \ldots, t_l)$: construct a new network (G', S, T, c'), where $V(G') = V(G) \cup \{S, T\}$, $E(G') = E(G) \cup (\bigcup_{i=1}^{k} (Ss_i)) \cup (\bigcup_{j=1}^{l} (t_jT))$ and $c'(e) = c(e)$ for edges of G, and $c'(Ss_i) \ge \sum_{v \in V(G)} c(s_iv)$, $\forall i = 1, \ldots, k$; $c'(t_jT) \ge \sum_{v \in V(G)} c(vt_j)$, $\forall j = 1, \ldots, l$.

2. Vertex capacities: If $v \in V(G)$ has capacity $c(v)$, then „pull v apart": instead of v add 2 new vertices v' and v'' and the edge $(v'v'')$ of capaciy $c(v)$, and instead of the edges (uv) the edges (uv'), instead of edges (vw) the edges $(v''w)$, with the same capacities.

3. Undirected edges: Instead of the undirected edge $\{u, v\}$ add the two (oppositely) directed edges (u, v) and (v, u) with the same capacities.