1 Network Flows Definitions

A **network** is a tuple $G = (V, E, c, s, t)$, where V is a set of vertices, E is a set of directed edges (parallel edges are allowed), $s \in V$ is the **source**, $t \in V$ is the **sink**, c is a **capacity** function: $c : E \to Z_+$.

For set of vertices $A \subseteq V$, define $\delta^+(A) = \{e \mid \mathit{tail}(e) \in A \wedge \mathit{head}(e) \notin A\}$, and $\delta^-(A) = \{e \mid \mathit{tail}(e) \notin A \wedge \mathit{head}(e) \in A\}$. We write $\delta^-(u)$ and $\delta^+(u)$ instead of $\delta^-(\{u\})$ and $\delta^+(\{u\})$ respectively.

Definition 1 A function $f : E \to R_+$ is called a *flow* if the following three conditions are satisfied:

1. *conservation of flow at interior vertices:* for all vertices u not in $\{s, t\}$,

$$\sum_{e \in \delta^-(u)} f(e) = \sum_{e \in \delta^+(u)} f(e) \ ;$$

2. *capacity constraints:* $f \leq c$ pointwise: i.e. for all $e \in E$,

$$f(e) \leq c(e) \ .$$

Definition 2 The value of a flow f, denoted by $|f|$, is defined to be

$$|f| = \sum_{e \in \delta^+(s)} f(e) - \sum_{e \in \delta^-(s)} f(e).$$

We say that e is **saturated** if $f(e) = c(e)$.

Definition 3 An *s-t cut* (or just *cut*, when s and t are understood) is a pair (A, B) of disjoint subsets of V whose union is V such that $s \in A$ and $t \in B$. The *capacity* of the cut (A, B), denoted by $c(A, B)$, is

$$c(A, B) = \sum_{e \in \delta^+(A)} c(e) \ .$$

If f is a flow, we define the *flow across the cut (A,B)* to be

$$f(A, B) = \sum_{e \in \delta^+(A)} f(e) - \sum_{e \in \delta^-(A)} f(e) \ .$$

Lemma 1.1 *For any $s - t$ cut (A, B), $f(A, B) = |f|$.*

Definition 4 Given a flow f on a network G define the *residual network* G_f as follows: For every $e \in E$ with $f(e) < c(e)$, add an edge e' in G_f with $c_f(e') = c(e) - f(e)$; e' is the *forward* edge obtained from e. For every $e \in E$ with $f(e) > 0$, add an edge $\overline{e}$ in G_f with $c_f(\overline{e}) = f(e)$; $\overline{e}$ is the *back* edge obtained from e. G_f has the same s and t.

Definition 5 Given a network G and flow f on G, an *augmenting path* is a directed path from s to t in the residual network G_f.

Lemma 1.2 *Given f' a flow in G_f, consider the function $\hat{f} : E \to R_+$ defined by $\hat{f}(e) = f(e) + f'(e') - f'(\overline{e})$, where e' and $\overline{e}$ are the forward and back edge obtained from e. Then $\hat{f}$ is a flow in G with $|\hat{f}| = |f| + |f'|$.*

Given $\hat{f}$ is a flow in G, consider the function $f' : E_f \to R_+$ defined as follows: if for edge $e \in E$ we have $f(e) < \hat{f}(e)$, then for the forward edge obtained from e we have $f'(e') = \hat{f}(e) - f(e)$, and if for edge $e \in E$ we have $f(e) > \hat{f}(e)$, then for the back edge obtained from e we have $f'(\overline{e}) = f(e) - \hat{f}(e)$, with f' being zero on the other edges of G_f. Then f' is a flow in G_f with $|f'| = |\hat{f}| - |f|$.

The main theorem in Network Flows is the following MaxFlow-MinCut Theorem:

Theorem 1.3 *Let $G = (V, E, c, s, t)$ be a network and f be a flow in G. The following three conditions are equivalent:*

1. *f is a maximum flow in G.*
2. *The residual network G_f contains no augmenting paths.*
3. *$|f| = c(A, B)$ for some s-t cut (A, B).*

The "flow-decomposition theorem" is:

Theorem 1.4 *Let f be a flow in G. Then there exists paths $s - t$ paths $P_1, \ldots, P_k$ with positive integers $\alpha_1, \ldots, \alpha_k$ and circuits $C_1, \ldots, C_q$ with positive integers $\beta_1, \ldots, \beta_q$, with $0 \le k + q \le |E|$, such that for all $e \in E$,*

$$f(e) = \sum_{i \in \{1,\ldots,k\} \wedge e \in P_i} \alpha_i + \sum_{j \in \{1,\ldots,q\} \wedge e \in P_j} \beta_j$$

and $|f| = \sum_{i=1}^{k} \alpha_i$.

2 The Ford-Fulkerson Algorithm

BELLMAN-FORD-FULKERSON(G,s,t)

for each edge $e \in E(G)$
  do $f(e) \longleftarrow 0$
Construct G_f
while there exists a path P from s to t in the residual net work G_f
  do $c_f(P) \longleftarrow \min_{e \in P} c_f(e)$
    for each edge a in P
      do if a is a forward edge: $a = e'$ for some $e \in E$
        $f(e) \longleftarrow f(e) + c_f(P)$
      else ($a = \overline{e}$ for some $e \in E$)
        $f(e) \longleftarrow f(e) - c_f(P)$

3 Matching Definitions

Given an undirected graph $G = (V, E)$, a **matching** is a subset $M \subseteq E$ such that no two edges in M share a vertex.

Definition 6 Given a matching M in $G = (V, E)$, an edge $e \in E$ is *matched* if $e \in M$, and *free* if $e \in E \setminus M$. A vertex v is *matched* if v has an incident matched edge, and *free* otherwise.

Definition 7 A *perfect matching* is a matching in which every vertex is matched.

Definition 8 Given a matching M in $G = (V, E)$, a path (cycle) in G is an *alternating path (cycle)* with respect to the matching M if it is simple (has no repeated vertices) and consists of alternating matched and free edges. An alternating path is an *augmenting path* (with respect to M) if its endpoints are free.

Definition 9 A graph $G = (V, E)$ is bipartite iff V can be partitioned in A and B such that every edge of E has one endpoint in A and one endpoint in B.

Fact 3.1 *A graph is bipartite iff it does not have any odd cycle.*

Definition 10 If $G = (V, E)$ is an undirected graph, a **vertex cover** of G is a subset of V where every edge of G is adjacent to one node in this subset. The minimum vertex cover problem asks for the size of the smallest vertex cover. An **edge cover** of G is a subset subset of E where every vertex of G is adjacent to one edge in this subset. The minimum edge cover problem asks for the size of the smallest edge cover.

Fact 3.2 *For any graph $G = (V, E)$, any $M \subseteq E$ matching in G and Q vertex cover in G, $|M| \leq |Q|$.*

Theorem 3.3 *In a bipartite graph, the size of a maximum matching equals the size of the minimum vertex cover.*