Section 4.2

The Ford and Fulkerson Approach

AUGMENTING PATH

One approach to finding the maximum flow in a network, is to start with some "path" between the source and sink and improve the flow along the path. Once this is done, we repeat the process on $N$ with its modified flow. We continue until we cannot find a path whose flow can be approved. This is known as the augmenting path technique.

CUTS AND CAPACITY

- Let $S$ be a subset set of $V$ such that $s \in S$ and $t \in \overline{S} = V - S$.
- $out(S) = \{e = u \rightarrow v \in E: u \in S \text{ and } v \in \overline{S}\};$ that is, the set of all arcs from $S$ to $\overline{S}$.
- $in(S) = \{e = v \rightarrow u \in E: u \in S \text{ and } v \in \overline{S}\};$ that is, the set of all arcs from $\overline{S}$ to $S$.
- $out(S) \cup in(S)$ is called the cut determined by $S$.
- For a set $S$ of vertices, we call $c(S) = \sum_{e \in out(S)} c(e)$ the capacity of the cut determined by $S$. 
**SOME PREPARATORY RESULTS**

**Lemma 4.2.1:** Given a network $N = (V, E, s, t, c)$ with flow $f$, then for every $S \subseteq V$ such that $s \in S$ and $t \in \overline{S}$,

$$F = \sum_{e \in \text{out}(S)} f(e) - \sum_{e \in \text{in}(S)} f(e).$$

**Proposition 4.2.1:** Given a network $N$, for every flow $f$ with total flow $F$ and for every $S \subseteq V$,

$$F \leq c(S).$$

**Corollary 4.2.1:** Given a network $N$ with flow $f$ and $S \subseteq V$ such that $S$ contains $s$ but not $t$, if $F = c(S)$, then $F$ is a maximum and the cut determined by $S$ is of minimum capacity.

---

**SLACK AND SATURATION**

- If the arc $e = x \rightarrow y$ is on an $s \rightarrow t$ path and we wish use $e$ to push more flow to $t$, then $e$ presently must not be up to capacity; that is $f(e) < c(e)$. The amount of improvement is limited to $\Delta(e) = c(e) - f(e)$ called the **slack** of $e$.
- If $f(e) = c(e)$, we say the arc $e$ is **saturated**.
- If the arc $e = y \rightarrow x$, then in order to increase the flow from $s$ to $t$, we must cancel some of the flow into $x$ on this arc (since the flow is away from $t$). Thus, there must already be some flow ($f(e) > 0$) on $e$ if we are to increase the total flow.

---

**AUGMENTING PATH**

An **augmenting path** is a (not necessarily directed) path from $s$ to $t$ that can be used to increase the flow from $s$ to $t$. 
FORWARD AND BACKWARD LABELING

- In the augmenting path technique, we label vertices.
- A forward labeling of vertex \( v \) using arc \( e = u \rightarrow v \) is done when \( u \) is labeled and \( v \) is not labeled and \( c(e) > f(e) \). The label \( e^+ \) is assigned to \( v \). Here \( \Delta(e) = c(e) - f(e) \).
- A backward labeling of vertex \( v \) using arc \( e = v \rightarrow u \) is done when \( u \) is labeled and \( v \) is not labeled and \( f(e) > 0 \). The label \( e^- \) is assigned to \( v \). Here \( \Delta(e) = f(e) \).

AUGMENTING PATHS AND MAXIMUM FLOW

Theorem 4.2.1: In a network \( N \) with flow \( f \), the total flow \( F \) is maximum if and only if no augmenting path from \( s \) to \( t \) exists.

THE FORD AND FULKERSON ALGORITHM

Algorithm 4.2.1 The Ford and Fulkerson Algorithm.

Input: A network \( N = (V, E, s, t, c) \) and a flow \( f \).
(Initially, we usually choose \( f(e) = 0 \) for every arc \( e \).)

Output: A modified flow \( f^* \) or the answer that the present flow is maximum.

Method: Augmenting paths.
FORD AND FULKERSON ALG.  
(CONCLUDED)
1. Label $s$ with $*$ and leave all other vertices unlabeled.
2. Find an augmenting path $P$ from $s$ to $t$.
3. If none exists,
   then halt, noting that the present flow is maximum;
   else compute and record the slack of each arc of $P$
   and compute the minimum slack $\lambda$. Now, redefine
   the flow $f$ by adding $\lambda$ to $f$ for all forward arcs of $P$
   and subtracting $\lambda$ from $f$ for all backward arcs of $P$.
4. Set $f^* = f$ and repeat this process for $N$ and the new
   flow $f^*$.

EDMONDS AND KARP TECHNIQUE

Edmonds and Karp were able to show that if a
breadth-first search is used in the labeling
algorithm and the shortest augmenting path
is always selected, then the algorithm will
terminate in at most $O(|V|^2 |E|)$ steps even if
irrational capacities are allowed.

SCAN

We will use the term scan to imply that a
breadth-first search is being done.
FINDING AN AUGMENTING PATH ALGORITHM

Algorithm 4.2.2 Finding an Augmenting Path.
Input: A network $N$ and a flow $f$.
Output: An augmenting path $P$ or a message the none exists.
Method: A modified breadth-first search.

FINDING AN AUGMENTING PATH (CONCLUDED)

1. Label $s$ with $\ast$.
2. If all labeled vertices have been scanned, then halt, noting that no augmenting path exists; hence the present flow is maximum, else find a labeled but unscanned vertex $v$ and scan as follows:
   For each $e = vu \in \text{out}(b)$, if $c(e) - f(e) > 0$ and $u$ is unlabeled, label $u$ with $e^\ast$. For each $e = uv \in \text{in}(v)$, if $f(e) > 0$ and $u$ is unlabeled, then label $u$ with $e^\ast$.
3. If $t$ has been labeled, then starting at $t$, use the labels to backtrack to $s$ along an augmenting path. The label at each vertex indicates its predecessor in the path. When you reach $s$ output the path and halt; else repeat step 2.

THE MAX FLOW-MIN CUT THEOREM

Theorem 4.2.2 (The Max Flow-Min Cut Theorem): In a network $N$, the maximum value of a flow equals the minimum capacity of a cut.