Ch. 4 - Networks & Flows

§1

Legal flows and capacity of cuts.

Def. A network is a 4-tuple $N = (G, s, t, c)$ where $G = (V, E)$ is a digraph with two distinguished vertices $s$ & $t$ (called the source & sink respectively) and $c : E \rightarrow \mathbb{R}^*$ is a function called the capacity function. ($\mathbb{R}^*$ = set of non-negative reals)

Ex. 1

Here an arrow "→" to indicate the source $s$ and a double circle to indicate the sink.

Def. A legal flow in a network $N$ is a function $f : E \rightarrow \mathbb{R}^*$ such that

(a) $f(e) \leq c(e)$ for each $e \in E$ (capacity constraint) &
(b) $\sum_{e \in \text{In}(v)} f(e) = \sum_{e \in \text{Out}(v)} f(e)$ for each $v \in V - \{s, t\}$, where $\text{In}(v)$ = set of all edges in $G$ coming into the vertex $v$ & $\text{Out}(v)$ = set of all edges in $G$ going out of the vertex $v$.

Def. The value of a legal flow, $f$, in a network $N$ is defined by

$\text{Val}(f) = \sum_{e \in \text{In}(t)} f(e) - \sum_{e \in \text{Out}(t)} f(e)$

So $\text{Val}(f)$ = net flow into $t$. 
Notice that for each of the vertices a, b, c, d the flow coming in is equal to the flow coming out. For example: \[
\sum_{e \in \text{In}(c)} f(e) = \sum_{e \in \text{Out}(c)} f(e) = f(sc) + f(tc) - f(c) = 9 + 4 - 13 = 0.
\]

Observe also that
\[
\text{Val}(f) = \sum_{e \in \text{In}(a)} f(e) - \sum_{e \in \text{Out}(a)} f(e) = f(bt) + f(dt) - f(t) = 10 + 10 - 4 = 16.
\]

Finally note that
\[
\sum_{e \in \text{Out}(s)} f(e) - \sum_{e \in \text{In}(s)} f(e) = f(sa) + f(sc) - 0 = 7 + 9 = 16 = \text{Val}(f)
\]

**Q:** Given a network N, how can we find a legal flow \( f_0 \) in N such that \( \text{Val}(f_0) \geq \text{Val}(f) \) for all other legal flows \( f \) in N?

**Def.** A source-separating set of vertices in a network N is any set of vertices \( U \subseteq V(G) \) such that \( s \notin U \) and \( t \notin U \). We define the complement \( \overline{U} \) of U by \( \overline{U} = V(G) - U \).

**Def.** Let \( U \) be a source-separating set of vertices in \( N \). The cut determined by \( U \) is defined by \( \text{cut}(U) = \text{In}(U) \cup \text{Out}(U) \) where \( \text{In}(U) = \text{set of all edges in } G \text{ from } U \text{ to } \overline{U} \), and \( \text{Out}(U) = \text{set of all edges in } G \text{ from } U \text{ to } \overline{U} \).
Let $N$ be a network and $U$ be a source-separating set of vertices. We define the capacity of the cut determined by $U$ by
\[ c(\text{Cut}(U)) = \sum_{e \in \text{Out}(U)} c(e) \] (i.e., sum of outward capacities).

We also define $\text{MinCut}(N)$ by
\[ \text{MinCut}(N) = \min \{ c(\text{Cut}(U)) : U \text{ is a source-separating set of vertices in } N \} \]

**Example 3**

Let $N$ be the network below.

(a) Then $U = \{ s, d \}$ is a source-separating set of vertices.

(b) $\text{Cut}(U) = \text{In}(U) \cup \text{Out}(U)$
\[ = \{ d \} \cup \{ s, a, b, t \} = \{ d, s, a, t \} \]

(c) $c(\text{Cut}(U)) = \sum_{e \in \text{Out}(U)} c(e) = c(sa) + c(dt) = 6 + 7 = 15$.

(d) $c(\text{Cut}\{s, a\}) = 23$, $c(\text{Cut}\{s, a, b\}) = 24$, $c(\text{Cut}\{s, b, d\}) = 25$
\[ c(\text{Cut}\{s, a, d\}) = 22, \quad c(\text{Cut}\{s, b, d\}) = 27, \quad c(\text{Cut}\{s\}) = 16 \]
\[ c(\text{Cut}\{s, d\}) = 15, \quad c(\text{Cut}\{s, a, b, d\}) = 21. \]

Since there are only 8 possible source-separating sets of vertices (all the subsets of $\{a, b, d\}$ plus $\{s\}$), it follows that
\[ \text{MinCut}(N) = 15. \]

**Remark:** If $G$ has $n+2$ vertices, then the network $N$ will have $2^n$ different source-separating sets of vertices. So it will be no easy task to find $\text{MinCut}(N)$ directly from the definition. Hence we need a fast algorithm for it.
Prop. 1 Let \( U \) be any source-separating set of vertices in a network \( N \) and \( f \) be a legal flow in \( N \). Then \[
\text{Val}(f) = \sum_{e \in \text{Out}(U)} f(e) - \sum_{e \in \text{In}(U)} f(e).
\]

Proof: Let \( U = \{s = x_1, x_2, \ldots, x_k\} \) & \( \overline{U} = \{t = y_1, y_2, \ldots, y_n\} \)

Then from the definition of \( \text{Val}(f) \) we have \[
\text{Val}(f) = \sum_{i=1}^{k} \left( \sum_{e \in \text{In}(y_i)} f(e) - \sum_{e \in \text{Out}(y_i)} f(e) \right) \quad \cdots (1)
\]

Also by the conservation of flow for \( y_1, \ldots, y_n \) we have \[
0 = \sum_{e \in \text{In}(y_i)} f(e) - \sum_{e \in \text{Out}(y_i)} f(e) \quad \cdots (2)
\]

\[
0 = \sum_{i=1}^{n} \left( \sum_{e \in \text{In}(y_i)} f(e) - \sum_{e \in \text{Out}(y_i)} f(e) \right) \quad \cdots (n)
\]

Adding equations (1), (2), \( \cdots, (n) \) we get \[
\text{Val}(f) = \sum_{i=1}^{n} \left( \sum_{e \in \text{In}(y_i)} f(e) - \sum_{e \in \text{Out}(y_i)} f(e) \right)
\]

Let \( A(U, f) = \sum_{e \in \text{out}(U)} f(e) - \sum_{e \in \text{In}(U)} f(e) \)

and \( B(U, f) = \sum_{i=1}^{n} \left( \sum_{e \in \text{In}(y_i)} f(e) - \sum_{e \in \text{Out}(y_i)} f(e) \right) \).

We want to show \( \text{Val}(f) = A(U, f) \). We will show that \( A(U, f) = B(U, f) \). Since \( \text{Val}(f) = B(U, f) \), it will follow that \( \text{Val}(f) = A(U, f) \). To show that \( A(U, f) = B(U, f) \), we will show that \( A(U, f) \) & \( B(U, f) \) agree about the net contribution of \( f(e) \) & \( -f(e) \) for each edge \( e \in E(G) \).
Let $e = \overline{uv}$ be any edge in $E(G)$ from $u$ to $v$. Then there are four cases.

**Case(i)** $u \in U$ & $v \in U$: In this case neither $f(e)$ nor $-f(e)$ appear in either of the expressions $A(U,f)$ or $B(U,f)$. So $A(U,f)$ & $B(U,f)$ agree about the net contribution of $f(e)$ & $-f(e)$.

**Case(ii)** $u \in U$ & $v \in \overline{U}$: In this case only $f(e)$ occurs in $A(U,f)$ and only $f(e)$ occurs in $B(U,f)$ also. So $A(U,f)$ & $B(U,f)$ agree about the net contribution of $f(e)$ & $-f(e)$ again.

**Case(iii)** $u \in \overline{U}$ & $v \in U$: In this case only $-f(e)$ occurs in $A(U,f)$ and only $-f(e)$ occurs in $B(U,f)$ also. So $A(U,f)$ & $B(U,f)$ agree about the net contribution of $f(e)$ & $-f(e)$ once more.

**Case(iv)** $u \in \overline{U}$ & $v \in \overline{U}$: In this case neither $f(e)$ nor $-f(e)$ occurs in $A(U,f)$. Also both $f(e)$ & $-f(e)$ occurs in $B(U,f)$. So $A(U,f)$ & $B(U,f)$ agree about the net contribution of $f(e)$ & $-f(e)$ once again.

Thus $A(U,f) = B(U,f)$ and so we get $Val(f) = A(U,f)$.

**Prop. 2** Let $U$ be a source-separating set of vertices in a network $N$, and $f$ be any legal flow in $N$. Then $Val(f) \leq c[Cut(U)]$.

**Proof:** From Proposition 1, we have

$$Val(f) = \sum_{e \in \text{Out}(U)} f(e) - \sum_{e \in \text{In}} f(e) \leq \sum_{e \in \text{Out}(U)} f(e) \quad \text{because} \quad f(e) \geq 0 \leq \sum_{e \in \text{Out}(U)} c(e) = c[Cut(U)] \quad \text{because} \quad f(u) \leq c(e).$$
§2. The Ford-Fulkerson Algorithm & MaxFlow-MinCut Theorem

Def. Let $N$ be a network. We define

$$\text{MaxFlow}(N) = \max \{ \text{Val}(f) : f \text{ is a legal flow in } N \}.$$ 

Recall also that

$$\text{MinCut}(N) = \min \{ c[\text{Cut}(U)] : U \text{ is a source-separating set of vertices in } N \}.$$ 

Observe that since $\text{Val}(f) \leq c[\text{Cut}(U)]$ for any legal flow $f$ & source-separating set of vertices in $N$, we immediately get $\text{MaxFlow}(N) \leq \text{MinCut}(U)$.

We will find a flow $f^*$ in $N$ along with a source-separating set of vertices $U^*$ such that $\text{Val}(f^*) = c[\text{Cut}(U^*)]$. From this it will follow that $\text{MaxFlow}(N) = \text{MinCut}(N)$.

Def. Let $f$ be a legal flow in a network $N$. The slack w.r.t. $f$ of an edge $e$ in a semi-path $P$ from $s$ to $t$ is defined by $s_l(e) = \text{maximum flow you can add to } e \text{ in the direction from } s \text{ to } t$.

An augmenting semi-path $P$ is any semi-path from $s$ to $t$ with $s_l(e_i) > 0$ for each $e_i$ in $P$.

Ex. 1

Below are two augmenting semi-paths with the slack of each edge.

1. $s \rightarrow d \rightarrow t$
   $$s_l(\overline{sd}) = 12, \ s_l(d\overline{t}) = 0$$

2. $s \rightarrow d \rightarrow b \rightarrow t$
   $$s_l(\overline{sd}) = 12, \ s_l(b\overline{t}) = 3, \ s_l(\overline{bd}) = 5$$
Ex. 1. If we use the first augmenting semi-path, we can modify the flow and increase its value by sending 6 units along the semi-path $s \rightarrow d \rightarrow t$.

We could have used the second augmenting semi-path to modify the flow and increase its value by sending 3 units along the semi-path $s \rightarrow d \rightarrow b \rightarrow t$ from $s$ to $t$.

Algorithm 1 (Ford-Fulkerson Algorithm)

**INPUT:** A network $N = (G, s, t, c)$

**OUTPUT:** A maximal flow $f^*: E(G) \rightarrow R^*$

1. For each edge $e \in E(G)$, let $f^*(e) \leftarrow 0$ & $i \leftarrow 1$
2. If there is no augmenting semi-path from $s$ to $t$, STOP; else, find an augmenting semi-path $P$ from $s$ to $t$.
3. Compute the slack (with respect to $f^*$) of each edge $e$ in the semi-path $P$ and let $\mu_i = \min \{ \text{sl}(e) : e \in P \}$
4. Let $f^*(e) \leftarrow f^*(e) + \mu_i$ for each forward edge $e$ of $P$, & $f^*(e) \leftarrow f^*(e) - \mu_i$ for each backward edge $e$ of $P$; $i \leftarrow i + 1$ & then go to step 2. (A forward edge $e$ in a semi-path $P$ from $s$ to $t$ is an edge which goes in the direction from $s$ to $t$. A backward edge $e$ in $P$ is an edge which goes in the direction from $t$ to $s$.)
Ex. 2 Let \( N \) be the network

Find a maximal flow \( f^* \) in \( N \) and a source-separating set of vertices \( U^* \) such that \( \text{Val}(f^*) = c[\text{Cut}(U^*)] \).

1st augmenting semi-path, \( P_1 \):
\[
\begin{align*}
S & \overset{(0,6)}{\rightarrow} a \overset{(0,5)}{\rightarrow} b \overset{(0,9)}{\rightarrow} d \overset{(0,5)}{\rightarrow} t
\end{align*}
\]
Slacks: \( 6 \quad 5 \quad 9 \quad 5 \quad \therefore M_1 = 5 \)

2nd augmenting semi-path, \( P_2 \):
\[
\begin{align*}
S & \overset{(0,10)}{\rightarrow} d \overset{(5,9)}{\rightarrow} b \overset{(0,7)}{\rightarrow} t
\end{align*}
\]
Slacks: \( 10 \quad 5 \quad 7 \quad \therefore M_2 = 5 \)

There are no more augmenting paths. Also
\[
\text{Val}(f^*) = \sum_{e \in \text{In}(t)} f^*(e) - \sum_{e \in \text{Out}(t)} f^*(e) = (5+5) - (0) = 10.
\]

Let \( U^* = \{ v \in V(G) : \text{there is an aug. semi-path from } s \text{ to } v \} \).
Then \( s \in U^* \) & \( t \notin U^* \) (because there is no more augmenting semi-paths from \( s \) to \( u \)). So \( U^* \) is a source-separating set of vertices in \( N \). Also \( U^* = \{ S, a, d \} \) and \( c[\text{Cut}(U^*)] = \sum_{e \in \text{Out}(u)} c(e) = c(ab) + c(dt) = 5 + 5 = 10 \).
Thus
\[
\text{Val}(f^*) = c[\text{Cut}(U^*)].
\]
Theorem 3: (MaxFlow - MinCut Theorem)
In any network, \( \text{MaxFlow}(N) = \text{MinCut}(N) \).

Proof: Let \( f^* \) be the flow obtained by the Ford-Fulkerson Algorithm. Then there is no augmenting semi-path from \( s \) to \( t \). So if we put \( U^* = \{ v \in V(G) : \text{there is an any semi-path from } s \text{ to } v \} \), then \( s \in U^* \) & \( t \notin U^* \). So \( U^* \) is a source-separating set of vertices in \( N \).

First, we know that
\[
\text{Val}(f^*) = \sum_{e \in \text{Out}(U^*)} f^*(e) - \sum_{e \in \text{In}(U^*)} f^*(e),
\]

Now consider an edge \( e = vw \) from \( \text{Out}(U^*) \). If \( f^*(e) \) was less than \( c(e) \), then we would be able to send some more flow from \( s \) to \( w \). But this is impossible, because \( v \in U^* \) & \( w \notin U^* \). So we must have \( f^*(e) = c(e) \) for each edge \( e \in \text{Out}(U^*) \).

Also consider an edge \( e = vw \) from \( \text{In}(U^*) \). If \( f^*(e) \) was non-zero, then we would be able to send some more flow from \( s \) to \( w \) by push back some along the backward edge \( vw \). But this is impossible, because \( w \notin U^* \). Hence \( f^*(e) = 0 \) for each edge \( e \in \text{In}(U^*) \). Thus
\[
\text{Val}(f^*) = \sum_{e \in \text{Out}(U^*)} f^*(e) - \sum_{e \in \text{In}(U^*)} f^*(e) = \sum_{e \in \text{Out}(U^*)} c(e) - \sum_{e \in \text{In}(U^*)} 0 = c[\text{Cut}(U^*)] - c[\text{Cut}(U^*)]
\]
Since \( \text{MaxFlow}(N) \leq \text{MinCut}(N) \) & we have \( \text{Val}(f^*) = c[\text{Cut}(U^*)] \), it follows that \( \text{MaxFlow}(N) = \text{MinCut}(N) \).
Ex. 3 Let $N$ be the network

Find a maximal flow $f^*$ in $N$, the associated source-separating set of vertices $U^*$ & check that $\text{Val}(f^*) = c[\text{Cut}(U^*)]$

1st augmenting semi-path, $P_1$:
\[
\begin{align*}
S &\rightarrow (0,10) \rightarrow a \rightarrow (0,6) \rightarrow b \rightarrow (0,8) \rightarrow t \\
\text{Slacks:} &\quad 10 \quad 6 \quad 8 \quad i: M_1 = 6
\end{align*}
\]

2nd augmenting semi-path, $P_2$:
\[
\begin{align*}
S &\rightarrow (0,10) \rightarrow a \rightarrow (0,4) \rightarrow t \\
\text{Slacks:} &\quad 4 \quad 4 \quad i: M_2 = 4
\end{align*}
\]

3rd augmenting semi-path $P_3$:
\[
\begin{align*}
S &\rightarrow (0,9) \rightarrow c \rightarrow (0,3) \rightarrow d \rightarrow (0,5) \rightarrow t \\
\text{Slacks:} &\quad 9 \quad 7 \quad 5 \quad M_3 = 5
\end{align*}
\]

There are no more augmenting semi-paths now. So $U^* = \{v \in V(G): \text{we can send some more flow from } S \text{ to } v\}$ = \{S, c, d, a\}

Thus $\text{Val}(f^*) = \sum_{e \in \text{In}(t)} f^*(e) - \sum_{e \in \text{Out}(t)} f^*(e) = f^*(at) + f^*(at) + f^*(dt) - 0$

\[
\begin{align*}
\text{Val}(f^*) &= 6 + 4 + 5 = 15 \\
c[\text{Cut}(U^*)] &= \sum_{e \in \text{Out}(U^*)} c(e) = c(ab) + c(dt) + c(dt) = 6+4+5 = 15
\end{align*}
\]