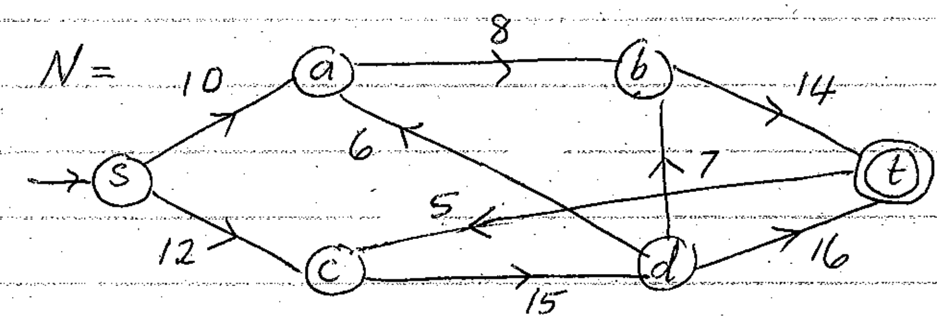
Ch. 4 - Networks & Flows

§1. Legal flows and capacity of cuts.

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

Ex. 1 $N =$



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 $f: E \to \mathbb{R}^{\#}$ such that

(a) $f(e) \le c(e)$ for each $e \in E$ (*capacity constraint*) &

(b) $\sum_{e\in In(v)} f(e) = \sum_{e \in Out(v)} f(e)$ for each $v \in V - \{s,t\}$, where (conservation of flow)

$In(v)$ = set of all edges in G coming into the vetex v

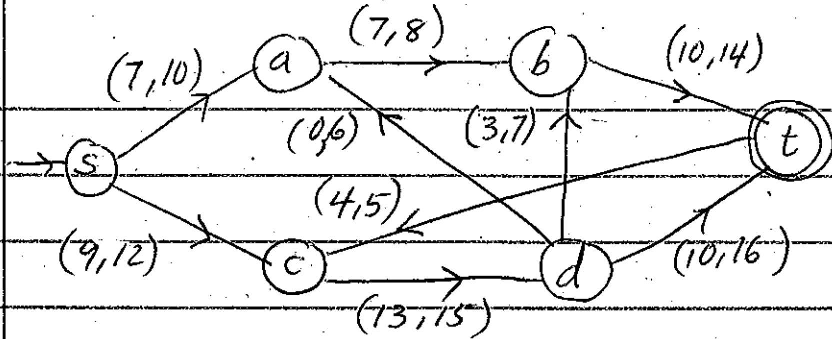
& $Out(v)$ = set of all edges in G going out of the vetex v.

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

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

So $Val(f)$ = net flow into t.

Ex. 2



Notice that for each of the vertices a, b, c, d the flow coming in is equal to the flow coming out. For example $\sum_{e \in In(c)} f(e) - \sum_{e \in Out(c)} f(e) = f(\vec{sc}) + f(\vec{tc}) - f(\vec{cd})$
$= 9 + 4 - 13 = 0.$

Observe also that

$$Val(f) = \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e) = f(\vec{bt}) + f(\vec{dt}) - f(\vec{tc})$$
$$= 10 + 10 - 4 = 16.$$

Finally note that

$$\sum_{e \in Out(s)} f(e) - \sum_{e \in In(s)} f(e) = f(\vec{sa}) + f(\vec{sc}) - 0 = 7 + 9 = \underbrace{16}_{= Val(f)}.$$

Qu: Given a network N, how can we find a legal flow f_0 in N such that $Val(f_0) \geq Val(f)$ for all other legal flows f in N?

Def. A <u>source-separating</u> set of vertices in a network N is any set of vertices $U \subseteq V(G)$ such that $s \in U$ and $t \notin U$. We define the <u>compliment</u> $\bar{U}$ of U by $\bar{U} = V(G) - U$.

Def. Let U be a source-separating set of vertices in N. The <u>cut</u> determined by U is defined by $cut(U) = In(U) \cup Out(U)$ where $In(U)$ = set of all edges in G from $\bar{U}$ to U, and
$Out(U)$ = set of all edges in G from U to $\bar{U}$.

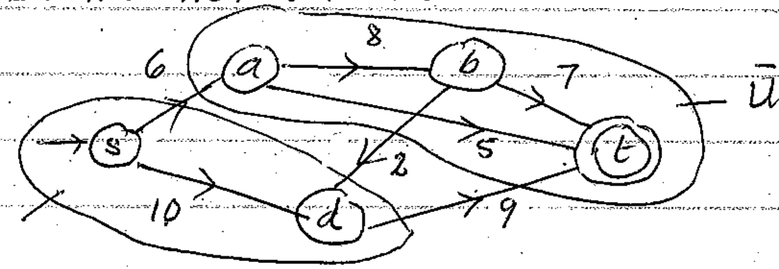
Def. Let N be a network and $\mathcal{U}$ be a source-separating set of vertices. We define the <u>capacity of the cut determined by $\mathcal{U}$</u> by

$$c\,[Cut(\mathcal{U})] = \sum_{e \in Out(\mathcal{U})} c(e) \quad \text{(i.e., sum of outward capacitie}$$

We also define <u>$MinCut(N)$</u> by

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

Ex. 3 Let N be the network below.



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

(b) $Cut(\mathcal{U}) = In(\mathcal{U}) \cup Out(\mathcal{U})$
$= \{\vec{bd}\} \cup \{\vec{sa}, \vec{dt}\} = \{\vec{bd}, \vec{sa}, \vec{dt}\}$.

(c) $c\,[Cut(\mathcal{U})] = \sum_{e \in Out(\mathcal{U})} c(e) = c(\vec{sa}) + c(\vec{dt}) = 6 + 9 = 15$.

(d) $c\,[Cut(\{s,a\})] = 23$, $c\,[Cut(\{s,a,b\})] = 24$, $c\,[Cut(\{s,b\})] = 25$
$c\,[Cut(\{s,a,d\})] = 22$, $c\,[Cut(\{s,b,d\})] = 27$, $c\,[Cut\{s\}] = 16$
$c\,[Cut(\{s,d\})] = 15$, $c\,[Cut(\{s,a,b,d\})] = 21$. Since there are only 8 possible source-separating sets of vertices (all the subsets of $\{a,b,d\}$ plus $\{s\}$), it follows that $MinCut(N) = 15$.

<u>Remark</u>: 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 $MinCut(N)$ directly from the definition. Hence we need a fast algorithm for it.

Prop. 1 Let $\mathcal{U}$ be any source-separating set of vertices in a network N and f be a legal flow in N. Then

$$Val(f) = \sum_{e \in Out(\mathcal{U})} f(e) - \sum_{e \in In(\mathcal{U})} f(e).$$

Proof: Let $\mathcal{U} = \{s = x_1, x_2, \ldots, x_k\}$ & $\bar{\mathcal{U}} = \{t = y_1, y_2, \ldots, y_n\}$

Then from the definition of $Val(f)$ we have

$$Val(f) = \sum_{e \in In(y_1)} f(e) - \sum_{e \in Out(y_1)} f(e) \quad \ldots \quad (1)$$

Also by the conservation of flow for $y_2, \ldots, y_n$ we have

$$0 = \sum_{e \in In(y_2)} f(e) - \sum_{e \in Out(y_2)} f(e) \quad \ldots \quad (2)$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$0 = \sum_{e \in In(y_n)} f(e) - \sum_{e \in Out(y_n)} f(e) \quad \ldots \quad (n)$$

Adding equations (1), (2), ..., (n) we get

$$Val(f) = \sum_{i=1}^{n} \left\{ \sum_{e \in In(y_i)} f(e) - \sum_{e \in Out(y_i)} f(e) \right\}.$$

Let $A(\mathcal{U}, f) = \sum_{e \in Out(\mathcal{U})} f(e) - \sum_{e \in In(\mathcal{U})} f(e)$ and

$$B(\mathcal{U}, f) = \sum_{i=1} \left\{ \sum_{e \in In(y_i)} f(e) - \sum_{e \in Out(y_i)} f(e) \right\}.$$

We want to show $Val(f) = A(\mathcal{U}, f)$. We will show that $A(\mathcal{U}, f) = B(\mathcal{U}, f)$. Since $Val(f) = B(\mathcal{U}, f)$, it will follow that $Val(f) = A(\mathcal{U}, f)$. To show that $A(\mathcal{U}, f) = B(\mathcal{U}, f)$, we will show that $A(\mathcal{U}, f)$ & $B(\mathcal{U}, f)$ agree about the net contribution of $f(e)$ & $-f(e)$ for each edge $e \in E(G)$.

Let $e = \overrightarrow{uv}$ be any edge in $E(G)$ from u to v. Then there are four cases.

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

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

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

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

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

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

Proof: From Proposition 1, we have

$$
\begin{aligned}
Val(f) &= \sum_{e \in Out(\mathcal{U})} f(e) - \sum_{e \in In} f(e) \\
&\leq \sum_{e \in Out(\mathcal{U})} f(e) \quad \text{because } f(e) \geq 0 \\
&\leq \sum_{e \in Out(\mathcal{U})} c(e) = c[Cut(\mathcal{U})] \quad \text{bec. } f(u) \leq c(e).
\end{aligned}
$$

§2. The Ford-Fulkerson Algorithm & MaxFlow-Min Cut 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

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

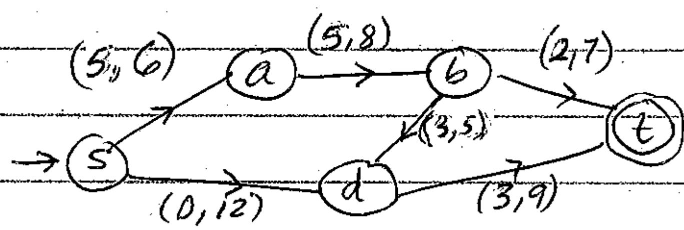
Observe that since $\text{Val}(f) \leq c[\text{Cut}(\mathcal{U})]$ for any legal flow f & source-sep. set of vertices in N, we immediately get $\text{MaxFlow}(N) \leq \text{MinCut}(\mathcal{U})$.

We will find a flow f^* in N along with a source-separting set of vertices $\mathcal{U}^*$ such that $\text{Val}(f^*) = c[\text{Cut}(\mathcal{U}^*)]$. 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 $sl(e)$ = maximum flow you can add to e in the direction from s to t.

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

Ex. 1



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

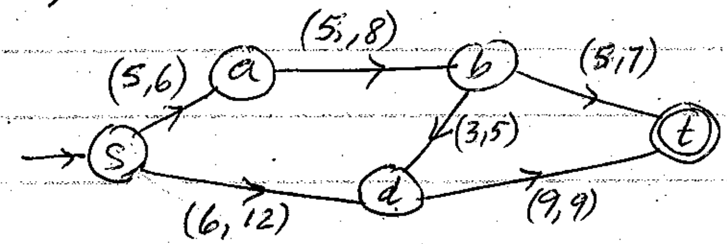
1. $s \xrightarrow{(0,12)} d \xrightarrow{(6,9)} t$

   $sl(\vec{sd}) = 12$, $sl(\vec{dt}) = 6$

2. $s \xrightarrow{(0,12)} d \xleftarrow{(3,5)} b \xrightarrow{(2,7)} t$

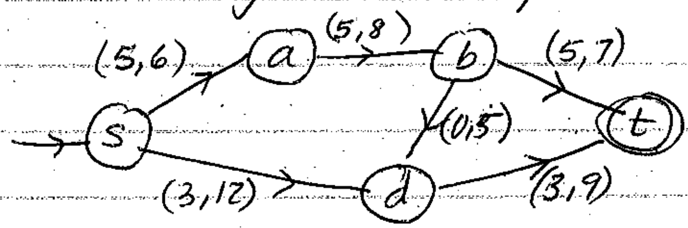
   $sl(\vec{sd}) = 12$, $sl(\vec{bd}) = 3$, $sl(\vec{bt}) = 5$.

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 $s \to d \to t$



$Val(\text{old } f) = 5$,
$Val(\text{new } f) = 11$.

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 $s \to d \leftarrow b \to t$ from s to t

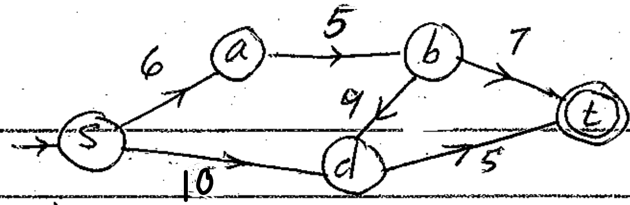


$Val(\text{old } f) = 5$,
$Val(\text{new } f) = 8$.

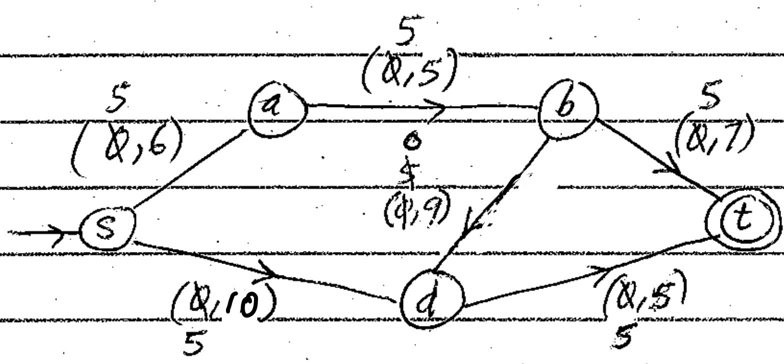
<u>Algorithm 1</u> (Ford-Fulkerson Algorithm)
<u>INPUT</u>: A network $N = \langle G, s, t, c \rangle$
<u>OUTPUT</u>: A maximal flow $f^*: E(G) \to \mathbb{R}^{\#}$.

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_i from s to t.
3. Compute the slack (with respect to f^*) of each edge e in the semi-path P_i and let
$$\mu_i = \min\{sl(e) : e \in P_i\}.$$
4. Let $f^*(e) \leftarrow f^*(e) + \mu_i$ for each forward edge e of P_i & $f^*(e) \leftarrow f^*(e) - \mu_i$ for each backward edge e of P_i, $i \leftarrow i+1$ & then go to step 2. (A <u>forward edge</u> e in a semi-path P from s to t is an edge e which goes in the direction from s to t. A <u>backward edge</u> e in P is an edge e 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 $\mathcal{U}^*$ such that $Val(f^*) = c[Cut(\mathcal{U}^*)]$.



1st augmenting semi-path, P_1:

$$s \xrightarrow{(0,6)} a \xrightarrow{(0,5)} b \xrightarrow{(0,9)} d \xrightarrow{(0,5)} t$$

Slacks: $\quad 6 \quad 5 \quad 9 \quad 5 \qquad \therefore \mu_1 = 5$

2nd augmenting semi-path, P_2:

$$s \xrightarrow{(0,10)} d \xleftarrow{(5,9)} b \xrightarrow{(0,7)} t$$

Slacks: $\quad 10 \quad 5 \quad 7 \qquad \therefore \mu_2 = 5$

There are no more augmenting paths. Also

$$Val(f^*) = \sum_{e \in In(t)} f^*(e) - \sum_{e \in Out(t)} f^*(e) = (5+5) - (0) = 10.$$

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

Thus

$$Val(f^*) = c[Cut(\mathcal{U}^*)].$$

Theorem 3: (MaxFlow - MinCut Theorem)
In any network, $MaxFlow(N) = MinCut(N)$.

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
$U^* = \{v \in V(G) : \text{there is an aug. semi-path from } s \text{ to } v\}$,
then $s \in U^*$ & $t \notin U^*$. So U^* is a source-separating set of vertices in N.

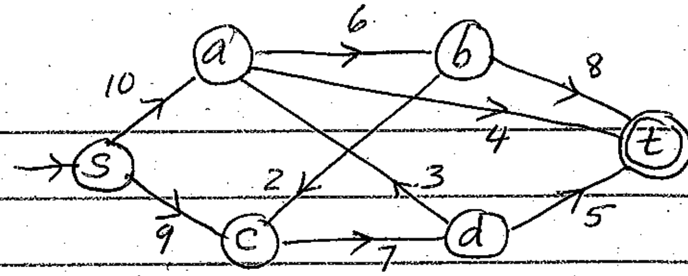
First, we know that
$$Val(f^*) = \sum_{e \in Out(U^*)} f^*(e) - \sum_{e \in In(U^*)} f^*(e),$$

Now consider an edge $e = \overrightarrow{vw}$ from $Out(U^*)$. 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 $e \in Out(U^*)$.

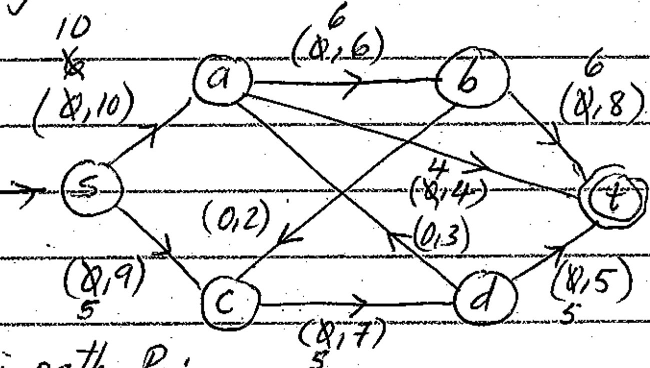
Also consider an edge $e = \overrightarrow{wv}$ from $In(U^*)$. If $f^*(e)$ was non-zero, then we would be able to send some more flow from s to w by push back some along the backward edge $\overrightarrow{wv}$. But this is impossible because $w \notin U^*$. Hence $f^*(e) = 0$ for each edge $e \in In(U^*)$. Thus
$$Val(f^*) = \sum_{e \in Out(U^*)} f^*(e) - \sum_{e \in In(U^*)} f^*(e) = \sum_{e \in Out(U^*)} c(e) - \sum_{e \in In(U^*)} 0 = c[Cut(U^*)]$$

Since $MaxFlow(N) \le MinCut(N)$ & we have $Val(f^*) = c[Cut(U^*)]$, it follows that $MaxFlow(N) = MinCut(N)$.

Ex.3 Let N be the network



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



1st aug. semi-path, P_1:

$s \xrightarrow{(0,10)} a \xrightarrow{(0,6)} b \xrightarrow{(0,8)} t$

Slacks: 10 6 8 $\therefore \mu_1 = 6$

2nd augmenting semi-path, P_2:

$s \xrightarrow{(6,10)} a \xrightarrow{(0,4)} t$

slacks: 4 4 $\therefore \mu_2 = 4$

3rd augmenting semi-path P_3:

$s \xrightarrow{(0,9)} c \xrightarrow{(0,7)} d \xrightarrow{(0,5)} t$

slacks 9 7 5 $\mu_3 = 5$

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

$= 6 + 4 + 5 = 15$

$c[Cut(U^*)] = \sum_{e \in Out(U^*)} c(e) = c(\vec{ab}) + c(\vec{at}) + c(\vec{dt}) = 6+4+5 = 15$ ✓