Menger's Theorem

Recall: $\mathcal{K}(x,y)$ = minimum size of an x,y-cut or xy-separator

$\lambda(x,y)$ = maximum no. of internally disjoint xy-paths.

An xy-cut must contain at least one vertex from each xy-path and no vertex can cut two internally disjoint xy-paths.

$\therefore \mathcal{K}(x,y) \geq \lambda(x,y)$ is again a dual-type of problem.

Proof is lengthy and will be ~~omitted,~~ left for reference only.

~~Proof of~~ Lemma 5.1: If P is an f-augmenting path with tolerance z, then changing flow by $+z$ on edges followed forward on P and by $-z$ on edges followed backward on P produces a feasible flow with $\mathrm{val}(f') = \mathrm{val}(f) + z$.

Proof: ~~We nee~~ NB:- we only need to check the edges on P, since other edges are not affected.

First, we need to check the capacity constraints:-

But, if e is a forward edge, then

$$0 \le \cancel{f'(e)} = f(e) + z = g(e) \le c(e),$$

and if e is backward, then

$$\cancel{c(e) \ge f'(e) = f(e) - z \to 0}$$

$$c(e) \ge f(e) \ge f(e) - z = g(e) \ge 0 \qquad (*)$$

$(*)$ by definition of z tolerance.

Next, we need to check the constraint $g^+(v) = g^-(v)$ on P.

There are four cases. (see diagram)

Case 1. $\left.\begin{aligned} g^+(v) &= f^+(v) + z \\ g^-(v) &= f^-(v) + z \end{aligned}\right| \implies g^+(v) = g^-(v)$

Case 2. $\left.\begin{aligned} g^+(v) &= f^+(v) + z \\ g^-(v) &= f^-(v) + z \end{aligned}\right| \implies g^+(v) = g^-(v)$

Simly. the remaining two cases.

Finally, we have $\mathrm{val}(g) = g^-(t) - g^+(t)$

$$= \begin{cases} f^-(t) + z - f^+(t) = f^-(t) - f^+(t) + z = \mathrm{val}(f) + z \\ f^-(t) - [f^+(t) - z] = f^-(t) - f^+(t) + z = \mathrm{val}(f) + z, \end{cases}$$

according as the final edge on P is forward or backward

Why does the Ford-Fulkerson Algorithm Work?

1. ~~NB:~~ Every s,t-path uses an edge from $[S,T]$, so intuitively the value of a feasible flow should be bounded (above) by the capacity of the cut $[S,T]$.

For any set of vertices U, we put

$f^+(U)$ = total flow on edges leaving U

$f^-(U)$ = total flow on edges entering U

The net flow out of U is then $f^+(U) - f^-(U)$.

We see: ~~$f^+(U)$~~ net flow out of U = sum of net flows of vertices in U.

In particular, if f is a feasible flow and $[S,T]$ is a source sink cut then net flow out of S and net flow into T equal val(f).

Proof: The claim is $f^+(U) - f^-(U) = \sum_{v \in U} (f^+(v) - f^-(v))$ (1)

We consider the contribution of the flow $f(xy)$ of an edge xy on both sides of U. If $x, y \in U$, then $f(xy)$ is not counted on the left, but it contributes +vely via x and negatively via y on right. If $x, y \notin U$, then $f(xy)$ doesn't contribute at all. If $xy \in [U, \bar{U}]$, then it contributes +vely on both sides. If $xy \in [\bar{U}, U]$, then it contributes negatively on both sides.

Summing over all edges, we get the equality (1)

When $[S,T]$ is a source-sink cut and f is a feasible flow, net flow from nodes of S sum to $f^+(s) - f^-(s)$ and net flow from nodes of T equals $f^+(t) - f^-(t) = -\text{val}(f)$, as reqd.

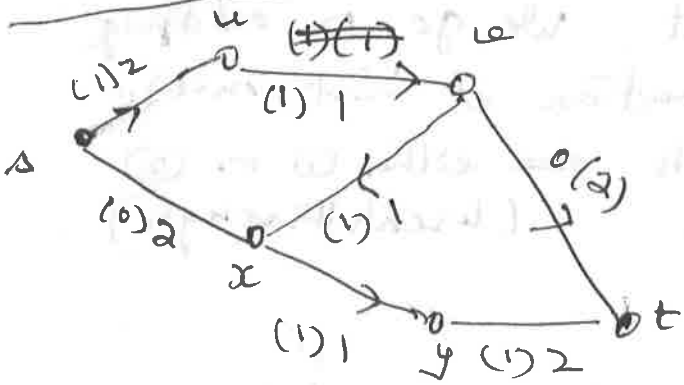
Corollary :- If f is a feasible flow, and $[S,T]$ is a source-sink cut, then $val(f) \leq cap[S,T]$.

By above: value of f = net flow out of S,

i.e. $val(f) = f^{+}(S) - f^{-}(S) \leq f^{+}(S)$, since the flow into S is not less than 0. ~~Since~~ capacity constraints require $f^{+}(S) \leq cap(S,T)$, we get the result.

From this we get the following :- out of all source-sink cuts, the one with min. capacity gives the best bound for $val(f)$, i.e. minimization of cap$([S,T])$ is dual to maximization of $val(f)$. If we get one with the equality, then we have got a solution to the problem.

Illustration of the algorithm



$R = \{s\}$, $S = \emptyset$

Stage 1:- Start with s — we have excess capacity on both u and x, so they are labelled as Reached from s.

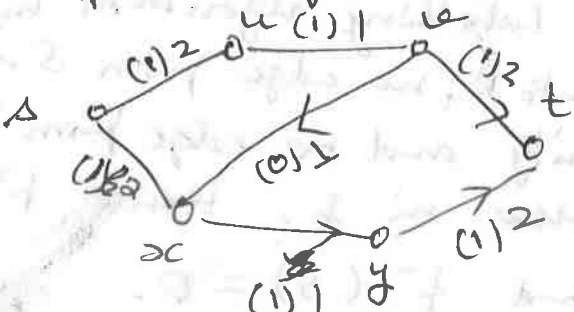
So now $R = \{s, u, x\}$, $S = \{s\}$.

There is no excess capacity on uv or xy:- so we can't get anything from u. But there is ~~excess capacity~~ non-zero flow on vx, so v can be added to R.

So now $R = \{s, u, x, v\}$, $S = \{s, u, x\}$.

Now v is the only element of $R - S$, and searching from v reaches t, so we have found an f-augmenting path: s, x, v, t.

The tolerance is 1. So we get a new feasible flow as follows:-



$R = \{s\}$, $S = \emptyset$

In this new flow, f', every edge has unit flow except $f'(vx) = 0$. When we run the ~~the~~ labelling algorithm again, we have excess capacity on su and sx, so we can add these to ~~$R - S$~~ R to get $R - S = \{s, u, x\}$. But from these we can label no others, so $R = \{s, u, x\}$, $S = \{s, u, x\}$, and we terminate. The cut is $[S, \bar{S}]$ with capacity 2

Proof of Theorem 5:- the zero flow is feasible and allows us to start. We go on adding vertices to S_2 (each vertex at most once), ~~and~~ terminates with ~~one~~ either ① or ②

① ~~[illegible]~~ :- $t \in R$ (breakthrough)

② ~~=~~ $S = R$.

In case ①, we have an f-augmenting path $\to$ we get a new feasible flow with increased flow - so we repeat. When the capacities are rational each augmentation increases the flow by a multiple of $\frac{1}{a}$ where a = lcd of denominators - so after finitely many iterations, the capacity of some cut is reached, so it terminates with $S = R$.

We now claim that $\text{cap}[S, \bar{S}] = \text{val}(f)$, hence by the earlier remark, this is the maximum possible flow.

It is a cut because $s \in S$ and $t \notin S$. Since applying the labelling algorithm introduces no node of $\bar{S}$ into R, no edge from S into $T = \bar{S}$ has excess capacity and no edge from T into S has non-zero flow in f. Hence, $f^+(S) = \text{cap}[S,T]$ and $f^-(S) = 0$.

But as we showed before
$$\text{val}(f) = f^+(S) - f^-(S) = \text{cap}[S,T]$$

$\to$