Residual network: for every edge (x, y) with (net) flow f_{xy},

$\bar{u}_{xy} = u_{xy} - f_{xy}$ (≥ 0 by capacity constraints)

$\bar{u}_{yx} = u_{yx} + f_{xy}$

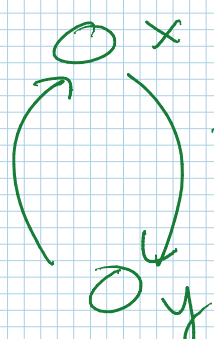
Augmenting path: directed path in residual network

Flow on augmenting path = min capacity of edge on the path in residual network.

Flow f on G + Flow f' on G_f (residual n/w wrt f)

= Flow $(f + f')$ on G

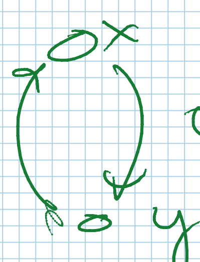
f sending α flow



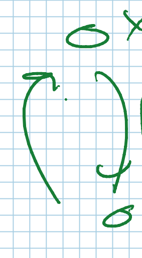
f' sending β flow

$\underline{\alpha \geq \beta}$ $\qquad$ $\underline{\beta \geq \alpha}$

$\longrightarrow$ $\alpha - \beta$



0 $\qquad$ 0



$\beta - \alpha$

✓ Flow conservation at x and y

✓ Capacity constraints: If $\alpha \geq \beta$, then follows from f

If $\beta \geq \alpha$, then $\beta \leq \alpha + u_{xy} \Rightarrow \beta - \alpha \leq u_{xy}$

Max flow - min cut Theorem

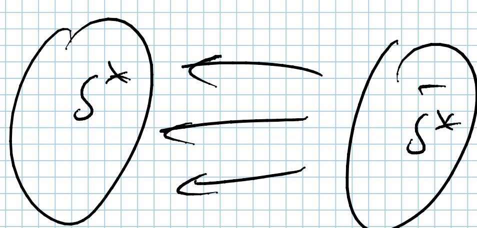
Max flow - min cut Theorem

Thm: The following are equivalent:

1. Flow from s to t is maximized
2. The residual network has no augmenting path with > 0 flow.
3. $\exists$ cut $(S^*, \bar{S}^*)$ s.t. $f(\bar{S}) = u(S^*, \bar{S}^*)$

Proof: $1 \Rightarrow 2$: Augmenting path can be added to increase flow as shown above, so flow cannot be max.

$2 \Rightarrow 3$: $S^* =$ all vertices reachable from s in residual network. Then $t \notin S^*$. So, $\bar{S}^* \neq \varphi$.



residual network

Hence all edges $(S^*, \bar{S}^*)$ are saturated and no flow on $(\bar{S}^*, S^*)$ edges.

$3 \Rightarrow 1$: For any cut $(S, \bar{S})$, $|f| \leq u(S, \bar{S})$ and $\exists$ cut $(S^*, \bar{S}^*)$

s.t. $|f| = u(S^*, \bar{S}^*)$. Therefore, f is maximized.

Ford-Fulkerson's Algorithm

Repeat until no augmenting path
→ Find augmenting path (by bfs)
→ Augment flow

Capacities	Running time
Integer	$O(m\|f\|)$ ← pseudo-poly time
Rational	converges
Real	may not converge

Pseudo-poly: poly in unary representation of input

Weakly poly: poly in binary representation of input

Strongly poly: poly in combinatorial parameters (# of input objects independent of their size)

Shortest Augmenting Path (Edmonds-Karp)

Algorithm: Use FF but augment on shortest (in terms of # of edges) path.

Lemma: For any vertex x, $d(s,x)$ is non-increasing in residual network.

Proof: Suppose among the vertices that came closer to s after augmentation, x is closest. Let y be predecessor of x on a shortest s-x path in residual network. Since y did not come closer, y must have been closer than x in residual network before augmentation. So, the edge (y,x) appeared in residual network because flow was sent on (x,y) edge. This is impossible since y was closer than x.

Theorem: Running time $= O(nm^2)$

Proof: At most $nm/2$ iterations since between two consecutive iterations using the same edge, the edge must have been used in the opposite direction, which, by the above lemma, implies that distance from s must have increased by at least 2 (and max distance is n). So, every edge in at most $n/2$ iterations.