Network Flows (Contd)

Algo2 $(G, \bar{c})$:

$\Delta \leftarrow$ threshold

$G(\Delta)$: • Keep the edges of G w/ capacity $\geq \Delta$.

• Delete the rest

• Find s-t paths in $G(\Delta)$

• Augment the flow by the bottleneck

• Construct a res. graph (and ensure edges have cap $\geq \Delta$)

• Repeat until there is no s–t path.

L iterations

$\geq L \cdot \Delta$

$m \cdot \log 400 \geq 400$

$m \cdot \log|C| \leq |C|$

$m \cdot n \leq 2^n \cdot e$

Algo2:

$O(\log|C| \cdot m \cdot O(m+n))$

$\log|C| \cdot \boxed{\text{max \# of iterations in each } \underset{2^i}{\Delta} \text{ phase}} \cdot O(m+n)$

$< 2m$

Algo 1:

$O(|C| \cdot (m+n))$

$O(|C| \cdot (m+n))$

$$\min \begin{cases} \sum_{s \to u \in E} c(s \to u) \\ \sum_{w \to t \in E} c(w \to t) \end{cases}$$

Init.

• $\Delta \leftarrow$ Largest power of 2 s.t it is smaller than $|C|$.

$$\frac{|C|}{2} \leq \Delta < |C|$$

$$\Delta = \max \{ 2^k \mid 2^k < |C| \}$$

• Construct the graph $G(\Delta)$:

1. Find a $s \leadsto t$ path in $G(\Delta)$ if it exists

YES → Update the flow.

$F \leftarrow F + \text{bottleneck}$. $F_{up} \geq F + \Delta$.

→ Compute the residual graph by removing edges of capacity $< \Delta$.

$O(m+n)$ bookkeeping
↳ to maintain flow and capacities w.r.t original graph.

→ Go to step 1.

NO → · Update $\Delta \leftarrow \frac{\Delta}{2}$.

· Update the graph (residual) by including edges of cap $\geq \Delta$

→ If $\Delta = 1$ and there are no s-t paths, return F.

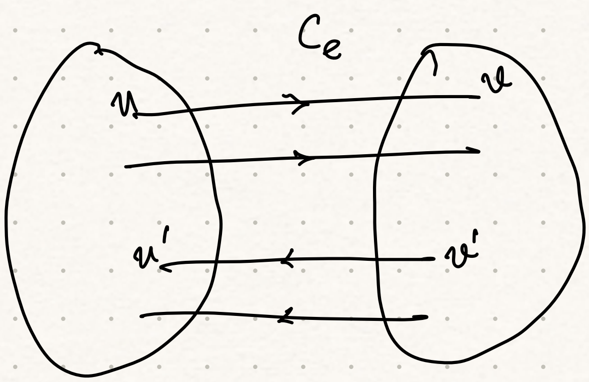
Need to bound the no. of iterations in each Δ phase.

Lemma: No. of augmentations in each $\frac{\Delta}{2}$-phase $\leq 2m$.

Claim: If F be the flow at the end of Δ-phase, then cap of the cut obtained at the end of Δ-phase in $G_F(\Delta)$ is at most $F + m \cdot \Delta$.

$u \to v \in E(G_{orig})$

$v' \to u' \in E(G_{orig})$



1. $C_e < f_e + \Delta$

If not, $C_e \geq f_e + \Delta$

⇒ there is a res. cap of $\geq \Delta$.

↳ Algo would not have terminated as this would lead to s-t path or v would be part of S itself.

← v is reachable from S in $G_F(\Delta)$ res.

2. Back edge $f_e < \Delta$

$$\text{Flow} = \sum_{\substack{\text{fwd edges} \\ s \to t}} f_e - \sum_{\substack{e': \\ \text{Back edges}}} f_{e'}$$

$$> \sum_{\substack{e \\ \text{fwd}}} (c_e - \Delta) - \sum_{\substack{e': \\ \text{back}}} \Delta .$$

$$> \underbrace{\sum_{\substack{e \\ \text{fwd}}} c_e} - \underbrace{\sum_{e,e'} \Delta}_{\leq m \cdot \Delta}$$

$$\underbrace{\sum_{e} c_e}_{\text{Capacity of the cut.}} < F + m \cdot \Delta .$$

$\leftarrow$ End of proof of claim.

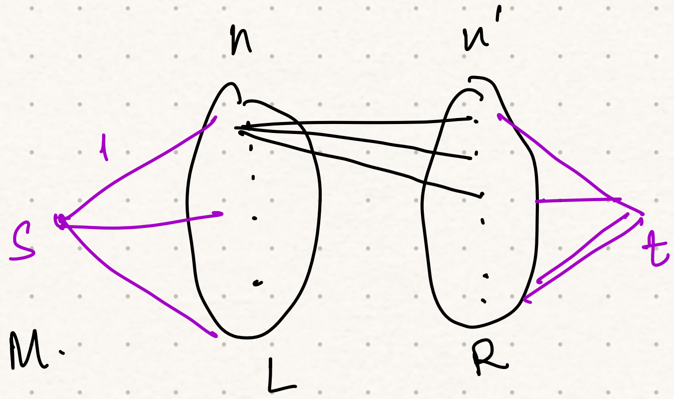
$F' \leftarrow$ be the augmented flow at the end of $\frac{\Delta}{2}$ phase.

$F \leftarrow$ Flow at the end of Δ-phase.

$$\underbrace{F + m \cdot \Delta > \underline{\underline{F'}}} \geq F + L \cdot \frac{\Delta}{2} . \Rightarrow \boxed{L < 2m}$$

Any feasible flow is at most cap. of any cut.

Bipartite Matching:



$M \subseteq E$

$v \in$ only one edge in the set M.

Perfect if every vertex has an incident edge in M.

$\hookrightarrow$ Add s,t, and add edges from s to every vertex in L

from every vertex in R to t.

↳ Assign cap of 1 to each edge.

Obs: Max flow gives max. matching.