CS 6381: Lecture #2 8.14.2020

(Strongly) Polynomial algorithms for Max-Flow

Promise #1 (CS 6363): Proof of E-K. alg & its Complexity

Recall: E-K selects paths with smallest number of edges in G_f from s to t at each step [done efficiently using BFS]

Now we show that this yields a strongly poly alg. even in the presence of irrational capacities (provided you can add, subtract numbers in one step)

This proof is from CLRS Ch 26:

Let $\delta_f(s,v)$ = the min. # of edges in a path from s to v in G_f.

Overview of the proof:

1. $\delta_f(s,v)$ is monotone non-decreasing as the algorithm progresses for all $v \in V$.
2. Let $P^f(s,t)$: represent the augmenting path in G_f from s to t.

   We say edge $(u,v) \in P^f(s,t)$ is Critical iff $c^f(P^f(s,t)) = c^f(u,v)$.

3. Each edge (u, v) is critical at most $\frac{|V|-1}{2}$ times (in the direction (u, v); and can at most equal number in the reverse direction)

4. (3) $\Rightarrow$ # of augmentations $\leq \frac{|E| \cdot (|V|-1)}{2}$
$= \Theta(|E| \cdot |V|)$

5. The effort in finding $P^f(s,t)$ is $\Theta(|E|+|V|)$ using BFS.

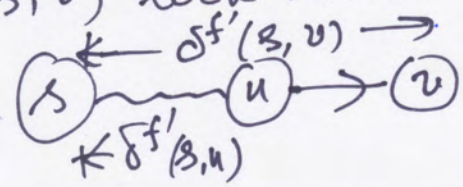
6. Hence, E.K. algorithm is strongly polynomial.

Details of the proof:

1) Suppose the claim is not true. At some step, suppose $\delta_f(s,v)$ decreases.
Let f be the flow before and f' .. .. after | such that $\delta^f(s,v) > \delta^{f'}(s,v)$

Among such vertices, let v one with min $\delta_{f'}(s,v)$

Let $P^{f'}(s,v)$ look like



| ~~$\delta_{f'}(s,v) = \delta_{f'}(s,u)+1$~~

Since $\delta_{f'}(s,u) < \delta_{f'}(s,v)$, $\delta_{f'}(s,u) \geq \delta_f(s,u)$

Claim $(u, v) \notin E^f$.

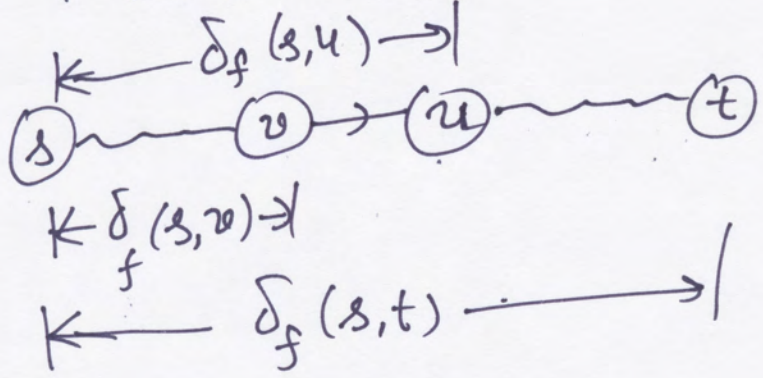
Pf Suppose not; then

$$\begin{aligned} \delta_f(s,v) &\le \delta_f(s,u) + 1 \\ &\le \delta_{f'}(s,u) + 1 \\ &= \delta_{f'}(s,v) \end{aligned}$$

which contradicts our choice of v: $\delta_{f'}(s,u) < \delta_f(s,v)$

Hence $(u,v) \notin E^f$, but $(u,v) \in E^{f'}$

Hence $P^f(s,t)$ looks like



$$\begin{aligned} \therefore \delta_f(s,v) &= \delta_f(s,u) - 1 \\ &\le \delta_{f'}(s,u) - 1 \\ &= \delta_{f'}(s,v) - 2 \end{aligned}$$

Hence contradiction to $\delta_f(s,u) < \delta_f(s,v)$

Completes Proof of (1).

Proof of B)

At each step, at least one edge is Critical.

If an edge (u, v) is critical at some step, this edge (in this direction) is not present in the next residual graph. Prior to its appearing again (in the same direction) at some later step in the residual graph, an augmenting path using (v, u) (in this direction) must have been used.

Suppose (u, v) is critical in G^f & $G^{f''}$

$$G^f \ldots\ldots G^{f'} \ldots\ldots G^{f''}$$

$\exists\ G^{f'}$ in which (v, u) is used.

$\delta_f(s, v) = \delta_f(s, u) + 1$ since (u, v) is Critical & hence in $P^f(s, t)$

$\delta_{f'}(s, u) = \delta_{f'}(s, v) + 1$ since $(v, u) \in P^{f'}(s, t)$

And $\delta_{f'}(s, u) \geq \delta_f(s, u)$

$\therefore \delta_{f'}(s, u) = \delta_{f'}(s, v) + 1$

$\geq \delta_f(s, v) + 1$

$= \delta_f(s, u) + 2$

Hence between two successive instances of (u,v) being critical, # of edges in the augmenting path increases by 2.

# of edges in an augmenting path (simple) $\leq |V|-1$.

Hence, a single edge (u,v) (in that direction) can not be critical more than $\frac{|V|-1}{2}$ times

Hence # of augmentations $\leq |E|\left(\frac{|V|-1}{2}\right)$

Hence algorithm is (strongly) polynomial

function does not depend on $C(u,v)$.

$|E|\left(\frac{|V|-1}{2}\right).(|E|+|V|) = O(|V|^5)$

BFS for dense graphs

Now we improve complexity in upcoming algorithms.

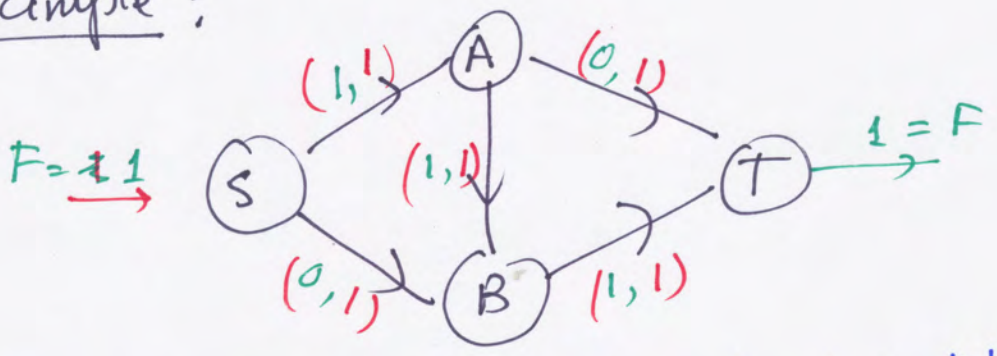
Material will be from S. Even, ~~et al~~ Ahuja et al etc

Layered Networks, Saturating Flows, E.A. Dinic / E.A. Dinitz

Saturating Flow Vs Maximum Flow

A feasible flow is Saturating if we can not increase F without decreasing flow on some edge.

Example:



This is a saturating flow that is NOT a maximum flow.

All maximum flows are saturating flows

~~Layered Network Construction~~: This reduces work in constructing G_f to fewer times

After we construct residual graph G_f, we donot need to restrict ourselves to augmentation along only one path. Any feasible flow on G_f can be super-imposed on f in G to get a new improved feasible flow in G.

One such feasible flow in G^f is a Saturating flow. A saturating flow needs no more than $|E|$ flow augmentations on paths since at each step one edge is saturated. This is true whether it is done on G or on G^f. This is the main importance of Saturating flows.

It turns out that there is a way to "use up" all paths having minimum # of edges in "one-shot". The next residual graph will have paths with larger number of edges at each step. This is the idea of E.A. Dinic using Layered Networks

Construction of Layered Networks

Input: Given a feasible flow f in G;

Step 1: Let $V_0 = \{s\}$ be the initial layer.

Step 2: Given V_i Construct V_{i+1} as follows

$$V_{i+1} = \{j : k \in V_i, (k,j) \in G^f\}$$

as long as $\bigcup_{\ell=0}^{i+1} V_\ell$ does not include t.

Else, if V_{i+1} includes t, redefine

$$V_{i+1} = \{t\}.$$

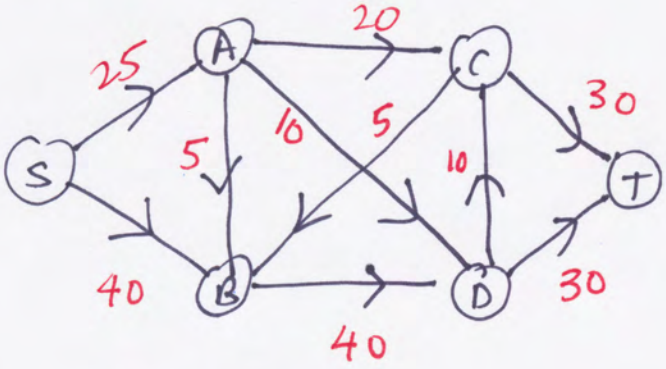
Step3: Edges in Layered network connect nodes in adjacent layers only and all of these are in E^f. Their capacities are the same as $c^f(u,v)$ in G^f.

Step4
If $t \notin$ Layered network; STOP; F is maximum.
Else, find a saturating flow in Layered Network from s to t. Superimpose on f in G and repeat step 1 - 4.

Clearly, this process produces flows that increase in value as the algorithm progresses and each saturating flow is found in polytime. The remaining part is to show that the number of layers increases at each step and since it is bounded by $|V|$, the algorithm itself is strongly polynomial. Before completing proof, we will use an example.

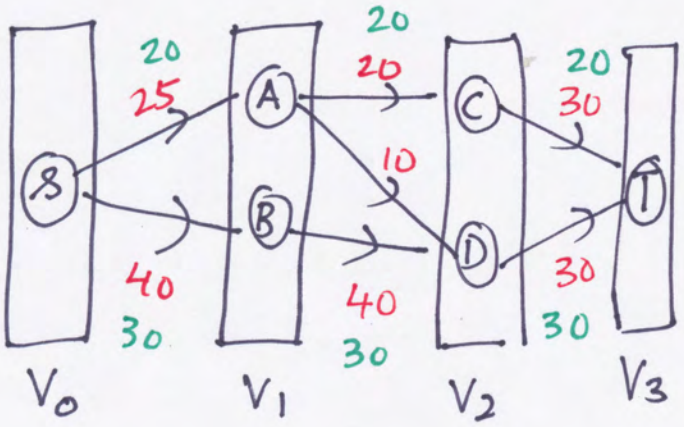
Note: All s-t paths in a layered network have the same length = # of layers.

Example



$= G = G_f^{(0)}$ ↓ $f \equiv 0$.

Layered Network (1)

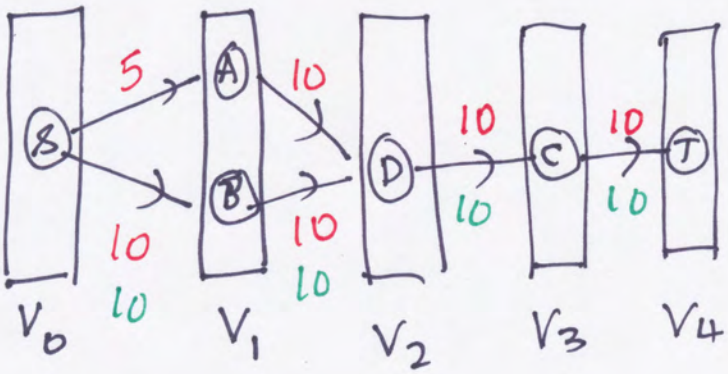


A Saturating flow in green

F = 50 so far

0th 1st layered network

Layered Network (2)



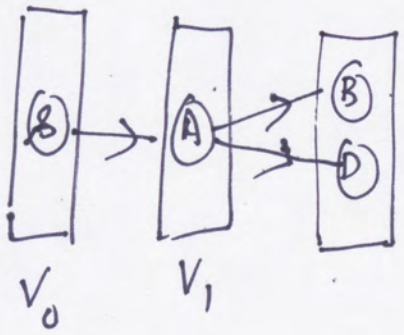
~~1st~~ 2nd L. N.

Saturating flow in green.

Residual capacities in red.

F = 60 so far.

Layered Network (3)



End

Min cut

$\{S, A, B, D\}, \{C, T\}$

Cut capacity in original graph

AC, DC, DT

~~20~~ 20 + 10 + 30 = 60

~~F = 6~~ F = 60 max

E.A. Dinic found a saturating flow in LN one path at a time and hence had a higher complexity

---

# of layered networks $\leq |V|$.

of ~~augm~~ augmenting paths in one LN: $\leq |E|$

Finding one path $\leq |V|$

$\therefore$ Complexity $|E||V|^2 \rightarrow |V|^4$ in dense graphs

$< |V|^5$ before.

EK

What remains is to show # of layers is Strictly increasing as algorithm progresses.

This is taken from S. Even's book p: 97-100

Pseudo-Codes 101-104

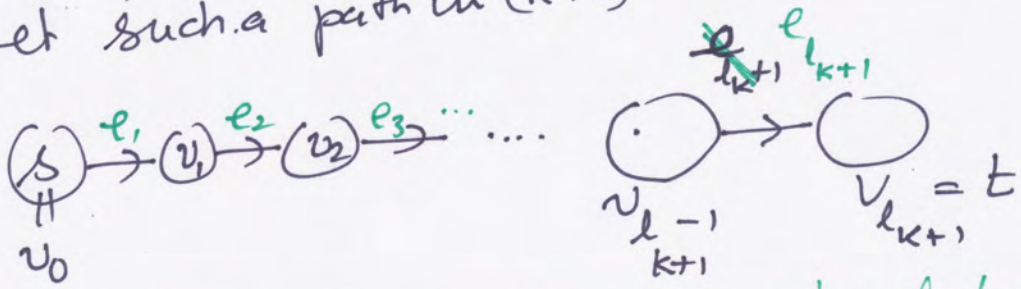
Let us call the part of the algorithm that starts with a flow f in G, ~~find~~ Constructs LN, finds a saturating flow $\tilde{f}^*$ in LN and Superimposes $\tilde{f}$ on f to get a new flow f' in G — one phase of the algorithm. Let l_k denote the length of LN in k^{th} phase.

Lemma: If $(k+1)^{st}$ phase is not the end of the algorithm (i.e last layer has t in it) then $l_{k+1} > l_k$.

Remark: This in turn implies # of phases $\leq |V|$.

Pf: Every path in k^{th} Layered network from s to t has length l_k ; $k = 1, 2, \ldots$

Let such a path in $(k+1)^{st}$ LN look like



§ **Case 1**: All nodes in this path also belong to LN in k^{th} phase [Please note that a LN need not contain all nodes]

Even though ⊃ is true, the layer number for a node need not be same in both k^{th} & $(k+1)^{st}$ LN.

Let V_j represent j^{th} layer in k^{th} LN

Claim

If $\# v_a \in V_b$ then $a \geq b$

Here the node v_a is in a^{th} layer of $(k+1)^{st}$ LN
And b^{th} .. .. k^{th} LN

Proof. By induction on a.

$a=0$, $v_a = s$ in

BASIS CASE $s \in 0^{th}$ layer in all LN.

$\therefore$ $b=0$ when $a=0$; $a \geq b$ holds

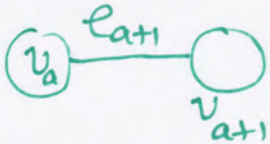
I.H: Result holds for a, show for a+1.

Suppose by way contradiction

$\# v_{a+1} \in V_c \neq \phi$

If $c \leq b+1$, since $a \geq b$, $a+1 \geq b+1 \geq c$

Result continues to hold.

If $c > b+1$, edge e_{a+1} has not been used in k^{th} phase, since it is not in k^{th} LN (edges in LN go only between successive layers). If 

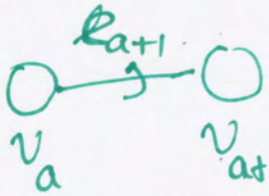
has not been used in k^{th} LN, and is in $(k+1)^{st}$ LN, it was in k^{th}, G f and hence v_{a+1} can not belong to V_c, $c > b+1$

Now, in particular $t \in V_{l_k}$, $t = v_{l_{k+1}}$

$\therefore l_{k+1} \geq l_k$ (**)

Moreover, equality can not hold in (**) since working backwards, equality holds for the entire path contradicting we saturated k^{th} LN.

Case 2: Not all vertices in path in $(k+1)^{st}$ LN appear in k^{th} LN.

Let 

be the first edge of the path in $(k+1)^{st}$ LN such that $v_a \in V_b$ in k^{th} LN

and v_{a+1} not in k^{th} LN.

Thus, e_{a+1} was not used in (k^{th}) phase and as such flows did not change on e_{a+1} in k^{th} phase.

The only possible reason for v_{a+1} not to belong to V_{b+1} is because $b+1 = l_k$

$\therefore a+1 \geq l_k \therefore l_{k+1} > l_k$. End of proof