CMSC 451 Nov. 2

Network Flow (III)

Ford-Fulkerson Algorithm

Repeat finding augment paths on the residual network.

Max-flow Min-cut Theorem

$$\max_{f:\text{flow}} |f| = \min_{(X,Y)\text{ cut}} c(X,Y)$$

and f is a max-flow if G_f has no path $s \to t$.

Time complexity of FF:

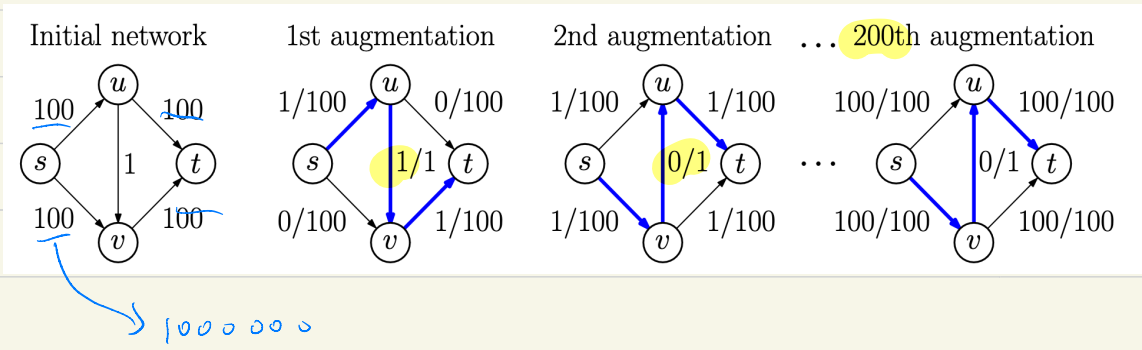
Lemma. If all capacities are integers, then the max flow is also an integer.

Proof. Running the FF algorithm, and all numbers occurred are integers.

- Each augmentation, flow value increase at least one
- At most C rounds
  - → max possible flow value, $C = \sum$ all capacities, n: max capacity.
- Time complexity: $O(m \cdot C)$ — pseudo-polynomial
  - → input length = $\log C$.

polynomial time algorithm: running time is polynomial of input length.

Is it tight?



FF: select ANY augment path.

Idea: select the largest (fattest) augment path.

Question: how to find the largest augment path?

$c[v] :=$ max capacity from s to v.

Dijkstra-like algorithm:

$c[s] = +\infty$, $c[v] = 0$ for other $v \in V$.

repeat

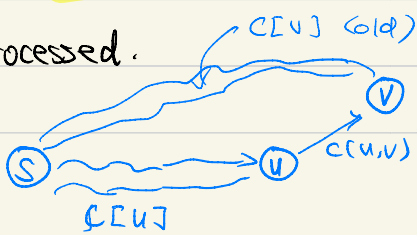
$u \leftarrow$ pop_max(Q) ← $c[u]$ is finished

for $(u,v) \in E$.

$c[v] = \max(c[v], \min(c[u], c(u,v)))$

until all vertices processed.

Time: $O(m \log n)$



Time complexity ? Each round : $O(m \log n)$ $O(m \log C)$

$\Rightarrow$ How many rounds the algo. need ?

Let f be the current flow, f^* be the max flow.

Δ be the max augment path's capacity on G_f

↑ hope this is large.

Claim: $\Delta \geq \frac{1}{m}(|f^*| - |f|)$

↳ what's left to improve.

> Theorem. Any flow can be decomposed into at most m paths.

Proof.

Given a flow f, :

Repeat the process.

1. Find a path $s \to t$ w/ non-zero flow
2. $c \leftarrow$ min flow value along the path
3. subtract c flow along the path.

Until zero flow left.

Each round:

- Find a path,
- at least one edge reduced to zero flow

→ at most m rounds.

Claim: $\Delta \geq \frac{1}{m}(|f^*| - |f|)$

Let $p_1, p_2, \dots, p_k$ ($1 \leq k \leq m$) paths found in the process decomposing $f^* - f$, $|f^*| - |f| = \sum_{i=1}^{k} |p_i|$

$$\Rightarrow \Delta \geq \max_{1 \leq i \leq k} |p_i| \geq \frac{1}{k}(|f^*| - |f|)$$

$$\geq \frac{1}{m}(|f^*| - |f|)$$

max/fattest augment path.

Claim: $\Delta \geq \frac{1}{m}(|f^*| - |f|)$

$\exists$ path in G_f, s.t. $s \to t$.
all edges in the path of cap. $\geq \Delta$

Alternative Proof using Min-Cut Theorem.

Let X be set of vertices reachable from s using only edges $> \Delta$.

Y be other vertices.

$\Rightarrow (X, Y)$ is a cut. $\Leftrightarrow t \notin X$

if $t \in X$, $\Rightarrow \exists$ aug. path $> \Delta$, contradiction.

$\Rightarrow$ all the edges (u, v), $u \in X$, $v \in Y$.
$c(u, v) \leq \Delta$
↳ in G_f

$\Rightarrow c(X, Y) \leq \Delta \cdot m$

$|f^*| - |f|$ = max flow on G_f $\leq c(X, Y) \leq \Delta \cdot m$

$\Rightarrow \quad \Delta \geq \frac{1}{m}(|f^*| - |f|)$

Time complexity.

Each round: before f, $\Rightarrow$ after, f'

$$|f^*| - |f'| \leq (1 - 1/m) \cdot (|f^*| - |f|)$$

After $O(m \log |f^*|)$ rounds,

$|f^*| - |f'| < 1 \Rightarrow$ max flow found.

Running time: $O(m^2 \log n \log C)$

Polynomial time.

still only weakly polynomial time !

weakly polynomial $\Rightarrow$ running time depends on # bit of input.

strongly polynomial $\Rightarrow$ running time does not depend on magnitude of numbers in the input.

Only related to m & n, not related to C, even $\log C$.

Capacity Scaling Algorithm.

Let U be the largest capacity.

Algorithm:

$\Delta \leftarrow$ max power of 2 s.t. $\leq U$.

repeat.

Run FF, but only consider edges in G_f w/ residual cap. $\geq \Delta$.

$\Delta \leftarrow \Delta/2$

until $\Delta < 1$

Claim: after Δ-round, $|f^*| - |f| \leq m \cdot \Delta$

Proof. consider G_f after the Δ-round.

max aug. path on it must $< \Delta$

cut (X, Y) s.t. $X =$ set of vert. reachable using $\geq \Delta$ edges from s.

$\Rightarrow \quad c(X, Y) \leq m \cdot \Delta$

Δ-round $\Rightarrow$ $|f^* - f| \le m \cdot \Delta$

next round, threshold $= \Delta/2$.

FF, each path improve at least $\Delta/2$

overall $m\Delta$ to improve $\Rightarrow$ at most $2m$ rounds.

overall, $\log U$ number of halving Δ.

$$\log U \cdot 2m = O(m \log U)$$

Time: $O(m^2 \log U)$

$C = \sum$ all capacities

$U =$ max capacity

$C \le U \cdot m$

Edmonds-Karp:

Find the shortest augment path. ← BFS

$O(nm)$ rounds

Time: $O(nm^2)$ ← strongly polynomial.