FLOWS

MAX-FLOW

Given a directed graph $G=(V,E)$ where every edge e has c_e or $c(e) \in \{1 \dots v\}$

Def. A flow [vector] $f \in \mathbb{R}^m$

1. $f \geq 0$
2. Conservation of flow: $\forall v \notin \{s,t\}$

$$\sum_{e=(u,v)} f_e = \sum_{e=(v,w)} f_e$$

3. Total conservation

$$\sum_{e=(s,u)} f_e = \sum_{e=(v,t)} f_e$$

Def. f is feasible when

$$\forall e \in E : f_e \leq c_e$$

LP formulation:

$$\max \sum_{e=(s,u) \in E} f_e$$

s.t.

$$f_e \leq c_e \quad \forall e$$

+ conservation constraints

$$f \geq 0$$

Runtime w/ IPM: $O(m^{7/2} L)$ where $L = \text{polylog}(U)$

STOC '13 Orlin: $O(mn)$ SP

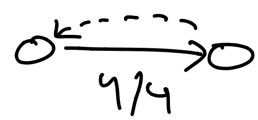
CP-free approach

Ford–Fulkerson. Runtime $O(m f^*)$ where f^* is value of max-flow.

FF:

G_f = res. graph

$f := 0$



while there is an augmenting path in G_f,
let f' be some flow on the path.
$f = f' + f$. Fix G_f.

Min Cut – Max Flow Thm:

Value of Max-flow = value of min-cut in G

Cut $= (S, T)$ $\quad S \cup T = V$, $s \in S$, $t \in T$

Runtime:

$O(m f^*)$ Pseudo polynomial

Capacity-Scaled FF:

G, each edge has c_e or $c(e)$

Largest capacity ~~thats~~ is bit-string of length $\lceil \log_2 U \rceil$

$$
\begin{array}{c|c|c|ccc}
\text{edges:} & & 8 & 4 & 2 & 1 \\
e_1 & 0 & 1 & 0 & 1 & 0 \\
e_2 & 0 & 1 & 1 & 1 & 0 \\
e_3 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

10

$f = 0$

worst case: I've appended 1 to $c'(e)$ for every e

the min-cut can't be larger than m-edges

$$\Rightarrow f^* \leq m$$

Every iter. I run FF $\rightarrow O(m^2)$

RT: $O(\log_2 U \cdot m^2)$ WP

Blocking flows: (Dinic 70's)

Def. For each v in the residual graph, let $\ell(v)$ be the length of shortest $s \rightsquigarrow v$ path.

Def. An edge $e = (u,v)$ is admissible if $\ell(v) = \ell(u) + 1$

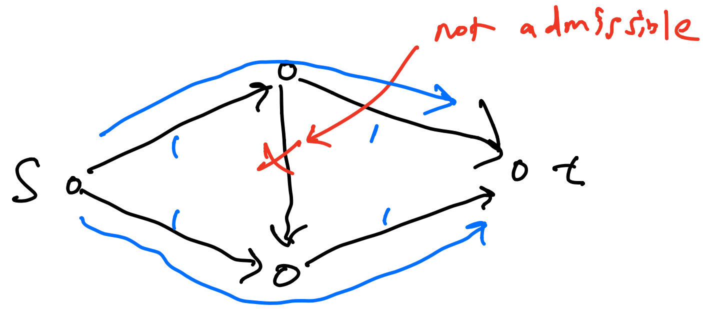
$\Rightarrow$

Def. The level graph $L \subset G_f$. L contains only the admissible edges.

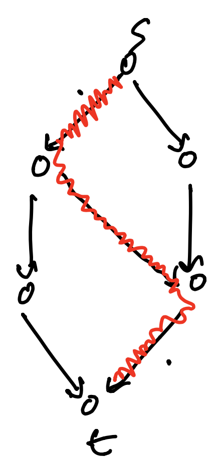
L = DAG.

Def. A path is admissible if all of its edges are admissible. A flow is admissible if its made up of admissible $s \rightsquigarrow t$ paths.

Def. A blocking flow is an admissible flow where on each $s \rightsquigarrow t$ path, at least one edge is saturated $(c_e = f_e)$.



unit-cap



BF = 1

MF = 2

Claim: After augmenting along a blocking flow, the shortest path distance $[d(s,t)]$ from $s \rightsquigarrow t$ in the new Level graph strictly increases.

$\Rightarrow$ RT: $O\left(n \cdot \left(\text{time to find a blocking flow}\right)\right)$

Prove:

L = level graph before augmentation

L' = " " after "

L contains only admissible edges.

What edges can appear in L'?

- some edges $L \leftarrow$
- $rev(L)$
- orange edges $\in G_f$ but not in L

$$d_{L'}(s,t) \geq d_L(s,t)$$

Q: can $d_{L'}(s,t) = d_L(s,t)$: **NO**!

$\Rightarrow d_{L'}(s,t) > d_L(s,t)$

Finding a blocking flow:

1) Special case: unit-capacity

modified DFS

1) advance
2) retreat
3) augment

$O(m)$

$\Rightarrow$ RT : unit-cap : $O(mn)$

General capacity graphs

1) advance,
after every n advances, then what?
$\rightarrow$ delete 1 edge
mn

2) retreat
$\leq m$ times

3) augment
delete ≤ 1 edge per aug.
$\leq m$ times

RT: $O(mn) + O(m) + O(m) = O(mn)$

Actual Runtime: $O(mn^2)$ SP

Thanks!

Hopcroft-Karp Bipartite-matching in $O(m\sqrt{n})$