Lecture 18: Max-Flow Min-Cut

Michael Dinitz

October 28, 2021
601.433/633 Introduction to Algorithms

Introduction

Flow Network:

- Directed graph $\mathbf{G = (V, E)}$
- *Capacities* $\mathbf{c : E \to \mathbb{R}_{\geq 0}}$ (simplify notation: $\mathbf{c(x, y) = 0}$ if $\mathbf{(x, y) \notin E}$)
- Source $\mathbf{s \in V}$, sink $\mathbf{t \in V}$

Today: flows and cuts

- Flow: "sending stuff" from $\mathbf{s}$ to $\mathbf{t}$
- Cut: separating $\mathbf{t}$ from $\mathbf{s}$

Turn out to be very related!

Today: some algorithms but not efficient. Mostly structure. Better algorithms Thursday.

Flows

Intuition: send "stuff" from $\mathbf{s}$ to $\mathbf{t}$

- Water in a city water system, traffic along roads, trains along tracks, ...

Definition

An *$(\mathbf{s},\mathbf{t})$-flow* is a function $\mathbf{f}:\mathbf{E}\to\mathbb{R}_{\geq 0}$ such that

$$\sum_{\mathbf{u}:(\mathbf{u},\mathbf{v})\in\mathbf{E}} \mathbf{f}(\mathbf{u},\mathbf{v}) = \sum_{\mathbf{u}:(\mathbf{v},\mathbf{u})\in\mathbf{E}} \mathbf{f}(\mathbf{v},\mathbf{u})$$

for all $\mathbf{v}\in\mathbf{V}\setminus\{\mathbf{s},\mathbf{t}\}$. This constraint is known as *flow conservation*.

Value of flow $|\mathbf{f}|$: "total amount of stuff sent from $\mathbf{s}$ to $\mathbf{t}$"

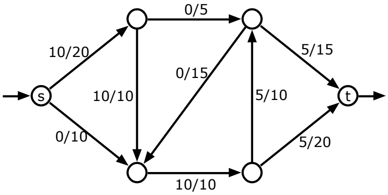
$$|\mathbf{f}| = \sum_{\mathbf{u}:(\mathbf{s},\mathbf{u})\in\mathbf{E}} \mathbf{f}(\mathbf{s},\mathbf{u}) - \sum_{\mathbf{u}:(\mathbf{u},\mathbf{s})\in\mathbf{E}} \mathbf{f}(\mathbf{u},\mathbf{s}) = \sum_{\mathbf{u}:(\mathbf{u},\mathbf{t})\in\mathbf{E}} \mathbf{f}(\mathbf{u},\mathbf{t}) - \sum_{\mathbf{u}:(\mathbf{t},\mathbf{u})\in\mathbf{E}} \mathbf{f}(\mathbf{t},\mathbf{u})$$

Feasible Flows

Capacity constraints: $\mathbf{0 \le f(u,v) \le c(u,v)}$ for all $\mathbf{(u,v) \in V \times V}$

Definitions:

- An $\mathbf{(s,t)}$-flow satisfying capacity constraints is a *feasible* flow.
- If $\mathbf{f(e) = c(e)}$ then $\mathbf{f}$ *saturates* $\mathbf{e}$.
- If $\mathbf{f(e) = 0}$ then $\mathbf{f}$ *avoids* $\mathbf{e}$.



An (s,t)-flow with value 10. Each edge is labeled with its flow/capacity.

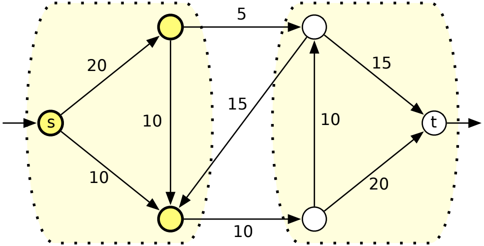
Problem we'll talk about: find feasible flow of maximum value (max flow)

Cuts

Definition

- An $(\mathbf{s},\mathbf{t})$*-cut* is a partition of $\mathbf{V}$ into $(\mathbf{S},\bar{\mathbf{S}})$ such that $\mathbf{s} \in \mathbf{S}$, $\mathbf{t} \notin \mathbf{S}$
- The *capacity* of an $(\mathbf{s},\mathbf{t})$-cut $(\mathbf{S},\bar{\mathbf{S}})$ is

$$\mathbf{cap}(\mathbf{S},\bar{\mathbf{S}}) = \sum_{(\mathbf{u},\mathbf{v}) \in \mathbf{E}: \mathbf{u} \in \mathbf{S}, \mathbf{v} \in \bar{\mathbf{S}}} \mathbf{c}(\mathbf{u},\mathbf{v}) = \sum_{\mathbf{u} \in \mathbf{S}} \sum_{\mathbf{v} \in \bar{\mathbf{S}}} \mathbf{c}(\mathbf{u},\mathbf{v})$$



Problem we'll talk about: find $(\mathbf{s},\mathbf{t})$-cut of minimum capacity (min cut)

Warmup Theorem

Theorem

Let $\mathbf{f}$ *be a feasible* $(\mathbf{s},\mathbf{t})$*-flow, and let* $(\mathbf{S},\bar{\mathbf{S}})$ *be an* $(\mathbf{s},\mathbf{t})$*-cut. Then* $|\mathbf{f}| \leq \mathbf{cap}(\mathbf{S},\bar{\mathbf{S}})$.

$$
\begin{aligned}
|f| &= \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) && \text{(definition)}\\
&= \sum_{u\in S}\left(\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u)\right) && \text{(flow conservation constraints)}\\
&= \sum_{u\in S}\left(\sum_{v\in \bar{S}} f(u,v) - \sum_{v\in \bar{S}} f(v,u)\right) && \text{(remove terms which cancel)}\\
&\leq \sum_{u\in S}\sum_{v\in \bar{S}} f(u,v) && \text{(flow is nonnegative)}\\
&\leq \sum_{u\in S}\sum_{v\in \bar{S}} c(u,v) = \mathrm{cap}(S,\bar{S}) && \text{(flow is feasible)}
\end{aligned}
$$

Max-Flow Min-Cut

Corollary

If $\mathbf{f}$ *avoids every* $\bar{\mathbf{S}} \to \mathbf{S}$ *edge and saturates every* $\mathbf{S} \to \bar{\mathbf{S}}$ *edge, then* $\mathbf{f}$ *is a maximum flow and* $(\mathbf{S}, \bar{\mathbf{S}})$ *is a minimum cut.*

Theorem (Max-Flow Min-Cut Theorem)

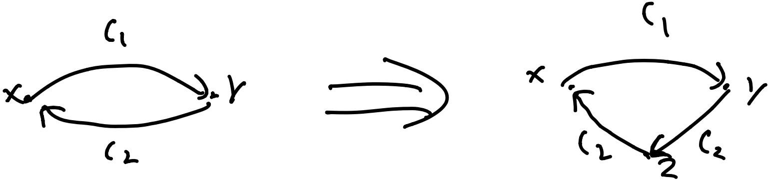
In any flow network, value of max $(\mathbf{s}, \mathbf{t})$*-flow* $=$ *capacity of min* $(\mathbf{s}, \mathbf{t})$*-cut.*

Spend rest of today proving this.

- Many different valid proofs.
- We'll see a classical proof which will naturally lead to algorithms for these problems.

One Direction

Cycles of length **2** will turn out to be annoying. Get rid of them.



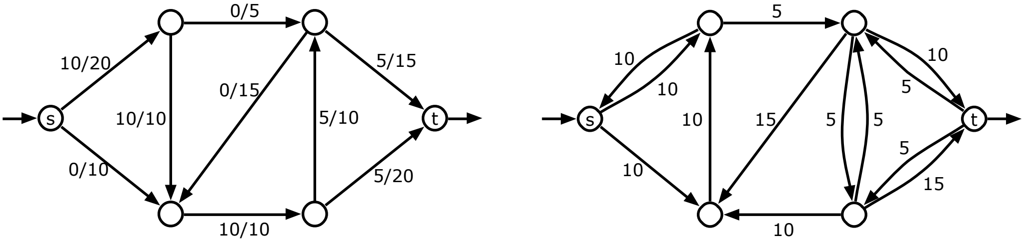
- Doesn't change max-flow or min-cut
- Increases #edges by constant factor, # nodes to original # edges.

Residual

Let $\mathbf{f}$ be feasible $(\mathbf{s},\mathbf{t})$-flow. Define *residual capacities*:

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \\ f(v,u) & \text{if } (v,u) \in E \\ 0 & \text{otherwise} \end{cases}$$

Residual Graph: $G_f = (V, E_f)$ where $(u,v) \in E_f$ if $c_f(u,v) > 0$.



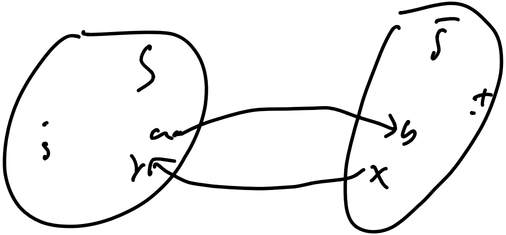
A flow f in a weighted graph G and the corresponding residual graph G_f.

Start of Proof

Let $\mathbf{f}$ be a max $(\mathbf{s},\mathbf{t})$-flow with residual graph $\mathbf{G_f}$.
Want to Show: There is a cut $(\mathbf{S},\bar{\mathbf{S}})$ with $\mathbf{cap}(\mathbf{S},\bar{\mathbf{S}}) = |\mathbf{f}|$.

Case 1: There is no $\mathbf{s} \to \mathbf{t}$ path in $\mathbf{G_f}$

Let $\mathbf{S} = \{\text{vertices reachable from } \mathbf{s} \text{ in } \mathbf{G_f}\}$



- $(\mathbf{S},\bar{\mathbf{S}})$ an $(\mathbf{s},\mathbf{t})$-cut. ✓
- $\mathbf{c_f(a,b) = 0}$
  $\implies \mathbf{c(a,b) - f(a,b) = 0}$
  $\implies \mathbf{c(a,b) = f(a,b)}$
- $\mathbf{c_f(y,x) = 0}$
  $\implies \mathbf{f(x,y) = 0}$

$\mathbf{f}$ saturates $\mathbf{S} \to \bar{\mathbf{S}}$ edges, avoids $\bar{\mathbf{S}} \to \mathbf{S}$ edges $\implies \mathbf{cap}(\mathbf{S},\bar{\mathbf{S}}) = |\mathbf{f}|$ by corollary

Case 2

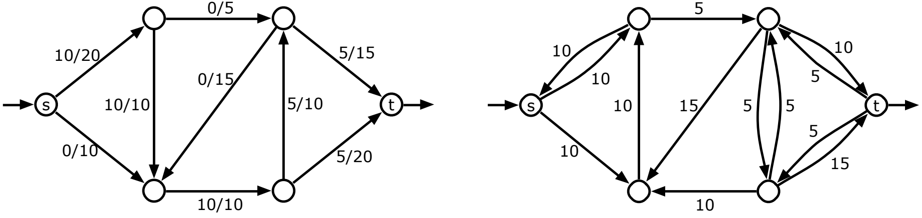
Suppose $\exists$ an $\mathbf{s} \rightarrow \mathbf{t}$ path $\mathbf{P}$ in $\mathbf{G_f}$.

- Called an *augmenting path*

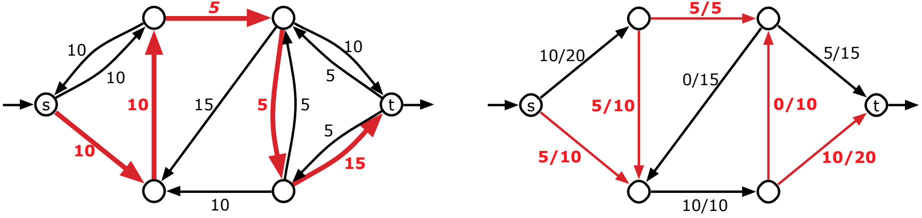
Idea: show that we can "push" more flow along $\mathbf{P}$, so $\mathbf{f}$ not a max flow. Contradiction, can't be in this case.

- Foreshadowing: augmenting path allows us to send more flow. Algorithm to increase flow!

Intuition



A flow f in a weighted graph G and the corresponding residual graph G_f.



An augmenting path in G_f with value $F = 5$ and the augmented flow f'.

Formalities

Let $\mathbf{P}$ be (simple) augmenting path in $\mathbf{G_f}$. Let $\mathbf{F} = \mathbf{min}_{\mathbf{e} \in \mathbf{P}} \mathbf{c_f}(\mathbf{e})$.

Define new flow $\mathbf{f'}$: for all $(\mathbf{u}, \mathbf{v}) \in \mathbf{E}$, let

$$\mathbf{f'}(\mathbf{u},\mathbf{v}) = \begin{cases} \mathbf{f}(\mathbf{u},\mathbf{v}) + \mathbf{F} & \text{if } (\mathbf{u},\mathbf{v}) \text{ in } \mathbf{P} \\ \mathbf{f}(\mathbf{u},\mathbf{v}) - \mathbf{F} & \text{if } (\mathbf{v},\mathbf{u}) \text{ in } \mathbf{P} \\ \mathbf{f}(\mathbf{u},\mathbf{v}) & \text{otherwise} \end{cases}$$

Claim: $\mathbf{f'}$ is a feasible $(\mathbf{s}, \mathbf{t})$-flow with $|\mathbf{f'}| > |\mathbf{f}|$.

Plan: prove (sketch) each subclaim individually

- $|\mathbf{f'}| > |\mathbf{f}|$
- $\mathbf{f'}$ an $(\mathbf{s}, \mathbf{t})$-flow (flow conservation)
- $\mathbf{f'}$ feasible (obeys capacities)

$|\mathbf{f'}| > |\mathbf{f}|$

Consider first edge of $\mathbf{P}$ (out of $\mathbf{s}$), say $(\mathbf{s}, \mathbf{v_1})$

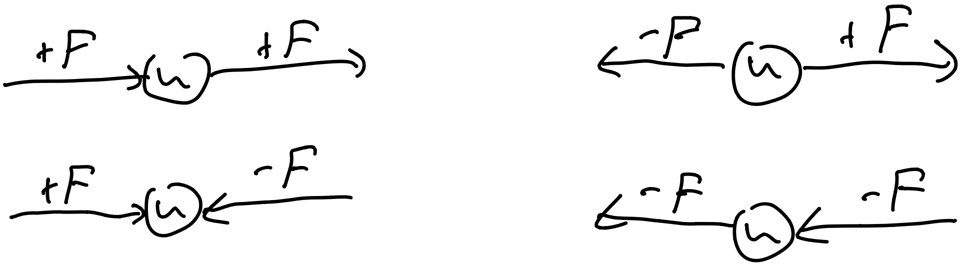
- If $(\mathbf{s}, \mathbf{v_1}) \in \mathbf{E}$, then $\mathbf{f'}(\mathbf{s}, \mathbf{v_1}) = \mathbf{f}(\mathbf{s}, \mathbf{v_1}) + \mathbf{F}$
- If $(\mathbf{v_1}, \mathbf{s}) \in \mathbf{E}$ then $\mathbf{f'}(\mathbf{v_1}, \mathbf{s}) = \mathbf{f}(\mathbf{v_1}, \mathbf{s}) - \mathbf{F}$

$$|\mathbf{f'}| = \sum_{\mathbf{u}} \mathbf{f'}(\mathbf{s}, \mathbf{u}) - \sum_{\mathbf{u}} \mathbf{f'}(\mathbf{u}, \mathbf{s}) = |\mathbf{f}| + \mathbf{F} > |\mathbf{f}|$$

$\mathbf{f}'$ obeys flow conservation

Consider some $\mathbf{u} \in \mathbf{V} \setminus \{\mathbf{s}, \mathbf{t}\}$.

- If $\mathbf{u} \notin \mathbf{P}$, no change in flow at $\mathbf{u} \implies$ still balanced.
- If $\mathbf{u} \in \mathbf{P}$, four possibilities:



f' obeys capacity constraints

Let $(u, v) \in E$

- If $(u, v), (v, u) \notin P$: $f'(u, v) = f(u, v) \le c(u, v)$

- If $(u, v) \in P$:

$$\begin{aligned} f'(u,v) &= f(u,v) + F \\ &\le f(u,v) + c_f(u,v) \\ &= f(u,v) + c(u,v) - f(u,v) \\ &= c(u,v) \end{aligned}$$

- If $(v, u) \in P$:

$$\begin{aligned} f'(u,v) &= f(u,v) - F \\ &\ge f(u,v) - c_f(v,u) \\ &= f(u,v) - f(u,v) \\ &= 0 \end{aligned}$$

Ford-Fulkerson Algorithm and Integrality

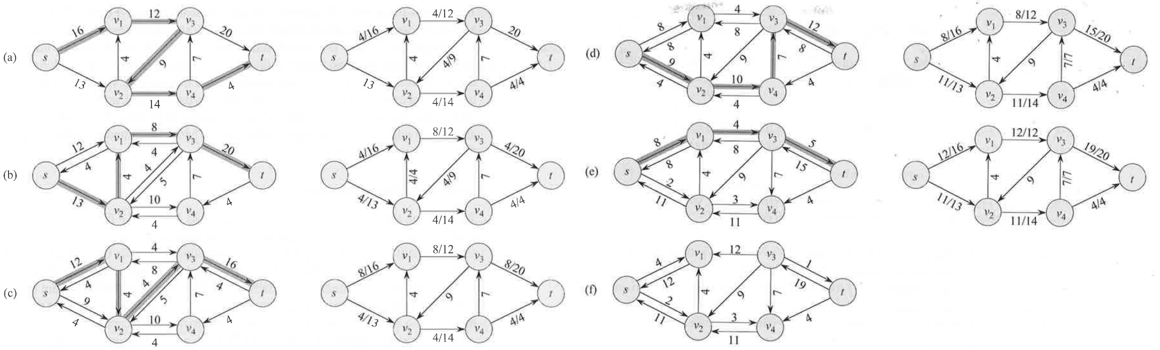
FF Algorithm

Obvious algorithm from previous proof: keep pushing flow!

$\mathbf{f} = \vec{\mathbf{0}}$
while($\exists \mathbf{s} \to \mathbf{t}$ path $\mathbf{P}$ in $\mathbf{G_f}$) {
    $\mathbf{F} = \mathbf{min}_{\mathbf{e} \in \mathbf{P}}\, \mathbf{c_f}(\mathbf{e})$
    Push $\mathbf{F}$ flow along $\mathbf{P}$ to get new flow $\mathbf{f'}$
    $\mathbf{f} = \mathbf{f'}$
}
return $\mathbf{f}$ or $\{\mathbf{v} \in \mathbf{V} : \mathbf{v}$ reachable from $\mathbf{s}$ in $\mathbf{G_f}\}$

Correctness: directly from previous proof

Example



Integrality

Corollary

If all capacities are integers, then there is a max flow such that the flow through every edge is an integer

Proof.

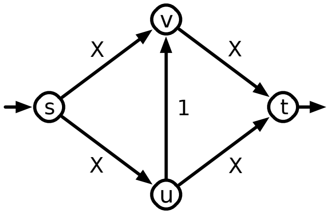
Induction on iterations of the Ford-Fulkerson algorithm: initially true, stays true $\implies$ true at end. $\square$

Running Time

Theorem

If all capacities are integers and the max flow value is $\mathbf{F}$, *Ford-Fulkerson takes time at most* $\mathbf{O(F(m+n))}$

Finding path takes $\mathbf{O(m+n)}$ time, increase flow by at least $\mathbf{1}$



A bad example for the Ford-Fulkerson algorithm.

Running time ≥ # iterations.
This example:

- Running time: $\mathbf{\Omega(x)}$
- Input size $\mathbf{O(\log x) + O(1)}$

$\implies$ Exponential time!