Michael Forbes
Miforbes@illinois.edu

2019-09-24.4 → 2019-10-10.1
2019-10-10.2 ←
cs473

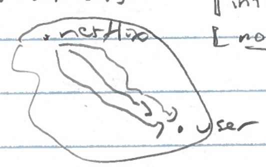
cs473 Algorithms: Lecture 13

logistics: - pset 5 due W10

last time: - algebraic finger-printing [Chandra]
- Frievald's trick
- Schwartz-Zippel lemma / polynomial identity testing

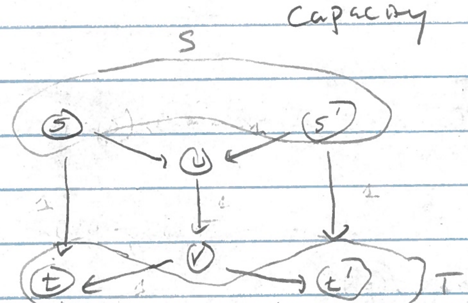
today: - network flows
- def
- basics
- network cuts

ex: [internet] [not true] Q: speed test?



def: network is directed graph $G=(V,E)$
w/ sources $S \subseteq V$
targets/sinks $T \subseteq V$ } $S \cap T = \emptyset$
capacity $c: E \to \mathbb{R}_{\geq 0}$ ← in general
$c: E \to \mathbb{Z}_{\geq 0}$ ← this course [we'll see why]

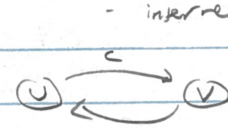
ex:



unit capacities: $c = 1$

[models: - road networks
- water networks
- internet]

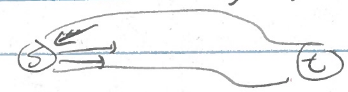
rmk: - bidirected edges possible



- assumptions: - capacities are integral
- no isolated vertices

def: network is single-source/single sink if $S = \{s\}$, $T = \{t\}$

rmk: - single source/sink is essentially the general case [compare to shortest paths problem]
[we will stick to this case]
- sources might have incoming edges

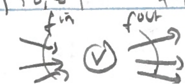


def: network $G=(V,E)$ $s,t \in V$ capacities c

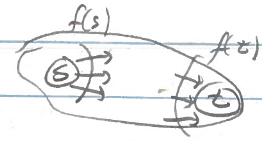
An (s,t) flow is $f: E \to \mathbb{R}_{\geq 0}$, st [allow non integral flow even if c integral]
- capacity constraints: $\forall e \quad 0 \leq f(e) \leq c(e)$ [capacity is respected]
- conservation constraints: [flow only created/absorbed at source/sink]

for $v \in V$
$$f^{in}(v) := \sum_{u \to v} f(u \to v)$$
$$f^{out}(v) := \sum_{v \to w} f(v \to w)$$
$$f(v) := f^{out}(v) - f^{in}(v)$$

then $\forall v \in V \setminus \{s,t\} \quad f(v) = 0$



Michael Forbes
mforbes@illinois.edu
2019-10-10.2 ⇆ 2019-10-10.2 → 2019-10-10.3
CS 473



The <u>value of f</u> is $|f| = f(s)$

<u>lem</u>: $f(t) = -f(s) \Rightarrow |f| = |f(t)|$

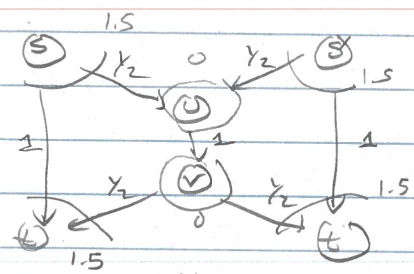
<u>pf</u>: $f(s) + f(t) = f(s) + f(t) + \sum_{v \neq s,t} f(v) = \sum_v \underbrace{f(v)}_{f^{out} - f^{in}}$

$= \sum_v \left[\sum_{v \to w} f(v \to w) - \sum_{u \to v} f(u \to v) \right]$

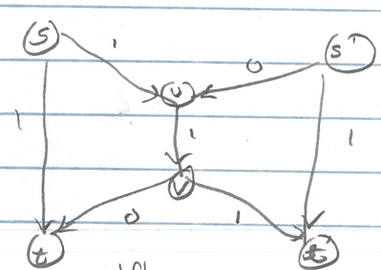
$= \sum_{e = u \to v} \left[f(e) - f(e) \right] = 0$ ∎

<u>ex</u>:

$c = 1$



[alternate view] $|f| = 3$



$|f| = 3$ [integral flow!]

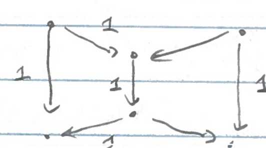
<u>def</u>. network $G = (V, E)$, $s, t \in V$, capacities c.

$P = \{ s \rightsquigarrow t$ paths in $G \}$ [exponentially large, in general]
(paths ← simple walks)

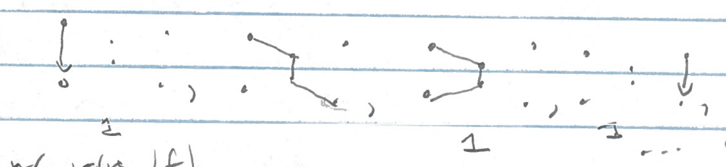
A (s,t) <u>flow</u> is $f : P \to \mathbb{R}_{\geq 0}$ st - capacity constraint $\forall e \in E$ $\sum_{p : p \ni e} f(p) \leq c(e)$

The <u>value</u> is $|f| := \sum_p f(p)$.
[conservation is implicit]

<u>ex</u>:



$P = \{$



$\ldots \}$

<u>lem</u>: $f : P \to \mathbb{R}_{\geq 0}$ a path flow w/ value $|f|$

$\Rightarrow$ exists $g : E \to \mathbb{R}_{\geq 0}$ an edge flow w/ value $|g| = |f|$.

<u>Pf</u>: define $\forall e \in E$ $g(e) = \underbrace{\sum_{p \ni e} \underbrace{f(p)}_{\geq 0}}_{\geq 0} \leq c(e) \Rightarrow g$ satisfies capacity constraints
(≤ c(e) ← def of path flow)

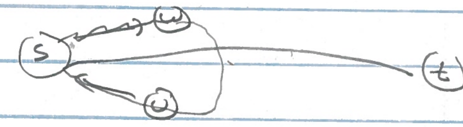
for $v \neq s, t$ $g(v) = g^{out}(v) - g^{in}(v) = \sum_{v \to w} \underbrace{g(v \to w)}_{= \sum_{p \ni v \to w} f(p)} - \sum_{u \to v} \overbrace{g(u \to v)}^{\sum_{p \ni u \to v} f(p)}$

$= \sum_{\substack{p \text{ s-t path} \\ \text{through } v}} f(p) \cdot (1 - 1) = 0$

$\Rightarrow$ conservation constraints

$g(s) = \sum_{s \to w} \underbrace{g(e)}_{} - \sum_{u \to v} \underbrace{g(e)}_{}$

$= \sum_{\substack{p \ni e \\ \text{s-t path}}} f(p)$ $\quad = \sum_{\substack{p \ni e \\ \text{s-t path}}} f(e)$ ← $\emptyset$, $= 0$



$= \sum_{s \to w} \sum_{p \ni (s \to w)} f(p)$

$= \sum_p f(p) = |f|$. $\Rightarrow$ values are the same ∎

2019-10-10.2 →
2019-10-10.4 ←
Michael Forbes
miforbes@illinois.edu
2019-10-10.3
CS473

rmk: this is "algorithmic", bottleneck is $|\{p : f(p) > 0\}|$

prop: $f : E \to \mathbb{R}_{\geq 0}$ edge flow. Then there exists path flow $g : P \to \mathbb{R}_{\geq 0}$ w/

- $|f| = |g|$
- $|\{p : g(p) > 0\}| \leq m = |E|$
- g computable in $O(m(m+n))$ time

pf idea: induct on $|\{e : f(e) > 0\}| \leq m$

algo: "on input: edge flow $f : E \to \mathbb{R}_{\geq 0}$

- initialize $g : P \to \mathbb{R}_{\geq 0}$ to be zero.
- while $\exists$ $s \rightsquigarrow t$ path p in (V, F), $F = \{e : f(e) > 0\}$
  - define $g(p) = \min_{e \in p} f(e) > 0$
  - $\forall e \in p$ $\quad f(e) \leftarrow f(e) - g(p)$

correctness/complexity:

Clm: loop invariants
- f valid edge flow
  - $0 \leq f(e) \leq c(e)$ [capacity]
  - conservation $f(v) \to f(v) + (-g(p)) - (-g(p))$
- $f + g = f_0$ ← original edge flow

edge flow ↗ f; path flow ⇒ edge flow ↖ g

$$\underbrace{f(e) + \sum_{p \ni e} g(p)}_{\leq c(e)} \mapsto \left(f(e) - g(p')\right) + \left(\sum_{p \ni e} g(p) + g(p')\right)$$

$$\Rightarrow \sum_{p \ni e} g(p) \leq c(e) \Rightarrow g \text{ valid path flow}$$

Cor: at end $|g| = |f_0|$ ← original flow

Pf: Prop: $|f| > 0 \Rightarrow s \rightsquigarrow t$ path in $F = \{e : f(e) > 0\}$

Pf: later today (*)

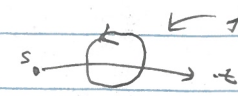
$\Rightarrow$ termination $\equiv$ no path in $F \Rightarrow |f| = 0 \Rightarrow |g| = \underbrace{|f_0|}_{= g + f}$ ☒

Clm: each loop iteration
- ≤ 1 new path added to g
- ≥ 1 edge removed from F

Cor: $\leq m$ iterations $\Rightarrow$
- $\leq m$ paths in g
- $O(\underbrace{m}_{\#\text{iterations}} \underbrace{(m+n)}_{\text{finding } s\text{-}t \text{ path in } F})$ runtime ☒

[exp size] [conservation is free] [compact]

rmk:
- path flow $\equiv$ edge flow, each is useful
- at end we may not have $|F| = 0$, eg



← flow cycle

Q (max flow): given network $G = (V, E)$ $s, t \in V$ capacities c [path/edge?] what is the maximum value of any flow?

=

[continuous space of feasible solutions]
[no a priori brute force]

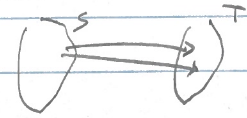
Michael Forbes
miforbes@illinois.edu
2019-10-10.4 ← 2019-10-10.3
→ 2019-10-15.1
cs473

def: network $G=(V,E)$, $s \neq t \in V$, capacities c

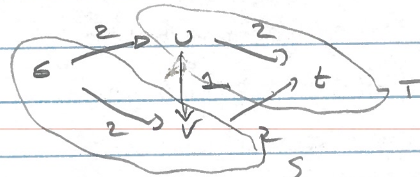
An (s,t) cut is a partition $V = S \sqcup T$ w/ $s \in S$, $t \in T$

the capacity of the cut $c(S,T) = \sum_{u \in S, v \in T} c(u \to v)$

rmk: edges $T \to S$ not included



ex:



$c(S,T) = 2+2 = 4$

Q (min-cut): for network what is the minimum capacity of any cut?
[only exp-many cuts so can brute force, but not efficient]

lem: network $G=(V,E)$ $s,t \in V$ capacity c
(s,t)-flow f, $S \sqcup T$ with $s \in S$, $t \in T$

then $|f| = \underbrace{\sum_{v \in S, w \in T} \underbrace{f(v \to w)}_{\le c(v\to w)}}_{\le c(S,T)} - \sum_{v \in S, u \in T} f(u \to v)$

pf: $|f| = f(s) = f(s) + \sum_{v \in S \setminus \{s\}} \overbrace{f(v)}^{=0} = \sum_{v \in S} f^{out}(v) - f^{in}(v)$

$= \sum_{v \in S} \left[\sum_{v \to w} f(v \to w) - \sum_{u \to v} f(u \to v) \right]$

$= \sum_{v \in S\, w \in T} f(v \to w) + \cancel{\sum_{v \in S\, w \in S} f(v \to w)}$

$- \cancel{\sum_{v \in S\, u \in S} f(u \to v)} - \sum_{v \in S, u \in T} f(u \to v)$ ☒

Cor: f (s,t) flow, C s,t cut $\Rightarrow |f| \le |C|$
$\Rightarrow$ max-flow $\le$ min-cut

Thm: $=$ [integrality of capacities] [gives algo a cert of optimality]

Cor: f (s,t) flow w/ $|f| > 0$, $F = \{e : f(e) > 0\}$
$\Rightarrow G = (V,F)$ has $s \rightsquigarrow t$ path

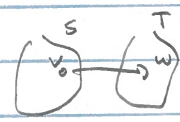
Pf: by contrapositive. If no $s \rightsquigarrow t$ path $S := \{v : s \rightsquigarrow v \text{ in } G=(V,F)\}$

so $s \in S$, $t \notin S \Rightarrow |f| = \sum_{v \in S, w \in T} \underbrace{f(v \to w)}_{=0} - \underbrace{\sum_{v \in S\, u \in T} f(u \to v)}_{\ge 0}$

$T := V \setminus S$

$\Rightarrow \underset{\substack{\vee\!\mid \\ 0}}{|f|} \le 0 \Rightarrow |f| = 0$

as



no $s \rightsquigarrow w$ path in $F = \{e : f(e) > 0\}$
but $s \rightsquigarrow v$ path in F
$\Rightarrow f(v \to w) = 0$

logistics: psets due W10
today: - flows
- cuts

next time: - max flow algorithms
- max flow = min cut