Michael Forbes
Miforbes@illinois.edu

2019-10-15.4 → 2019-10-17.1
2019-10-17.1 ←
CS 473

CS 473 Algorithms: Lecture 15

logistics: - pset 6 due W10
- midterm 1 grades out

last time: Ford Fulkerson
- efficient flow algo

today:

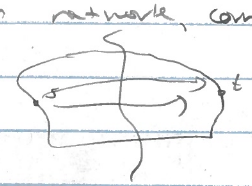
def: network $G=(V,E)$ $s \neq t \in V$ capacities $c: E \to \mathbb{Z}_{\geq 0}$ [integral]

$(s,t)$-flow $f: E \to \mathbb{R}_{\geq 0}$ - capacity constraints
- conservation constraints

value $|f| = f^{out}(s) - f^{in}(s)$

$(s,t)$-cut $V = \underset{s}{\overset{}{S}} \sqcup \underset{t}{T}$ capacity $c(S,T) = \sum_{u \in S, v \in T} c(u \to v)$

Q: given network, compute - max flow
- min cut



idea: increase flow greedily allowing flow to reverse

def: network $G$, capacities $c$, flow $f$ in $G$

The residual network $G_f = (V, E_f)$ with residual capacities $c_f$, where

- forward edges [excess capacity]: $(u,v) \in E$ if $f(u \to v) < c(u \to v)$ $c_f(u \to v) = c(u \to v) - f(u \to v) > 0$
- backward edges [reverse flow]: $(u,v) \in E$ if $f(u \to v) > 0$: $c_f(v \to u) = f(u \to v)$

ex: $u \xrightarrow{3/5} v \longmapsto u \underset{3}{\overset{2}{\rightleftarrows}} v$

rmk: $G_f$ has $\leq 2m$ edges

def: network $G$ flow $f$. An augmenting path is a simple $s \to t$ path $p$ in $G_f$, has capacity $|p| = \min_{e \in p} c(e)$

lem: $p$ augmenting path in $G_f \Rightarrow f+p$ flow in $G$

[conservation] [capacity] [value] w/ value $|f+p| = |f| + |p| > |f|$

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

```
f(e) ← 0   ∀e ∈ E   [initialize]
initialize G_f
while ∃ augmenting path p in G_f
    f ← f + p
    G_f ← G_{f+p}
return f
```

Michael Forbes
miforbes@illinois.edu
2019-10-17.2 ⇆ 2019-10-17.1
→ 2019-10-17.3
CS473

prop: loop invariants: – flow $f$ integral [[nontrivial fact]] ] uses that $c$ integral
– residual capacities integral

complexity: – each iteration takes $O(m+n)$ time [[graph reachability]]
– $\leq |f^*| \leq \underbrace{\sum_{e\in E} c(e)}_{=:C}$ many iterations [[each iteration increases flow value by 1]]
↑ max flow value ↑ uses integrality of $c$
– $O(mC)$ time

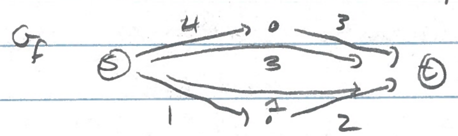
correctness: termination ⇒ no $s \rightsquigarrow t$ path in $G_f$ [[augmenting path]]
⇒ cut w/ $|c(S,T)| = |f|$
⇒ $f$ is max flow, and max flow = min cut

rmk: – input size to problem is # bits to describe – $G=(V,E)$ $O(m \lg n)$
but for an algo to be efficient in runs – $s \neq t \in V$ $O(\lg n)$
in polynomial ~~time~~ in input size – $c: E \to \mathbb{Z}_{\geq 0}$ $O(m \lg n + m \lg C)$ ← capacities in binary
⇒ Ford Fulkerson not efficient if $C$ large
eg: $2^{100}$ is 101 bits in length ← small, easy to do arithmetic
but FF w/ $C = 2^{100}$ takes $\underbrace{\geq 2^{100} \text{ steps}}$ in worst case [[can happen]]
↖ large

– if $c: E \to \mathbb{Z}_{\geq 0}$ has capacities expressed in unary, FF is efficient
↳ size $O(mC)$ ↳ pseudo polynomial time

Q: better algo?
idea: FF allows any augmenting path, choose "good" ones
def: network $G$ flow $f$.
An maximum capacity augmenting path is a (simple) $s$-$t$ path in $G_f$ w/ maximum capacity $|p| = \min_{e\in p} c(e)$.

ex: $G_f$




Prop: can find a maximum capacity augmenting path in $O(m \lg C)$ steps [[sum of capacities]]
Pf. idea: binary search
let $G_f^{\geq K}$ be $G_f$ where $\overbrace{\text{edges in } G_f \text{ with capacity} < K \text{ are dropped}}^{\text{cannot be used in capacity } K \text{ path}}$
algo: find $\max_{0\leq K\leq C} \{ G_f^{\geq K} \underbrace{\text{has } s \rightsquigarrow t \text{ path}} \}$
$O(m)$ steps [[no isolated vertices]]
$O(\lg C)$ steps in binary search
return path in $G_f^{\geq K_{max}}$ with capacity $K_{max}$
→ correctness: if $G_f^{\geq K}$ has $s \rightsquigarrow t$ path then so does $G_f^{\geq K'}$ for $K' \leq K$ ⇒ 0 —— $K_{max}$ —— $C$ [[monotonic]]
has $s \rightsquigarrow t$ path / has no $s \rightsquigarrow t$ path

Michael Forbes
mforbes@illinois.edu


algo (Ford Fulkerson w/ max capacity augmenting path)
  network $G=(V,E)$ $s\neq t\in V$ capacities $c$
  initialize – $f(e)\leftarrow 0$ $\forall e\in E$
    – $G_f$
  while $s\rightsquigarrow t$ path in $G_f$ $\leftarrow O(m)$
    find maximum capacity augmenting path $p$ in $G_f$ $\leftarrow O(m \lg C)$
    $f\leftarrow f+p$ $O(n)$
    $G_f \leftarrow G_{f+p}$ $O(n)$

Prop: algorithm is correct
  algo terminates, takes $O(\overbrace{I}^{\#\text{iterations}}\cdot m\lg C)$ steps [what is it?]

Prop: $f, f'$ flows in $G \Rightarrow$ exists flow of value $|f'|-|f|$ in $G_f$

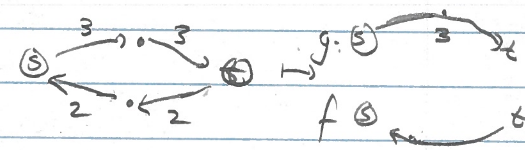
Pf: pset 6

Cor: $f$ flow in $G \Rightarrow$ exists flow of value $|f^*|-|f|$ in $G_f$ ← max flow in $G$

Prop (edge flow $\mapsto$ path flow) $f: E\to \mathbb{R}_{\geq 0}$ edge flow. Then there is a path flow $g:\{\text{all } s\rightsquigarrow t \text{ paths}\}\to\mathbb{R}_{\geq 0}$ w/
 – $|g|\geq |f|$ [previously claimed '=', but '≥' is what proved]
 – $|\{p : g(p)>0\}|\leq m$
 – $g$ computable in $O(m^2)$ time

Sketch: while $|f|>0$
  find $t\rightsquigarrow s$ path $p$ in $G_f$ ← reverse of FF
  $\equiv$ $s\rightsquigarrow t$ path in $G$ w/ $p\leq f$
  $f\leftarrow f-p$
  add $p$ to $g$

rmk: at end $|f|<0$ is possible, eg




Cor: network $G$ flow $f$. let $K_{max} = \max_K \{ s\rightsquigarrow t \text{ path in } G_f^{\geq K}\}$ [max capacity of any augmenting path]
 Then $K_{max} \geq \frac{|f^*|-|f|}{m}$

Pf: $G_f$ has $s\rightsquigarrow t$ edge flow of value $|f^*|-|f|\geq 0$
 $\Rightarrow$ $G_f$ has path flow of value $\geq$ (only $\leq m$ paths) $\sum g(p)$
 $\Rightarrow$ some path in path flow has $\geq \frac{|f^*|-|f|}{m}$ value
 $\Rightarrow$ all edges in have capacity $\geq$
 $\Rightarrow$ $G_f^{\geq \frac{|f^*|-|f|}{m}}$ has $s\rightsquigarrow t$ path
 $\Rightarrow$ $K_{max}\geq \frac{|f^*|-|f|}{m}$ $\square$

Michael Forbes
Miforbes@illinois.edu
2019-10-17.4 ← 2019-10-17.3
→ 2019-10-22.1
cs473

network $G$ capacities $c: E \to \mathbb{Z}_{\geq 0}$ $C = \sum_{e \in E} c(e)$

Cor: maximum capacity Ford Fulkerson takes $O(m \lg C)$ iterations

Pf: let $\underset{\underset{0}{||}}{f_0}, f_1, \ldots, f_i, \ldots$ be flows of algo, all are integral

then $f_i = f_{i-1} + p_i$ ← ith augmenting path $\quad |f_i| = |f_{i-1}| + \underbrace{|p_i|}_{\geq \frac{|f^*| - |f_{i-1}|}{m}}$

[progress measure]

$$|f^*| - |f_i| = |f^*| - |f_{i-1}| - |p_i|$$
$$\leq \left(1 - \tfrac{1}{m}\right)\left(|f^*| - |f_{i-1}|\right)$$
$$\leq \underbrace{\left(1 - \tfrac{1}{m}\right)^i}_{\leq e^{-i/m}} \Big( \underbrace{|f^*|}_{\leq C} - \underbrace{|f_0|}_{=0} \Big)$$
$$\leq C \cdot e^{-i/m}$$
$$< 1 \quad \text{if } i > m \ln C$$

is integral by max flow = min cut and integral capacities

$|f^*|$ and $|f_i|$

⟹ $|f_i| = |f^*|$ as flow values are integral

⟹ termination. ∎

Cor: max flow in $O(m^2 \lg^2 C)$ time

rmk: - can run this algorithm on real-valued capacities - may not terminate - will converge in value

- this is actually poly time
- Ford Fulkerson w/ shortest length augmenting paths $O(nm^2)$ time (# edges)
  - runtime independent of $C$ [assuming $O(1)$ cost arithmetic]
  "strongly polynomial time"
  - terminates for all capacities
  - proves max flow = min cut in general
- ... can achieve $O(nm)$ time max flow
  $O(nm)$ time edge → path flow [only $m^2$ in lecture]

summary:

| | | | |
|---|---|---|---|
| Ford Fulkerson | $O(mC)$ | pseudo polynomial time | |
| Ford Fulkerson w/ max capacity | $O(m^2 \lg^2 C)$ | (weakly) poly time | Edmonds Karp |
| FF w/ shortest augment path | $O(nm^2)$ | strongly poly time | Edmonds Karp/Dinitz |
| | $O(nm)$ | Orlin 2013 | |

logistics: pset6 due W10
today: efficient flow algo
next time: applications of flow