Michael A. Forbes
MForbes@illinois.edu
2024-02-05.1

CS473 Algorithms: Lecture 7 (2024-02-06)

lec 6

logistics - - pset2 due F17

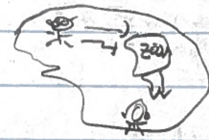
last lecture: dynamic programming
Bellman-Ford - shortest paths w/o negative cycles
- negative cycle detection

lec 6

today: flows

reading: KT 7.0, 7.1

Q: [pandemic] zoom meetings in [high definition]? [internet bandwidth]



[me, glasses]
[zoom]
[you, headphones]

Q: how much [flow] through network? [automobile traffic] [water networks]

def: directed graph $G=(V,E)$ is **simple** if
- no isolated vertices
- no loops
- no parallel edges

[wlog, for simplicity] | simple

rmk: [allow] anti parallel edges

def: **capacitated graph** is simple directed graph $G=(V,E)$ w/ edge capacities $(c_e)_{e\in E}$, $c_e \in \mathbb{N}$ [non-neg integers] $\neq 0$, $c_e \geq 0$

for s [source], t [sink] $\in V$, **(s,t)-flow over G** is $f=(f_e)_{e\in E}$, $f_e \in \mathbb{R}_{\geq 0}$ [non-neg reals] [models continuous quantities, eg water]

st: - **capacity constraints**: $0 \leq f_e \leq c_e$, $\forall e$ [no overflow]
- **conservation constraints**: $f^{in}(v) = \sum_{e:\cdot\to v} f_e$ [in flow]

$$f^{out}(v) = \sum_{e: v\to\cdot} f_e \quad \text{[out flow]}$$

notation

$$f(v) = f^{out}(v) - f^{in}(v) \quad \text{[net flow]}$$

$$\forall v \in V\setminus\{s,t\} \quad f(v)=0 \quad \text{[conservation]}$$

value of (s,t)-flow is $|f| = f(s)$ [net flow at source] [intuitively = flow $s \rightsquigarrow t$]

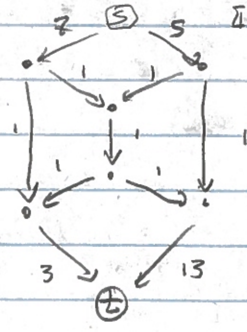
justify

max (s,t)-flow problem is to compute $\widehat{\arg}\max_{f\ (s,t)\text{ flow}} |f|$ [omit s,t when clear]

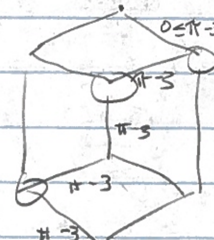
Q: compute max flow?

Q: — " — via brute force? [[not] clear: flow values are continuous, [cannot] enumerate]

eg: [capacities]

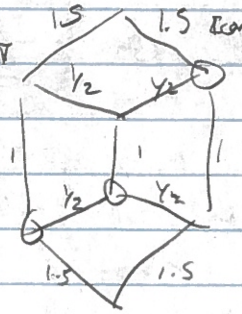


[flows]
[irrational]
$0 \leq \pi - 3 \leq 1$ [capacity,
[conservation]



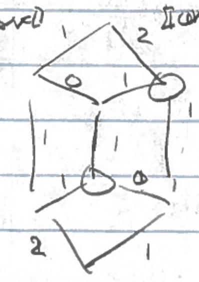
$|f| = \pi - 3$
f irrational

[conserved]



$|f| = 3$
f rational

[conserved]



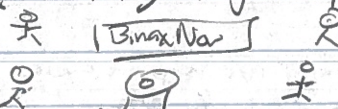
$|f| = 3$
f integral [!]

Michael A. Forbes
miforbes@illinois.edu
2024-02-05.2

review

metafact - integrality of optimal solutions is significant issue

eg: [BinaryNav]



[one could test]
[two roommates; both need test for travel]
[no splitting! ⟹ issue]
[can split toilet roll ⟹ less of a problem]

eg

[other max flow may be non integral]

[next lecture]
thm: [some] max flow is integral

cor: max flow is computable

sketch: finitely many $(f_e)_{e\in E}$ over $\boxed{\mathbb{N}}$ satisfying capacity constraint
↳ try all - test conservation
- maximize $|f|$

□

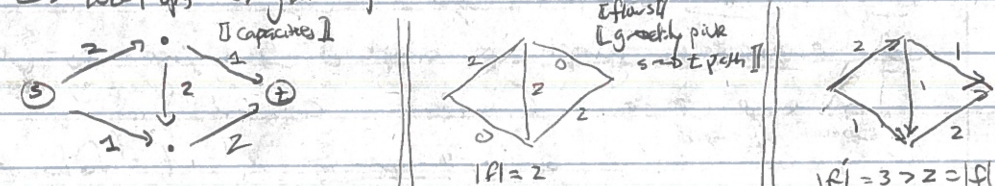
Q: compute max flow [efficiently]?

notation [inside] **idea**: dynamic programming ‖? [no obvious algo?]

idea: greedy algs?
↳ local opt ⟹ global opt

water

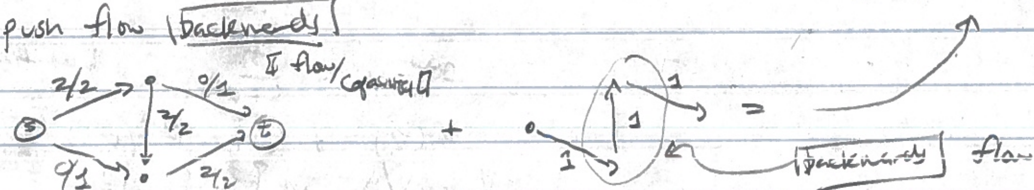
eg -



idea: push flow along s⇝t path

[local] opt = no s⇝t path w/ unused capacity ⟹ [not] global opt
⟹ greedy [fails]

idea: push flow [backwards]



[formalize]
def: f (s,t)-flow over capacitated $G=(V,E)$. $G^f=(V^f,E^f)$ is capacitated **residual graph** w/ - $V^f = V$

- $E^f = \{e : f_e < c_e, e\in E\}$ ← forward edges

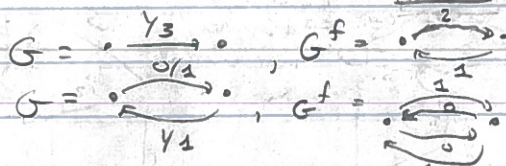
f_e $\cup\{\underline{-e} : 0 < f_e, e\in E\}$ ← backward edges

↖ reverse orientation

- residual capacities: $e\in E^f$ $(c^f)_e = \begin{cases} c_e - f_e \boxed{>0}, & e \text{ forward} \\ f_e \boxed{>0}, & e \text{ backward}\end{cases}$ [unused capacity] [flow to reverse]

0 capacity edges don't count

eg:



parallel edge [≤ 2 parallel edge]

simple

[capacitated]
def: f (s,t)-flow over G, G^f residual. An *augmenting path* p is [simple] [no repeat vertices]

\>0 $s⇝t$ path in G^f. The value is $|p| = \min_{e\in p} c^f_e$ [bottleneck] >0

residual

G_f parallel

f_e vs f_e

Augmenting f by p yields $f+p$, $(f+p)_e = \begin{cases} f_e + |p|, & e\in p \text{ [e forward]} \\ f_e - |p|, & -e\in p \text{ [e backward]} \\ f_e & \text{else [p avoids e]}\end{cases}$

Michael A. Forbes
MForbes@illinois.edu
2024-02-05.3

prop : $f \pm p$ valid flow, value $|f| \pm |p|$ — easy $\Rightarrow$ prop

pf: capacity : $e \in E$ — $\underline{e, e \notin p}$: $0 \leq f_e \leq c_e$ [f flow]

$\overset{\shortparallel}{(f \pm p)_e}$ [p does not touch e]

[e forward in G^f]

$\underline{e \in p}$: $(c^f)_e = c_e - f_e \geq \min_{e' \in p} c^f_{e'} = |p|$

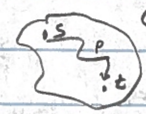
$(f \pm p)_e = \underbrace{f_e}_{\geq 0} + \underbrace{\overbrace{|p|}^{\leq c_e - f_e}}_{\geq 0} \leq c_e$, ≥ 0

[backwards]

$\underline{\overleftarrow{e} \in p}$: $(c^f)_{\overleftarrow{e}} = f_e \geq \min_{e' \in p} (c^f_{e'}) = |p|$

$(f \pm p)_e = \overset{\geq 0}{f_e} - \underbrace{\overset{\geq 0}{|p|}}_{\leq f_e} \leq f_e \leq c_e$, ≥ 0

conservation : sketch. — sketch



for $v \neq s, t$, p visiting v

4 cases of orientation

- $\xrightarrow{f_e} \cdot \xrightarrow{f_{e'}}$ [2 forward] $\longmapsto$ $\xrightarrow{f_e + |p|} \cdot \xrightarrow{f_{e'} + |p|}$ $\Rightarrow$ conserved
- $\xrightarrow{f_e} \cdot \xleftarrow{f_{e'}}$ [forward, backward] $\longmapsto$ $\xrightarrow{f_e + |p|} \xleftarrow{f_{e'} - |p|}$ $\Rightarrow$ conserved
- = ″
- = ″

[$\Rightarrow$ $f \pm p$ valid flow]

value : sketch : $p : s \rightsquigarrow t$ path — sketch

p simple $\Rightarrow$ visits s once — verify simple

e forward: $(s) \xrightarrow{f_e}$ $\longmapsto$ $(s) \xrightarrow{f_e + |p|}$ — else simple ↳ 2022 M.Forbes...

e backward : $(s) \xleftarrow{f_e}$ $\longmapsto$ $(s) \xleftarrow{f_e - |p|}$

$|f|$ $\longmapsto$ $|f| + |p|$

idea : repeated augment flow — greedy

algo (Ford Fulkerson)

- $f_e \leftarrow 0$, $e \in E$ ← valid flow
- initialize G^f
- while exists augmenting path p in G^f
  - $f \leftarrow f \pm p$
  - $G^f \leftarrow G^{f \pm p}$
- return f

Q : complexity? [termination?] correctness? [return clear]

Michael A. Forbes
mforbes@illinois.edu
2024-02-05.4

no isolated vertices

prop: each iteration takes $O(m+n) = O(m)$ time

sketch: clm: constructing G^f from f takes $O(m)$ time [apply edge updates in $O(1)$ time]

sketch: iterate edges via adjacency list

clm: finding p takes $O(m+n)$ time

idea: depth first search

clm: augmenting $f \mapsto f+p$ takes $O(m)$ time

sketch: iterate ~~over~~ edges in p

compute $\min_{e \in p} (c^f)_e = |p|$ ☒

sketch

vs sketch

Q: # iterations?

remember [wlog] omit s,t

prop: flow f in G, residual G^f

$\boxed{\text{if}}$ capacities $(c_e)_e$ integral [yes, per convention]

flow (f_e) integral [induction]

then G^f has integral capacities [equations preserve integrality]

$|p|$ integral, for p augmenting path

$f+p$ integral

$(c^f_e) = \begin{cases} c_e - f_e & \text{—''—} \\ f_e & \text{—''—} \end{cases}$

$|p| = \min_{e \in p} c_e$

$(f+p)_e = \begin{cases} f_e + |p| & \text{—''—} \\ f_e - |p| & \text{—''—} \\ f_e & \text{—''—} \end{cases}$

sketch: ☐

$(c^f)_e$

$(c^f)_e$

f_e vs f_e

[induction] cor: $\boxed{\text{if}}$ integral capacities, in FF, flows and residual capacities are integral

prop: —''— , $|f+p| = |f| + \underbrace{|p|}_{>0} \geq |f| + 1$ integrality

[trivial ub] prop: $\boxed{\text{any}}$ flow f in G has $|f| \leq \sum_{e \in E} c_e$

pf: $|f| \overset{\text{''}}{=} f^{out}(s) - f^{in}(s)$

$\leq f^{out}(s) = \sum_{e: s \to \cdot} \underbrace{f_e}_{\leq c_e} \leq$ [$c_e \geq 0$]

Cor: integral capacities, FF has $\leq \sum_e c_e = C$ iterations

sketch: start w/ $|f| = 0$, increase by ≥ 1 each iteration, $|f| \leq C$ always

Cor: —''— FF runs in $O(mC)$ time.

Q: correctness?

today: flows
- def
- residual graph
- augmenting paths
- integrality
- FF
  - algo
  - complexity

reading: KT 7.0, 7.1

next lecture: flow

logistics: pset 2 due Fri