Lecture 11 Network Flows I: Augmenting Paths, Max-Flow Min-Cut Theorem	L11.1 6.046 03/31/2020

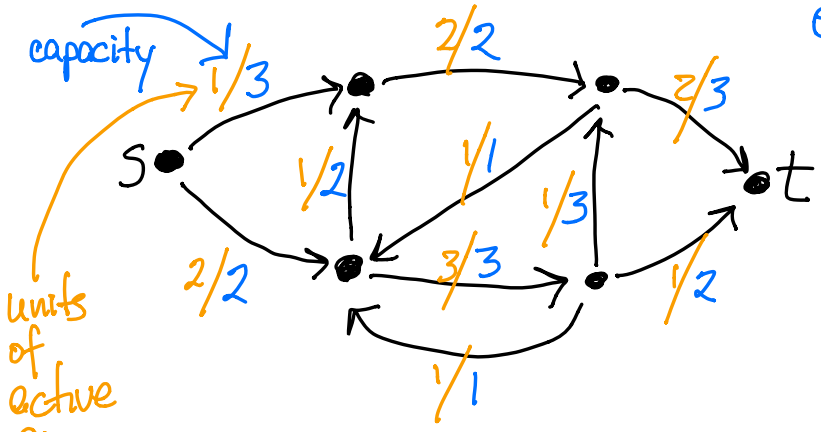
Today:
- Network Flow
- Flow decomposition
- s-t cuts
- residual graph
- augmenting paths
- max-flow min-cut theorem
- Ford-Fulkerson algorithm

Reading:
- CLRS 26.1, 26.2

Maximum Flow Problem: Central problem in graph th.
- common in logistics, routing, connectivity analysis, infrastructure development, etc.
- surprisingly versatile: many other problems can be reduced to it.

Setup: Network = Directed graph $G = (V, E)$
with source vertex $s \in V$
sink vertex $t \in V$
and edge capacities $c: E \to \mathbb{R}^{\geq 0}$
define $c(u,v) = 0$ if $(u,v) \notin E$

Intuition:
- Each edge is one-way road
- Capacity = number of lanes
- Question: what is the maximum rate traffic can flow from s to t, expressed as a number of lanes?



Is this a valid flow?
- No edge capacity should be exceeded
- All vertices other than s & t need to conserve flow (no buildup or depletion)

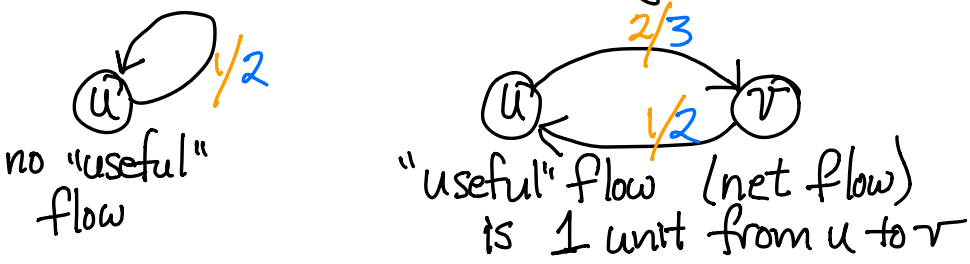
Note: Total flow value is total flow leaving source or entering sink (same for valid flow)
[Here it is 3]

More formally: "gross" flow: $g: E \to \mathbb{R}^{\geq 0}$

$g(u,v)$ = amount of flow on edge (u,v)

$\Rightarrow$ want
- $0 \leq g(u,v) \leq c(u,v) \quad \forall (u,v) \in E$
- $\sum_u g(u,v) - g(v,u) = 0 \quad \forall v \neq s,t$

But: This definition is unwieldy because admits flow cycles of length 1 and 2:



no "useful" flow

"useful" flow (net flow) is 1 unit from u to v

SO WE WILL ADOPT A MORE USEFUL DEFINITION

Net flow: $f: V \times V \to \mathbb{R}$ ← can be negative

With (defined for any u & v):

- $f(u,v) \le c(u,v) \quad \forall u,v \in V$ feasibility
- $\sum_u f(u,v) = 0 \quad \forall v \neq s,t$ flow conservation
- $f(u,v) = -f(v,u) \quad \forall u,v \in V$ skew symmetry

Skew symmetry implies

- $f(u,u) = 0 \quad \forall u$ ← eliminates cycles of length 1
- $f(u,v)$ = net flow from u to v
  ⇒ if $f(u,v) < 0$, flow of $|f(u,v)|$ from v to u

eliminates flow cycles of length 2

Note: $f(u,v) = 0$ if $(u,v) \notin E$ and $(v,u) \notin E$

Definition: Value of flow: $|f| = \sum_v f(s,v)$

Maximum flow problem:

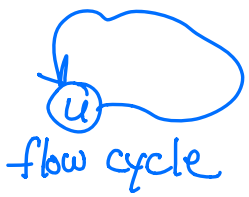
Given a network $G = (V, E, s, t, c)$

find a flow in G of maximum value

Some legitimate flows:

1. Trivially, $f = 0$ is an allowed flow in any network.
2. Any cycle that doesn't exceed the capacity of any edge is a valid flow, although its value is zero.
3. If there is a path $s \rightsquigarrow t$, then any flow along that path of value less then its least capacity is valid.

Useful construct: Any flow can be decomposed into a collection of s-t paths and flow cycles



Formally: Let $\text{supp}_f(G)$ = subgraph of G of edges (u,v) with $f(u,v) > 0$ (that is, a graph of the gross flow of f)

Flow Decomposition Lemma: For any flow f with $|f| \geq 0$, $\text{supp}_f(G)$ can be decomposed into a collection of s-t paths and flow cycles.

Proof: By induction on # of edges in $\text{supp}_f(G)$

(1) $\text{supp}_f(G)$ has no edges $\Rightarrow$ claim trivially holds

(2) $\text{supp}_f(G)$ has some edges

(*) Crucial observation: By flow conservation, if $(u,v) \in \text{supp}_f(G)$ & $v \neq t$, then $(v,u') \in \text{supp}_f(G)$ for some u'.

Case (a): $(s,v) \in \text{supp}_f(G)$ for some v. Use (*), possibly many times, to find an s-t path or an s-s cycle P in $\text{supp}_f(G)$. Remove P from $\text{supp}_f(G)$ by reducing each edge $e \in P$ by bottleneck capacity $c_P(f) = \min_{e \in P} f(e)$. This removes at least one edge from $\text{supp}_f(G)$ and keeps $|f| \geq 0$. Use inductive claim on what remains.

Case (b): $(v,t) \in \text{supp}_f(G)$ for some v. Follow this edge backwards, analogously to case (a), remove the resulting s-t path or cycle, and use inductive claim on what remains. (This case not strictly needed, because it can be shown that if no

edge leaves s in $\text{supp}_f(G)$, then no edge enters t is $\text{supp}_f(G)$. Prove this as an exercise. (Would still need to deal with t-t cycles.)

Case (c): All edges (u,v) in $\text{supp}_f(G)$ have $u \neq s$ and $v \neq t$. $|f| = 0$ (because $|f| \geq 0$), so there are no edges adjacent to s or t. Use (*), possibly many times, on u to find a u-u cycle, proceed as above to remove it and use inductive claim. ∎

Exercise 1: Show WLOG flow decomposition leads to
$\#\ s$-t paths $+ \#$ flow cycles $\leq \#$ edges of non-zero capacity

Exercise 2: If $|f| > 0$ then there has to be ≥ 1 $s \rightsquigarrow t$ flow path (can't have only cycles).

Now let's work towards how we would find a maximum flow.

Let f^* = a max flow in G (f^* need not be unique)
Let $F^* = |f^*|$ = max flow value

How can we check if $F^* > 0$?

- Let G^+ = subgraph of G of edges with positive capacity
- If exists $s \rightsquigarrow t$ path P in $G^+ \Rightarrow F^* > 0$ because P can support positive flow
- By flow decomposition lemma, this is not only sufficient but necessary. (Also, Exercise 2)

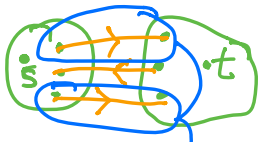
Can we certify that $F^*=0$? (i.e., that $\nexists$ s-t path in G^f)

Yes, by using a cut

- Let $\hat{S} = \{v \in V \mid \exists \text{ s-v path in } G^f\}$
- $s \in \hat{S}$
- if $F^*=0$, then $t \notin \hat{S}$ and instead $t \in V \setminus \hat{S}$

Thus, $\hat{S}$ is an s-t cut that separates s from t.

Definition: An s-t cut is a cut $(S, V \setminus S)$ s.t. $s \in S$ and $t \in V \setminus S$



Definition: The capacity of a cut

$$c(S) = c(S, V \setminus S) = \sum_{u \in S} \sum_{v \in V \setminus S} c(u,v)$$

- $c(S)$: TOTAL CAPACITY of edges that leave S
- $\sum_{u \in S} \sum_{v \in V \setminus S}$: set notation $\Rightarrow$ implicit sum
- only edges leaving S contribute to $c(S)$, not those entering.

So, $c(S)=0 \iff$ cut separates s from t

Summary: $F^*=0 \iff \nexists$ s-t path in $G^f \iff \exists$ s-t cut S w/ $c(S)=0$

- Existence of separating s-t cut $\iff$ non-existence of s-t path
- s-t paths and s-t cuts are dual to each other

The s-t cut problem: Given $G=(V, E, s, t, c)$, find an s-t cut of minimum capacity

Let S^* be a min capacity s-t cut

i.e. $S^* = \underset{\forall S \text{ s-t cuts}}{\arg\min}\ c(S)$

min s-t cut may not be unique

For s-t cut S and flow f, $f(S) = f(S, V \setminus S) = \sum_{u \in S} \sum_{v \in (V \setminus S)} f(u,v)$

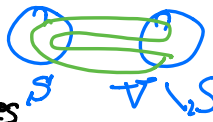
$f(S)$ is the net flow across the cut

Includes negative values that flow toward source side, unlike capacity.

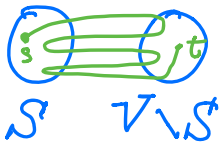
By feasibility, $f(S) \leq c(S)$ for any s-t cut S

Claim I: $f(S) = f(S')$ for any two s-t cuts S and S'

Proof:
- Flow decomposition on $f \rightarrow$ collection of s-t paths and flow cycles.
- Flow cycles contribute zero to $f(S)$ and $f(S')$.
  zig-zags cancel out.



- Every s-t flow path contributes same amount of flow to $f(S)$ and $f(S')$
  all but one zig-zag cancel out, resulting in one net s→t flow.



∎

Because $|f| = f(\{s\}) \overset{\text{by claim I}}{\Longrightarrow} |f| = f(S)$ for any s-t cut S

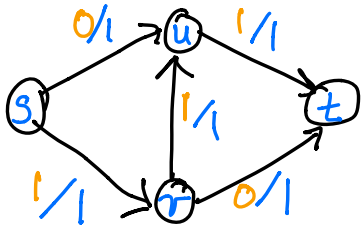
$\Rightarrow$ Max flow $\leq$ min s-t cut because S^* can be any s-t cut & we can choose one of min. cap.

$F^* = |f^*| = f^*(S^*) \leq c(S^*)$

(Weak duality of flows and s-t cuts → will see in Lec 9)

To find max flow, maybe we can iteratively increase value by identifying and adding an s-t path to current flow.

We will find this is a good idea but need to overcome the following issue:

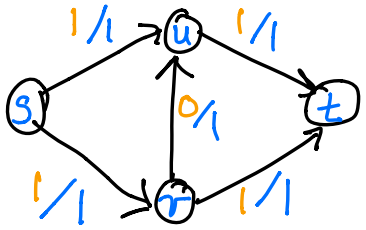


- Value of $|f| = 1$
- Can't just add s-t path
- But there is a flow with value 2

✳ Need to undo some of existing flows!

→ If we could decrease the flow along (v,u), the net $u \to v$ flow will increase, but there is no edge (u,v) along which to push this flow. We can achieve this conceptually all the same.

→ Increase flow on (s,u) and (v,t) by 1 and decrease flow on (v,u) by 1, producing new $|f'| = 2$



→ Can we increase further?
No! By weak duality of flows and s-t cuts

$$2 = |f'| \leq F^* \leq c(S^*) \leq c(\{s\}) = 2$$

$\Rightarrow F^* = 2$ and f' is a max flow

↑ Example cut with capacity = 2. Could have used $\{s,u\}$ or $\{s,u,v\}$.

→ Can we implement this algorithmically?

Residual Network $G_f(V, E_f, s, t, c_f)$
of flow f in network G

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

= how much extra net $u \to v$ flow can be sent

by feasibility of f $\quad 0 \le c_f(u,v) \le c(u,v) + c(v,u)$

Edge $(u,v) \in E_f$ whenever $c_f(u,v) > 0$
discards saturated edges

can also be expressed as just $c(u,v) - f(u,v)$

Note: If f is a flow in G and f' a flow in G_f, then $f + f'$ is a flow in G. (why?)

→ Reduces improving a flow f to finding a non-zero flow in G_f.

If can't find non-zero flow in G_f, $\exists$ s-t cut S with $c_f(S) = 0$ ← residual capacity of S

$$c_f(S) = \sum_{u \in S} \sum_{v \in (V \setminus S)} c_f(u,v)$$

Note: For any s-t cut S

$$c_f(S) = \sum_{u \in S} \sum_{v \in V \setminus S} c_f(u,v) = \sum_{u \in S} \sum_{v \in V \setminus S} c(u,v) - f(u,v)$$

$$= c(S) - f(S)$$ ← Claim I

So if $c_f(\hat{S}) = 0$ then $c(\hat{S}) = f(\hat{S}) = |f|$

and $c(\hat{S}) = |f| \le F^* \le c(S^*) \le c(\hat{S}) \Rightarrow |f| = F^* = c(S^*) = c(\hat{S})$

$\Rightarrow$ f is a max flow and $\hat{S}$ is the min s-t cut !!

11.11

Summary:

① f is a max flow: $|f| = F^*$
iff max flow value of G_f is 0
(iff $\nexists$ s-t path in G_f)

② Max-flow Min-cut Thm: $F^* = c(S^*)$
strong duality of flows and s-t cuts → Lect. 9

To create max flow algorithm:
Augmenting path = directed s-t path in G_f
→ Can push additional flow along augmenting path up to bottleneck capacity.

Ford-Fulkerson [1956]

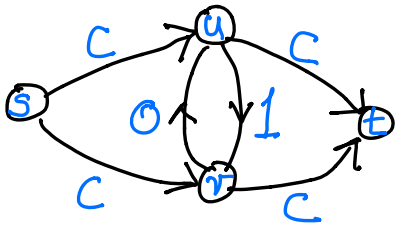
- Start with zero flow
- Find augmenting path P (e.g. via DFS) ← $O(E)$ each time
- Augment flow by pushing $c_f(P)$ along P ←
- Repeat until no more augmenting paths

Correctness follows from idea, f is max flow iff $\nexists$ augmenting path.

Running Time

① If capacities are integers $\in [0, C]$
then $\leq |f| \leq c(\{s\}) \leq |V| \cdot C$ augmentations
because $c_f(P) \geq 1$ always ← $\leq |V|$ edges leaving s

Such bad cases are possible:



- Alternate augmenting paths
  $s \to u \to v \to t$
  $s \to v \to u \to t$

Total running time $O(E \cdot V \cdot C)$

→ pseudopolynomial algorithm = polynomial in sum of input numbers (unary and not binary encoding)

Flow integrality thm: If G has all capacities integral, then $\exists$ max flow that is integral too.

② If capacities rational
⇒ running time still finite and pseudopolynomial

③ If capacities real
⇒ potentially infinite runtime