Lecture #8

Last Time: Flow Decomposition and Augmenting Paths

decompose any flow into s-t paths, and cycles used it to prove min-cut = max flow

Today: Capacity Scaling and Min Cost Matching

Ford-Fulkerson:

while $\exists$ s-t path W in $\underset{\text{residual graph}}{H}$ with $\underset{\text{bottleneck}}{\Delta_H(W)} > 0$

augment along W by $\Delta_H(W)$ (means pushing more flow, or undoing flow)

update residual

Complications: How do we choose <u>which</u> path W?

capacities are irrational $\Rightarrow$ might never terminate, not even converge to max flow

pseudo polynomial, not polynomial in bit complexity of U

capacities are integers bdded by U $\overset{*}{\Rightarrow}$ number of iterations is at most mU (or max flow)

* always finds integer-valued flow, but not all max-flows are integer

Today we will cover capacity scaling

Augment along fattest paths first (no need to find <u>the</u> fattest)

The D-residual graph is the residual graph but where only edges with capacity at least D are kept

Capacity Scaling:

Set $D = U$

while $D \geq 1$

loop #1

Let $H_f(D)$ be the D-residual graph (for current flow)

while $\exists$ s-t path W in $H_f(D)$

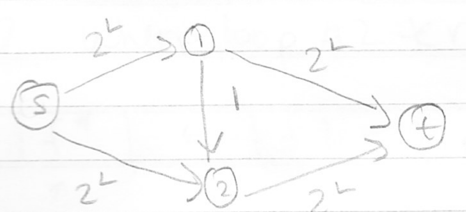
loop #2

Augment along W by $\Delta_{H_f(D)}(W)$

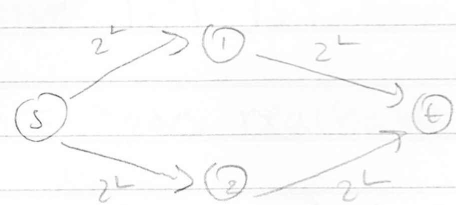
Update D-residual

Set $D = D/2$

How does this handle an example from last time?



The 2^L-residual is



and in two steps, capacity scaling would terminate

Let $n = |V|$, $m = |E|$

Theorem: [Edmonds-Karp], [Dinitz] Capacity scaling computes a maximum flow (when it terminates) and can be implemented in $O(m^2(1+\log U))$ time

exponentially better dependence on U

Lemma 1: At termination, f is a maximum flow

Proof: when $D=1$ then $H_f(D) = H$ the residual graph.

proved this last time

If there are no s-t paths in H, f is a max flow ☒

This establishes correctness, but how about bounds on number of iterations?

Lemma 2: When loop #2 terminates (for some D),

$$|f^*| \leq |f| + Dm$$

max flow

Proof: there is an s-t flow in residual H of value $|f^*| - |f|$.

Let S = nodes reachable from s in $H_f(D)$.

Then $cap_H(S, V/S) \leq Dm$ but by max flow ≤ min-cut this bounds max flow in H ☒

Hence loop #2 iterates at most $2m$ times (why? Because at end of loop #2 $|f^*| \le |f| + m\Delta$ but next scaling is $\Delta/2$)

And loop #1 iterates $1 + \log_2 U$ times

Finally time to find an s-t path in residual is $O(m)$, this completes the proof of thm.

<u>weakly polynomial</u>: run time is polynomial in the description length of the problem

<u>strongly polynomial</u>: run time is polynomical in the number of values $O(\underset{\text{edges}}{m}+\underset{\text{vertices}}{n})$, not allowed to depend on size of value

e.g. choose shortest (i.e. number of arcs) augmenting path runs in $O(m^2 n)$ time (we won't prove this)

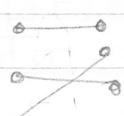
Minimum cost flow

<u>input</u>: Instance of max flow + costs $c: A \to \mathbb{R}$

<u>goal</u>: among all maximum s-t flows, find the one of minimum cost

Today we will focus on a special case of it

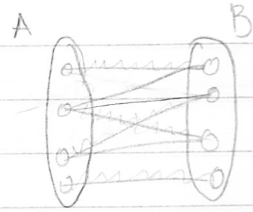
Recall, a <u>matching</u> M is a set of edges

e.g. 

where every node has at most one incident edge

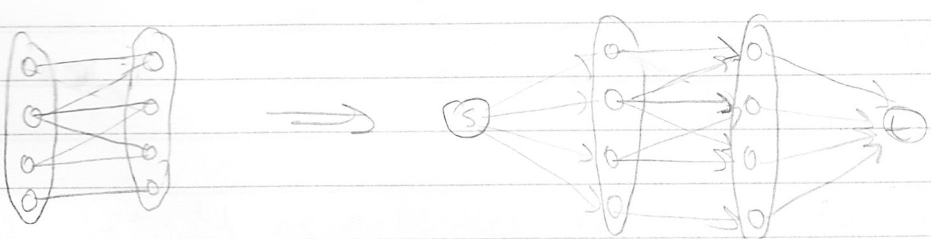
In bipartite perfect matching we are given a bipartite graph

$G = (V, E)$ with $V = A \cup B$



Is there a matching that covers all nodes?

As many of you have seen before it can be reduced to maximum flow



all arcs unit capacity

Let $|A| = |B| = n$, then there is an s-t flow of value n iff there is a perfect matching.

e.g. matching buyers and houses

But what if edges have costs (or values)? Can we find perfect matching with minimum cost?

Given some perfect matching M we can construct a residual H_M where

$G=(V,E)$ $\qquad\qquad$ H

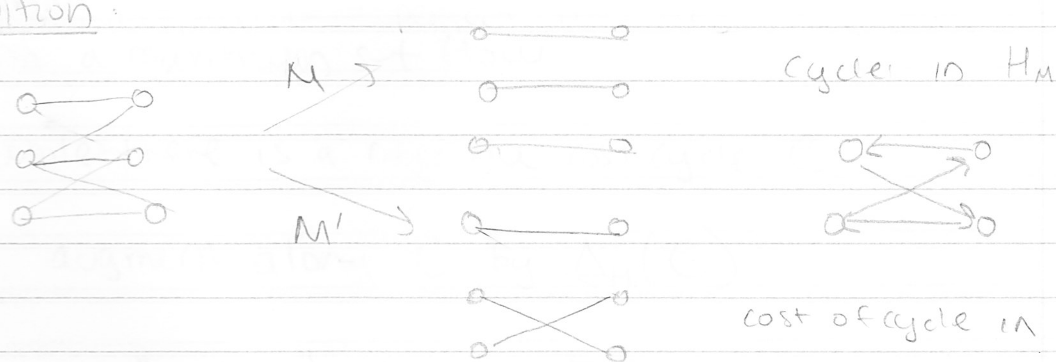
$a \text{ --- } b \quad \Longrightarrow \quad a \longrightarrow b$

$(a,b) \notin M$ $\qquad\qquad$ $c_H(a,b) = c(a,b)$

$a \text{ --- } b \quad \Longrightarrow \quad a \longleftarrow b$

$(a,b) \in M$ $\qquad\qquad$ $c_H(b,a) = -c(a,b)$

Intuition:



cost of cycle in $H_{M'}$ is $c(M') - c(M)$

Lemma: Let M be a perfect matching. Then H_M has a cycle of negative total cost iff M is not the min cost perfect matching.

Proof: ($\Longrightarrow$) Augmenting along $\mathcal{C}$ gives a new perfect matching M' with

$$c(M') = c(M) + c(\mathcal{C}) < c(M)$$

($\Longleftarrow$) Let M' be any perfect matching whose cost is strictly less. Consider

$$\underbrace{M \triangle M'}_{\text{symmetric difference}} \equiv \text{edges in } M \text{ or } M' \text{ but } \underline{\text{not}} \text{ both}$$

$M \Delta M'$ is a node-disjoint set of cycles in H_M and

$$c_H(e_1) + c_H(e_2) \ldots + c_H(e_k) = c(M') - c(M)$$

thus if $c(M') < c(M)$, at least one cycle must have negative cost $\boxtimes$

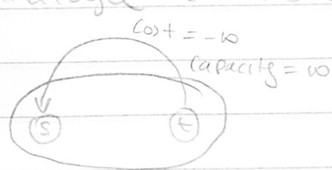
<u>Klein cycle cancelling:</u>

Find a maximum s-t flow

while there is a negative cost cycle e

augment along e by $\Delta_H(e)$

This is the "analogue" of Ford-Fulkerson



How do we choose <u>which</u> negative cost cycle?

any negative cost cycle $\Rightarrow$ pseudopolynomial time

via Bellman Ford

most negative cost $\Rightarrow$ uh oh! NP-hard

For minimum cost perfect matching, not too hard to find a good rule (next time) and we will then study general case