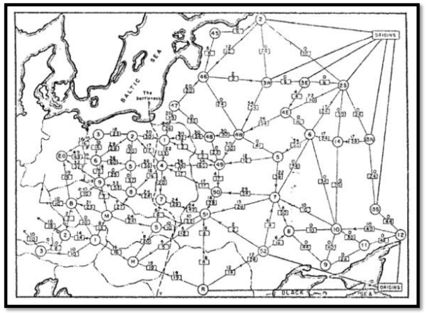
CSE 421: Algorithms

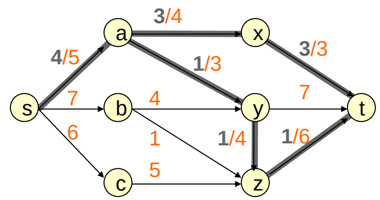
Winter 2014

Lecture 18: The max-flow/min-cut theorem

Reading:
Sections 7.1-7.2



finding maximum-flows (integer capacities)



Ford-Fulkerson method

Start with $f = 0$ for every edge

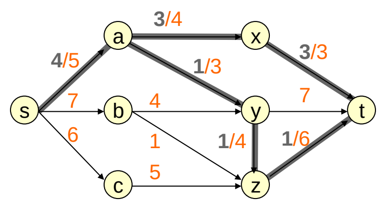
While G_f has an augmenting path, augment.

Questions:

- Does it halt?
- Does it find a maximum flow?
- How fast?

residual capacity

- The **residual capacity** (w.r.t. f) of (u,v) is $c_f(u,v) = c(u,v) - f(u,v)$ if $f(u,v) \leq c(u,v)$ and $c_f(u,v)=f(v,u)$ if $f(v,u) > 0$



- e.g. $c_f(s,b)=7$; $c_f(a,x) = 1$; $c_f(x,a) = 3$

residual graph & augmenting paths

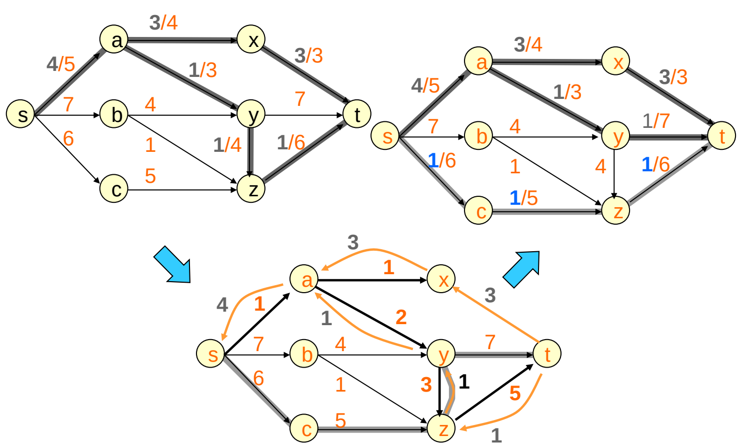
- The **residual graph** (w.r.t. **f**) is the graph $G_f = (V, E_f)$, where $E_f = \{ (u,v) \mid c_f(u,v) > 0 \}$
  - Two kinds of edges
    - Forward edges
      - $f(u,v) < c(u,v)$ so $c_f(u,v) = c(u,v) - f(u,v) > 0$
    - Backward edges
      - $f(u,v) > 0$ so $c_f(v,u) \geq -f(v,u) = f(u,v) > 0$

- An **augmenting path** (w.r.t. **f**) is a simple $s \rightarrow t$ path in G_f.

augmenting a flow along a path

augment(**f**,**P**)
    $c_P \leftarrow \min_{(u,v)\in P} c_f(u,v)$ "bottleneck(P)"
    for each e∈**P**
        if **e** is a forward edge then
            increase **f**(**e**) by c_P
        else (**e** is a backward edge)
            decrease **f**(**e**) by c_P
        endif
    endfor
    return(**f**)

augmenting a flow



last time

Lemma:
If G_f has an augmenting path **P**, then the function **f'=augment(f,P)** is a legal flow.

always halts

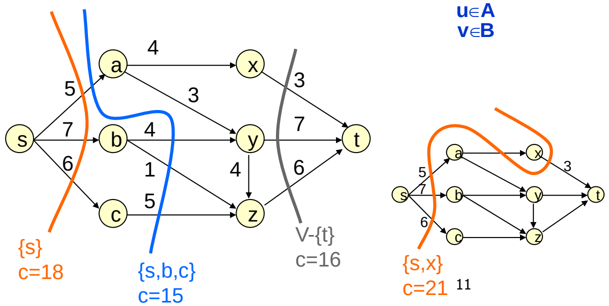
- At every stage the capacities and flow values are always integers (if they start that way)
- The flow value $\nu(f') = \nu(f) + c_P > \nu(f)$ for
  $f' =$ augment(f,P)
  - Since edges of residual capacity **0** do not appear in the residual graph
- Let $C = \sum_{(s,u)\in E} c(s,u)$
  - $\nu(f) \leq C$
  - **F-F** does at most **C** rounds of augmentation since flows are integers and increase by at least **1** per step

running time

- For $f = 0$, $G_f = G$
- Finding an augmenting path in G_f is graph search $O(n+m)=O(m)$ time
- Augmenting and updating G_f is $O(n)$ time
- Total $O(mC)$ time
- **Does it find a maximum flow?**
  - Need to show that for every flow **f** that isn't maximum G_f contains an **s-t**-path

cuts

- A partition **(A,B)** of **V** is an *s-t-cut* if $s \in A$, $t \in B$
- **Capacity** of cut **(A,B)** is $c(A,B) = \sum_{\substack{u\in A\\ v\in B}} c(u,v)$

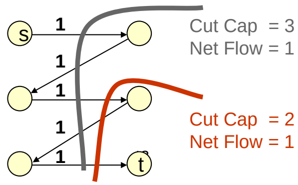


convenient definitions

- $f^{out}(A) = \sum_{v\in A,\, w\notin A} f(v,w)$
- $f^{in}(A) = \sum_{v\in A,\, u\notin A} f(u,v)$

claims

- For any flow **f** and any (s,t)-cut **(A,B)**,
  - the net flow across the cut equals the total flow: $\nu(f) = f^{out}(A) - f^{in}(A)$, and
  - the net flow across the cut cannot exceed the capacity of the cut: $f^{out}(A) - f^{in}(A) \leq c(A,B)$
- **Corollary:**
  Max flow ≤ Min cut



proof of claim

- Consider a set **A** with $s \in A$, $t \notin A$
- $f^{out}(A) - f^{in}(A) = \sum_{v \in A,\, w \notin A} f(v,w) - \sum_{v \in A,\, u \notin A} f(u,v)$
- We can add flow values for edges with both endpoints in **A** to **both** sums and they would cancel out so
- $f^{out}(A) - f^{in}(A) =$
- $\nu(f) = f^{out}(s)$ and $f^{in}(s) = 0$

proof of claim

$$\begin{aligned}
\nu(f) &= f^{out}(A) - f^{in}(A) \\
&\leq f^{out}(A) \\
&= \textstyle\sum_{v \in A,\, w \notin A} f(v,w) \\
&\leq \textstyle\sum_{v \in A,\, w \notin A} c(v,w) \\
&\leq \textstyle\sum_{v \in A,\, w \in B} c(v,w) \\
&= c(A,B)
\end{aligned}$$

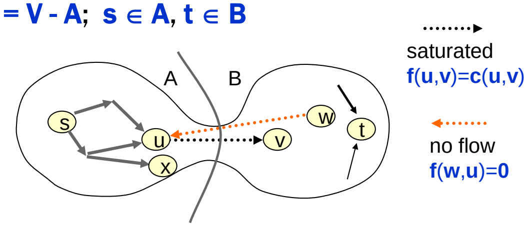
max flow/min cut theorem

Theorem: For any flow **f**, if G_f has no augmenting path then there is some **s**-**t**-cut **(A,B)** such that $\nu(f) = c(A,B)$ (proof on next slide)

- We know by previous claims that any flow **f'** satisfies $\nu(f') \leq c(A,B)$ and we know that F-F runs for finite time until it finds a flow **f** satisfying conditions of the theorem. Therefore for any flow **f'**, $\nu(f') \leq \nu(f)$
- **Corollary:**
  - (1) F-F computes a maximum flow in **G**
  - (2) For any graph **G**, the value **n(f)** of a maximum flow = minimum capacity **c(A,B)** of any **s**-**t**-cut in **G**

proof of the theorem

Let $A = \{ u \mid \exists$ a path in G_f from s to $u \}$
$B = V - A$; $s \in A$, $t \in B$



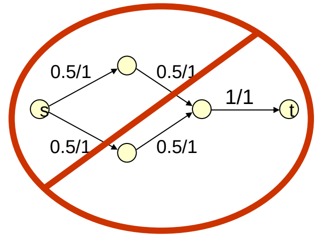
This is true for **every** edge crossing the cut:
$\nu(f) = f^{out}(A) - f^{in}(A) = c(A,B)$ and $f^{in}(A) = 0$ hence

$$f^{out}(A) = \sum_{\substack{u \in A \\ v \in B}} f(u,v) = \sum_{\substack{u \in A \\ v \in B}} c(u,v) = c(A,B)$$

flow integrality theorem

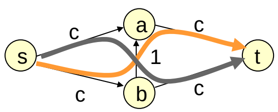
If all capacities are integers

- The max flow has an integer value
- Ford-Fulkerson method finds a max flow in which $f(u,v)$ is an integer for all edges (u,v)



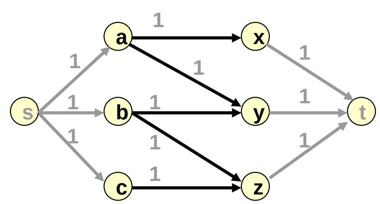
corollaries & facts

- If Ford-Fulkerson terminates, then it has found a max flow.
- It will terminate if $c(e)$ integer or rational (but may not if they're irrational).
- However, may take exponential time, even with integer capacities:



$c = 10^9$, say

bipartite matching



Integer flows implies each flow is just a subset of the edges

Therefore flow corresponds to a matching

$O(mC)=O(nm)$ running time

next time: capacity-scaling algorithm

- General idea:
  - Choose augmenting paths **P** with ‘large’ capacity c_P
  - Can augment flows along a path **P** by any amount $\Delta \leq c_P$

    Ford-Fulkerson still works
  - Get a flow that is maximum for the high-order bits first and then add more bits later