Algorithms

End of MST
Intro to Flows

Recap

- Next HW: Oral grading on Monday, 11/4 & Tuesday, 11/5 (already posted)
- Reading due Monday
- Reading due Wednesday
  - ↳ Note: Larger than usual!
- No reading next Friday

MST Recap

- Use when the goal is minimum connection of all vertices

Key: Greedy!

We saw 3 ways:

1. Find all safe edges. Add them & recurse.
   (Borůvka): $O(E \log V)$

2. Keep a single connected component.
   At each iteration, add 1 safe edge.
   (Jarník/Prim): $O(E + V \log V)$

3. Sort edges & loop through them.
   If edge is safe, add it.

   (Kruskal): $O(E \underbrace{\log E}_{= \log V})$

Which is best?? $E < \binom{V}{2}$

Ch 10: Flows

Motivation:

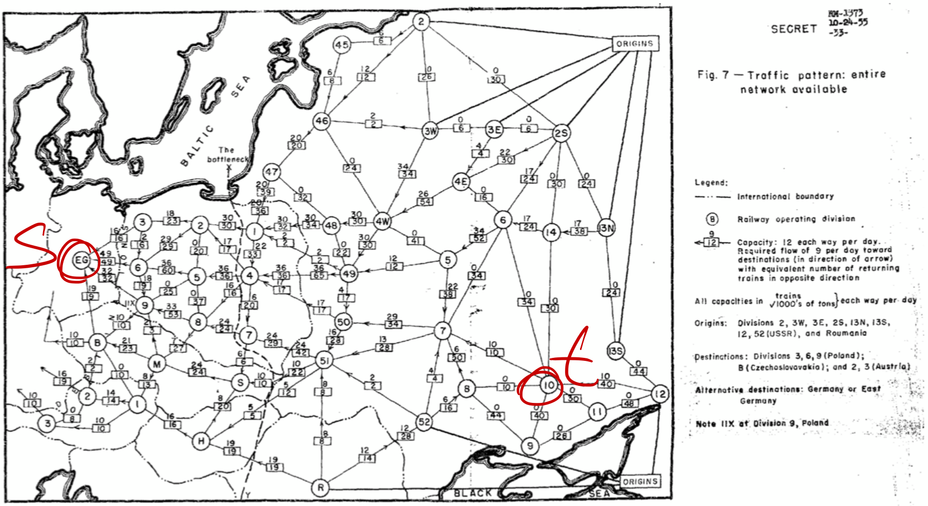


Figure 10.1. Harris and Ross's map of the Warsaw Pact rail network. (See Image Credits at the end of the book.)

How to send from one vertex to another?

How to divide one vertex from another?

More formally:

Given a directed graph with two designated vertices, s and t.

Each edge is given a capacity $c(e)$.

Assume:

- No edges enter s.
- No edges leave t.
- Every $c(e) \in \mathbb{Z}$.

integer capacity

Goal:

Max flow: find the most we can ship from s to t without exceeding any capacity

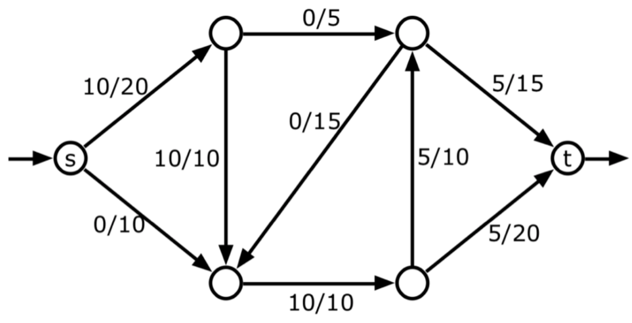
Min cut: find smallest set of edges to delete in order to disconnect s + t

Flows:

A **flow** is a function $f : E \to \mathbb{R}^+$, where $f(e)$ is the amount of flow going over edge e.

Must satisfy 2 things:

- Edge constraints:
  $$0 \le f(e) \le c(e)$$
- Vertex constraints:
  for $v \neq s, t$: $$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$



An (s,t)-flow with value 10. Each edge is labeled with its flow/capacity.

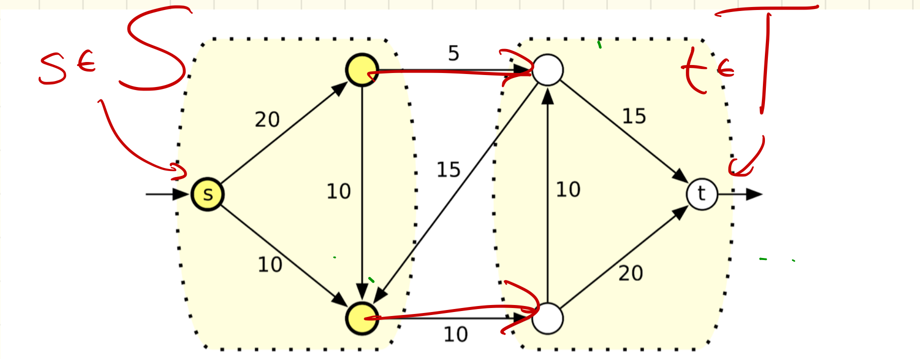
$$\text{Value}(f) = \sum_{e \text{ out of } s} f(e) = \sum_{e \text{ into } t} f(e)$$

Cuts:

An s-t cut is a partition of the vertices into 2 sets, S and T, so that:

- $s \in S$
- $t \in T$
- $S \cap T = \emptyset$, $S \cup T = V$

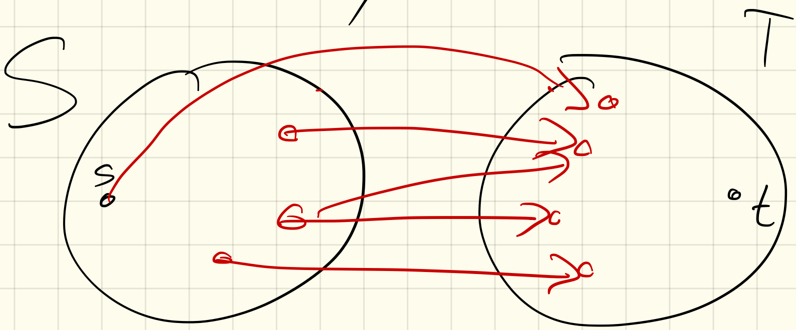
The capacity of a cut is $\displaystyle\sum_{\substack{\vec{uv} \in E \\ \text{with } u \in S, v \in T}} c(\vec{uv})$



An (s, t)-cut with capacity 15. Each edge is labeled with its capacity.

Intuitively, these are connected:

Consider <u>any cut</u>:



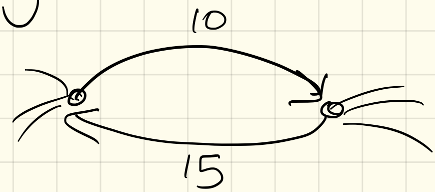
No flow can send more than capacity of the cut.

<u>any flow $\leq$ any cut</u>

Note:

We'll assume every pair of vertices has at most one edge.

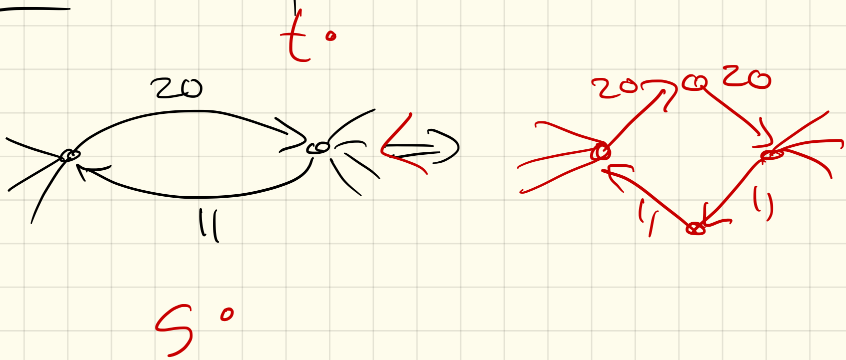
So no:



Why?

- Makes calculations easier!

How? Simple transformation:



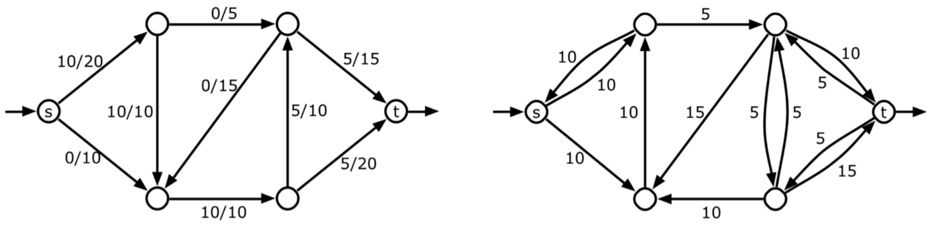
Thm: (Ford-Fulkerson '54, Elias-Feinstein-Shannon '56)
The max flow value
= min cut value

One way is easy:

Any flow $\leq$ any cut.
Why? (See 2 slides back)

Key tool in proof:

Residual network G_f:



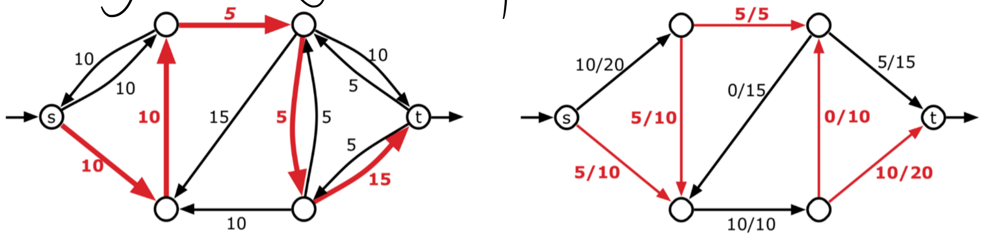
A flow f in a weighted graph G and the corresponding residual graph G_f.

Intuitively:

Shows how much more (or less) flow can be pushed through an edge.

Q: Could G have no ability to send more flow, but there is a larger flow?

Augmenting a path:



An augmenting path in G_f with value $F = 5$ and the augmented flow f'.

This is just an s-t path in G_f.

Then, find min capacity edge on that path.

Claim: I can build a new flow whose value is bigger than f's :