Flow Networks

- A **flow network** is a digraph $G = (V, E)$, together with a **source** vertex $s \in V$, a **sink** vertex $t \in V$, and a **capacity function** $c: E \rightarrow \mathbb{N}$.
Flow in a Network

- Given a flow network \( G = (V, E, s, t, c) \), a flow in \( G \) is a function \( f : E \to \mathbb{N} \) that satisfies
  - Every \( e \in E \) satisfies \( f(e) \leq c(e) \).
  - Every \( v \in V \setminus \{s, t\} \) satisfies
    \[
    \sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)
    \]

Example: Flow

- The capacities are in red.
- The flow is in blue.
Reminder: Cuts

- A cut is a partitioning of the vertices of the flow network into two sets $S, T$ such that $s \in S$ and $t \in T$.
- The size of a cut is the sum of the capacities of the edges from $S$ to $T$.

Reminder: Max Flow – Min Cut

- Max flow – min cut theorem. In every flow network, the size of the minimum cut is equal to the size of the maximum flow.
Warm-up: Anti-Parallel Edges

• Two directed edges are said to be anti-parallel if they have the same endpoints, but are in opposite directions.

• Problem. Consider a flow network \((V, E, s, t, c)\), and let \(e, e' \in E\) be anti-parallel edges. Prove that there exists a maximum flow in which at least one of \(e, e'\) has no flow through it.

Solution

• Consider a maximum flow \(f\).
  ◦ If either \(e\) or \(e'\) has no flow through it in \(f\), we are done.
  ◦ Assume, WLOG, that \(f(e) \leq f(e')\).
  ◦ By decreasing \(f(e')\) by \(f(e)\) and then setting \(f(e) = 0\), we obtain a valid flow of the same size.
    • Let \(e = (v, u)\). Both the incoming and the outgoing flows of \(v\) and \(u\) were decreased by the same amount, so the constraints are still satisfied.
Recall: The **Rand corporation** studied the Soviet train system.

They studied the Soviet ability to transport things from the Asian side to the European side.
They also studied the **minimum cut**.

**Problem.** In this scenario there are **several sources and sinks**!

---

**Problem 2: Several Sources and Sinks**

**Problem.** We are given a flow network with **several sources** and **several sinks**. Explain how to use an algorithm for finding a maximum flow (in a standard flow network) for this case.
Solution

- We add a "super source" $S$, and add an edge from it to each of the sources. Each of these edges has an infinite capacity.
- We symmetrically add a "super sink" $T$.
- Run the original algorithm from $S$ to $T$.

Correctness

- There is a bijection between the flows of the original network and the flows of the new network (and the original edges have the same flow in both cases).
  - A flow in the new network yields a flow in the original one by removing the source and the sink, and vice versa.
  - Two corresponding flows have the same size.
  - A max flow in the new network corresponds to a max flow in the original one.
Problem 3: Even Flow

• **Problem.** Given a flow network \((V, E, s, t, c)\) such that all of the capacities are even, prove that the size of the maximum flow is even.

The capacities are **even**.

Every cut has an **even size**.

The minimum cut has an **even size**.

The maximum flow has an **even size**.
Problem 4: Edge-disjoint Paths

• **Problem.** Given a digraph $G = (V, E)$ and vertices $s, t \in V$, describe an algorithm that finds the maximum number of edge-disjoint paths from $s$ to $t$.

Solution: Edge-disjoint Paths

• We give every edge a capacity of 1. A network with only 1-capacities is called a 0-1 network (max flow can be computed more efficiently in such networks).
• Find a max flow in the resulting 0-1 network.
Correctness of Solution

- Given a 0-1 flow network with a max flow $f$ and maximum number of edge-disjoint paths $k$, we need to prove $|f| = k$.
- $|f| \geq k$: By having a flow of 1 through every disjoint path, we obtain a flow of size $k$.
- $|f| \leq k$: Proof by induction on $|f|$.
  - Induction basis: obvious when $|f| = 0$.

Correctness of Solution (2)

- Induction step (show that $m = |f| \leq k$).
  - Consider a maximum flow (of size $m$).
  - Remove every edge with 0 flow through it.
  - Find a path from $s$ to $t$ (it exists since $|f| > 0$).
  - Remove the edges of the path, to obtain a network with a flow of size $m - 1$.
  - By the induction hypothesis, there are $m - 1$ edge-disjoint paths in this network.
  - Bring back the removed path, to obtain at least $m$ edge-disjoint paths.
Perhaps RAND were also studying the minimum number of train tracks needed to be destroyed to prevent any transportation from the Asian side to the European side?

Problem 5: Disconnecting Edges

**Problem.** Given a digraph $G = (V, E)$ and vertices $s, t \in V$, describe an algorithm that finds the minimum number of edges needed to be removed so that no path from $s$ to $t$ remains.
Solution: Disconnecting Edges

- As before, we give every edge a capacity of 1, to obtain a 0-1 network.
- Find a max flow in the network.
- We already proved that the max flow equals the maximum number of edge-disjoint paths.
- It remains to prove that the max number of edge-disjoint paths equals the min number of edges needed to disconnect $s$ from $t$.

Proof

- $k$ – maximum number of edge disjoint paths.
- $\ell$ – minimum number of disconnecting edges.

- $\ell \geq k$: There are $k$ edge-disjoint paths, and we need to remove at least one edge from each.
- $\ell \leq k$: The min cut is a set of disconnecting edges. By the max flow – min cut theorem, there are $k$ edges in the min cut.
Menger’s Theorem

- The idea that the min number of disconnecting edges is equal to the max number of edge-disjoint paths is called Menger’s Theorem, and is from 1927.

Problem 6: Vertex-disjoint Paths

- **Problem.** Given a digraph $G = (V, E)$ and vertices $s, t \in V$, describe an algorithm that finds the maximum number of vertex-disjoint paths from $s$ to $t$. 
Solution: Vertex-disjoint Paths

- We split every vertex of $V \setminus \{s, t\}$ as follows:

- In the resulting graph, paths are edge-disjoint if and only if they are vertex-disjoint.
- As before, we add capacities of 1, to obtain a 0-1 network.

Solution: Vertex-disjoint Paths (2)

- Algorithm:
  - We build a 0-1 network as described in the previous slide.
  - Find max flow in the resulting network.

- It remains to prove:
  - There is a bijection between sets of vertex-disjoint paths in the original graph and sets of edge-disjoint paths in the new network.
Proof

- Every path in the new network is of the form $s \rightarrow v_{in} \rightarrow v_{out} \rightarrow u_{in} \rightarrow \cdots \rightarrow w_{in} \rightarrow w_{out} \rightarrow t$.
- It corresponds to the path in the original graph: $s \rightarrow v \rightarrow u \rightarrow \cdots \rightarrow w \rightarrow t$.
- A set of vertex-disjoint paths in the original graph corresponds to a set of vertex-disjoint paths in the new network, and these are edge-disjoint.
- In the new network, a set of edge-disjoint paths are also vertex-disjoint, and thus also the corresponding paths in the original graph.

The End