Network Flows and Graph Cuts

August 23, 2025

Maximum Bipartite Matching

Question:

Give a bipartite graph, how can we find a largest matching.

Solution:

We can find a largest matching by max-flow algorithms.

Minimum Cut Problem

Input

- A directed $G = (V, E)$;
- A cost $C(e) \geq 0$ for each edge e;
- A source $s \in V$ and a sink $t \in V$

Goal

Find a cut S such that $s \in S$ and $t \notin S$ minimizing the cost.

Max-Flow Min-Cut Theorem

Theorem

Maximum Flow is equal to Minimal Cut.

Upper Bound Flow Value by Cut Cost

Lemma

Let f be a s-t flow, and $(S, \overline{S})$ a s-t cut. Then

$$v(f) = \sum_{e=(u,v):u \in S, v \notin S} f(e) - \sum_{e=(u,v):u \notin S, v \in S} f(e).$$

Proof

Proof.

$$v(f) = \sum_{e=(s,\cdot)} f(e) = \sum_{v\in S, e=(v,\cdot)} f(e) - \sum_{v\in S, e=(\cdot,v)} f(e)$$

□

Maximum Flow Algorithms

Greedy Algorithm.

Maximum Flow Algorithms

Residual Capacity

- Forward edge $e = (u, v)$: with residual capacity $c'(e) = c(e) - f(e)$
- Backward edge $e = (v, u)$: with residual capacity $c'(e) = f(e)$

Residual Graph and Augmented Paths

Ford–Fulkerson Algorithm

Correctness of Ford–Fulkerson

- We prove it is a flow
- We prove it is the maximum flow

Correctness of Ford–Fulkerson

Correctness of Ford–Fulkerson

Correctness of Ford–Fulkerson

Thanks!