Lecture 12: Flows and LPs

- Algorithm (FF)
  - Max Flow Min Cut
- Min cost max flows
- LPs

\[ G = (V, E) \]

1. \( 0 \leq f_{e} \leq c_{e} \)
2. Flow conservation
3. No flow entering \( s \), no flow leaving \( t \)
\[ \text{val}(f) = \sum_{\text{paths } P \in \text{decomp}} \phi(P) \leq \sum_{P \in \text{decomp}} \text{bottleneck}(P) \]

\[ \leq \sum_{\text{edges } e \in C: P \models e} \phi(P) \leq \sum_{P \text{ hub } P} \text{edge in } C \leq \text{value}(C) \]

S-t Cut: set of edges that "block" every s-t path.
Thm [Menger]: \( \exists \) flow \( f^* \) \( \exists \) cut \( C^* \) \( \max \) valued \( f^* \) \( \leq \) \( \text{value}(C^*) \)

- if all caps are integers then \( f^* \) can also be integer valued

found in poly time (always)

poly \((n,m,\log \text{num_w})\)

Suppose each edge has a cost \( c_e \)

Want a flow \( f^* \)

that has

1. max value and

2. min cost among all max value flows.

Thm! can also find this in polytime

If caps are integer, then \( \exists \) an integer valued \( f^* \) satisfying
Give an algo for max flow. (Outline the MPMC proof)

Ford - Fulkerson
Let $L =$ all vertices reachable from $s$

$R =$ all the rest

Claim: original edges going from $L$ to $R$ form a min-cut
Linear Programs:

- Powerful (gen. Min Cost Maxflow)
- Versatile
- Easy to use
- Machine is
  - may not return integer solns (even if input is integer)
  - slower (maybe)