Network Flows, Matching

Shayan Oveis Gharan
Applications of Max Flow: Bipartite Matching
Bipartite Matching Problem

Given an undirected bipartite graph $G = (X \cup Y, E)$
A set $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
**Goal**: find a matching with largest cardinality.

![Graph Example](image)
Create digraph $H$ as follows:

- Orient all edges from $X$ to $Y$, and assign infinite (or unit) capacity.
- Add source $s$, and unit capacity edges from $s$ to each node in $L$.
- Add sink $t$, and unit capacity edges from each node in $R$ to $t$.
Bipartite Matching: Proof of Correctness

**Thm.** Max cardinality matching in $G$ = value of max flow in $H$.

**Pf.** 

Given max matching $M$ of cardinality $k$.
Consider flow $f$ that sends 1 unit along each of $k$ edges of $M$.
f is a flow, and has cardinality $k$. $\blacksquare$
Bipartite Matching: Proof of Correctness

**Thm.** Max cardinality matching in $G = \text{value of max flow in } H$. 

**Pf. (of $\geq$)** Let $f$ be a max flow in $H$ of value $k$. Integrality theorem $\Rightarrow$ $k$ is integral and we can assume $f$ is 0-1.

Consider $M = \text{set of edges from } X \text{ to } Y \text{ with } f(e) = 1$.
- each node in $X$ and $Y$ participates in at most one edge in $M$
- $|M| = k$: consider $s$-$t$ cut $(s \cup X, t \cup Y)$
Perfect Bipartite Matching
Perfect Bipartite Matching

**Def.** A matching $M \subseteq E$ is **perfect** if each node appears in exactly one edge in $M$.

**Q.** When does a bipartite graph have a perfect matching?

Structure of bipartite graphs with perfect matchings:
- Clearly we must have $|X| = |Y|$.
- What other conditions are necessary?
- What conditions are sufficient?
Perfect Bipartite Matching: N(S)

**Def.** Let $S$ be a subset of nodes, and let $N(S)$ be the set of nodes adjacent to nodes in $S$.

**Observation.** If a bipartite graph $G$ has a perfect matching, then $|N(S)| \geq |S|$ for all subsets $S \subseteq X$.

**Pf.** Each $v \in S$ has to be matched to a unique node in $N(S)$. 
Marriage Theorem

Thm: [Frobenius 1917, Hall 1935] Let $G = (X \cup Y, E)$ be a bipartite graph with $|X| = |Y|$.
Then, $G$ has a perfect matching iff $|N(S)| \geq |S|$ for all subsets $S \subseteq X$.

Pf. $\Rightarrow$
This was the previous observation.
If $|N(S)| < |S|$ for some $S$, then there is no perfect matching.
Marriage Theorem

**Pf.** \( \exists S \subseteq X \) s.t., \( |N(S)| < |S| \) \( \iff \) G does not a perfect matching

Formulate as a max-flow and let \((A, B)\) be the min s-t cut

G has no perfect matching \( \implies v(f^*) < |X|. \) So, \( \text{cap}(A, B) < |X| \)

Define \( X_A = X \cap A, X_B = X \cap B, Y_A = Y \cap A \)

Then, \( \text{cap}(A, B) = |X_B| + |Y_A| \)

Since min-cut does not use \( \infty \) edges, \( N(X_A) \subseteq Y_A \)

\[ |N(X_A)| \leq |Y_A| = \text{cap}(A, B) - |X_B| = \text{cap}(A, B) - |X| + |X_A| < |X_A| \]
Bipartite Matching Running Time

Which max flow algorithm to use for bipartite matching?

Generic augmenting path: \( O(m \ \text{val}(f^*)) = O(mn) \).
Capacity scaling: \( O(m^2 \log C) = O(m^2) \).
Shortest augmenting path: \( O(m \ n^{1/2}) \).

Non-bipartite matching.

Structure of non-bipartite graphs is more complicated, but well-understood. [Tutte-Berge, Edmonds-Galai]
Blossom algorithm: \( O(n^4) \). [Edmonds 1965]
Best known: \( O(m \ n^{1/2}) \). [Micali-Vazirani 1980]
Edge Disjoint Paths
Edge Disjoint Paths Problem

Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find the max number of edge-disjoint $s$-$t$ paths.

**Def.** Two paths are edge-disjoint if they have no edge in common.

**Ex:** communication networks.
Max Flow Formulation

Assign a unit capacity to every edge. Find Max flow from s to t.

**Thm.** Max number edge-disjoint s-t paths equals max flow value.

**Pf.** ≤
Suppose there are k edge-disjoint paths $P_1, ..., P_k$.
Set $f(e) = 1$ if e participates in some path $P_i$; else set $f(e) = 0$.
Since paths are edge-disjoint, f is a flow of value k. □
**Thm.** Max number edge-disjoint s-t paths equals max flow value.

**Pf.** ≥ Suppose max flow value is k

Integrality theorem ⇒ there exists 0-1 flow f of value k.

Consider edge (s, u) with f(s, u) = 1.

- by **conservation**, there exists an edge (u, v) with f(u, v) = 1
- continue until reach t, always choosing a new edge

This produces k (not necessarily simple) edge-disjoint paths.

We can return to u so we can have cycles. But we can eliminate cycles if desired.
Network Connectivity
Network Connectivity

Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find min number of edges whose removal disconnects $t$ from $s$.

**Def.** A set of edges $F \subseteq E$ disconnects $t$ from $s$ if all $s$-$t$ paths uses at least one edge in $F$.

**Ex:** In testing network reliability
Thm. [Menger 1927] The max number of edge-disjoint s-t paths is equal to the min number of edges whose removal disconnects t from s.

Pf.

i) We showed that max number edge disjoint s-t paths = max flow.

ii) Max-flow Min-cut theorem => min s-t cut = max-flow

iii) For a s-t cut (A,B), cap(A,B) is equal to the number of edges out of A. In other words, every s-t cut (A,B) corresponds to cap(A,B) edges whose removal disconnects s from t.

So, max number of edge disjoint s-t paths = min number of edges to disconnect s from t.