Lectures 16
Maximum Flow
• Applications of Max Flow
  • Bipartite matching
  • Edge-disjoint paths

Sofya Raskhodnikova

S. Raskhodnikova; based on slides by A. Smith and K. Wayne.
Last time: Ford-Fulkerson

1. Find max s-t flow & min s-t cut in $O(mnC)$ time
   - All capacities are integers from 1 to $C$

2. **Duality**: Max flow value = min cut capacity

3. **Integrality**: If capacities are integers, then FF algorithm produces an integral max flow
Today: Applications when C=1

- Maximum bipartite matching
  - Reducing MBM to Max Flow
  - Hall’s theorem

- Edge-disjoint paths
  - another reduction to Max Flow

Still to Finish: faster algorithm for large C (capacity scaling)
7.5 Bipartite Matching

Application of Max Flow With $C=1$
Matching

- Input: undirected graph $G = (V, E)$.
- $M \subseteq E$ is a matching if each node appears in at most 1 edge in $M$.
- Max matching: find a maximum cardinality matching.
Bipartite Matching

Bipartite matching.
- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a maximum cardinality matching.

Matching:
$1-2', 3-1', 4-5'$
Bipartite Matching

Bipartite matching.

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most edge in $M$.
- Max matching: find a maximum cardinality matching.
Reductions

Roughly: Problem A reduces to problem B if there is a simple algorithm for A that uses an algorithm for problem B as a subroutine.

Usually:
- Given an instance $x$ of problem A
  - we find an instance $x'$ of problem B
- Solve $x'$
- Use the solution to build a solution to $x$

Useful skill: quickly identifying problems where existing solutions may be applied.
- Good programmers do this all the time
Reduction: Given a bipartite graph \( G = (L \cup R, E) \),

- Direct all edges in \( E \) from \( L \) to \( R \), and assign capacity 1 to each edge.
- Add source \( s \), and capacity 1 edges from \( s \) to each node in \( L \).
- Add sink \( t \), and capacity 1 edges from each node in \( R \) to \( t \).
- Output the resulting digraph \( G' = (L \cup R \cup \{s, t\}, E') \).
Theorem. Cardinality of max matching in $G = \text{value of max flow in } G'$.

Proof: We need two statements

- max matching in $G \leq \text{max flow in } G'$
  
  Equivalently: every matching of cardinality $k$ in $G$ can be transformed into flow of value $k$ in $G'$.

- max matching in $G \geq \text{max flow in } G'$
  
  Equivalently: every flow of value $k$ in $G'$ can be transformed into a matching of cardinality $k$ in $G$. 
Theorem. Cardinality of max matching in $G = \text{value of max flow in } G'$.

Pf. $\leq$

- Given max matching $M$ of cardinality $k$.
- Consider flow $f$ that sends 1 unit along each of $k$ paths.
- $f$ is a flow, and has value $k$. □
Theorem. Cardinality of max matching in $G$ = value of max flow in $G'$.

Pf. $\geq$

- Let $k$ be the max flow value in $G'$.
- **Integrality theorem** $\Rightarrow$ there is an integral flow $f$ of value $k$ in $G'$
- All capacities are 1 $\Rightarrow$ $f$ is 0-1.
- Consider $M$ = set of edges from L to R with $f(e) = 1$.
  - each node in L and R participates in at most one edge in $M$
  - $|M| = k$: consider flow across cut $(L \cup s, R \cup t)$

---

**Bipartite Matching: Proof of Correctness**

Graphical representation of $G'$ and $G$.
Exercises

• Give an example where the greedy algorithm for MBM fails.

• How bad can the greedy algorithm be, i.e. how far can the size of the maximum matching (global max) be from the size of the greedy matching (local max)?

• What do augmenting paths look like in this max-flow instance?
Perfect 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 $|L| = |R|$.
- What other conditions are necessary?
- What conditions are sufficient?
**Notation.** 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 $G = (L \cup R, E)$ has a perfect matching, then $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.

**Pf.** Each node in $S$ has to be matched to a different node in $N(S)$.

---

**No perfect matching:**

$S = \{ 2, 4, 5 \}$

$N(S) = \{ 2', 5' \}$. 
Marriage Theorem. [Frobenius 1917, Hall 1935] Let $G = (L \cup R, E)$ be a bipartite graph with $|L| = |R|$. Then, $G$ has a perfect matching iff

$$|N(S)| \geq |S| \text{ for all subsets } S \subseteq L.$$ 

**Pf.** ⇒ This was the previous observation.

No perfect matching:
$S = \{ 2, 4, 5 \}$

$N(S) = \{ 2', 5' \}$. 

---

**Marriage Theorem**
Proof of Marriage Theorem

Pf. Suppose \( G \) does not have a perfect matching.

- Formulate as a max flow problem with \( \infty \) constraints on edges from \( L \) to \( R \) and let \( (A, B) \) be min cut in \( G' \).

- **Key property \#1** of \( G' \): min-cut cannot use \( \infty \) edges.
  So  \( \text{cap}(A, B) = |L \cap B| + |R \cap A| \)

- **Key property \#2**: integral flow still corresponds to a matching
  - By max-flow min-cut, \( \text{cap}(A, B) < |L| \).

- Choose \( S = L \cap A \).
  - Since min cut can't use \( \infty \) edges: \( N(S) \subseteq R \cap A \).
  - \( |N(S)| \leq |R \cap A| = \text{cap}(A, B) - |L \cap B| < |L| - |L \cap B| = |S|. \)

\[ \begin{align*}
  G' \\
  S = \{2, 4, 5\} \\
  L \cap B = \{1, 3\} \\
  R \cap A = \{2', 5'\} \\
  N(S) = \{2', 5'\}
\end{align*} \]
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 (not covered in class): $O(m n^{1/2})$.

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]
- Recently: better algorithms for dense graphs, e.g. $O(n^{2.38})$ [Harvey, 2006]
A bipartite graph is k-regular if $|L|=|R|$ and every vertex (regardless of which side it is on) has exactly k neighbors.

Prove or disprove: every k-regular bipartite graph has a perfect matching.
7.6 Disjoint Paths

Application of Max Flow With $C=1$
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.
Disjoint path 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 unit capacity to every edge.

**Theorem.** Max number edge-disjoint s-t paths equals max flow value.
Max flow formulation: assign unit capacity to every edge.

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

Pf. \( \leq \)

- Suppose there are \( k \) edge-disjoint paths \( P_1, \ldots, P_k \).
- 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 \( k \).
Max flow formulation: assign unit capacity to every edge.

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

**Pf.** $\geq$

- Suppose max flow value is $k$.
- **Integrality** theorem $\Rightarrow$ 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
- Produces $k$ (not necessarily simple) edge-disjoint paths.

---

can eliminate cycles to get simple paths if desired
Network connectivity problem. 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 each $s$-$t$ paths uses at least one edge in $F$. (That is, removing $F$ would make $t$ unreachable from $s$.)
Edge-Disjoint Paths and Network Connectivity

**Theorem.** [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. ≤**
- Suppose the removal of $F \subseteq E$ disconnects t from s, and $|F| = k$.
- All s-t paths use at least one edge of F. Hence, the number of edge-disjoint paths is at most k. ▪
Theorem. [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. ≥

- Suppose max number of edge-disjoint paths is k.
- Then max flow value is k.
- Max-flow min-cut ⇒ cut (A, B) of capacity k.
- Let F be set of edges going from A to B.
- |F| = k and disconnects t from s. •