Learning Goals

- Reduction from edge-disjoint paths to integer network flow
- Removal of cycles from flows

Directed Edge-Disjoint Paths

Definition

Two paths are *edge-disjoint* if they share no common edge. A set of paths are edge-disjoint if any two are edge-disjoint.

Directed Edge-Disjoint Paths

Definition

Two paths are *edge-disjoint* if they share no common edge. A set of paths are edge-disjoint if any two are edge-disjoint.

Problem statement:

- Input: a directed graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.

Directed Edge-Disjoint Paths

Definition

Two paths are *edge-disjoint* if they share no common edge. A set of paths are edge-disjoint if any two are edge-disjoint.

Problem statement:

- Input: a directed graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.

Natural idea: Reduce it to a network flow problem.
Let s be the source, t the sink; for all $e \in E$, let c_e be 1.

Directed Edge-Disjoint Paths

Definition

Two paths are *edge-disjoint* if they share no common edge. A set of paths are edge-disjoint if any two are edge-disjoint.

Problem statement:

- Input: a directed graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.

Natural idea: Reduce it to a network flow problem.
Let s be the source, t the sink; for all $e \in E$, let c_e be 1.

Theorem

The value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.

Directed Edge-Disjoint Paths

Definition

Two paths are *edge-disjoint* if they share no common edge. A set of paths are edge-disjoint if any two are edge-disjoint.

Problem statement:

- Input: a directed graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.

Natural idea: Reduce it to a network flow problem.
Let s be the source, t the sink; for all $e \in E$, let c_e be 1.

Theorem

The value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.

Running time: Ford-Fulkerson takes time $O(mn)$.

Review of last lecture

- Hall's theorem: a bipartite graph has no complete matching if and only if there is a set of nodes on the left hand side having strictly fewer neighbors than its size.
- Edge-disjoint paths from s to t in a directed graph
- Reduction from edge-disjoint paths to flow: add capacity 1 on all edges; the value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.
  - Disjoint paths $\Rightarrow$ flow ✓
  - Integer-valued Flow $\Rightarrow$ disjoint paths: to be proved.

Proof of theorem

Theorem

The value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.

Proof of theorem

Theorem

The value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.

Proof.

1. For every set S of edge-disjoint paths, there is a flow f with $|f| = |S|$.

Proof of theorem

Theorem

The value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.

Proof.

1. For every set S of edge-disjoint paths, there is a flow f with $|f| = |S|$.
2. For every integer-valued flow f, there is a set S of edge-disjoint paths with $|S| = |f|$.

Proof of theorem

Theorem

The value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.

Proof.

1. For every set S of edge-disjoint paths, there is a flow f with $|f| = |S|$.
2. For every integer-valued flow f, there is a set S of edge-disjoint paths with $|S| = |f|$.

Lemma

Given any flow f, it takes polynomial time to find another flow f' without cycle such that $|f'| = |f|$.

Proof of theorem

Theorem

The value of the maximum flow is equal to the maximum number of edge-disjoint paths in G.

Proof.

1. For every set S of edge-disjoint paths, there is a flow f with $|f| = |S|$.
2. For every integer-valued flow f, there is a set S of edge-disjoint paths with $|S| = |f|$.

Lemma

Given any flow f, it takes polynomial time to find another flow f' without cycle such that $|f'| = |f|$.

Corollary

In every flow network there is a max flow with no cycles.

Removing cycles from a flow

Lemma

Given any flow f, it takes polynomial time to find another flow f' without cycle such that $|f'| = |f|$.

Removing cycles from a flow

Lemma

Given any flow f, it takes polynomial time to find another flow f' without cycle such that $|f'| = |f|$.

Proof sketch:

- Use BFS to find a simple cycle in f. Find the edge e in the cycle carrying the smallest flow, reduce the flow along the cycle by $f(e)$. Then repeat.

Removing cycles from a flow

Lemma

Given any flow f, it takes polynomial time to find another flow f' without cycle such that $|f'| = |f|$.

Proof sketch:

- Use BFS to find a simple cycle in f. Find the edge e in the cycle carrying the smallest flow, reduce the flow along the cycle by $f(e)$. Then repeat.
- What remains after the operation remains a flow with the same value as before.

Removing cycles from a flow

Lemma

Given any flow f, it takes polynomial time to find another flow f' without cycle such that $|f'| = |f|$.

Proof sketch:

- Use BFS to find a simple cycle in f. Find the edge e in the cycle carrying the smallest flow, reduce the flow along the cycle by $f(e)$. Then repeat.
- What remains after the operation remains a flow with the same value as before.
- The number of edges carrying positive flow strictly decreases after each iteration, so this can go on at most $O(m)$ rounds.

Extension: Undirected Edge-Disjoint Paths

- Input: an undirected graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.

Extension: Undirected Edge-Disjoint Paths

- Input: an undirected graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.
- Natural idea: Construct directed graph G' by replacing each edge $\{u, v\} \in E$ by a pair of edges (u, v) and (v, u) in G', and solve the edge-disjoint path problem for G'.

Extension: Undirected Edge-Disjoint Paths

- Input: an undirected graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.
- Natural idea: Construct directed graph G' by replacing each edge $\{u, v\} \in E$ by a pair of edges (u, v) and (v, u) in G', and solve the edge-disjoint path problem for G'.
- Caveat: What if (u, v) and (v, u) are used by two different paths in G'? Two such paths are not edge-disjoint in G!

Extension: Undirected Edge-Disjoint Paths

- Input: an undirected graph $G = (V, E)$. Two nodes $s, t \in V$.
- Output: the maximum number of edge-disjoint paths from s to t.
- Natural idea: Construct directed graph G' by replacing each edge $\{u, v\} \in E$ by a pair of edges (u, v) and (v, u) in G', and solve the edge-disjoint path problem for G'.
- Caveat: What if (u, v) and (v, u) are used by two different paths in G'? Two such paths are not edge-disjoint in G!
- Solution: This can happen only if the flow in G' has a 2-cycle. This cannot arise if we remove all cycles from the flow.

Menger's Theorem

The following theorem immediately follows the discussion above:

Theorem (Menger's theorem)

In a graph, the smallest number of edges whose removal disconnects node s from t is equal to the largest number of edge-disjoint paths from s to t.

Exercise: Prove the version of the theorem for vertex disjoint paths, i.e., the smallest number of vertices whose removal disconnects node s from t is equal to the largest number of vertex-disjoint paths from s to t.