Menger’s Theorem

Vassos Hadzilacos

(This document describes optional material for this course.) In our discussion of the problem of finding a maximum cardinality set of edge-disjoint \( s \to t \) paths in a digraph we developed results that we can leverage to prove Menger’s Theorem, a well-known result in graph theory. It is one of those max-this-equals-min-those results, like max-flow-equals-min-cut and bipartite-max-matching-equals-min-vertex-cover. All these results are instances of a general phenomenon known as “linear programming duality”. I didn’t have time to discuss Menger’s Theorem in class, so here it is, for those who are interested.

**Menger’s Theorem.** Let \( G = (V,E) \) be a digraph and \( s,t \in V \). The maximum number of edge-disjoint \( s \to t \) paths in \( G \) is equal to the minimum number of edges whose removal from \( G \) disconnects \( t \) from \( s \).

**Proof.** Let \( P \) be any set of edge-disjoint \( s \to t \) paths, and \( D \) be any set of edges whose removal from \( G \) disconnects \( t \) from \( s \). By the pigeonhole principle, \(|P| \leq |D|\): Every path in \( P \) (pigeon) must use an edge in \( D \) (pigeonhole); otherwise, the removal of \( D \) from \( G \) does not disconnect \( t \) from \( s \). So, there is a function \( \phi : P \to D \) such that path \( p \in P \) uses edge \( \phi(p) \in D \). Since \( P \) is edge-disjoint, \( \phi \) must be one-to-one. So, by the pigeonhole principle \(|P| \leq |D|\). This immediately implies the following:

**Fact.** If \(|P| = |D|\) then (a) \( P \) is a maximum cardinality set of edge-disjoint \( s \to t \) paths, and (b) \( D \) is a minimum cardinality set of edges whose removal disconnects \( t \) from \( s \).

Let \( P \) be a maximum cardinality set of edge-disjoint \( s \to t \) paths. Let \( F \) be the flow network obtained from \( G \) by removing all edges into \( s \) and from \( t \), and assigning capacity 1 to every edge. Let \( f \) be a maximum flow of \( F \), \((S,T)\) be a minimum \((s,t)\)-cut of \( F \), and \( D = \text{out}(S) \cap \text{in}(T) \); i.e., \( D \) is the set of edges that cross the cut from \( S \) to \( T \). By definition, the removal of \( D \) from the graph of \( F \) disconnects \( t \) from \( s \), and so the removal of \( D \) from \( G \) also disconnects \( t \) from \( s \) (why?). We have:

\[
|P| = V(f) \quad \text{[proved in class in discussion of max edge-disjoint path problem]}
= c(S,T) \quad \text{[by max-flow-min-cut]}
= |D| \quad \text{[by definition of \( D \) and the fact that all edges have capacity 1]}
\]

By part (b) of the above Fact, \( D \) is a minimum cardinality set of edges whose removal from \( G \) disconnects \( t \) from \( s \).

So, we proved that the maximum number of edge-disjoint \( s \to t \) paths in \( G \) is equal to the minimum number of edges whose deletion from \( G \) disconnects \( t \) from \( s \). \( \square \)

This proof of Menger’s theorem immediately suggests an algorithm that, given a directed graph \( G \) and nodes \( s,t \), finds a minimum cardinality set of edges whose deletion from \( G \) disconnects \( t \) from \( s \):

1. Construct \( F \) from \( G \), \( s \), and \( t \)
2. Find a maximum flow \( f \) of \( F \)
3. Using \( f \), find a minimum cut \((S,T)\) of \( F \)
4. Return the set of edges \( \text{out}(S) \cap \text{in}(T) \)

This takes \( O(mn) \) time, where \( m \) is the number of edges of \( G \) and \( n \) is the number of nodes of \( G \).