Time Spent: 1 hour

Collaborators and Resources: Nathaniel MacArthur-Warner and I discussed what max flow says about the graph

---

Problem 2

In this problem we're given the definition of edge connectivity and our task is to find a way to compute edge connectivity (defined as the minimum number k of edges that must be removed to disconnect G) by running a maximum-flow algorithm on a collection of flow networks, each having O($|V|$) vertices and O($|E|$) edges. The way we conceptualized this was to say that we look at each path from s to t_i where $t_i \neq s$ and then we run a max-flow algorithm on each case ...

1. Given an undirected graph G = (V,E)
2. Initialize an empty array A
3. For each edge e = (s,t) ∈ E
   (a) Remove e
   (b) Create two directed edges $e_1 = (s, t)$ and $e_2 = (t, s)$ and set each of their capacities to 1
4. Denote one of the vertices s to be our starting point
5. For each vertex $t_i \in V$ such that $t_i \neq s$
   (a) Run Ford-Fulkerson to find the maximum flow f* between s and t_i
   (b) Add f* to A
6. Return min(A)

Claim 1. The algorithm works correctly and returns the minimum number k of edges that must be removed to disconnect G (the edge connectivity)

Proof. Divide the nodes of G into two sets, A and B where s $\in$ A and s $\notin$ B. Select any vertex in B and call it t. The algorithm I have described simply determines the maximum flow between s and t using the Ford-Fulkerson algorithm. Now, having computed the maximum value f* of an s-t flow we know that this is equal to the minimum capacity of an s-t cut (K-T 7.13). Furthermore, since all the capacities are 1 we know that the minimum capacity of an s-t cut is equal to the minimum number of edges needed to disconnect s from t. Taking the minimum out of all values of f* is going to give us the way to disconnect the graph removing the fewest number of edges possible. So we know that the algorithm works. □

Claim 2. The algorithm runs in $O(|V| \cdot |E|^2)$ time.

Proof. Because the Ford-Fulkerson algorithm runs in O($|E|$f*) time where m is the number of edges and f* is the maximum flow found, we have one part of our run-time. Because the Ford-Fulkerson algorithm is applied to every pair of vertices (s, t_i), it's going to run a maximum of $|V|$ times since there are V-1 choices for t_i. The maximum flow f* is always upper bounded by the number of edges $|E|$ and so we know that in total the algorithm will run in $O(|V| \cdot |E|^2)$ time. Constructing the directed graph also takes O($|E|$) time, however as the graph gets larger we will see that this initial step becomes inconsequential in terms of running time. □