1 Unreliable Nodes

Given an undirected graph $G = (V, E)$, where some nodes are non-reliable (in N) and some are reliable (in R), and two reliable nodes s and t, find the minimum number of edges + non-reliable nodes to remove to disconnect s and t:

1.1 Solution

First, define a flow network G'. For each undirected edge in G, create two directed edges in G', one for each direction.

Once this is done, modify G' such that every node n in N becomes two nodes, here called n_a and n_b. Create an edge (n_a, n_b). Redirect all of the incoming edges of n to n_a and create edges for all of the outgoing edges of n which go from n_b to the destination of the corresponding edge.

Every edge in G should receive a capacity of 1 unit.

The purpose of this graph modification is to recast the problem as a minimum cut problem. For every edge $v_a \rightarrow v_b$ which is cut, it represents removing the vertex v in N. For every other edge which is cut, it represents simply removing the edge in E. Since all edge weights are 1, removing an edge counts the same as removing a non-reliable vertex. The algorithm will not be able to "remove" reliable edges because they are represented as a single vertex in G'.

To finish, run a max-flow algorithm on G'. This will yield the same result as the value of the minimum flow due to the maxflow-mincut theorem.

1.2 Time Complexity

Graph creation: Every non-reliable node will be traversed once, and each edge converted into two exactly once and redirected at most once, so the graph generation time is $O(|V| + |E|)$.

Graph size: The new graph will have $O(|V|)$ vertices because each vertex is turned into at most 2 vertices. The graph will have $O(|N| + |E|)$ edges because each edge is turned into 2 edges, and then $|N|$ edges added.

Graph algorithm: Maximum flows can be computed in $O(VE)$ time. For our flow network, this translates to $O(|V|(|N| + |E|))$ time.