# HW2 - Problem 3.2

Consider the following problem, which we'll call the "Sender-Receiver Problem": Given a graph $G = (V, E)$ with two disjoint subsets of $V$ named $S$ (senders) and $R$ (receivers), find a minimum cost subgraph of $G$ that has a path connecting each receiver to any sender.

When $S \cup R = V$, we'll call this the "Sender-Receiver I Problem". When $S \cup R \neq V$, we'll call this the "Sender-Receiver II Problem".

Show that the Sender-Receiver I Problem is in P, that the Sender-Receiver II Problem is in NP-hard, and give a factor 2 approximation algorithm to the Sender-Receiver II Problem.

## 1 Show that the Sender-Receiver I Problem is in P

We will give a polynomial time algorithm to solve this problem (based on the hint provided in Vazirani):

*Sender-Receiver Algorithm:* Add a new vertex which is connected to each sender by a zero cost edge. Consider the new vertex and all receivers as required vertices in $T$ and the remaining vertices as Steiner, and find an approximation to the minimum cost Steiner tree using the "MST-based algorithm":

*MST-Based Algorithm:* For every pair of vertices $(u, v)$ in the set $T$ of required vertices, find the shortest path between $u$ and $v$. Create a graph $G'$ containing all terminal vertices with edges representing the shortest paths between the vertices on the original graph. Find the MST of $G'$. Return the union of all edges in $G$ that correspond to an edge taken by the MST of $G'$.

### The algorithm gives a feasible solution

Because the only way to get to $f$ is to traverse through a sender, everything connected to $f$ must also be connected to a sender. We know that all vertices in $R$ are connected to $f$, because all these vertices are in the MST created by the algorithm. Thus, every receiver is connected to a sender, making the algorithm's output feasible.

### The algorithm gives an optimal solution (to Sender-Receiver I)

Because we found the MST of $G'$, we know that there's no other set of edges that could connect the corresponding vertices in $G$ at less cost (unless other vertices were added or removed). Because every vertex in this problem is either a sender or receiver, there are also no additional vertices that could added, and there are no current vertices that can be dropped to reduce costs.

### The algorithm runs in polynomial time

We know that computing an MST runs in polynomial time, as does finding the shortest paths between vertices (using dynamic programming) and adding an additional vertex and corresponding edges.

The algorithm finds an optimal solution to the Sender-Receiver I problem in polynomial time. Sender-Receiver I is in P.

## 2 Show that the Sender-Receiver II Problem is in NP-hard

To show this, we will show that there is a polynomial time reduction from the Steiner tree problem (which is NP-hard) to the Sender-Receiver problem.

To perform the reduction, take the required vertices $T$ from the Steiner tree problem and turn one of them into a "Sender" vertex. Turn all other vertices in $T$ into "Receiver" vertices. All other vertices (those not in $T$) remain unchanged.

The Steiner tree problem already was trying to connect all terminal nodes to each other at minimum cost. Changing the problem so that one of the terminal nodes (the sender) *must* connect to all the other terminal nodes (the receivers) does not further constrain the solution space of original problem. It is simply repeating a constraint that was already in place. Additionally, all the actions taken to turn the problem from a Steiner tree into a Sender-Receiver II problem (labeling the terminals sender/receiver) can be performed in polynomial time. Thus, the Steiner tree problem can be reduced to the Sender-Receiver II problem in polynomial time.

Because we already know that the Steiner tree problem is NP-hard, and we know that it can be reduced to the Sender-Receiver II problem in polynomial time, the Sender-Receiver II problem must also be NP-hard. (If the Sender-Receiver II problem was not NP-hard but was instead, for example, in P, we could convert any Steiner tree problem into a Sender-Receiver II problem, and then use our Sender-Receiver II algorithm to solve the Steiner tree problem in polynomial time, which we know is not possible.)

## 3 Give a factor 2 approximation algorithm for the Sender-Receiver II Problem

Our algorithm is already given in the first part of this problem (the "Sender-Receiver Algorithm"). To show that it is a 2-approximation, we use Vazirani's Theorem 3.3, which states that the cost of an MST on the required vertices is within 2*OPT of the Steiner tree problem. From the first part of this problem, we've shown how to convert a Sender-Receiver problem into a Steiner tree problem, so the optimal solution to the Steiner tree problem is the same as the optimal solution to the corresponding Sender-Receiver problem. Because we use the union of the paths given by the MST shortest paths, our algorithm output is less than MST, which is less that 2*OPT. Thus, our algorithm is a 2-approximation of the Sender-Receiver II problem.