1 Problem Description

Prove that Steiner Tree problem is NP-complete.

2 Solution

We prove that the Steiner Tree problem is NP-complete via a reduction of the Set Cover problem to the Steiner Tree problem.

Given an instance of Set Cover with universe set U of size n, a set $M \subseteq 2^U$ of size m, a target size k, and an algorithm $F(T, G, n)$ for determining if a Steiner Tree with n edges exists in G where T is the set of terminal nodes, which runs in polynomial time in the size of the input graph, the Set Cover instance can be solved as follows:

Construct a graph G, which has 3 sets of vertices: one set V_1 corresponding to the elements of U, one set V_2 corresponding to the elements of M, and one final set of one vertex v_f.

Place an edge between two vertices u and v exactly when:

$$u \in V_1 \wedge v \in V_2 \wedge u \in v$$

or

$$u \in V_2 \wedge v = v_f$$

Run the algorithm

$$F(V_1 \cup \{v_f\}, G, k + n)$$

and return the result.

Graph construction takes $O(n^2 m)$, since in the worst case, for each pair of vertices considered for edge adding, we do a linear search through the corresponding subset of U, which is size $O(n)$. The graph has $1 + m + n$ vertices, and $O(nm)$ edges. Thus, the whole algorithm will run in polynomial time.

Because the Set Cover problem is given as an NP-complete problem, and given a polynomial time algorithm to solve Steiner Tree, we can solve Set Cover in polynomial time, Steiner Tree is also an NP-complete problem.

3 Proof of Correctness of Reduction

Claim 1: given an instance of Set Cover, if there is a solution of size k, there is a solution of $k + n$ edges to the corresponding constructed Steiner Tree instance.

Over the elements of the subsets in this solution, each element of U will be included at least once. These live in the graph as $V_U \subseteq V_2$. Therefore, each vertex in V_1 will have a connection to at least one of the k V_U vertices. A valid Steiner tree is the terminal nodes V_1, the selected subset nodes V_2, the final vertex v_f, exactly one edge connected to each of V_1 where the other end is in V_2, and $(v, v_f) \forall v \in V_2$. The graph connects all the terminal nodes and has $k+n$ edges and is thus a $k + n$ Steiner tree.

Claim 2: Given an instance of Set Cover, if there is a solution with $k + n$ edges to the corresponding constructed Steiner Tree instance, there is a solution to the set cover instance of size k.

First, note that at least n edges of this solution tree must reside in between V_1 and V_2, because each vertex in V_1 must be connected to the rest, and there are no direct edges between the vertices of V_1.

If there are more than n edges in this section of the tree, we see that some edges redundantly cover a V_1 vertex (type-1), and the rest connect a V_2 vertex to v_f (type-2). For any type-1, I will argue that it can either be disregarded or swapped out for a type-2. Since our solution is a tree, removing the edge splits it into two sub-trees. These subtrees can be reconnected by using one of the unused type-2 edges. There is guaranteed to be at least one of these because if there weren't, the tree would have had a loop and not have been a tree in the first place. This procedure can be continued until the current Steiner tree has k type-2 edges.

At this point, a set cover of size k can be easily shown to exist. Since there are no redundant type-1 edges, each of the V_2 vertices connected to V_1 vertices in the tree must be connected to v_f through the type-2 edges. There are k type-2 edges, so there must be k V_2 vertices in the Steiner tree. Between these, they are guaranteed to include all of the elements in U.