NP-Completeness of the Steiner Tree Problem

Simone Bianco

Sapienza Università di Roma, Italy

January 6, 2025

Abstract

The Steiner Tree (ST) problem is an optimization problem in graph theory with applications in network design and computational biology. This essay proves the NP-Completeness for the decision version of the problem in undirected graphs by showing membership in NP and constructing a polynomial-time reduction from the NP-Complete Exact Cover by 3-Sets (X3C) problem

Contents

1 The Steiner Tree problem

The Steiner Tree problem is an optimization problem that arises in various fields, including network design, computational biology, and VLSI design. While there are many formulations of this problem, they all require to find an optimal interconnection for a given set of objects, called *terminals*, for a predefined objective function, which usually asks to minimize the length of such interconnections. This problem has many applications, especially when planning a connecting structure among different terminal points to minimize overall cost or distance. [**shortest˙network**]

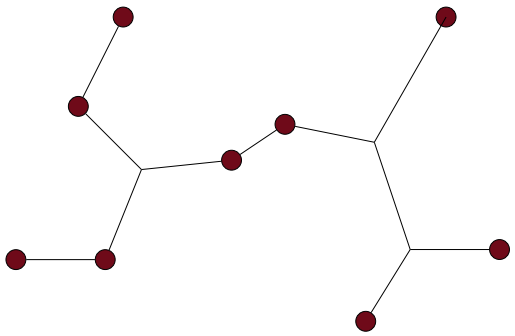

Figure 1: A minimal length Steiner Tree

The usual formulation of this problem is achieved through graphs, where each node of the graph acts as an interconnection point. The terminals are also represented as nodes of the graph. A Steiner Tree must cover all the designed terminal nodes.

Definition 1. *Given a graph G and a subset of nodes $S \subseteq V(G)$, called terminal set, a Steiner Tree (ST) for S in G is a sub-tree T such that covers all the nodes of S, i.e. $S \subseteq V(T)$.*

In particular, we notice that, by definition, a Steiner Tree can also contain nodes that aren't part of the terminal set. These nodes are called *Steiner points*. The Steiner Tree problem asks to find an ST for a given set of terminal nodes with the lowest possible number of edges.

Problem 1. *Given a graph G and a subset of nodes $S \subseteq V(G)$, find a Steiner Tree with the minimum number of edges.*

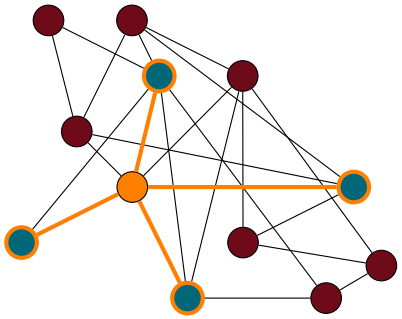

Figure 2: Minimum Steiner Tree for the given terminals (blue nodes)

An interesting thing to notice is that this formulation of the Steiner Tree problem acts as a generalization two well-known combinatorial optimization problems: the *Shortest Path problem* and the *Minimum Spanning Tree problem*. In particular, finding the minimum Steiner Tree on two terminals is equivalent to finding the shortest path between them. When all vertices in the graph are terminals, instead, the problem becomes equivalent to finding the minimum spanning tree of the graph.

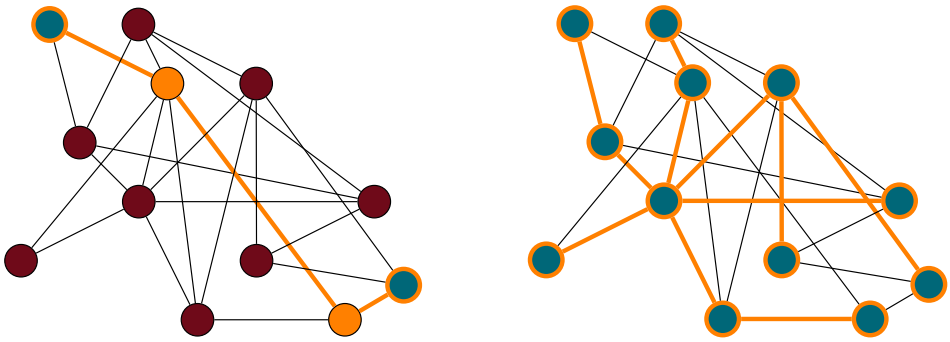

Figure 3: The ST problem on two nodes (left) and all nodes (right)

While both of these problems can be solved in polynomial time, the Steiner Tree problem is significantly more challenging. In fact, the Steiner Tree problem results to be "intractable" unless $\mathsf{P} = \mathsf{NP}$. This is due to the NP-Completeness of the decision version of this problem, which implies that the standard optimization variant is NP-Hard [**np˙complete**]. The decision version of the problem is defined as follows.

Problem 2. *Given a graph G, a subset of nodes $S \subseteq V(G)$ and an integer $k > 0$, determine if there is a Steiner Tree for S with at most k edges.*

2 NP-Completeness of Steiner Tree

It's easy to see that this decision problem is a member of NP, since the requested Steiner Tree can act as a witness. In particular, we can verify a witness T through this simple certification procedure, whose steps can all be done in time $O(|V(G)| + |E(G)|)$:

1. Check that T is a subgraph of G, i.e. that $V(T) \subseteq V(G)$ and $E(T) \subseteq E(G)$.
2. Check that T is a tree, i.e. that it contains no cycles
3. Check that T covers all the terminals, i.e. that $S \subseteq V(T)$
4. Check that T has at most k edges, i.e. that $|E(T)| \leq k$

To show the NP-Hardness of the Steiner Tree problem, we have to find an NP-Hard problem that can be reduced in polynomial time to it. Since the concept of Minimal Steiner Tree reduces to the concept of "terminal coverage", it's easy to see that the former should be somehow related to the *Set Coverage problem.*

Given an universe set $U = \{x_1, \ldots, x_n\}$ and a collection of subsets $\mathcal{C} = \{C_1, \ldots, C_m\}$ where $C_i \subseteq U$, a Set Cover of U over C is a subcollection $\mathcal{C}^* \subseteq \mathcal{C}$ that covers U, that is $U = \bigcup_{C_i \in \mathcal{C}^*} C_i$. The Set Cover (SC) problem asks to find the minimum cardinality Set Cover.

$$U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$$

$$C_1 = \{x_1, x_2, x_4, x_7\} \quad C_2 = \{x_1, x_3, x_5, x_6\} \quad C_3 = \{x_3, x_5, x_7\}$$

Figure 4: Example of a Set Cover problem instance

Similarly to what we deed for the ST problem, the decision version of the Set Coverage problem would ask to decide if there is a Set Cover of at most k subsets.

Problem 3. *Given a set $U = \{x_1, \ldots, x_n\}$, a collection of subsets $\mathcal{C} = \{C_1, \ldots, C_m\}$ where $C_i \subseteq U$ and an integer $k > 0$, determine if there is a cover of U with at most k subsets.*

This decision problem is NP-Complete, where the NP-Hardness comes a reduction from the Vertex Cover problem (VC).

After defining the two decision problems, we're ready to give a reduction from Set Cover to Steiner Tree. Given an instance $\langle U, \mathcal{C}, k \rangle$ of SC, the reduction proceeds as follows:

1. For every subset $C_i \in \mathcal{C}$, we add a node u_i in G.
2. For every element $x_j \in U$, we add a node v_i in G. For every $i \in [m]$, if $x_j \in C_i$ then we add the edge (u_i, v_j) in G.
3. We add a special node z and an edge (z, u_i) in G for all $i \in [m]$.
4. Let $S = \{z, x_1, \ldots, x_{3q}\}$.
5. Return $\langle G, S, 2n \rangle$.

This procedure can easily be computed in polynomial time. The reduction is based on the idea that in the generated graph the required Steiner Tree will be composed of the edges incident to all the nodes corresponding th the original cover. However, this reduction is actually invalid: in some cases we get a terminal cover that contains cycles, meaning that it isn't a tree.

$$U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$
$$C_1 = \{x_1, x_2, x_3, x_5, x_5\} \quad C_2 = \{x_1, x_3, x_5, x_6\}$$
$$C_3 = \{x_3, x_5\} \quad C_4 = \{x_1, x_4, x_6\}$$

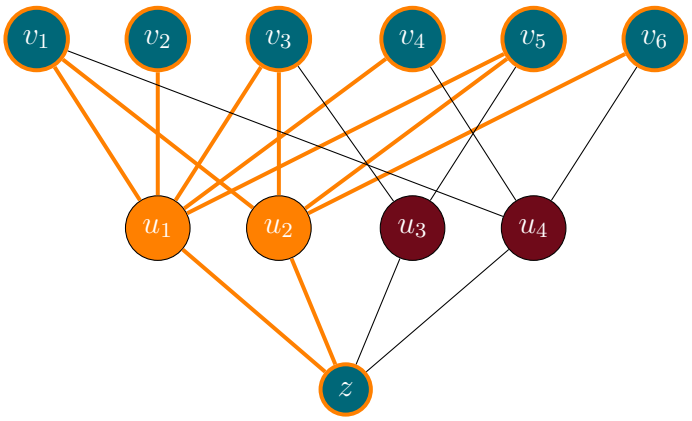

Figure 5: Example of how the reduction from SC to ST fails.

To fix this issue, we can use a variant of the Set Cover problem, that being the *Exact Cover problem* (XC). In this variant, we require that the sub-collection is not only a cover but that it also is pairwise disjoint. In other words, the cover must be a partition.

Problem 4. *Given a set $U = \{x_1, \ldots, x_n\}$, a collection of subsets $\mathcal{C} = \{C_1, \ldots, C_m\}$ where $C_i \subseteq U$ and an integer $k > 0$, determine if there is a partition U with at most k subsets.*

Through this additional constraint, the reduction is guaranteed to generate an acyclic graph, i.e. a tree. However, we still have an issue: the number of edges of the output tree is too variable. For instance, consider the following two cases:

- Given $|U| = n$ with $\mathcal{C} = \{\{x_1, \ldots, x_{n-1}\}, \{x_n\}\}$ and $k = n$ then T will have $n + 2$ edges
- Given $|U| = n$ with $\mathcal{C} = \{\{x_1\}, \ldots, \{x_n\}\}$ and $k = n$ then T will have $2n$ edges

This means that we cannot find a "good" common value k' for the output triple $\langle G, S, k' \rangle$ – in fact, this was an issue even in the Set Cover case.

To fix this final issue, we use a stronger variant of the exact cover problem: the *Exact Cover by 3-Sets* (X3C) problem. This new problem requires that the universe set has $3q$ elements, for some integer $q > 0$, and that each of the subsets inside the collection has 3 elements.

We notice that since $|U| = 3q$, any exact cover will be made of exactly q subsets, meaning that the optimization version of the problem becomes a simple search problem. The decision version of this problem, defined as follows, is NP-Complete, where the hardness is obtained through a reduction from the 3D Matching problem.

Definition 2. *Given a set $U = \{x_1, \ldots, x_{3q}\}$ and a collection of subsets $\mathcal{C} = \{C_1, \ldots, C_m\}$ such that $C_i \subseteq U$ and $|C_i| = 3$, determine if there is a sub-collection $\mathcal{C}^* \subseteq \mathcal{C}$ that is a partition of U.*

To better understand how the procedure works, the following X3C instance is transformed into the graph below it:

$$U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

$$C_1 = \{x_1, x_2, x_4\} \quad C_2 = \{x_1, x_3, x_5\} \quad C_3 = \{x_3, x_5, x_6\} \quad C_4 = \{x_1, x_4, x_6\}$$

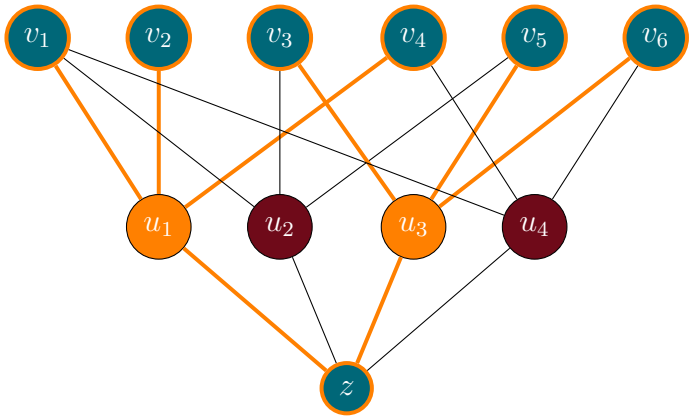

Figure 6: Example of how the reduction from X3C to ST works.

The correctness of the reduction follows from the claim below, concluding the proof of the NP-Completeness of the Steiner Tree problem.

Claim 1. *U has an Exact Cover by 3-Sets over $\mathcal{C}$ iff G has a Steiner Tree for S with $4q$ edges.*

Proof. Suppose that there is an X3C $\mathcal{C}^* \subseteq \mathcal{C}$ of U. Since $\mathsf{U} = 3q$ and each subset inside $\mathcal{C}^*$ has 3 elements, we get that $|\mathcal{C}|^* = q$. Without loss of generality, assume that $\mathcal{C}^* = \{C_1, \ldots, C_q\}$. Consider the tree T made of the edges that have $u_1, \ldots, u_q$ as endpoints. Since each $u_1, \ldots, u_q$ has degree 4 and $C_1, \ldots, C_q$ are pairwise disjoint, T is a tree with $4q$ edges. Moreover, since $\mathcal{C}^*$ is a cover of U, each node of U is covered by T. Since the node z is also covered, we conclude that T is a Steiner Tree for S with $4q$ edges.

Vice versa, suppose that there is a Steiner Tree T' for S with $4q$ edges. By construction of the graph G, in order for T to be a Steiner Tree it must contain $u_1, \ldots, u_t$. Moreover, since T has at most $4q$ edges and each $u_1, \ldots, u_t$ has degree 4, it must hold that $t \leq q$. By way of contradiction, suppose that $t < q$. Since T is a tree, each terminal v_j is reached by only one u_i. Hence, if $t < q$ there must be at least one terminal that isn't covered by T, which is a contradiction. Thus, we have that $t = q$ and that $C_1, \ldots, C_t$ is an X3C of U. □