Project: Exploring the Steiner Tree Problem

Overview

This project involves researching the Steiner Tree NP-Complete Problem, exploring various approximation solutions for it within polynomial time. The goal is to develop an educational application that offers insights into the best-performing approximation algorithms, runtimes, proofs of NP-completeness, and approximation guarantees. The application also provides a platform to visually compare approximation algorithms to exponential algorithms.

Problem Definition

The graphical Steiner tree problem involves finding a tree of minimum weight in an undirected graph with non-negative edge weights. The tree must contain a subset of vertices (terminals) and minimize the total weight of its edges. There are two versions of the problem: in the optimization problem associated with Steiner trees, the task is to find a minimum-weight Steiner tree; in the decision problem the edge weights are integers and the task is to determine whether a Steiner tree exists whose total weight does not exceed a predefined natural number k. This website will focus on the optimization problem.

Proof of NP-Completeness

To prove the graphical steiner tree problem is NP, we will obeserve the decision version of the problem (i.e. finding a steiner tree with a maximum total weight less than a natural number K).

Proof of NP

It is clear that a proposed solution to the graphical steiner tree problem can be verified in polynomial time. Suppose for a graph G(V,E) and set of steiner nodes S, a proposed solution P(E) $\subset$ E is given. All one must do to prove that this solution is a valid steiner tree less than k is

1. sum the weights of each edge e $\in$ P(E) and verify that this sum is less than k. (O(E))
2. Ensure that every node in S is connected to every other node in S. (O((S+E)*S))

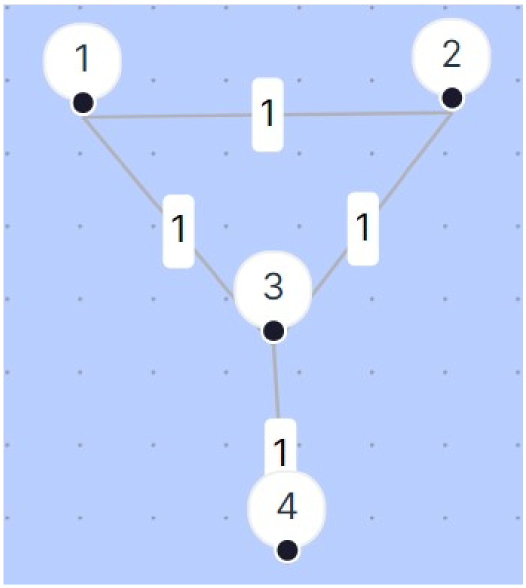
Clearly this can be checked in polynomial time.

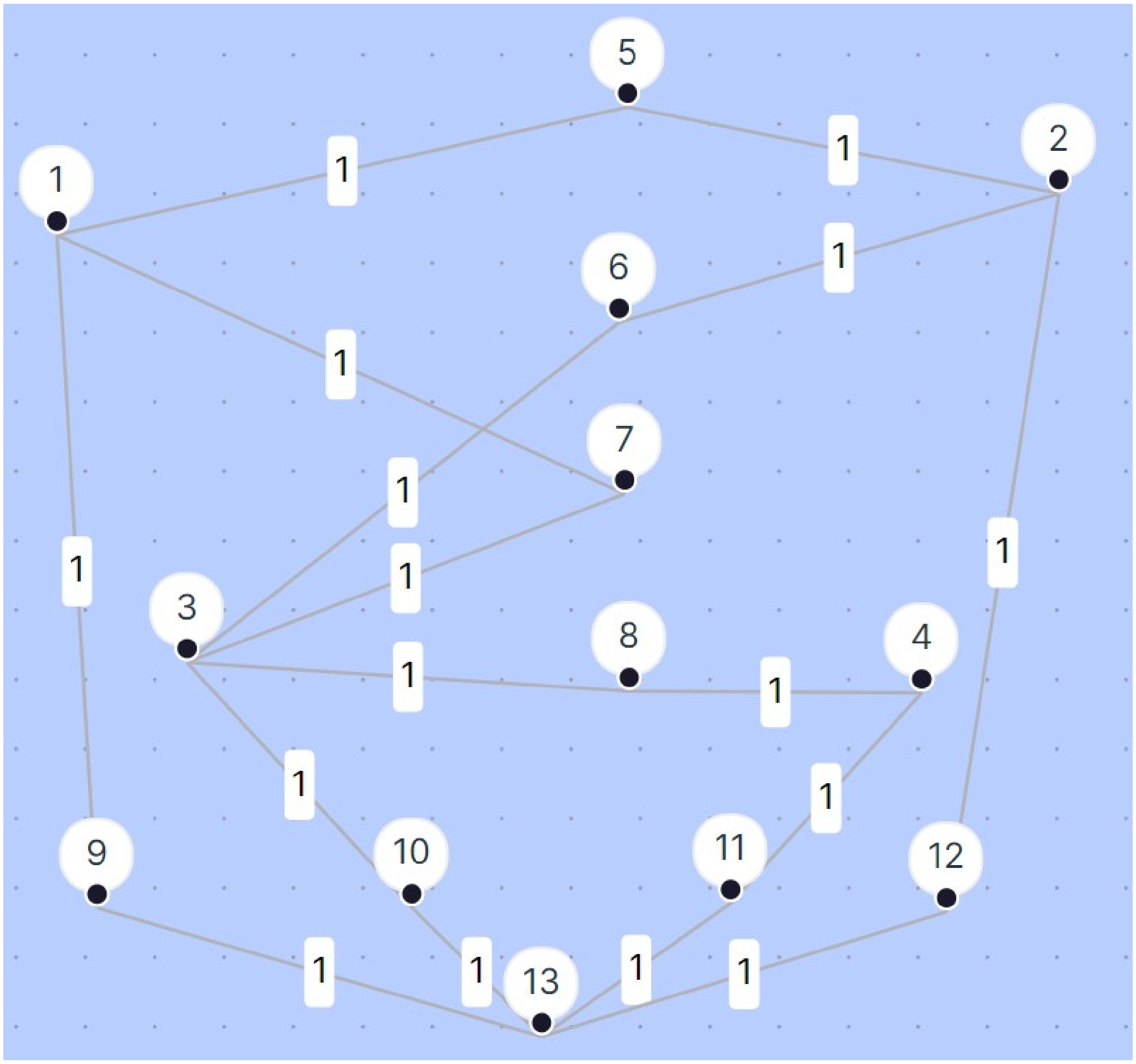
Proof of NP-Hard

We will use the vertex cover problem, which can be defined as, given an undirected graph G(V,E), finding a V' $\subset$ of V such that for all (u,v) $\in$ E, u $\in$ V' or v $\in$ V'. such that |V'| $\leq$ k. This is a very well-known NP-Complete problem. We will now polynomially reduce an arbitrary instance of the vertex cover problem I_v (G(V,E), k) into an instance of the graphical steiner tree problem I_s (H(V,E), S, k'). This will be done the following way:
Create a new graph H based on G with the following additions:

1. Insert a new intermediate node w_e in the middle of each edge in G.
2. Create a new vertex q_v for each vertex v in G and a new edge (q_v, v).
3. Create a new vertex s and add an edge from each q_v to s.
4. Set the edge weight of each edge in H to have a weight of 1.

For demonstration, the images below will show the conversion from some G to an H (edge weights of 1 do not matter for set cover):





In this example, nodes {1,2,3,4} are from the nodes of G. We will refer to this as $H(V)_v$. Nodes {5,6,7,8} are from the insertions of nodes for each edge of G. We will refer to these nodes as $H(V)_e$. Nodes {9,10,11,12} are from the creating the q_v for each vertex v in G. We will refer to these nodes as $H(V)_0$. Node 13 is node s.

Furthermore, we will define $X = \{s\} \cup H(V)_v \cup H(V)_e$ and define $k' = ||H(V)_e|| + 2(k) + ||H(V)_v - 2||$. We will let the steiner nodes S = X.

Polynomial Reduction

Using this construction of I_s (H(V,E), S, k'), we now solve this instance of the graphical steiner tree problem. If a steiner tree can be created with weight less than k', then it is true that there is a vertex cover of G of size k. This will be proven below with two theorems:

1. If there exists a vertex cover of G of size k, then there exists a graphical steiner tree of weight k' for H.

Suppose Graph G has a vertex cover of size k. Let $X' \subset H(V)_v$ be a set of k vertices that is a vertex that accomplish this vertex cover. We will construct the graphical steiner tree F (in polynomial time) with the following format:

1. Create graph H as usual
2. Add one edge from each $u \in H(V)_e$ to some $v \in H(V)_v$ to F
3. For each $v \in X'$, add two edges to F: One that connects v to q_v and one that connects q_v to s
4. Add one edge from from each $u \in H(V)_v \setminus X'$ to some $v \in H(V)_e$

Note that F will clearly be a steiner tree for the terminals $S = H(V)_e$. Furthermore, the total number of edges will be $||H(V)_e|| + 2(k) + ||H(V)_v - k||$.

2. If there exists a graphical steiner tree of weight k' for H, then there exists a vertex cover of G of size k .

Suppose X' is a graphical steiner tree of weight k' for H. Since the tree is connected, all vertices must be connected to S, including $H(V)_v$. Note that each node in $H(V)_v$ must either be connected "forward" or "backward". Forward means that the node is connected to the steiner tree through a node in $H(V)_e$. Backward means that the node is connected to the steiner tree through $H(V)_0$. Let $H(V)_b \subset H(V)_v$ be the nodes that are connected backward. Let $k'_b = ||H(V)_b||$. Consider the nodes X in G(V) that correspond to $||H(V)_b||$. Note that by the steiner tree construction, every node in $H(V)_e$ is connected to some node in $H(V)_b$, which would make the nodes X a vertex cover of G. Next, we must show that $|X| < k$. Therefore $k'_b < k$. Assume for contradiction that $k'_b > k$. $||X|| \geq ||H(V)_e|| + 2(k'_b) + ||H(V)_v - k'_b|| > ||H(V)_e|| + 2(k) + ||H(V)_v - k|| = k'$. This is a contradiction, since ||X|| is at most k by construction.

Thus by these two theorems, there exists a graphical steiner tree of weight k' for H iff there exists a vertex cover of G of size k. Therefore, we can polynomially reduce the vertex cover problem to the graphical steiner tree problem. Therefore, this shows that the graphical steiner tree problem is NP Hard.

Thus, since the graphical steiner tree problem is both NP and NP-Hard, it must also be NP-Complete.

Adapted from: https://mostafatouny.github.io/post/steiner-tree-np-complete/

Future Steps

The future steps for this project involve exploring similar problems to the graphical Steiner tree (Euclidean, rectilinear) and providing educational material on these problems.

Metric Steiner Tree Approximation Algorithms

2-approximation Algorithm

- Use an MST of terminal nodes.
- Approximation factor is bounded by 2.
- Performance Proof

Approach

Conversion from P1 to P2

1. Define P1 as the Steiner tree problem and P2 as the metric Steiner tree problem.
2. Convert the instance of the Steiner tree problem I1 to an instance of metric complete Steiner tree problem I2.
3. Ensure that OPT_P2(I2) ≤ OPT_P1(I1) by using metric closure.
4. Construct a solution S2 to the metric Steiner tree problem P2.
5. Convert the solution of the metric Steiner tree problem S2 into a solution for the regular Steiner tree problem S1.

Note: All conversions must be in polynomial time for the approach to work. If an alpha-approximation algorithm exists for P2, then an alpha-approximation algorithm also exists for P1.

Mapping and Proof

- Show that there is a mapping from P1 to P2 that takes polynomial time.
    - To convert I1 to I2, we will use the metric closure: Every c(u,v) is equal to the shortest path from u to v (shortest path algorithm). This is clearly polynomial time conversion
- Show that OPT_P2(I2) ≤ OPT_P1(I1).
    - This is clearly true, since edge costs only decrease when computing metric closure. Thus OPT(I2) must be $\leq$ OPT(I1).
- Show that there is a mapping from S2 to S1.
    - For each edge e in s2:
      If e is in I1, then add it to s1
      If not, then that means that there was a corresponding shortest path
      (since all edges are from metric closure), so add the edges that made up the shortest path instead, leaving out edges that are already in s1
    - Clearly, cost of s1 <= cost of s2, since every edge in s1 is either in s2, or is part of a path that was represented by an edge in s2, making it less than the edge in s2
    - Note that s1 may not be a tree (it may contain a cycle), but this is alright, as it still a structure that solves the problem approximately

Now Create the approximation for Metric Steiner Tree:
Assumption:
G is complete
Cost of each edge is metric: c(u,v) <= c(u,w)+c(w,v) for all triplets u,v,w

2-approximation:
Use an MST of terminal nodes
Proof that Approximation Factor is bounded by 2:
Suppose T* is the optimal steiner tree for a graph G
Duplicate every edge of T*, making an Eulerian multigraph-call this B*.
Clearly, B* has cost 2*OPT, and B* has a cycle
Since every node in B* has even edges, there exists a Eulerian tour E* = 2* OPT
Create a Hamiltonian path H from E* where we “short-cut” Steiner vertices and already visited terminals
Cost of H <= Cost of E* = 2 * OPT True by the triangle inequality
Note that H must be a spanning tree of the terminal nodes
Thus, the MST of the terminal nodes must have a cost less than or equal to the cost of H <= Cost of E* = 2 * OPT .
Thus MST of the terminal nodes <= 2 * OPT.

Simulated Annealing

- Use of simulated annealing for optimization.
- Paper
- Video

Rectilinear Steiner Tree Problem

The rectilinear Steiner tree problem is a variant of the geometric Steiner tree problem. It arises in the physical design of electronic design automation, specifically in wire routing for VLSI circuits.

Approximation Algorithm

- F. K. Hwang. "On Steiner minimal trees with rectilinear distance." SIAM Journal on Applied Mathematics, 30:104–114, 1976.

Euclidean Steiner Tree Problem

The Euclidean Steiner tree problem involves connecting N points in the plane with lines of minimum total length, forming a tree. It has applications in VLSI design for wire routing.

Applications

- VLSI Design
- Telecommunications

Note: Customize this Markdown file further based on your project's specific details and requirements.