The Prize-Collecting Steiner Forest Problem

Kiss Bendegúz
Supervisor: Király Tamás

January 2025

Introduction

Steiner Forest Problem:

- Given an undirected graph with non-negative edge costs and specific vertex pairs called demands.
- Goal: Find a forest, F connecting the demands while minimizing the edge costs.

Prize-Collecting Version (PCSF):

- Adds penalties for unmet demands.
- Objective: Minimize the sum of edge costs and penalties.

Mathematical Formulation

Given:

- Graph $G(V, E)$, edge costs $c : E \to \mathbb{R}_0^+$.
- Demands $D = \{(v_1, u_1), \dots, (v_m, u_m)\}$.
- Penalties $\pi : V \times V \to \mathbb{R}_0^+$.

Goal:

- Find forest F and subset of demands $Q \subseteq D$ to minimize:

$$\sum_{e \in F} c_e + \sum_{(i,j) \in Q} \pi_{ij}$$

A 3-Approximation Algorithm

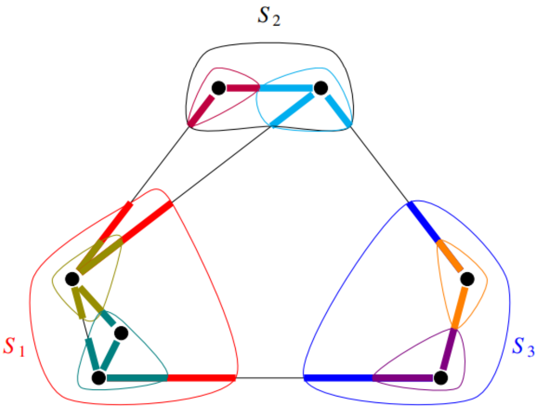
- Due to NP-hardness we deal with approximation algorithms

Steiner Forest Algorithm:

1. Start with an empty forest.
2. Add edges using static coloring:
   - Active sets color their cutting edges.
   - Merge sets when edges are fully colored.
3. Remove unnecessary edges.

PCSF Algorithm: The base structure remains the same, but we introduce a much more difficult coloring scheme called dynamic coloring to deal with the penalties.

Static coloring



Observations

- Both algorithms guarantee a 3-approximation solution.
- The running time of the algorithm is polynomial because the number of possible active sets is linear and the subroutines called during the process run in polynomial time.

Work So Far and Future Plans

- Complete the Python implementation of the 3-approximation algorithm.
- Compare with heuristic methods for performance evaluation.
- Get familiar with the 2-approximation algorithm.

References

Ahmadi, Ali, et al. *2-approximation for prize-collecting Steiner forest.* Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).

Thank you for your attention!