The Prize-Collecting Steiner Forest Problem

Kiss Bendegúz
Supervisor: Király Tamás

January 2026

Introduction

- **Work during the semester**:
    - Improving and simplifying the implementation of the approximation algorithm
    - Studying and implementing a new heuristic approach because the previous one gave mostly trivial solutions, usually minimum spanning trees
- **PCSF**: finding a forest in a graph minimizes edge costs and penalties for disconnected node pairs.
- Generalizes the Prize-Collecting Steiner Tree Problem (PCST).

Problem Definition

Given:

- Undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}^+$
- Penalty function $\pi : V \times V \to \mathbb{R}^+$
- $\pi_{ij} = 0$, if $i \geq j$

Goal: Find a forest F minimizing:

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

where Q is the set of disconnected node pairs in F.

3-Approximation Algorithm

1. Starting with each node as an active set
2. These active sets start “coloring” their outgoing edges.
3. When an edge is fully colored (the coloring duration equals its cost), we add it to the forest.
4. We also “color” node pairs: if the color reaches their penalty, they become tight. This coloring procedure is much more complicated and difficult to imagine visually.
5. Sets that cut only tight pairs are removed from active sets. We "deactivate" them.
6. Repeat until no active sets remain.
7. Finally, we keep only those edges, which we use to connect the non-tight pairs.

2-Approximation Algorithm

1. Run the 3-approximation algorithm on the original graph
2. Collect the node pairs for which we paid penalties
3. We run the algorithm again with $\pi_{ij} = 0$ for those pairs
4. Evaluate the output by the original penalties
5. Return the better of the two solutions

Application in biological networks

- We examine a disease and gain as much biological information as possible at experimental cost.
- Interactomes: graphs representing biological networks
- Nodes represent biomolecules, edges represent interactions.
- Edge costs are the experimental cost (time, substance, etc.) of checking an interaction
- Penalties can be imagined as information loss if we ignore a protein interaction.
- To find the necessary interactions, we can solve a PCSF problem with the given edge costs and penalties

New heuristic Algorithm

Three main phases

- : Preparation phase:
  - terminal nodes with at least one positive penalty
  - D matrix, containing the cost of the shortest paths between each node pair
- Clustering phase:
  - We start each cluster with a random terminal node
  - Add an u node to a cluster represented by v if $\alpha \cdot D_{u,v} \leq \pi_{u.v}$
  - Merge two clusters if their union spans a tree with less weight
- MST phase:
  - Make the metric closure of the clusters
  - Add an r root node to G and connect it to every node using an edge with ω cost
  - Find an MST in the modified graph
  - If an uv edge is in the MST interpolate the $u - v$ path in G
  - Find an MST or minimum spanning forest

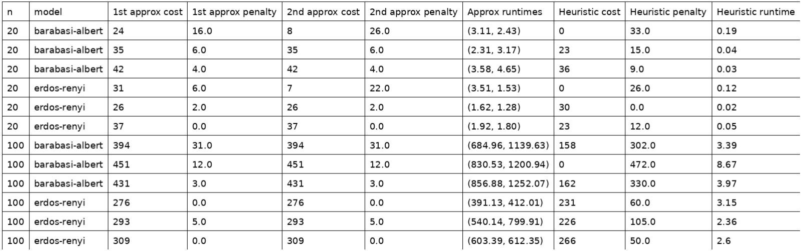
About implementation

- Language: Python
- Efficient data structure for coloring durations
- Used built-in functions from `networkx` library for:
    - Max-flow computation
    - Shortest paths
    - Deciding the connectivity of two nodes, etc.

Testing

- Testing on different random graph models: Erdős-Rényi, Barabási-Albert
- Parameters for $n = 100$
  - Edge costs: Uniform integers from $[1, 20]$
  - Penalties on 10% of the node pairs, penalty value $= 1$
  - Erdős-Rényi: $p = \frac{1}{10}$
  - Barabási-Albert $m = 2$

Results Summary



n	model	1st approx cost	1st approx penalty	2nd approx cost	2nd approx penalty	Approx runtimes	Heuristic cost	Heuristic penalty	Heuristic runtime
20	barabasi-albert	24	16.0	8	26.0	(3.11, 2.43)	0	33.0	0.19
20	barabasi-albert	35	6.0	35	6.0	(2.31, 3.17)	23	15.0	0.04
20	barabasi-albert	42	4.0	42	4.0	(3.58, 4.65)	36	9.0	0.03
20	erdos-renyi	31	6.0	7	22.0	(3.51, 1.53)	0	26.0	0.12
20	erdos-renyi	26	2.0	26	2.0	(1.62, 1.28)	30	0.0	0.02
20	erdos-renyi	37	0.0	37	0.0	(1.92, 1.80)	23	12.0	0.05
100	barabasi-albert	394	31.0	394	31.0	(684.96, 1139.63)	158	302.0	3.39
100	barabasi-albert	451	12.0	451	12.0	(830.53, 1200.94)	0	472.0	8.67
100	barabasi-albert	431	3.0	431	3.0	(856.88, 1252.07)	162	330.0	3.97
100	erdos-renyi	276	0.0	276	0.0	(391.13, 412.01)	231	60.0	3.15
100	erdos-renyi	293	5.0	293	5.0	(540.14, 799.91)	226	105.0	2.36
100	erdos-renyi	309	0.0	309	0.0	(603.39, 612.35)	266	50.0	2.6

Future Plans

- More testing with different parameters
- Try to find LP-based algorithms

References I

- Ali Ahmadi, et al. *2-Approximation for Prize-Collecting Steiner Forest.*
  Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2024.
- Akhmedov, Murodzhon, et al. "A fast prize-collecting steiner forest algorithm for functional analyses in biological networks." International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Cham: Springer International Publishing, 2017.

Thank you for your attention!