The Prize-Collecting Steiner Forest Problem

Kiss Bendegúz
Supervisor: Király Tamás

June 2025

Introduction

- **Work during the semester**:
  - Implement and test the 2-approximation algorithm I studied
  - Compare the results with a heuristic method that I developed
- **PCSF**: find a forest in a graph minimizing 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 gets fully colored (the coloring duration equals to its cost), we add it to the forest.
4. We also "color" node pairs: if the color reaches their penalty, they become tight.
5. Sets that only cut tight pairs are removed from the active sets.
6. Repeat until no active sets remain.
7. Finally, we keep only the edges needed to connect the non-tight pairs.

2-Approximation Algorithm

1. Run the 3-approximation algorithm on the original graph
2. Mark pairs we paid penalties for
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

Heuristic Algorithm

- Categorize v nodes as:
  - **Necessary:** sum of penalties including $v \geq$ maximal outgoing edge cost
  - **Useless:** sum of penalties including $v \leq$ minimal outgoing edge cost
- Other nodes:
  - If at least half of the penalties containing v are 0 v becomes useless
  - Otherwise it becomes necessary
- Build MST over necessary nodes

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

Testing

- Erdős-Rényi graphs with 100 nodes, $p = 0.1$
- Edge costs: Uniform integers from $[1, 10]$
- Penalties on 20% of the node pairs, penalty value $= 1$
- Compared algorithms: heuristic, 3-approx, 2-approx

Results Summary

#	m	Heuristic			3-Approx			2-Approx		
		Cost	Penalty	Time (s)	Cost	Penalty	Time (s)	Cost	Penalty	Time (s)
1	481	175	0	0.14	183	49	444.16	168	7	552.44
2	500	154	0	0.07	163	47	393.37	163	0	397.29
3	494	168	0	0.09	169	43	461.58	124	21	426.65
4	481	166	0	0.09	181	54	472.60	118	23	421.72
5	464	180	0	0.09	198	59	591.07	190	6	477.45
6	456	164	0	0.07	175	57	399.08	118	19	352.72
7	498	137	0	0.08	157	48	320.73	114	16	274.05
8	475	166	0	0.08	173	52	390.46	173	0	371.17
9	496	174	0	0.07	184	56	396.20	103	31	314.22
10	497	154	0	0.06	168	47	348.00	164	1	336.47
11	505	177	0	0.06	197	64	432.69	189	3	480.04
12	514	178	0	0.09	189	57	716.42	183	6	613.20
13	489	151	0	0.12	168	41	547.06	168	0	554.86
14	510	144	0	0.10	157	48	506.97	154	2	543.36
15	517	172	0	0.11	178	64	551.24	170	11	619.86

Future Plans

- Improve running time of my code
- Explore new heuristics (e.g. penalty/edge cost ratio)
- Investigate real-world applications of PCSF like protein interactions

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.

Thank you for your attention!