

Reflection on CS170 Project

Weixiao Zhan, Weishi Li

Apr. 2020

Introduction

From lecture/textbook, we learned that there are general three way to solve or approximate an NP in a methodological perspective. Brute search on small data set, Approximate algorithms or continue optimizing algorithms.

For the min-average-cost dominate tree problem(MDT), we decide to go for the second approach. The first reason is the large or medium graphs are already too big for brute search approach. Plus, the function of finding MDT has poor continuity since the dominate tree can vary so much because of a tiny weight change in some corner cases.

Our approximate algorithm takes on two step. The first step is to find the n min-cost spanning trees of G. Second step is go from a ST, we greedily to using DP to delete some edges to approach MDT.

Generate ST in cost-increasing order

The reason why we reduce a graph to ST is because a ST can at least guarantee the output is a valid DT even if not a minimum one, and the MDT are more likely consist of lighter weighted edges. And the minimum cost n ST increasing the probability of having the real MDT in one of the ST we generated.

The genST algorithm is from Sörensen Kenneth and Janssens Gerrit, Generating all spanning trees of a graph in order of increasing cost, in 2000. we didn't find a usable code or library of this algorithm. So we implemented it based on networkx.

remove edges to approach MDT