BTP Final Presentation

Fast Prize-collecting Steiner Tree Construction in Distributed setting

By Hemant Yadav (160101034)

Steiner Tree

- Given a connected undirected graph G = (V,E) and a weight function w : E → R+, and a set of vertices Z ⊆ V, known as the set of terminals, the goal of the ST problem is to find a tree T'= (V',E') such that $\sum_{e \in E'} w(e)$ is minimized subject to the conditions that Z ⊆ V' ⊆ V and E'⊆ E.

Prize Collecting Steiner Tree

- Given a connected weighted graph G = (V,E,p,w) where V is the set of vertices, E is the set of edges, p : V → R+ is a vertex prize function and w : E → R+ is an edge weight function, the goal is to find a tree T = (V',E'), where V'⊆ V and E'⊆E that minimizes the following quantity:

$$W(T) = \sum_{e \in E'} w_e + \sum_{v \notin V'} p_v$$

Objective

1) TO PROPOSE A FRAMEWORK FOR SOLVING PCST PROBLEM IN DISTRIBUTED SETTING.

2) COMPARE THE RESULTS OF OUR PROPOSED FRAMEWORK WITH ALREADY ESTABLISHED GOEMANS-WILLIAMSON ALGORITHM.

Goemans-Williamson Algorithm

Consists of two phases namely growth and pruning

In growth phase, calculate values that represent cost incurred if a set of vertices is along with the edges is included the prize part and penalty incurred if a set of vertices is included in the Steiner part.

Of these two values, the minimum one is chosen, and corresponding action is taken.

In pruning phase, remove as many edges as possible without violating certain conditions.

Proposed Framework

Given a connected undirected graph G = (V,E) and a weight function w : E → R+ , and a set of vertices Z ⊆ V , known as the set of terminals. There are 4 steps in the Faster PCST algorithm.

1) Choose a set of terminal Z ⊆ V by Median Based Approach or by Incident Edge Method.

2) Construct a Steiner Tree by applying Distributed Steiner Tree algorithm in G = (V,E) with set of terminal Z which has been chosen in step 1.

3) Calculate PCST value for ST found in step 2.

4) Repeat step 1,2 & 3 until there is no scope for further improvement in PCST value or √n times where n is no. of vertices in G. We will take the minimum of the two. So, if have run the algorithm √ n times then we shall terminate the algorithm without considering improvement.

Median Based Approach

Choose a subset of vertices having prize value greater than the median of prize values of all the vertices.

Since, we are working in distributed setting in our framework, this approach is also implemented in distributed setting.

Implementation Details

Following slides shed some light on how these ideas are implemented in the following order

1) Random graph generation

2) Goemans-Williamson Algorithm

3) Median Based Approach

Random Graph Generation

Graphs used for testing will be generated using this.

Input will be the number of vertices and number of edges(optional).

Output will be a connected graph with the given number of vertices and edges (in case edges are not specified, graph will use minimum required edges)

At an intermediate stage, an edge will be created between two vertices one of which will be chosen randomly from the connected set and other one from the rest.

Goemans-Williamson

- Input will be a graph G=(V,E) and a root node ‘r’.
- Output will be the prize part and PCST value.
- A 2 dimensional vector is used to store the graph
- To merge two sets(union), the trees representing the two sets are joined in a way that one becomes the child of another.
- Initially, all the vertices will be in their own tree.
- As the algorithm progresses, prize part will be stored in the tree containing the root node.

Median Based Approach

Input will be a graph G=(V,E) and a root node ‘r‘ similar to the G-W algorithm input.

Output will be the set of vertices all having value higher than the median chosen as the initial set of terminal vertices.

Uses the distributed sort to sort the prize values.

Thank You All

The End