An Algorithm for the Approximate Treedepth Problem

Aman Singal
Computer Science and Engineering Undergraduate
Indian Institute of Technology Dharwad
Dharwad, Karnataka, India
http://www.iitdh.ac.in
180030004@iitdh.ac.in

Abstract

The *treedepth* problem is known from different names such as vertex ranking, centered coloring, elimination tree height with multiple similar definitions. The tree-depth of a graph G is the minimum height of any rooted forest F such that G ⊆ closure(F). A fairly good heuristic approach to solve treedepth problem is presented with comparison of various different algorithms involving usage of graph related parameters such as connected components,cut-vertex and graph centralities.

2012 ACM Subject Classification Theory of computation→ Design and analysis of algorithms→ Graph algorithms analysis; Theory of computation→ Design and analysis of algorithms→ Approximation algorithms analysis

Keywords and phrases Treedepth, Vertex Coloring, Tree Decomposition, Graph Algorithms, Elimination Tree Height, Heuristic Algorithm

Supplementary Material Source Code: Python→ https://github.com/AmanSingal/pace-2020-submission1
C++ → https://github.com/AmanSingal/pace-2020-submission2

Acknowledgements A special thanks to Prof. Sandeep R.B for introducing me to this problem and guiding the way to approach it further.

1 Introduction

This paper presents a heuristic approach to fairly estimate the treedepth measure of a graph. Treedepth is a measure which captures the similarity of a graph to a star. The treedepth of a graph G is the minimum depth of any treedepth decomposition of G.

A *tree decomposition* of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$, where T is a tree and $\{X_i\}$ is a collection of subsets of V , such that

- $\bigcup_{i \in I} X_i = V$.
- For all $(v, w) \in E$, there exists an $i \in I$ with $v, w \in X_i$.
- For all $i, j, k \in I$, if j is on path from i to k in T, then $X_i \cap X_k \subseteq X_j$

OR

The treedepth of a graph G with connected components G_1, G_2, . . . , G_l is defined as follows:

$$\mathbf{td}(G) = \begin{cases} 1 & if\ |V(G)| = 1 \\ max_{1 \le i \le l}\ \mathbf{td}(G_i) & l > 1 \\ 1 + min_{v \in V(G)}\ \mathbf{td}(G - v) & otherwise \end{cases}$$

Structure of Paper. The whole paper is divided in four sections. Section 2 introduces to key concepts, definitions and notations which will be used in the further sections. Section 3 highlights the procedures and methods used for solving the problem. Section 4 concludes by comparing the methods and highlights which method proved to be most effective.

2 Preliminaries

Let $G = (V, E)$ be a graph, where $|V|$ *and* $|E|$ denotes no of vertices and edges respectively. The *neighbourhood* $\mathbf{N_{G(v)}}$ of a vertex v is the set of vertices that are adjacent to v. The notation $G - v$ is used for the removal of one vertex and its incident edges, that is, $G - v = G[V(G)\backslash\{v\}]$.

Cut Vertex - A cut vertex is a vertex that when removed (with its boundary edges) from a graph creates more components than previously in the graph.

Centrality - This concept is used in graph theory to identify the important nodes. Each node can be important from an angle depending on how "importance" is defined. There are various types on centralities such as Degree, Closeness, Betweenness, Eigen Vector Centrality.

Degree Centrality - In a non-directed graph, degree of a node is defined as the number of direct connections a node has with other nodes. In a directed graph (each edge has a direction), degree of a node is further divided into *In-degree* and *Out-degree*.
In-degree refers to the number of edges/connections incident on it.
Out-degree refers to the number of edges/connections from it to other nodes.
Degree Centrality metric defines importance of a node in a graph as being measured based on its degree i.e the higher the degree of a node, the more important it is in a graph.

Closeness Centrality - Closeness centrality metric defines the importance of a node in a graph as being measured by how close it is to all other nodes in the graph. For a node, it is defined as the sum of the *geodesic distance* between that node to all other nodes in the network.
The *Geodesic distance d* between two nodes a and b is defined as the number of edges/links between these two nodes on the shortest path(path with minimum number of edges) between them.

Betweenness Centrality- This metric defines and measures the importance of a node in a network based upon how many times it occurs in the shortest path between all pairs of nodes in a graph. A sample application of BC is to find bridge nodes in graphs. Nodes having high BC are the nodes that are on the shortest paths between a large number of pair of nodes and hence are crucial to the communication in a graph as they connect a high number of nodes with each other. Removing these nodes from the network would lead to huge disruption in the linkage or communication of the network.

Eigen Vector Centrality - This metric measures the importance of a node in a graph as a function of the importance of its neighbors. If a node is connected to highly important nodes, it will have a higher Eigen Vector Centrality score as compared to a node which is connected to lesser important nodes. One sample application of EVC is the calculation of *Page Rank* or Page Rank algorithm used by Google and many other companies to rank web pages on the internet by relevance.

3 Methods and Algorithms

The recursive definition of treedepth involves choosing a vertex everytime from a graph component and making it the root for the subgraph introduced by removing it. In regard to this the following methods were tried out:-

- **Based on Number of Connected Components[Method 1].** At every iteration the vertex which on removing from the graph gives the maximum no of components is selected and the procedure is then recursively continued. If there is no vertex which is cut vertex then Method 2 is followed.
- **Based on Degree[Method 2].** At every iteration the vertex with the maximum degree in a connected component is selected and removed and the procedure is then recursively continued. If the degree is same then the vertex with the lower index is selected.
- **Based on Degree and Number of Connected Components [Method 3].** This method involves following [Method - 2] except that first some of the vertices(example: 4 to 6) are marked which have the maximum degree then the vertex which gives the maximum no of components on removal is selected. If no such vertex is found then [Method 2] is followed.
- **Based on Closeness Centrality[Method 4].** At every iteration the vertex which gives the maximum value for closeness is selected and the normal recursive procedure is followed.
- **Based on Betweenness Centrality[Method 5].** At every iteration the vertex which gives the maximum value for betweenness is selected and the normal recursive procedure is followed.
- **Based on EigenVector Centrality[Method 6].** At every iteration the vertex with the maximum value for eigen vector parameter is selected and the normal recursive procedures is followed.

All the described methods here aim to increase the no of child for a vertex in a tree decomposition. Increasing the no of childs for every vertex helps in reducing the height of such tree decomposition thereby giving a fair approximation of treedepth. The following illustration depicts the overall idea:-

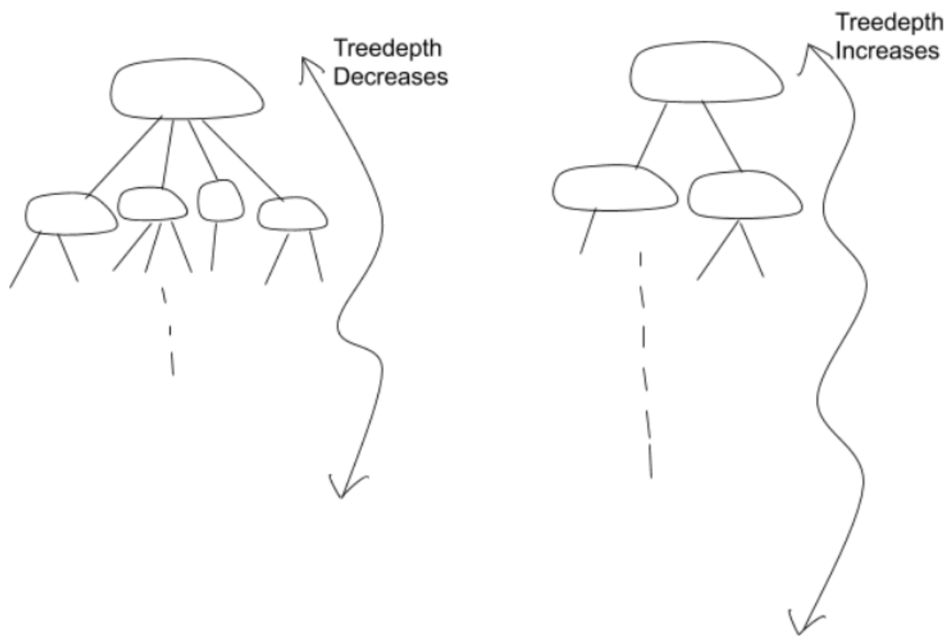

Figure 1 Illustration of the general idea

4 Observations and Conclusion

The above mentioned procedures were tried and tested and the order obtained for their effectiveness is listed below in ascending order:-
Method 1 $<$ Method 3 $<$ Method 2 $<$ Method 6 $<$ Method 4 $<$ Method 5
The effectiveness is judged on the basis of no of instances in which they give better results and the margin by which the tree depth obtained by these methods differ. **Betweenness measure** proved to be the best measure to use for approximating tree depth among all of the others mentioned. *For large graphs, to obtain the results in a bounded time* **approximate version of betweenness measure** *is used.*

References

1 Brandes, U. (2001). *A faster algorithm for betweenness centrality.* The Journal of Mathematical Sociology, 25(2), 163–177.

2 Girvan, M., & Newman, M. E. J. (2002). *Community structure in social and biological networks (Vol. 99).*

3 Knuth, D. E. (1993). *The Stanford GraphBase: a platform for combinatorial computing (Vol. 37).* Reading: Addison-Wesley.

4 Chia-Hao Lu, Bang Ye Wu. *Heuristic algorithms for tree-depth of social networks.* https://pdfs.semanticscholar.org/8ff7/a32fdb9f6c68f35846feb20b5fb52690ecd3.pdf\protect\@normalcr\relax_ga=2.16500341.236139305.1590415851-894515482.1590415851.

5 Linton C. Freeman. *A set of measures of centrality based on betweenness.* Sociometry 40: 35–41, 1977. http://moreno.ss.uci.edu/23.pdf

6 Ulrik Brandes. *On Variants of Shortest-Path Betweenness Centrality and their Generic Computation.* Social Networks 30(2):136-145, 2008. http://www.inf.uni-konstanz.de/algo/publications/b-vspbc-08.pdf.

7 Alejandro A. Schäffer. *Optimal node ranking of trees in linear time.* Inf. Process. Lett.,33(2):91–96, 1989. https://doi.org/10.1016/0020-0190(89)90161-0,doi:10.1016/0020-0190(89)90161-0.

8 Jatin Bhasin *Graph Analytics — Introduction and Concepts of Centrality.* Medium