15-400 Project Proposal: Searching for a Root in a Graph Streaming Situation

Amulya Musipatla (amusipat)

November 11, 2018

1 Introduction

Graphs are a very valuable tool when it comes to representing relationships between objects, and are commonly used in many applications to gain a deeper understanding of these relationships. However, in real applications, these graphs can grow to be extremely large. In these cases, it is not always feasible to store everything about the graph, so it is essential to minimize the amount of information that is stored in order while still maintaining the necessary knowledge. In addition, there are also many cases in which applications have to be able to process real time information. These applications tend to have very quickly growing graphs, and there is the added complexity of having to be able to adapt to more information as it comes.

These problems are often approached from a streaming perspective. In general, this involves a stream of elements where the algorithm is able to see a fixed number of elements in this stream at once, and needs to adapt any existing representation it has to take these new elements into consideration before discarding them and looking at different elements. With graphs these elements would be edges.

2 Project Description

I will be working with Professor David Woodruff for this project in 15-400. Professor Woodruff's research generally works with streaming and graph algorithms, and in this case we'll be working on a graph streaming problem.

The set up is as follows: given a stream of directed edges under the guarantee of the edges forming a tree, we want to be able to dynamically determine the root of the tree. As of now, Professor Woodruff has already established a lower bound of $O(n)$ for this problem via a reduction from the indexing problem.

Because we already have this lower bound, we want to work on different variations of this problem instance and see how our bounds change and if we can come up with a more efficient algorithm than expected. Other variations of this problem that we're particularly interested in involve taking multiple passes on the stream, working with a specific type of tree, or requiring some sense of order in the stream of edges.

We also want to work on characterizing easy and difficult instances- what exactly makes one type of tree or stream easier to find the root in, and what makes it difficult?

We found that this problem would be interesting to work on because usually with graph streaming problems, we work on trying to find algorithms using n bits of space rather than less than n bits of space, making this problem different from more common ones. Finding an algorithm using less than n bits of space could provide us with more insight on problems of this type. In addition, because being able to find the root in a tree is a fundamental problem, we could apply what we learn from this problem to other more complex problems in the field.

3 Prior Work and Literature

As stated above, most of the work in graph streaming algorithms has involved trying to find algorithms using n bits of space. In particular, the biggest problems in graph streaming involve trying to maintain a measure of connectivity to keep track of various aspects of the graph such as matchings, distances, and more. These problems don't directly relate to the problem Professor Woodruff and I are exploring, but we can use some of the methods as a starting point.

To help me with my project as of now, I have been reading papers that give an overview of graph streaming algorithms and common problems [1][2][3]. These papers give a high level overview of problems of this type, but to get even more familiar later on, I will read more in depth papers about specific problems rather than survey papers.

Other than that, there isn't much previous work in this problem, since it was proposed recently to Professor Woodruff at a conference. Because of this, we want to first work on gaining more intuition on the problem and easy/difficult instances of it before delving into algorithms and proving lower bounds.

4 Project Goals

The ideal 125% goal would be to find a tight lower and upper bound for some variation of the problem via a reduction and an algorithm. Actually implementing the algorithm to see how it runs in real life would give more insight for this problem as well.

Realistically, as our 100% goal, we want to be able to find a lower bound for the variation we choose to work on and come up with an algorithm or method that gets close to this bound.

As our 75% goal if things go slower than expected, we hope to at least be able to characterize difficult and easy instances and use this knowledge to come up with an algorithm that uses less than $O(n)$ space.

5 Project Milestones

First Milestone by end of 15-300: In order to prepare for research in this field next semester, I need to get up to date with common graph streaming algorithms and methods used. To solidify my knowledge here and gain more intuition, I hope to get more familiar with reductions in graph streaming and come up with some reductions to the problem we are focusing on.

February 1st: Read more papers about directed graph streaming algorithms as well as problems involving less than n space to gain more intuition and learn about methods we could use for this problem.
February 15th: Make observations about difficult instances and easy instances: what characterizes them, can we adapt the algorithms we already have for easy instances to work for harder ones?
March 1st: Work on improving the basic algorithm for difficult instances with different variations based on what we learned.
March 22nd: Work on establishing a lower bound for the variation we choose to work on (multi-pass, certain type of graph).
April 5th: Continue to work on establishing a lower bound.
April 19th: Based on the lower bound we established, try to further improve the algorithm we had established. Potentially start on implementing the algorithm if we choose to go down this path.
May 3rd: Apply what we learned from these harder instances and this problem to more general graphs and other graph streaming problems in general.

6 Resources Needed

There are no external resources required for this project.

7 Website

https://amusipatla.github.io/RootFinding/

References

[1] Ahn, Kook Jin, et al. Analyzing Graph Structure via Linear Measurements. *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, 2012, pp. 459467., doi:10.1137/1.9781611973099.40.

[2] Feigenbaum, Joan, et al. Graph Distances in the Data-Stream Model. *SIAM Journal on Computing*, vol. 38, no. 5, 2009, pp. 17091727., doi:10.1137/070683155.

[3] Mcgregor, Andrew. Graph Stream Algorithms. *ACM SIGMOD Record* vol. 43, no. 1, 2014, pp. 920., doi:10.1145/2627692.2627694.