Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs
Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, Shang-Hua Teng
Interim report for CS 6150

Christopher Mertin Seyed Majid Rasouli Pichahi

October 20, 2015

1 Background and Result

This paper uses a new technique to approximate the maximum s-t flow and the minimum s-t cut in time better than what is currently known $\mathcal{O}\left(n^{1.5}\right)$ by Even and Tarjan [1]. It does this in an intuitive way by treating the graph as a *simple circuit* and using basic physics equations to solve the problem.

This work expands on work by Goldberg and Rao [2] which developed an algorithm for computing the exact minimum s-t flows, and the work by Benczúr and Karger [3]. They use a multiplicative weights update to optimize the computation time from these as well, where the method is defined in [4, 5].

The multiplicative weights method uses a definition of congestion which is the "flow" from each edge over the total number of paths. This gives a ratio of how much traffic/current can flow through that path. This is important for the interpretation of the electrical flow method to solve the problem. For a simple circuit, the amount of current that can flow through a segment is based on the potential divided by the resistance, so the larger the resistance the less current can pass through. The max s-t flow wants to find the path that gets the most current through from s to t, so to do this, the resistance of each edge is related to the congestion of it. The higher the congestion, the less "current" that can flow through that edge, making it a bad path to take. In our paper, we will talk about the physical interpretations of the equations to relate how the authors came to the

The problem is approximately computing a minimum cut of a graph in time $\widetilde{\mathcal{O}}\left(m\sqrt{n}\epsilon^{-1}\right)$. The maximum flow of a graph can be used to obtain a minimum cut of the graph. The problem of computing maximum flows subject to capacity constraints can be reduced to the problem of computing electrical flows in resistor networks.

First, the notion of (e, p) oracle is defined to find a flow which has some desired properties. In this definition, p is called the width of the oracle. First, a simplified version of the approximate maximum-flow algorithm is explained that has running time $\widetilde{\mathcal{O}}\left(m^{3/2}\epsilon^{-3}\right)$. (Pseudocode 1)

The key point in analyzing this algorithm is that the total weight on G does not grow too quickly, due to the average congestion constraint on the flows returned by the oracle O.

By using Theorem 3.2 in the paper, the problem is thus reduced to designing an efficient oracle that has a small width. To build such an oracle, we set a resistance for each edge, and we use the procedure from Theorem 2.3 to approximate the electrical flow. (Pseudocode 2)

For some graphs, it is possible for the electrical flow returned by the oracle to exceed the edge capacities. In these kinds of graphs, there are some edges which we call "bad" edges. If one removes a "bad" edge from the graph, the electrical flow on the resulting graph is much better behaved, but the value of the maximum flow is only very slightly reduced. This demonstrates a phenomenon that will be central to the improved algorithm. So, Pseudocode 1 and 2 are improved by utilizing this idea.

The resulting algorithm is combined with Karger's idea of "graph smoothing" to improve its running time, that allows one to use random sampling to speed up an exact or $(1-\epsilon)$-approximate flow algorithm.

Finally, a variant of this algorithm is presented that computes approximately minimum s-t cuts in time $\widetilde{\mathcal{O}}\left(m + n^{4/3}\epsilon^{-8/3}\right)$. In which, The *Max Flow-Min Cut Theorem* [6, 7] is utilized which states that the capacity of the minimum s-t cut is equal to F^*, the value of the maximum s-t flow.

2 Plan

Some of the things that need to be explored from this paper so that we can fully understand it and write our final paper on it are as follows:

- Becoming familiar with the references listed below that the paper cited
- Looking at the sort of topics of papers that have cited this paper
- Working through the proofs and algorithms in section 4 for the improved algorithm and possibly for the first algorithm. As well as fully understanding the other iterations and approximations they made to get to the improved step
- Karger's method of graph smoothing

References

[1] S. Even and R. E. Tarjan. Network flow and testing graph connectivity. *SIAM Journal on Computing*, 4(4):507-518, Dec. 1975

[2] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *J. ACM*, 45(5):783-797, 1988

[3] A. A. Benczúr and D. R. Karger. Approximating s-t minimum cuts in $\mathcal{O}\left(n^2\right)$ time. In *STOC'96: Proceedings of the* 28^{th} *annual ACM Symposium on Theory of Computing*, pages 47-55, New York, NY, USA, 1996, ACM

[4] S. Aroira, E. Hazan, and S. Kale. The multiplicative weights update method: A meta-algorithm and applications. Available at: `http://www.cs.princeton.edu/~arora/pubs/MWsurvey.pdf`

[5] S. A. Plotkin, D. B. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20:257-301, 1995

[6] Samuel I. Daitch and Daniel A. Spielman, Faster approximate lossy generalized flow via interior point algorithms, In *Proceedings of the 40th annual ACM symposium on Theory of Computing*, pages 451-460, 2008

[7] Lisa K. Fleischer, Approximating Fractional Multicommodity Flow Independent of the Number of Commodities, *SIAM Journal on Discrete Mathematics*, 13(4):505-520, 2000