# CS267 HW0

Jonah Sherman

February 7, 2013

## 1 Bio

I'm a CS graduate student in the Theory group, advised by Umesh Vazirani. I work on graph algorithms for network flows, routing, and partitioning. As a theorist, I mainly concern myself with algorithms that have provable guarantees on the quality of output and asymptotic running time.

I'm aware that for many of these problems I work on, what's done in practice is actually quite different than what's done in theory, and I'm actually quite unfamiliar with how some of these problems are solved in practice. I hope to learn how some of the problems I design algorithms for are actually solved in practice.

## 2 Parallel Graph Partitioning

Given a graph, we aim to partition it into relatively independent pieces in the sense that not too many edges cross between them. The problem is useful by itself for clustering, VLSI layout, and machine scheduling. It also frequently occurs as a sub-problem in solving many other interesting problems. There has been substantial theoretical work on this problem, and it is known that a partition can be found with a cost of at most $O(\sqrt{\log n})$ times the optimal[1]. Furthermore, such a partitioning can be found in essentially the same amount of time it takes to compute max-flow[4].

On the other hand, it seems that in practice quite different methods are used. One popular heuristic is METIS[2], which works by finding a heavy matching, contracting the edges, and then recursing. This algorithm is extremely fast, and, while it provides no theoretical guarantees whatsoever on the quality of its output, it apparently produces good partitions in practice. A parallel version, ParMETIS, is frequently used to do partitioning on very large graphs[3]. ParMETIS uses the MPI library and is written in C to be portable to most parallel computers supporting MPI. Different variants of ParMETIS have different scaling properties, and essentially trade quality of partition for parallelization. The most naive, a random partition, clearly scales by $p$ on $p$ processors, but will give a poor partition. The authors suggest a tradeoff that results in a $O(\sqrt{p})$ improvement factor using $p$ processors. Measurements of their implmentation on up to 128 processors were found to match the theoretical scaling behavior predicted.

# References

[1] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, STOC '04, pages 222–231, New York, NY, USA, 2004. ACM.

[2] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, December 1998.

[3] George Karypis and Vipin Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *J. Parallel Distrib. Comput.*, 48(1):71–95, January 1998.

[4] Jonah Sherman. Breaking the multicommodity flow barrier for $o(\sqrt{\log n})$-approximations to sparsest cut. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 363–372, Washington, DC, USA, 2009. IEEE Computer Society.