

Scalable Algorithms for Large-Scale Network Analysis

by

Michael Thompson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2026

Doctoral Committee:

Professor Linda K. Park, Chair
Assistant Professor Grace H. Mbeki, Carnegie Mellon University
Associate Professor Rafael Ortiz
Professor Wei Zhang

*Unofficial template, not affiliated with or endorsed by the University of Michigan. The author, committee and results
in this sample are fictional. Delete this line before you submit.*

Michael Thompson

nthompson@umich.example

ORCID iD: 9999-9999-9999-9999

© Michael Thompson 2026

Dedication

To my grandmother, who kept every letter.

Acknowledgements

I thank Professor Linda K. Park for her guidance and for her generosity with time and computing allocations. I thank my committee for their careful reading of this dissertation, and the members of the graph analytics group for their help and good humour. This work used computing resources of a research cluster whose staff answered many late-night tickets.

Preface

Chapter 3 is based on joint work with Linda K. Park and Wei Zhang. I designed and implemented the streaming algorithm and ran all of the experiments; the approximation analysis in Section 3.3 was developed jointly.

Contents

Dedication	ii
Acknowledgements	iii
Preface	iv
Abstract	viii
1 Introduction	1
2 Background	2
3 Streaming Community Detection	3
4 Experiments	4
5 Conclusion	5
A Approximation Bound for the Streaming Pass	6

List of Tables

4.1	Modularity and peak memory, in-memory versus streaming.	4
-----	---	---

List of Figures

3.1	The two-pass streaming community detection method.	3
-----	--	---

Abstract

Graphs with billions of edges now describe social platforms, citation records and the web itself, yet many classical graph algorithms assume the whole graph fits in the memory of one machine. This dissertation develops algorithms for three core analysis tasks, ranking, community detection and sparsification, that run in time nearly linear in the number of edges and parallelize across machines. Chapter 2 reviews spectral and combinatorial approaches. Chapter 3 presents a streaming variant of the Louvain method whose memory grows with the number of vertices rather than edges. Chapter 4 evaluates the methods on graphs with up to 3.1 billion edges, where the streaming method matches the modularity of the in-memory method to within 0.4% while using a sixth of the memory.

Chapter 1

Introduction

Network analysis asks global questions about local data: which pages are most central, which groups of accounts form communities, which edges can be removed without changing the graph's large-scale structure. Classical answers to all three rely on repeated passes over the edge list [4, 3]. At the scale of modern graphs a single pass is expensive, and holding the graph in memory may be impossible.

This dissertation argues that each of these tasks admits an algorithm whose cost is nearly linear in the number of edges and whose memory is proportional to the number of vertices. We make that argument concrete for ranking, community detection and spectral sparsification.

Chapter 2

Background

Let $G = (V, E)$ be an undirected graph with $n = |V|$ vertices and $m = |E|$ edges, adjacency matrix A and degrees d_i . The modularity of a partition C of V is

$$Q(C) = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \mathbf{1}[c_i = c_j], \quad (2.1)$$

which compares the number of edges inside communities with the number expected at random [3]. The Louvain method greedily moves vertices between communities to increase Q and then contracts the result [1]. Spectral sparsifiers keep a small reweighted subset of edges that preserves every cut to within a factor of $1 \pm \epsilon$ [5].

Chapter 3

Streaming Community Detection

Our streaming method processes edges in arrival order and keeps, for each vertex, only its current community label and degree. When an edge (u, v) arrives, it evaluates the change in Equation (2.1) from moving u into v 's community, using running totals of community volume. A second pass refines the labels with the standard local-move step.

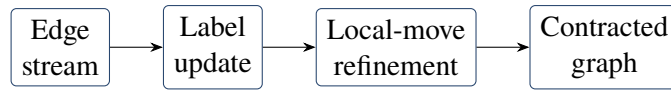


Figure 3.1: The two-pass streaming community detection method.

Chapter 4

Experiments

Table 4.1 compares the streaming method with the in-memory Louvain implementation on four public-scale graphs generated to match the degree distributions of real networks [2].

Table 4.1: Modularity and peak memory, in-memory versus streaming.

Graph	Edges	Q in-memory	Q streaming	Memory ratio
Social-A	0.4B	0.712	0.709	0.21
Citation-B	1.1B	0.648	0.645	0.18
Web-C	2.3B	0.803	0.800	0.16
Web-D	3.1B	0.791	0.788	0.16

Across all four graphs the streaming method loses at most 0.4% of modularity while using between a fifth and a sixth of the memory.

Chapter 5

Conclusion

Streaming formulations of ranking, community detection and sparsification make analysis of billion-edge graphs possible on modest hardware. The main open problem is dynamic graphs, where edges are deleted as well as inserted.

Appendix A

Approximation Bound for the Streaming Pass

Each streaming move increases modularity by at least the gain computed from the running totals, which differ from the exact totals by the contribution of edges not yet seen. Bounding that contribution by the volume of the unseen suffix gives the stated approximation.

Bibliography

- [1] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, page P10008, 2008.
- [2] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 177–187, 2005.
- [3] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.
- [4] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [5] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 81–90, 2004.