Scalable Algorithms for Data and Network Analysis

Shang-Hua Teng
Computer Science and Mathematics
University of Southern California
shanghua.teng@gmail.com

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Theoretical Computer Science

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

S.-H. Teng. *Scalable Algorithms for Data and Network Analysis*. Foundations and Trends® in Theoretical Computer Science, vol. 12, no. 1-2, pp. 1–274, 2016.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-131-3

Foundations and Trends® in Theoretical Computer Science
Volume 12, Issue 1-2, 2016

Editorial Board

Editorial Scope

Topics

Foundations and Trends® in Theoretical Computer Science publishes surveys and tutorials on the foundations of computer science. The scope of the series is broad. Articles in this series focus on mathematical approaches to topics revolving around the theme of efficiency in computing. The list of topics below is meant to illustrate some of the coverage, and is not intended to be an exhaustive list.

- Algorithmic game theory
- Computational algebra
- Computational aspects of combinatorics and graph theory
- Computational aspects of communication
- Computational biology
- Computational complexity
- Computational geometry
- Computational learning
- Computational Models and Complexity
- Computational Number Theory
- Cryptography and information security
- Data structures
- Database theory
- Design and analysis of algorithms
- Distributed computing
- Information retrieval
- Operations research
- Parallel algorithms
- Quantum computation
- Randomness in computation

Information for Librarians

Foundations and Trends® in Theoretical Computer Science, 2016, Volume 12, 4 issues. ISSN paper version 1551-305X. ISSN online version 1551-3068. Also available as a combined paper and online subscription.

Foundations and Trends® in
Theoretical Computer Science
Vol. 12, No. 1-2 (2016) 1–274

DOI: 10.1561/0400000051

now
the essence of knowledge

Scalable Algorithms for Data and Network Analysis

Shang-Hua Teng
Computer Science and Mathematics
University of Southern California
shanghua.teng@gmail.com

Contents

Abstract

In the age of Big Data, efficient algorithms are now in higher demand more than ever before. While Big Data takes us into the asymptotic world envisioned by our pioneers, it also challenges the classical notion of efficient algorithms: Algorithms that used to be considered efficient, according to polynomial-time characterization, may no longer be adequate for solving today's problems. It is not just desirable, but essential, that efficient algorithms should be *scalable*. In other words, their complexity should be nearly linear or sub-linear with respect to the problem size. Thus, *scalability*, not just polynomial-time computability, should be elevated as the central complexity notion for characterizing efficient computation.

In this tutorial, I will survey a family of algorithmic techniques for the design of *provably-good* scalable algorithms. These techniques include local network exploration, advanced sampling, sparsification, and geometric partitioning. They also include spectral graph-theoretical methods, such as those used for computing electrical flows and sampling from Gaussian Markov random fields. These methods exemplify the fusion of combinatorial, numerical, and statistical thinking in network analysis. I will illustrate the use of these techniques by a few basic problems that are fundamental in network analysis, particularly for the identification of significant nodes and coherent clusters/communities in social and information networks. I also take this opportunity to discuss some frameworks beyond graph-theoretical models for studying conceptual questions to understand multifaceted network data that arise in social influence, network dynamics, and Internet economics.

S.-H. Teng. *Scalable Algorithms for Data and Network Analysis*. Foundations and Trends® in Theoretical Computer Science, vol. 12, no. 1-2, pp. 1–274, 2016.
DOI: 10.1561/0400000051.

Preface

In 1997, I attended an invited talk given by Shafi Goldwasser at the *38th Annual Symposium on Foundations of Computer Science*. It was a very special talk. The title of her talk, printed in the conference program, "New Directions in Cryptography: Twenty Some Years Later," was modest. However, the talk was beautiful and poetic. In particular, the talk's subtitle, "Cryptography and Complexity Theory: a Match Made in Heaven," has stayed with me after all these years.

The rise of the Internet, digital media, and social networks has introduced another wonderful match in the world of computing. The match between *Big Data and Scalable Computing* may not be as poetic as the match between *Cryptography and Complexity Theory*: Big Data is messier than cryptography and scalable computing uses more heuristics than complexity theory. Nevertheless, this match — although practical — is no less important: "Big Data and Scalable Computing: a Pragmatic Match Made on Earth."

Reasons to Write and People to Thank

I would like start by thanking Madhu Sudan and James Finlay for inviting me to write a survey for *Foundations and Trends in Theoretical Computer Science*, and for their patience, support, and guidance during this long process.

When Madhu and James first reached out to me in the February of 2012 to write a survey on *graph sparsification*, I was noncommittal and used my busy schedule as the chair of a large department as my excuse. When they came back to me in the Fall of 2012 — knowing that I had successfully become a former chair — I did not reply until the June of 2013, when I had received the confirmation that the USC Daycare finally accepted my 8 month-old daughter Sonia off the wait-list. But during the span of these 16 months, many things happened that were relevant to their initial invitation.

- Nisheeth Vishnoi completed a wonderful and comprehensive survey, titled, $\mathbf{Lx} = \mathbf{b}$ [344], that appeared in the May issue of *Foundations and Trends in Theoretical Computer Science*.

- Joshua Batson, Dan Spielman, Nikhil Srivastava, and I completed our long overdue 8-page article, "Spectral Sparsification of Graphs: Theory and Algorithms," [43] for the *Research Highlights* of *Communications of the ACM*. That article appeared in August 2013.

- Several exciting new results emerged in spectral graph theory that were enabled or inspired by spectral sparsification and scalable Laplacian solvers.

These developments had reduced the need for another longer survey solely devoted to (spectral) sparsification. But I got unexpected encouragement to write a survey from researchers outside my usual theory community. Yan Liu, my machine learning/data mining colleague at USC, invited me to present my work on spectral graph theory and network analysis at the 2012 *SIAM Data Mining Conference*. I gave a talk titled, "Algorithmic Primitives for Network Analysis: Through the Lens of the Laplacian Paradigm," based on my joint work with Dan Spielman. Although the talk was a typical theoretical computer science talk, I was excited by the reception that I received from the Big Data experts: Huan Liu, Joydeep Ghosh, Vipin Kumar, Christos Faloutsos, and particularly Yan Liu, who strongly encouraged me to write a survey on these scalable algorithmic techniques for readers beyond theoretical computer science.

It was a tall order! But given this potential interest from the Big Data community, I reconnected with Madhu and James and proposed a tutorial to further expand my talk. My goal was to survey some basic theoretical developments (on scalable algorithms) and their techniques that might be useful for practical data/network analysis. The plan was to select a collection of fundamental and illustrative topics of potential practical relevances. I naturally favor problems and algorithms whose rigorous mathematical analysis are clean enough to present for researchers outside theory. Towards this end — in the survey — I selectively encapsulate some "heavy duty" mathematical materials, state them as theorems without proofs, and only expose the relevant essentials aiming to make the survey readable without losing its rigor.

Here, I would like to thank Yan Liu for initiating all this and her valuable feedback on the draft. I thank Madhu and James again for supporting this changed plan and their advice on how to proceed with this writing. I thank Amy Schroeder of the USC Viterbi Engineering Writing Program for editing this monograph, and the anonymous referee and my Ph.D. student Yu Cheng for valuable feedback. I thank Dan Spielman and all my collaborators who have directly contributed to this survey: To Nina Amenta, Reid Andersen, Nina Balcan, Joshua Batson, Marshall Bern, Christian Borgs, Michael Brautbar, Mark Braverman, Jennifer Chayes, Paul Christiano, Wei Chen, Xi Chen, Dehua Cheng, Yu Cheng, Siu-Wing Cheng, Ken Clarkson, David Eppstein, Tamal Dey, John Dunagan, Herbert Edelsbrunner, Matthias Eichstaedt, Michael Elkin, Yuval Emek, Michael Facello, Alan Frieze, Daniel Ford, John Gilbert, Rumi Ghosh, John Hopcroft, Kamal Jain, Jon Kelner, Marco Kiwi, James Lee, Tobin Lehman, Kristina Lerman, Xiangyang Li, Yan Liu, Qi Lu, Aleksander Mądry, Adrian Marple, Gary Miller, Vahab Mirrokni, Richard Peng, Greg Price, Heiko Röglin, Horst Simon, Nikhil Srivastava, Carl Sturtivant, Dafna Talmor, Bill Thurston, Steve Vavasis, Konstantin Voevodski, Noel Walkington, Yu Xia, and Xiaoran Yan, thank you!

Theory and Practice

While I believe in the importance of provably-good algorithms in data and network analysis, my own experiences at Intel, NASA, and Akamai have also taught me the limitation of "provably-good algorithms." The gap between proof-based theory and relevance-based practice is beyond the limitation of the worst-case or average-case analyses. The theory-practice gap exists also because essentially all measures of qualities — from the centrality of a node in a network, to the similarity between two datasets/networks, to the coherence of a network cluster and community — may have their limitations.

> *Many conceptual questions that arise in modeling real-world data are fundamentally challenging.*

Thus, as much as I would like — in this survey — to make an algorithmic connection between theory and practice in the area of data and network analysis, I would also like readers to approach this survey with an open mind: Theory is usually my guide for understanding practical problems, but theoretical thinking has too often been the obstacle that makes me struggle in connecting with practice. On the few occasions that theoretical thinking provided me with the insight to make a connection, I was elated. But indeed, I usually found myself unable to balance the connection between theory and practice, and then decided to do theory for its mathematical beauty.

For example, in theoretical computer science, we have also encountered the notion of clusterability in various settings, including VLSI layout, parallel processing, network clustering, community identification, and more generally, the design of divide-and-conquer algorithms for matrix/graph problems. However, as illustrated throughout this survey, I remain unsure about how to evaluate and validate the relevance of various mathematical notions of clusterability, particularly the conductance or cut-ratio measures that I have been studying for more than a decade. We use these partition/clusterability measures because they appear to be reasonable, and because, using them, we have obtained mathematical proofs that are aligned with our intuition.

My Ph.D. advisor Gary Miller once said to me, "coping with the uncertainty between theory and practice gives rise to plausible and sometimes good research questions, but questioning the certainty can lead to excellent questions." So I do think it is more than reasonable to question and challenge every notion one chooses.

Discrete vs Continuous and Beyond

I would like to conclude the preface by remarking that several subjects of this survey are at the intersection between combinatorial optimization and numerical analysis. Thus, I think they serve as good examples of the interplay between combinatorial thinking and numerical thinking [330]. While it is more conventional to view many network analysis problems as graph-theoretical problems, it can often be constructive to view them as numerical, statistical, or game-theoretical subjects as well. Network data is richer than its graph representation, and network science is beyond graph theory.

Numerical thinking is also more than numerical analysis — it is a creative process of discovering useful numerical connections that may not be apparent [330]. For example, in the 70s, Hall [162], Donath and Hoffman [115, 116], and Fiedler[1] [133, 134] made insightful connection between graphs and matrices — beyond just the matrix representation of graphs — which set the stage for spectral graph theory. The field of algorithm design has benefited greatly from the deep connection between graph properties (such as connectivity, conductance, and mixing time) and algebraic properties (spectral bounds of Laplacian/adjacency/random-walk matrices [82]). Over the last decade, scientists have made even broader and deeper connections between numerical solutions and network solutions [211, 73], between numerical representations and digital representations [110, 108], between

[1] To the memory of Miroslav Fiedler (1926 – 2015): It is difficult to overstate the impact of Fiedler's work to spectral graph theory. His paper, "Algebraic Connectivity of Graphs," [133] established a far-reaching connection between graph theory and linear algebra. Fiedler's spectral theorem, together with Koebe's disk-packing characterization of planar graphs [213] (see Theorem 5.34), Sperner's lemma for Brouwer's fixed-point theorem [310], and Cheeger's inequality [82] (see Theorem 4.2), are my favorite mathematical results — they beautifully connect continuous mathematics with discrete mathematics. I have always cherished my only meeting with Fiedler. After my talk, "The Laplacian Paradigm: Emerging Algorithms for Massive Graphs," [329] at *7th Annual Conference on Theory and Applications of Models of Computation*, Jaroslav Nešetřil (my former officemate at Microsoft Research Redmond) thoughtfully invited my wife Diana (a US historian) and I to a dinner with him and Fiedler. I still vividly remembered that special evening in the beautiful Prague on June 10, 2010. At age eighty four, Miroslav was charming and talkative, not just about mathematics but also about history.

numerical methods and statistical methods [111, 118], between numerical concepts and complexity concepts [314], and between numerical formulations and privacy formulations [121]. Numerical analysis has played an increasing role in data analysis through dimension reduction [111] and in machine learning through optimization.

In the examples of this survey, the Laplacian paradigm [319] has not only used numerical concepts such as preconditioning to model network similarity and graph sparsification, but also used combinatorial tools to build scalable solvers for linear systems [313, 318, 317, 319], Gaussian sampling [86], and geometric median [98]. Scalable techniques for PageRank approximation [65] have also led to algorithmic breakthroughs in influence maximization [64, 325, 324] and game-theoretical centrality formulation [84]. These results illustrate the rich connection between network sciences and numerical analysis.

A Quick Guide of the Survey

The survey begins with two background chapters. Chapter 1 discusses scalability measures for evaluating algorithm efficiency. It also introduces the basic characterizations of scalable algorithms, which are the central subjects of this survey. Chapter 2 reviews mathematical models for specifying networks, and highlights a few basic problems in data and network analysis. We will use these problems as examples to illustrate the design and analysis of scalable algorithms in the technical chapters that follow the background chapters. In Section 1.5, I will give a more detailed outline of these technical chapters.

Acknowledgements

The writing of this article is supported in part by a Simons Investigator Award from the Simons Foundation and the NSF grant CCF-1111270.

Shang-Hua Teng
Los Angeles

1

Scalable Algorithms

In light of the explosive growth in the amount of available data and the diversity of computing applications, efficient algorithms are in higher demand now more than ever before.

1.1 Challenges of Massive Data

Half a century ago, the pioneers of Theoretical Computer Science began to use asymptotic analysis as the framework for complexity theory and algorithm analysis [167]. At the time, computers could only solve small-scale problems by today's standards. For example, linear programs of fifty variables, linear systems with a hundred variables, or graphs with a thousand vertices, were considered large scale. In those days, the world's most powerful computers (e.g., the IBM main frames) were far less capable than the iPhones of today. Even though these pioneers were mindful that constant factors did matter for practical computing, they used asymptotic notations — such as Big-O — to define complexity classes such as P and NP to characterize efficient algorithms and intractable problems [122, 100, 233, 189, 141]. Asymptotic complexity simplifies analyses and crucially puts the focus on the order of the leading complexity terms of algorithms [103, 209, 307]. However, with respect to the size of practical inputs at the time, the asymptotic world seemed to belong to the distant future.

Our pioneers persisted because they had a vision that one day computational problems would be massive, and therefore, the rate of complexity growth in relation to the growth of the inputs would be essential for characterizing the efficiency of an algorithm as well as the computational difficulty of a problem.

The asymptotic world has arrived with the rise of the Internet!

Today, the Web has grown into a massive graph with trillions of nodes; social networks and social media have generated an unbounded amount of digital records; smart phones have produced billions of images and videos; and tasks of science and engineering simulation have created equations and mathematical programs that involve hundreds of millions of variables. Even our computing devices have grown rapidly in size and complexity: In the middle 90s, the Intel Pentium processor had 2 millions transistors; today's PCs contain more than a billion.

While *Big Data* has taken algorithm design into the asymptotic world envisioned by our pioneers, the explosive growth of problem size has also significantly challenged the classical notion of *efficient algorithms*, particularly the use of *polynomial time* as the characterization of *efficient computation* [122]: Algorithms that used to be considered efficient (in the classification according to P) — such as a neat $O(n^2)$-time or $O(n^{1.5})$-time algorithm — may no longer be adequate for solving today's problems.

> *Therefore, more than ever before, it is not just desirable, but essential, that efficient algorithms should be* scalable. *In other words, their complexity should be nearly linear or sub-linear with respect to the problem size. Thus, scalability — not just polynomial-time computability — should be elevated as the central complexity notion for characterizing efficient computation.*

1.2 The Scalability of Algorithms

The *scalability* of an algorithm measures the growth of its complexity — such as the running time — in relation to the growth of the problem size. It measures the capacity of an algorithm to handle *big* inputs.

Suppose Π is a computational problem[1] whose input domain is Ω. For each instance $x \in \Omega$, let size(x) denote the *size* of input x. The input domain Ω can be viewed as the union of a collection of subdomains $\{..., \Omega_n, ...\}$, where Ω_n denotes the subset of Ω with input size n.

Now, suppose A is an algorithm for solving Π. For $x \in \Omega$, let $T_A(x)$ denote the time complexity for running $A(x)$. Instead of directly using *instance-based complexity* $T_A(x)$ to measure the performance of algorithm A for solving x, we consider the following related quality measure:

Definition 1.1 (Instance-Based Scalability). The *scalability* of an algorithm A for solving an instance $x \in \Omega$ is given by:

$$\text{scalability}(A, x) = \frac{T_A(x)}{\text{size}(x)} \tag{1.1}$$

We now *summarize* the instanced-based scalability of algorithm A over all instances in Ω_n as:[2]

$$\text{scalability}_A(n) := \sup_{x \in \Omega_n} \text{scalability}(A, x) = \sup_{x \in \Omega_n} \frac{T_A(x)}{\text{size}(x)}.$$

Then, scalability$_A(n)$ is a function that measures the growth of the complexity of A in relation to the growth of the problem. Let $T_A(n) = \sup_{x \in \Omega_n} T_A(x)$ denote the (worst-case) complexity of algorithm A on inputs of size n. Note that:

$$\text{scalability}_A(n) = \frac{T_A(n)}{n}.$$

Thus, A is a polynomial-time algorithm iff scalability$_A(n)$ is polynomial in n. However, the scalability measure puts the focus on scalable algorithms:

---

[1] See Appendix A.1.1 for a review of the basic types of computational problems.

[2] We may also use other *beyond worst-case* formulae for performance summarization [314]. For more discussion, see Section 8.4.

Definition 1.2 (Scalable Algorithms). An algorithm A is *scalable* if there exists a constant $c > 0$ such that:

$$\text{scalability}_A(n) = O(\log^c n).$$

In the special case when $c = 0$, we say A is *linearly-scalable*. We say algorithm A is *super-scalable*, if $\text{scalability}_A(n) = o(1)$. Super-scalable algorithms have a complexity that is sub-linear in problem size. Thus, necessarily, these algorithms must find solutions *without examining the entire input data set*. Sampling and local data/network exploration are two basic tools for designing super-scalable algorithms, and will be the subjects of Chapters 3 and 4.

***Remark**: One may say that* $\text{scalability}_A(n)$ *encodes no more information than* $T_A(n)$ *about algorithm A, because* $\text{scalability}_A(n) = T_A(n)/n$. *For example, A is scalable if and only if* $T_A(n)$ *is* nearly linear[3] *in* n *as referred to in [313]. However, the former identifies scalability as an essential concept for the characterization of efficient algorithms.*

> *The scalability measure puts the emphasis not on polynomial-time algorithms, but on scalable algorithms. It highlights the* exponential *gap between* $O(\log^c n)$ *and* n*, and between scalable algorithms and quadratic-time algorithms.*

To capture the essence of scalable algorithms, throughout the article, we will adopt the following commonly-used asymptotic notation.

Definition 1.3 ($\widetilde{O}$-Notation). For a given function $g(n)$, we denote by $\widetilde{O}(g(n))$ the set of functions:

$$\widetilde{O}(g(n)) = \{f(n) : \exists \text{ constant } c > 0, \text{ such that } f(n) = O(g(n)\log^c g(n))\}$$

For any positive integer n, we also use $\widetilde{O}_n(1)$ to denote:

$$\{f(n) : \exists \text{ constant } c > 0, \text{ such that } f(n) = O(\log^c n))\}.$$

---

[3] We say A is an *almost* linear-time algorithm (or is *almost scalable*) if $T_A(n) = O(n^{1+o(1)})$, i.e., $\text{scalability}_A(n) = n^{o(1)}$.

In other words, $\widetilde{O}$ is a variant of the asymptotic O-notation that hides additional poly-logarithmic factors. For example, $n \log^3 n = \widetilde{O}(n)$. With this notation, an algorithm is scalable if its scalability measure on an input of size n is $\widetilde{O}_n(1)$. In other words, its complexity on an input of size n is $\widetilde{O}(n)$.

Remarks: *The scalability analysis of algorithms is not unique to Big Data. For example, in parallel processing [71, 223], the notion of scalability is used to measure the efficiency of a parallel algorithm in utilizing parallel machines: Let $T_A(p,n)$ denote the parallel complexity of an algorithm A on a machine with p processors. So, the* speed-up *of this parallel algorithm with respect to the sequential one is $\frac{T_A(n)}{T_A(p,n)}$.*

Then, $\frac{T_A(n)}{T_A(p,n)} \cdot \frac{1}{p}$ measures the ratio of the achievable speedup to maximum-possible speedup by running A on p processors. In this context, a parallel algorithm A is linearly-scalable *if this ratio is bounded from below by a constant. Although the focus of scalability analysis of parallel algorithms is different from the scalability analysis of sequential algorithms, we can draw on insights from previous studies.*

1.3 Complexity Class S

A basic step in algorithm analysis and complexity theory is to characterize the family of problems that have efficient algorithmic solutions. In the world of *Big Data*, instead of using the traditional polynomial-time as the criterion for efficient algorithms, we require that efficient algorithms must be scalable.

Definition 1.4 (Complexity Class S)**.** We denote by S the set of computational problems that can be solved by a deterministic scalable algorithm.

In other words, a computational problem Π is in class S if there exists an algorithm A that solves Π with $\text{scalability}_A(n) = \widetilde{O}_n(1)$.

Complexity class S is analogous to the traditional complexity classes P and FP. In complexity theory, one usually classifies computational problems into three basic types [307, 103]: *decision problems*, *search* *problems*, and *optimization problems* (see Appendix A.1.1 for a quick review). Formalism is needed to precisely define complexity classes in order to address the subtlety among different types of computational problems. For example, to use polynomial time as the benchmark for efficient computation, the class P is usually reserved only for decision problems that can be solved in polynomial time without randomization [307]. The search version of complexity class P is known as FP, i.e., the class of functions that can be deterministically computed in polynomial time. In this article, we intend to be less formal in this regard, so that the focus will be on the notion of scalability rather than the difference between decision, search, and optimization problems.

Definition 1.5 (Complexity Class RS)**.** We denote by RS the set of computational problems that can be solved by a randomized scalable algorithm.

Randomization also introduces its own subtlety, which has given rise to classes such as BPP, RP, and ZPP for decision problems. For example, ZPP denotes the set of problems that can be solved by a *Las Vegas* algorithm with an expected polynomial runtime, which makes no errors in all instances. BPP and RP relax the latter condition by allowing the polynomial-time randomized algorithms to make bounded errors. Algorithms for RP are allowed to make errors only for YES instances, but algorithms for BPP are allowed to make errors for both YES and NO instances.

Again, with regard to randomization, we intend to be less formal as well so the focus will be on the notion of scalability rather than different types of errors.

***Remarks**: In computational complexity theory, one has to first define a computational model in order to define a complexity class. Commonly used models are Turing machines and random-access machines (RAM). It's well known that complexity classes, such as* P *and* FP*, are essentially robust with respect to these models.*

The scalable classes S *and* RS *can be formally defined according to these computational models. Their robustnesses with respect to computational models require further investigation, but are outside the scope of* *this article. As universal as these computational models are, data and network analysis programs on Turing machines or abstract random-access machines could be cumbersome.*

However, as Sipser said [307], "Real computers are quite complicated — too much so to allow us to set up a manageable mathematical theory of them directly." In the world of Big Data, massive networks, and large-scale optimization, without getting bogged down with details, I encourage readers to think about the real RAM model of Blum, Shub, and Smale [54], but with unit-cost computation of basic rational operations over reals at a given machine precision $\epsilon_{machine}$.

1.4 Scalable Reduction and Algorithmic Primitives

Algorithm design for scalable computing is like building a software library. Once we develop a new scalable algorithm, we can add it to our scalable library, and use it as a subroutine to design the next wave of scalable algorithms.

At the heart of this perspective is the notion of *scalable reduction.*

Definition 1.6 (Scalable Reduction). A computational problem Π (over domain Ω) is S-reducible to another computational problem Π' (over domain Ω'), denoted by $\Pi \leq_{\mathsf{S}} \Pi'$, if the following is true: Given any solver B for Π', there exists an algorithm A for solving Π such that for every instance $x \in \Omega$, $A(x)$ takes $\widetilde{O}\left(\text{size}(x)\right)$ steps including (i) generating a collection of instances $y_1, ..., y_{L(x)} \in \Omega'$ with:

$$\sum_{i=1}^{L(x)} \text{size}(y_i) = \widetilde{O}\left(\text{size}(x)\right)$$

and (ii) making calls to B on these instances.

In other words, $\Pi \leq_{\mathsf{S}} \Pi'$ if Π has a scalable *Turing-Cook-reduction* to Π'. In this definition, A is assumed to be a deterministic algorithm. We say $\Pi \leq_{\mathsf{RS}} \Pi'$ if we use a randomized algorithm A in the definition above. Directly from the definition, we have:

Proposition 1.7. (1) $\Pi \leq_{\mathsf{S}} \Pi'$ and $\Pi' \leq_{\mathsf{S}} \Pi''$ imply $\Pi \leq_{\mathsf{S}} \Pi''$. (2) $\Pi \leq_{\mathsf{RS}} \Pi'$ and $\Pi' \leq_{\mathsf{RS}} \Pi''$ imply $\Pi \leq_{\mathsf{RS}} \Pi''$.

Proposition 1.8. (1) If $\Pi \leq_{\mathsf{S}} \Pi'$ and $\Pi' \in \mathsf{S}$, then $\Pi \in \mathsf{S}$. (2) If $\Pi \leq_{\mathsf{RS}} \Pi'$ and $\Pi' \in \mathsf{RS}$, then $\Pi \in \mathsf{RS}$.

The field of computing has produced a number of remarkable scalable algorithms in various applicational domains. The following are a few examples:

- **Basic Algorithms**: FFT, merge sort, median selection, Huffman codes
- **Graph Algorithms**: minimum (maximum) spanning trees, shortest path trees, breadth-first search, depth-first search, connected components, strongly connected components, planar separators
- **Optimization Algorithms**: linear programming in constant dimensions
- **Probabilistic Algorithms**: VC-dimension based sampling, quick sort
- **Data structures**: many wonderful data structures
- **Numerical and Geometric Algorithms**: the multigrid method, the fast multipole method, 2D Delaunay triangulations and Voronoi diagrams, 3D convex hulls, quadtrees and its fixed dimensional extensions, ϵ-nets, nearest neighbors, and geometric separators in any fixed dimensions

More recently, *property testing* [154, 288], a subfield of theoretical computer science — inspired by PAC learning [340], holographic proofs

[37], and the PCP theorem [31] — has led to thriving developments of sub-linear-time algorithms [291]. These results, together with the work of Vapnik-Chervonenkis [342] and Johnson-Lindenstrauss [180], have demonstrated the power of sampling in scalable algorithm design. On the practical side, a rich body of scalable algorithms has been developed in fields of network science [184, 183, 230, 357, 354, 349, 64, 325, 324, 84], machine learning [182, 218, 143], and numerical computing [68, 190].

1.5 Article Organization

In this article, we will survey scalable algorithmic techniques, particularly those based on rapid progress in spectral graph theory, numerical analysis, probabilistic methods, and computational geometry. Many of these techniques are simple on their own, but together they form a powerful toolkit for designing scalable algorithms. After a brief review of network models in Chapter 2, we will proceed with the technical chapters of this article as follows:

In Chapter 3, "Significant Nodes: Sampling — Making Data Smaller," we will start by focusing on the smallest structures in big networks — ***nodes***. We will discuss results from [65] that incorporate efficient local network exploration methods into advanced sampling schemes.

Both sampling and local network exploration are widely used techniques for designing efficient algorithms. Here, the combination of the two leads to the first super-scalable algorithm for identifying *all* nodes with significant PageRank values in *any* network [65].[4] Leaving the details of local network exploration for the next chapter, this short chapter will focus on an *annealing* approach to construct robust PageRank estimators. This approach uses a multi-precision sampling scheme to

[4]This result was covered by Richard Lipton and Kenneth Regan, under title "Shifts In Algorithm Design," on their popular blog *Gödel's Lost Letter and P=NP* (https://rjlipton.wordpress.com/2014/07/21/shifts-in-algorithm-design/). The surprising conclusion that one can in fact identify all nodes with significant PageRank values without examining the entire network also landed this result on the list of "Top ten algorithms preprints of 2012" by David Eppstein (http://11011110.livejournal.com/260838.html).

guide a local *statistics-gathering algorithm*. We will show that this annealing method can build a robust PageRank estimator by visiting only a sub-linear number of nodes in the network. The reverse-structural techniques of this algorithm have been used in subsequent scalable algorithms for computing other network centrality measures [84], and for influence maximization [64, 325, 324].

> **Keywords**: *Markov processes; PageRank; personalized PageRank; PageRank matrix; multi-precision sampling; multi-precision annealing; Riemann estimator.*

In Chapter 4, "Clustering: Local Exploration of Networks," we will turn our attention to slightly larger structures in networks — ***clusters***. We will survey the basic framework of *local network exploration algorithms* introduced in [318]. These algorithms conduct *locally expandable searches* of networks: Starting from a small set of input nodes of a network, local algorithms iteratively expands its knowledge about the "hidden" network by only exploring the neighborhood structures of the explored nodes.

We first discuss a family of provably-good local clustering algorithms[5] [318, 22, 93, 24, 23]. We will then analyze two scalable algorithms for personalized-PageRank approximation. These local algorithms highlight the usefulness of graph-theoretical concepts, such as random walks, personalized PageRank, and conductance, for analyzing network structures, both mathematically and algorithmically. We then conclude the section with a study of the interplay between network structures and dynamic processes (such as random walks, social influence, or information propagation). In particular, we will discuss the framework introduced in [146] for quantifying the impact of this interplay on the *clusterability* of subsets in the network, and prove a parameterized Cheeger's inequality [82].

This chapter, together with Section 5.8 Chapter 6 and Chapter 7, also provide us with a quick tour through spectral graph theory.

---

[5]Several of these local clustering algorithms have been implemented by Konstantin Voevodski of Google (http://gaussian.bu.edu/lpcf.html) and used in the context of protein network analysis.

Keywords: *clusterability measures; conductance; Laplacian matrix; Cheeger's inequality; sweep with eigenvectors; local network-analysis algorithms; local clustering; power methods; random-walk sampling; interplay between dynamic processes and networks; spectral graph theory.*

In Chapter 5, "Partitioning: Geometric Techniques For Data Analysis," we will focus on ***networks defined by geometric data.*** We illustrate the power of geometric techniques — such as spatial decomposition and divide-and-conquer — in scalable data analysis. The geometric structures, such as nearest neighborhood graphs, also offer potentially useful measures of clusterability and cluster stability [38], which have both structural and algorithmic consequences. We will discuss a family of geometric partitioning techniques,[6] and apply them to the computation of nearest neighborhood graphs and geometric clusters. These techniques lead to powerful scalable geometric divide-and-conquer schemes [252, 253]. They also provide a beautiful bridge between spectral graph theory and network analysis, involving a popular spectral partitioning method [315].

Keywords: *nearest neighborhood graphs; geometric graphs; geometric partitioning; spectral partitioning; separator theorems; centerpoints; evolutionary algorithms, conformal maps; geometric divide-and-conquer; VC dimensions; random projection; spectral projection.*

In Chapter 6, "Spectral Similarity: Sparsification — Making Networks Simpler," we will focus on what is means to say "one network is similar to another network." We will address three basic questions:

- **Conceptual Question**: *How should we measure the similarity between two given networks?*
- **Mathematical Question**: *Does every graph have a "similar" sparse graph?*

[6] Many algorithms discussed in this chapter have been implemented [148] and are available in MESHPART, a Matlab mesh partitioning and graph separator toolbox (http://www.cerfacs.fr/algor/Softs/MESHPART/).

- **Algorithmic Question**: *Is there a scalable algorithm for constructing a good sparsifier?*

This is my favorite subject in spectral graph theory.

> **Keywords**: *spectral similarity; cut-similarity; effective resistances; spectral sparsification; low-stretch spanning tree; matrix sampling; conjugate gradient; PageRank completion of networks.*

In Chapter 7, "Electrical Flows: Laplacian Paradigm for Network Analysis," we survey a ***family of scalable algorithmic toolkits***. At the heart of these toolkits is a scalable solver for Laplacian linear systems and electrical flows [313, 318, 317, 319]. This scalable Laplacian solver has initiated and enabled many new scalable algorithms for spectral approximation [319], geometric and statistical approximation [98], graph embedding [335], machine learning [358], numerical methods, random-walk approximation, and Gaussian sampling in graphical models [86].

These applications illustrate a powerful, general algorithmic framework — called the *Laplacian paradigm* — for network analysis. In this framework, we attempt to reduce an optimization problem to one or more linear algebraic or spectral graph-theoretical problems that can be solved efficiently by the Laplacian solver or primitives from this scalable family. In addition to the applications above, this framework has also led to several recent breakthrough results — including scalable algorithms for max-flow/min-cut approximation [303, 197, 275, 243, 282] and sparse Newton's method [86] — that have gone beyond the original scalable Laplacian linear solvers [319, 215, 200, 97, 217].

These success stories point to an exciting future for scalable algorithm design in data and network analysis. I hope that continuing advancements will help enrich our scalable algorithmic library, and inspire new efficient solutions for data and network analysis.

> **Keywords**: *SDD primitive; electrical flows; Laplacian linear systems; spectral approximation; graph embedding; machine learning; Gaussian sampling; Gaussian Markov random fields; random walk sparsification; sparse Newton's method; Laplacian paradigm.*

In Chapter 8, "Remarks and Discussions," we conclude this article with a few "inconclusive" remarks. The inconclusiveness reflects the conceptual challenges that we usually face in data and network analysis. We also discuss a few frameworks beyond the commonly-used graph-theoretical network models to address fundamental conceptual questions in data analysis and network science.

Keywords: *centrality; clusterability; k-means methods; multifaceted network data; beyond graph-based network models; incentive networks; interplay between influence processes and social networks; social choice theory; axiomatization; game theory; cooperative games; Shapley value; behaviors of algorithms; beyond worst-case analysis.*

A.1 Appendix to the Chapter

A.1.1 Basic Types of Computational Problems

In complexity theory, one usually classifies basic computational problems into three types [307, 103]:

Decision Problems: A decision problem concerns the membership of a language $L \subseteq \{0,1\}^*$: Given an input string $x \in \{0,1\}^*$, one is asked to determine if $x \in L$. The output of a decision problem has constant complexity as the answer is either YES or NO (or DON'T KNOW, when randomization is used). This family of problems is commonly used to capture the computational challenge in deciding whether or not an input problem has a feasible solution.

Search Problems (or function problems:) A search problem typically works with a binary relation, $R \subseteq \{0,1\}^* \times \{0,1\}^*$: Given an input x, one is asked to determine if there exists y such that $(x,y) \in R$, and furthermore, when such a y exists, one must also produce an element from $\text{solution}_R(x) = \{y \in \{0,1\}^* | (x,y) \in R\}$. A search problem has two basic size measures: the size of the input x and the size of an output y. Thus, the complexity for solving a search problem is measured by a function in terms of either or both of these sizes. If the scalability of an algorithm A (for a search problem Π) is bounded by a poly-logarithmic function with respect to the size of the output it produces, then we say that A is *output-scalable*.

Optimization Problems: In a basic constrained optimization problem, we have a single utility/cost function u, whose value depends on multiple decision parameters $(x_1, x_2, ..., x_n)$. Each x_i has its own domain $x_i \in \Omega^{(i)}$, and the feasible region is given by a global constraint:

$$(x_1, x_2, ..., x_n) \in C.$$

The optimization problem could either be a maximization or a minimization problem:

$$\begin{array}{ll} \text{optimize} & u(x_1, x_2, ..., x_n) \\ \text{subject to} & (x_1, x_2, ..., x_n) \in C \quad \text{and} \quad x_i \in \Omega^{(i)}, \ \forall\ i. \end{array}$$

An input instance of an optimization problem is given by a representation of $(u, C, \Omega^{(1)} \times \cdots \times \Omega^{(n)})$. Like search problems, the complexity of an optimization problem (or an optimization algorithm) can be measured in terms of either or both input and output sizes.

Beyond Decision, Search, and Optimization: Other types of computational problems exist. For a binary relation R, for example, when given an input x, the *counting problem* aims to determine the number of solutions in $|\text{solution}_R(x)|$, the *enumeration problem* needs to identify all members in $\text{solution}_R(x)$, while the *sampling problem* generates an element from $\text{solution}_R(x)$, chosen according to uniform or a given distribution over $\text{solution}_R(x)$. The *multi-objective optimization problem* captures the potential trade-offs among several — possibly competing — objective functions in *multiple-criteria* decision-making:

$$\begin{array}{ll} \text{optimize} & u_1(x_1, x_2, ..., x_n), ..., u_k(x_1, x_2, ..., x_n) \\ \text{subject to} & (x_1, x_2, ..., x_n) \in C \quad \text{and} \quad x_i \in \Omega^{(i)}, \ \forall\ i. \end{array}$$

A basic solution concept for multi-objective optimization is the *Pareto set*, which contains every feasible solution not strictly dominated by other feasible solutions.

The *game-theoretical problem* captures possible compromises among multiple decision makers in strategic decision-making, where each decision maker has his/her own utility function and can only determine his/her own subset of decision parameters. The basic solution concept in game theory is the *Nash Equilibrium* [259, 258]. Schematically, there are n players. The i^{th} player has utility function u_i, and controls only input parameter $x_i \in \Omega^{(i)}$. These players have to jointly set their decision parameters in order to satisfy a global constraint: $(x_1, ..., x_n) \in C$. Each player's utility usually depends on all decision parameters. The domain $\Omega^{(i)}$ is referred to as the *strategy space* for player i. The simplest example of a game is a two-player matrix game [256, 259, 258].

We can formulate the computation of *Pareto points* and *Nash equilibria* as search problems [104, 268, 269, 272, 287]. However, we can capture more complex real-world phenomena using multi-objective optimization and game theory than with search and optimization [271, 269, 270].

A.1.2 Convention for Basic Notation

In this article, we will largely follow the conventions below for mathematical notation [314]:

- *Lower-case English and Greek letters*:
  - Scalar constants, variables, and functions
  - Vertices in a graph
- *Upper-case English and Greek letters*:
  - Sets and graphs
  - Distributions and events
  - Constants (occasionally)
- *Lower-case English and Greek letters in bold font*:
  - Vectors

    If an n-dimensional vector is denoted by $\boldsymbol{v}$, then we assume $\boldsymbol{v} = (v_1, v_2, ..., v_n)^T$, where T denotes the transpose operator. We use v_i or $\mathbf{v}[i]$ to denote the i^{th} entry of $\boldsymbol{v}$. We always assume $\mathbf{v}$ is a column vector. Thus, its transpose, $\mathbf{v}^T$, is a row vector.

  - Permutation (mostly in Greek letters)
    If $\boldsymbol{\pi}$ denotes a permutation of n elements, then we use π_i or $\boldsymbol{\pi}[i]$ to denote its i^{th} element.

- *Upper-case English and Greek letters in bold font*:
  - Matrices
    If an $m \times n$-matrix vector is denoted by $\boldsymbol{M}$, then we use $m_{i,j}$ or $\mathbf{M}[i,j]$ to denote its $(i,j)^{th}$ entry.

- *Special matrices and vectors*:
  - $\mathbf{I}_n$ denotes the identity matrix in n dimensions. When it is clear from the context, we use $\mathbf{I}$ to denote the identity matrix of the assumed dimensions.
  - $\mathbf{1}$ and $\mathbf{0}$, respectively, denote the vectors of all 1s or 0s in the assumed dimensions.
  - For $v \in [n]$, $\mathbf{1}_v$ denotes the n-place vector that has 1 at entry v and 0 everywhere else.
  - For $S \subset [n]$, $\mathbf{1}_S$ denotes the n-place vector that has 1 at entries in S and 0 everywhere else.

- $[n]$ or $[1:n]$ denotes the set of integers between 1 and n. More generally, for integers $a \leq b$, $[a:b]$ denotes the set of integers between a and b.

- *Matrix entry-wise inequality*
  - For two $m \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$, and parameters $\epsilon > 0$ and $c > 0$, we use $\mathbf{A} \leq c \cdot \mathbf{B} + \epsilon$ to denote $a_{i,j} \leq c \cdot b_{i,j} + \epsilon$, $\forall i \in [m], \forall j \in [n]$.

- Respectively, log and ln denote the logarithm base 2 and the natural logarithm.

- The indicator random variable for an event A is $\mathbf{I}[A]$ or $[A]$.

- Respectively, $\Pr_D[A]$ and $\mathrm{E}_D[X]$ denote the probability of event A and the expectation of variable X, over a distribution D.

References

[1] K. V. Aadithya, B. Ravindran, T. Michalak, and N. Jennings. Efficient computation of the Shapley value for centrality in networks. In *Internet and Network Economics*, volume 6484 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2010.

[2] E. Abbe and C. Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. *CoRR*, abs/1503.00609, 2015.

[3] E. Abbe and C. Sandon. Recovering communities in the general stochastic block model without knowing the parameters. *CoRR*, abs/1506.03729, 2015.

[4] I. Abraham, M. Babaioff, S. Dughmi, and T. Roughgarden. Combinatorial auctions with restricted complements. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 3–16, 2012.

[5] I. Abraham, Y. Bartal, and O. Neiman. Nearly tight low stretch spanning trees. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 781–790, Oct. 2008.

[6] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 2:781–793, 2002.

[7] A. Aho, J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[8] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Elsevier North-Holland, Inc., New York, NY, USA, 1989.

[9] D. Aldous and J. Fill. *Reversible Markov chains and random walks on graphs.* Berkeley, 2002. unifinished monograph.

[10] N. Alon. Problems and results in extremal combinatorics - I. *Discrete Mathematics*, 273(1-3):31–53, Dec. 2003.

[11] N. Alon, R. M. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k-server problem. *SIAM Journal on Computing*, 24(1):78–100, Feb. 1995.

[12] N. Alon and V. D. Milman. λ_1, Isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory*, 38(Series B):73–88, 1985.

[13] N. Alon, P. Seymour, and R. Thomas. A separator theorem for graphs with an excluded minor and its applications. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, STOC '90, pages 293–299, 1990.

[14] N. Alon and J. H. Spencer. *The Probabilistic Method.* Wiley, 3rd edition, 2008.

[15] A. Altman and M. Tennenholtz. Ranking systems: The PageRank axioms. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, EC '05, pages 1–8, 2005.

[16] S. Ambikasaran and E. Darve. An $O(N \log N)$ fast direct solver for partial hierarchically semi-separable matrices. *Journal of Scientific Computing*, 57(3):477–501, 2013.

[17] N. Amenta, M. Bern, D. Eppstein, and S.-H. Teng. Regression depth and center points. *Discrete & Computational Geometry*, 23(3):305–323, 2000.

[18] R. Andersen. A local algorithm for finding dense subgraphs. *ACM Transactions on Algorithms*, 6(4):60:1–60:12, Sep. 2010.

[19] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: An axiomatic approach. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 199–208, 2008.

[20] R. Andersen, C. Borgs, J. Chayes, J. Hopcroft, K. Jain, V. Mirrokni, and S. Teng. Robust PageRank and locally computable spam detection features. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '08, pages 69–76, 2008.

[21] R. Andersen, C. Borgs, J. T. Chayes, J. E. Hopcroft, V. S. Mirrokni, and S.-H. Teng. Local computation of PageRank contributions. *Internet Mathematics*, 5(1):23–45, 2008.

[22] R. Andersen, F. Chung, and K. Lang. Using PageRank to locally partition a graph. *Internet Mathematics.*, 4(1):1–128, 2007.

[23] R. Andersen, S. O. Gharan, Y. Peres, and L. Trevisan. Almost optimal local graph clustering using evolving sets. *Journal of the ACM*, 63:Article No. 15, 2015.

[24] R. Andersen and Y. Peres. Finding sparse cuts locally using evolving sets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 235–244, 2009.

[25] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, Jan. 2008.

[26] S. Arora, A. Bhaskara, R. Ge, and T. Ma. Provable bounds for learning some deep representations. In *Proceedings of the International Conference on Machine Learning*, ICML'14, pages 584–592, 2014.

[27] S. Arora, R. Ge, Y. Halpern, D. M. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference on Machine Learning*, ICML"13, pages 280–288, 2013.

[28] S. Arora, R. Ge, T. Ma, and A. Moitra. Simple, efficient, and neural algorithms for sparse coding. *CoRR*, abs/1503.00778, 2015.

[29] S. Arora, R. Ge, A. Moitra, and S. Sachdeva. Provable ICA with unknown Gaussian noise, and implications for Gaussian mixtures and autoencoders. *Algorithmica*, 72(1):215–236, May 2015.

[30] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[31] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[32] K. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58, 1950.

[33] K. J. Arrow. *Social Choice and Individual Values.* Wiley, New York, 2nd edition, 1963.

[34] D. Arthur, B. Manthey, and H. Röglin. Smoothed analysis of the k-means method. *Journal of the ACM*, 58(5):19:1–19:31, Oct. 2011.

[35] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke. Optimal oblivious routing in polynomial time. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, STOC '03, pages 383–388, 2003.

[36] L. Babai. Graph isomorphism in quasipolynomial time. *CoRR*, abs/1512.03547, 2015.

[37] L. Babai, L. Fortnow, L. A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, 1991.

[38] M.-F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 1068–1077, 2009.

[39] M. F. Balcan, C. Borgs, M. Braverman, J. T. Chayes, and S.-H. Teng. Finding endogenously formed communities. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 767–783, 2013.

[40] M.-F. Balcan and M. Braverman. Finding low error clusterings. In *Conference on Learning Theory*, 2009.

[41] J. F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[42] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[43] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng. Spectral sparsification of graphs: Theory and algorithms. *Communications of the ACM*, 56(8):87–94, Aug. 2013.

[44] J. D. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, STOC, pages 255–262, 2009.

[45] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, 1st edition, 1998.

[46] A. Bavelas. Communication patterns in task oriented groups. *Journal of the Acoustical Society of America*, 22(6):725–730, Nov. 1950.

[47] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[48] P. Belleflamme. Stable coalition structures with open membership and asymmetric firms. *Games and Economic Behavior*, 30(1):1–21, 2000.

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

[50] M. Bern, D. Eppstein, and J. R. Gilbert. Provably good mesh generation. In *The 31st Annual Symposium on Foundations of Computer Science*, pages 231–241. IEEE, 1990.

[51] M. Bern, D. Eppstein, and S.-H. Teng. Parallel construction of quadtrees and quality triangulations. *International Journal of Computational Geometry & Applications*, 9(06):517–532, 1999.

[52] P. Biswal, J. R. Lee, and S. Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *Journal of the ACM*, 57(3):13:1–13:23, Mar. 2010.

[53] A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.

[54] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society, N. S.*, 21(1):1–46, Jul. 1989.

[55] B. Bollobás. *Modern graph theory.* Springer-Verlag, 1998.

[56] E. G. Boman, D. Chen, O. Parekh, and S. Toledo. On factor width and symmetric H-matrices. *Linear Algebra and Its Applications*, 405:239–248, 2005.

[57] E. G. Boman, B. Hendrickson, and S. Vavasis. Solving elliptic finite element systems in near-linear time with support preconditioners. *SIAM Journal Numerical Analysis*, 46(6):3264–3284, Oct. 2008.

[58] P. Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[59] P. Bonacich. Simultaneous group and individual centralities. *Social Networks*, 13(2):155 – 168, 1991.

[60] O. N. Bondareva. Some applications of the methods of linear programming to the theory of cooperative games. *Problemy Kibernet.*, 10:119–139, 1963.

[61] R. B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 280–285, 1987.

[62] S. P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[63] S. P. Borgatti and M. G. Everett. A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, 2006.

[64] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 946–957, 2014.

[65] C. Borgs, M. Brautbar, J. Chayes, and S.-H. Teng. Multi-scale matrix sampling and sublinear-time PageRank computation. *Internet Mathematics*, 10(1-2):20–48, 2014.

[66] C. Borgs, J. T. Chayes, A. Marple, and S. Teng. An axiomatic approach to community detection. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA*, pages 135–146, 2016.

[67] S. J. Brams, M. A. Jones, and D. M. Kilgour. Dynamic models of coalition formation: Fallback vs. build-up. In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '03, pages 187–200, 2003.

[68] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, Apr. 1977.

[69] A. Brandt. *Multilevel computations: Review and recent developments, edited by McCormick, S.F.*, pages 35–62. Marcel-Dekker, 1988.

[70] M. Braverman, Y. K. Ko, A. Rubinstein, and O. Weinstein. ETH hardness for densest-k-subgraph with perfect completeness. *CoRR*, arXiv:1504.08352, 2015.

[71] R. Brent. The parallel evaluation of general arithmetic expressions. *Journal of the ACM*, 21(2):201–208, Apr. 1974.

[72] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial (2Nd Ed.)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[73] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[74] A. Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*, SEQUENCES '97, pages 21–29, 1997.

[75] L. E. J. Brouwer. Ü ber Abbildung von Mannigfaltigkeitn. *Math. Annale*, 71:97–115, 1912.

[76] Z. Burda, J. Duda, J. M. Luck, and B. Waclaw. Localization of the maximal entropy random walk. *Physical Review Letters*, 102(16):160602, Apr. 2009.

[77] M. Caesar and J. Rexford. BGP routing policies in ISP networks. *Netwrk. Mag. of Global Internetwkg.*, 19(6):5–11, Nov. 2005.

[78] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM*, 42(1):67–90, Jan. 1995.

[79] D. Chakrabarti and C. Faloutsos. *Graph mining: laws, tools, and case studies.* Synthesis lectures on data mining and knowledge discovery. Morgan & Claypool, 2012.

[80] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16:361–368, 1996.

[81] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388, 2002.

[82] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in Analysis, Edited by R. C. Gunning*, pages 195–199. Princeton University Press, 1970.

[83] W. Chen and S.-H. Teng. Interplay between influence processes and social networks II: Clusterability and personalized Shapley values. 2016.

[84] W. Chen and S.-H. Teng. Interplay between social influence and network centrality: Shapley values and scalable algorithms. *CoRR*, abs/1602.03780, 2016.

[85] X. Chen and S. Teng. A complexity view of markets with social influence. In *Proceedings in Innovations in Computer Science - Tsinghua University, Beijing, China*, ICS, pages 141–154, 2011.

[86] D. Cheng, Y. Cheng, Y. Liu, R. Peng, and S.-H. Teng. Efficient sampling for Gaussian graphical models via spectral sparsification. In *Proceedings of the 28th Conference on Learning Theory*, COLT '05, 2015.

[87] S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, and S.-H. Teng. Silver exudation. *Journal of the ACM*, 47(5):883–904, Sept. 2000.

[88] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[89] P. Chew. There are planar graphs almost as good as the complete graph. *JCSS*, 39:205–219, 1989.

[90] P. Chin, A. Rao, and V. Vu. Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. In *Proceedings of The 28th Conference on Learning Theory*, pages 391–423, 2015.

[91] P. Christiano, J. A. Kelner, A. Mądry, D. A. Spielman, and S.-H. Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 273–282, 2011.

[92] F. Chung. The heat kernel as the PageRank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, Dec. 2007.

[93] F. Chung. A local graph partitioning algorithm using heat kernel Pagerank. *Internet Mathematics*, 6(3):315–330, Jan. 2009.

[94] F. Chung and L. Lu. *Complex Graphs and Networks (CBMS Regional Conference Series in Mathematics)*. American Mathematical Society, Boston, MA, USA, 2006.

[95] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, Feb. 1997.

[96] K. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with and without linear programming. In *Proceedings of 9th ACM Symposium on Computational Geometry*, pages 91–98, 1993.

[97] M. B. Cohen, R. Kyng, G. L. Miller, J. W. Pachocki, R. Peng, A. B. Rao, and S. C. Xu. Solving sdd linear systems in nearly $m \log^{1/2} n$ time. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC, pages 343–352, 2014.

[98] M. B. Cohen, Y. T. Lee, G. L. Miller, J. W. Pachocki, and A. Sidford. Geometric median in nearly linear time. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, STOC, 2016.

[99] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1988.

[100] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971.

[101] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[102] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, 9(3):251–280, Mar. 1990.

[103] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

[104] D. Johnson and C.H. Papadimitriou and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

[105] S. I. Daitch and D. 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.

[106] L. Danzer, J. Fonlupt, and V. Klee. Helly's theorem and its relatives. *Proceedings of Symposia in Pure Mathematics, American Mathematical Society*, 7:101–180, 1963.

[107] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the 20th Annual Symposium on Computational Geometry*, SCG '04, pages 253–262, 2004.

[108] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[109] E. David and K. Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY, USA, 2010.

[110] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 145–156, 2000.

[111] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[112] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, May 1994.

[113] J. Ding, J. R. Lee, and Y. Peres. Cover times, blanket times, and majorizing measures. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, STOC '11, pages 61–70, 2011.

[114] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 57–66, 2001.

[115] W. E. Donath and A. J. Hoffman. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. *IBM Technical Disclosure Bulletin*, 15:938 – 944, 1972.

[116] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *Journal Research Developments*, 17:420 – 425, 1973.

[117] L. Donetti, P. I. Hurtado, and M. A. Munoz. Entangled networks, synchronization, and optimal network topology. *CoRR*, Oct. 2005.

[118] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, Apr. 2006.

[119] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 1984.

[120] J. Dunagan, D. A. Spielman, and S.-H. Teng. Smoothed analysis of condition numbers and complexity implications for linear programming. *Math. Program.*, 126(2):315–350, Feb. 2011.

[121] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC'06, pages 265–284, 2006.

[122] J. Edmonds. Maximum matching and a polyhedron with $0, 1$ vertices. *Journal of Research at the National Bureau of Standards*, 69 B:125–130, 1965.

[123] Y. Eidelman and I. Gohberg. Inversion formulas and linear complexity algorithm for diagonal plus semiseparable matrices. *Computers & Mathematics with Applications*, 33(4):69 – 79, 1997.

[124] P. Elias, A. Feinstein, and C. E. Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2, 1956.

[125] M. Elkin, Y. Emek, D. A. Spielman, and S.-H. Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 32(2):608–628, 2008.

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

[127] K. Faust. Centrality in affiliation networks. *Social Networks*, 19(2):157–191, Apr. 1997.

[128] U. Feige, M. Feldman, N. Immorlica, R. Izsak, B. Lucier, and V. Syrgkanis. A unifying hierarchy of valuations with complements and substitutes. *CoRR*, abs/1408.1211, 2014.

[129] U. Feige, N. Immorlica, V. S. Mirrokni, and H. Nazerzadeh. Pass approximation: A framework for analyzing and designing heuristics. *Algorithmica*, 66(2):450–478, 2013.

[130] U. Feige and R. Izsak. Welfare maximization and the supermodular degree. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 247–256, 2013.

[131] U. Feige and J. Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 674, 1998.

[132] U. Feige and R. Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM Journal on Computing*, 31:1090–1118, 2002.

[133] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298 – 305, 1973.

[134] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czechoslovak Mathematical Journal*, 25(100):619 – 633, 1975.

[135] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[136] J. G. F. Francis. The QR transformation a unitary analogue to the LR transformation: Part 1. *The Computer Journal*, 4(3):256 – 271, 1961.

[137] L. C. Freeman. A set of measures of centrality based upon betweenness. *Sociometry*, 40:35–41, 1977.

[138] L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[139] A. M. Frieze, G. L. Miller, and S.-H. Teng. Separator based parallel divide and conquer in computational geometry. In *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '92, pages 420–429, 1992.

[140] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[141] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

[142] H. Gazit and G. L. Miller. Planar separators and the Euclidean norm. In *Proceedings of the International Symposium on Algorithms*, SIGAL '90, pages 338–347, 1990.

[143] R. Ge. *Provable algorithms for machine learning problems.* PhD thesis, Princeton University, 2013.

[144] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

[145] R. Ghosh and K. Lerman. Predicting influential users in online social networks. In *Proceedings of KDD workshop on Social Network Analysis (SNAKDD)*, May 2010.

[146] R. Ghosh, S.-H. Teng, K. Lerman, and X. Yan. The interplay between dynamics and networks: Centrality, communities, and Cheeger inequality. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1406–1415, 2014.

[147] J. R. Gilbert, J. P. Hutchinson, and R. E. Tarjan. A separator theorem for graphs of bounded genus. *Journal of Algorithms*, 5(3):391–407, Sept. 1984.

[148] J. R. Gilbert, G. L. Miller, and S.-H. Teng. Geometric mesh partitioning: Implementation and experiments. *SIAM Journal on Scientific Computing*, 19(6):2091–2110, Nov. 1998.

[149] J. R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in Matlab: design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, Jan. 1992.

[150] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 518–529, 1999.

[151] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, Nov. 1995.

[152] I. Gohberg, T. Kailath, and I. Koltracht. Linear complexity algorithms for semiseparable matrices. *Integral Equations and Operator Theory*, 8(6):780–804, 1985.

[153] O. Goldreich. Introduction to testing graph properties. In O. Goldreich, editor, *Property Testing*, pages 105–141. Springer-Verlag, Berlin, Heidelberg, 2010.

[154] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, July 1998.

[155] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins Press, 2nd edition, 1989.

[156] D. Gómez, E. Gonzalez-Arangüena, C. Manuel, G. Owen, M. del Pozo, and J. Tejada. Centrality and power in social networks: a game theoretic approach. *Mathematical Social Sciences*, 46(1):27–54, 2003.

[157] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, Dec. 1987.

[158] L. F. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. PhD thesis, Yale University, New Haven, CT, USA, 1987.

[159] B. Grofman and G. Owen. A game theoretic approach to measuring degree of centrality in social networks. *Social Networks*, 4(3):213 – 224, 1982.

[160] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, USA, 1989.

[161] W. Hackbusch. A sparse matrix arithmetic based on H-matrices. part I: Introduction to H-matrices. *Computing*, 62(2):89–108, 1999.

[162] K. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17:219 – 229, 1970.

[163] J. Hammersley and P. Clifford. *Markov fields on finite graphs and lattices*. Unpublished manuscript, 1971.

[164] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[165] S. Hanneke and E. P. Xing. Network completion and survey sampling. In D. A. V. Dyk and M. Welling, editors, *AISTATS*, volume 5 of *JMLR Proceedings*, pages 209–215, 2009.

[166] S. Hart and M. Kurz. Endogenous formation of coalitions. *Econometrica*, 51(4):1047–1064, 1983.

[167] J. Hartmanis and R. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285 – 306, 1965.

[168] T. Haveliwala. Topic-sensitive Pagerank: A context-sensitive ranking algorithm for web search. In *IEEE Transactions on Knowledge and Data Engineering*, volume 15(4), pages 784–âĂŞ796, 2003.

[169] K. He, S. Soundarajan, X. Cao, J. E. Hopcroft, and M. Huang. Revealing multiple layers of hidden community structure in networks. *CoRR*, abs/1501.05700, 2015.

[170] M. T. Heath and W. A. Dick. Virtual prototyping of solid propellant rockets. *Computing in Science and Engineering*, 2(2):21–32, Mar. 2000.

[171] J. Heinonen. *Lectures on analysis on metric spaces*. Universitext. Springer, 2001.

[172] M. Hubert and P. J. Rousseeuw. The catline for deep regression. *J. Multivariate Analysis*, 66:270–296, 1998.

[173] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, 1952.

[174] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, 1998.

[175] M. O. Jackson. *Social and Economic Networks*. Princeton University Press, Princeton, NJ, USA, 2008.

[176] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2(1):18–21, 1973.

[177] G. Jeh and J. Widom. Scaling personalized Web search. In *WWW*, pages 271–279, 2003.

[178] M. Jerrum and A. Sinclair. Conductance and the rapid mixing property for Markov chains: the approximation of permanent resolved. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, STOC, pages 235–244. ACM, 1988.

[179] M. Johnson, J. Saunderson, and A. Willsky. Analyzing hogwild parallel Gaussian Gibbs sampling. In *Advances in Neural Information Processing Systems 26*, pages 2715–2723. Curran Associates, Inc., 2013.

[180] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(1-1.1):189–206, 1984.

[181] M. I. Jordan. *Learning in Graphical Models*. The MIT press, 1998.

[182] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[183] U. Kang, D. H. Chau, and C. Faloutsos. Mining large graphs : Algorithms , inference , and discoveries. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 243—-254, 2011.

[184] U. Kang and C. Faloutsos. Big graph mining: Algorithms and discoveries. *SIGKDD Explorations Newsletter*, 14(2):29–36, Apr. 2013.

[185] R. Kannan and S. Vempala. Spectral algorithms. *Foundations and Trends® in Theoretical Computer Science*, 4(3-4):157–288, 2009.

[186] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, May 2004.

[187] D. R. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, STOC '02, pages 741–750, 2002.

[188] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, STOC '84, pages 302–311, 1984.

[189] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[190] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1), Dec. 1998.

[191] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, Mar. 1953.

[192] M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 253–260, 2001.

[193] M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

[194] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, Nov. 1994.

[195] J. Kelner, J. R. Lee, G. Price, , and S.-H. Teng. Metric uniformization and spectral bounds for graphs. *Geom. Funct. Anal.*, 21(5):1117–1143, 2011.

[196] J. A. Kelner. Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus. *SIAM Journal on Computing.*, 35(4):882–902, Apr. 2006.

[197] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'14, pages 217–226, 2014.

[198] J. A. Kelner and A. Levin. Spectral sparsification in the semi-streaming setting. *Theory of Computing Systems*, 53(2):243–262, 2013.

[199] J. A. Kelner and A. Mądry. Faster generation of random spanning trees. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 13–21, 2009.

[200] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 911–920, 2013.

[201] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, pages 137–146. ACM, 2003.

[202] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979.

[203] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *SDM*, pages 47–58. SIAM / Omnipress, 2011.

[204] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm. *SIAM Journal on Computing.*, 15(1):287–299, Feb. 1986.

[205] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[206] M. Kiwi, D. A. Spielman, and S.-H. Teng. Min-max-boundary domain decomposition. *Theoretical Computer Science*, 261:2001, 1998.

[207] J. Kleinberg. An impossibility theorem for clustering. In *NIPS*, pages 463–470, 2002.

[208] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. *Journal of the ACM*, 51(2):263–280, Mar. 2004.

[209] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[210] J. Kleinberg and E. Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, pages 295–304, 2008.

[211] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, Sept. 1999.

[212] R. Kleinberg. Geographic routing using hyperbolic space. In *Proceedings of the 26th Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM 2007, pages 1902–1909, 2007.

[213] P. Koebe. Kontaktprobleme der konformen abbildung. *Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Kl.*, 88:141âĂŞ164, 1936.

[214] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computa tion and Machine Learning*. The MIT Press, 2009.

[215] I. Koutis, G. Miller, and R. Peng. A nearly-mlogn time solver for SDD linear systems. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, FOCS, pages 590–598, 2011.

[216] I. Koutis, G. L. Miller, and R. Peng. Approaching optimality for solving SDD systems. In *Proceedings of the 51st Annual Symposium on Foundations of Computer Science*, FOCS, pages 235–244, 2010.

[217] R. Kyng and S. Sachdeva. Approximate gaussian elimination for laplacians: Fast, sparse, and simple. *CoRR*, abs/1605.02353, 2016.

[218] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, 2009.

[219] Y. T. Lee and A. Sidford. Following the path of least resistance : An Õ(m sqrt(n)) algorithm for the minimum cost flow problem. *CoRR*, abs/1312.6713, 2013.

[220] Y. T. Lee and A. Sidford. Matching the universal barrier without paying the costs : Solving linear programs with Õ(sqrt(rank)) linear system solves. *CoRR*, abs/1312.6677, 2013.

[221] Y. T. Lee and H. Sun. Constructing linear-sized spectral sparsification in almost-linear time. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 250–269, Oct 2015.

[222] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, EC '01, pages 18–28, 2001.

[223] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Array, Trees, Hypercubes*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[224] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, Nov. 1999.

[225] C. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers*, 34(10):892–901, Oct. 1985.

[226] C. E. Leiserson. *Area-Efficient VLSI Computation*. MIT Press, Cambridge, MA, USA, 1997.

[227] K. Lerman and R. Ghosh. Network structure, topology and dynamics in generalized models of synchronization. *Physical Review E*, 86(026108), 2012.

[228] K. Lerman, S.-H. Teng, and X. Yan. Network composition from multilayer data. USC Information Sciences Institute.

[229] J. Leskovec. *Dynamics of Large Networks*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2008. AAI3340652.

[230] J. Leskovec and C. Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 497–504, 2007.

[231] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, pages 29–123, 2009.

[232] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of 17th International World Wide Web Conference (WWW2008)*, 2008.

[233] L. Levin. Universal sorting problems. *Problems of Information Transmission*, 9:265 – 266, 1973.

[234] X. Y. Li and S.-H. Teng. Sliver-free three dimensional Delaunay mesh generation. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 28–37, 2001.

[235] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM J. on Numerical Analysis*, 16, 1979.

[236] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36:177–189, 1979.

[237] Y. Liu, O. Kosut, and A. S. Willsky. Sampling from Gaussian graphical models using subgraph perturbations. In *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, Jul. 7-12, 2013*, 2013.

[238] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, Sept. 2006.

[239] L. Lovász and M. Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *Proceedings: 31st Annual Symposium on Foundations of Computer Science*, pages 346–354, 1990.

[240] L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *RSA: Random Structures & Algorithms*, 4:359–412, 1993.

[241] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[242] A. Mądry. Fast approximation algorithms for cut-based problems in undirected graphs. In *FOCS'10: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 245–254, 2010.

[243] A. Mądry. Navigating central path with electrical flows: from flows to matchings, and back. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, FOCS, pages 253–262, 2013.

[244] G. A. Margulis. Explicit group theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators. *Problems of Information Transmission*, 24(1):39–46, Jul. 1988.

[245] F. Masrour, I. Barjesteh, R. Forsati, A.-H. Esfahanian, and H. Radha. Network completion with node similarity: A matrix completion approach with provable guarantees. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '15, pages 302–307. ACM, 2015.

[246] T. Mattson, D. A. Bader, J. W. Berry, A. Buluç, J. J. Dongarra, C. Faloutsos, J. Feo, J. R. Gilbert, J. Gonzalez, B. Hendrickson, J. Kepner, C. E. Leiserson, A. Lumsdaine, D. A. Padua, S. W. Poole, S. P. Reinhardt, M. Stonebraker, S. Wallach, and A. Yoo. Standards for graph algorithm primitives. *CoRR*, abs/1408.0393, 2014.

[247] N. Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31(1):114–127, Jan. 1984.

[248] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[249] T. P. Michalak, K. V. Aadithya, P. L. Szczepanski, B. Ravindran, and N. R. Jennings. Efficient computation of the Shapley value for game-theoretic network centrality. *J. Artif. Int. Res.*, 46(1):607–650, Jan. 2013.

[250] G. L. Miller. Riemann's hypotheis and tests for primality. *Journal of Computer and System Sciences*, 13(3):300–317, Dec. 1976.

[251] G. L. Miller, D. Talmor, S.-H. Teng, and N. Walkington. A Delaunay based numerical method for three dimensions: Generation, formulation, and partition. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, STOC '95, pages 683–692, 1995.

[252] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *Journal of the ACM*, 44(1):1–29, Jan. 1997.

[253] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Geometric separators for finite-element meshes. *SIAM Journal on Scientific Computing*, 19(2):364–386, Mar. 1998.

[254] N. Mishra, R. Schreiber, I. Stanton, and R. Tarjan. Finding strongly-knit clusters in social networks. *Internet Mathematics*, pages 155–174, 2009.

[255] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in three dimensions. In *Proceedings of the ACM Computational Geometry Conference*, pages 212–221, 1992. Also appeared as Cornell C.S. TR 92-1267.

[256] O. Morgenstern and J. von Neumann. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[257] R. Narayanam and Y. Narahari. A Shapley value-based approach to discover influential nodes in social networks. *Automation Science and Engineering, IEEE Transactions on*, PP(99):1 –18, 2010.

[258] J. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of the USA*, 36(1):48–49, 1950.

[259] J. Nash. Noncooperative games. *Annals of Mathematics*, 54:289–295, 1951.

[260] M. Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[261] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 2006.

[262] N. Nisan and A. Ronen. Algorithmic mechanism design (extended abstract). In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, STOC '99, pages 129–140, 1999.

[263] G. Niu, B. Recht, C. Re, and S. J. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Conference on Neural Information Processing Systems*, NIPS, 2011.

[264] E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai network: A platform for high-performance Internet applications. *SIGOPS Operating Systems Review*, 44(3):2–19, Aug. 2010.

[265] L. Orecchia, S. Sachdeva, and N. K. Vishnoi. Approximating the exponential, the lanczos method and an Õ(m)-time spectral algorithm for balanced separator. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1141–1160, 2012.

[266] L. Page, S. Brin, R. Motwani, and T. Winograd. The Pagerank citation ranking: Bringing order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, 1998.

[267] I. Palacios-Huerta and O. Volij. The measurement of intellectual influence. *Econometrica*, 72:963–977, 2004.

[268] C. Papadimitriou. On graph-theoretic lemmata and complexity classes. In *Proceedings 31st Annual Symposium on Foundations of Computer Science*, pages 794–801, 1990.

[269] C. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, pages 498–532, 1994.

[270] C. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, STOC, pages 749–753, 2001.

[271] C. Papadimitriou, A. Schaffer, and M. Yannakakis. On the complexity of local search. In *Proceedings of the 22th Annual ACM Symposium on Theory of Computing*, pages 438–445, 1990.

[272] C. H. Papadimitriou and M. Yannakakis. Multiobjective query optimization. In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '01, pages 52–59, 2001.

[273] D. Peleg and A. A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13:99–116, 1989.

[274] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM Journal on Computing.*, 18(4):740–747, Aug. 1989.

[275] R. Peng. Approximate undirected maximum flows in O(mpolylog(n)) time. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1862–1867, 2016.

[276] L. S. Penrose. The elementary statistics of majority voting. *Journal of the Royal Statistical Society*, 109(1):53–57, 1946.

[277] M. Piraveenan, M. Prokopenko, and L. Hossain. Percolation centrality: Quantifying graph-theoretic impact of nodes during percolation in networks. *PLoS ONE*, 8(1), 2013.

[278] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, May 1990.

[279] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[280] M. O. Rabin. *Probabilistic Algorithms.* Academic Press, New York, 1976.

[281] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, pages 255–264, 2008.

[282] H. Räcke, C. Shah, and H. Täubig. Computing cut-based hierarchical decompositions in almost linear time. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 227–238, 2014.

[283] D. Ray. *A game-theoretic perspective on coalition formation.* The Lipsey lectures. Oxford University Press, Oxford, 2007.

[284] Y. Rekhter and T.Li. A Border Gateway Protocol 4. *RFC 1771, Internet Engineering Task Force*, 1995.

[285] J. Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM Journal on Optimization*, 5:5–3, 1995.

[286] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 61–70, 2002.

[287] H. Röglin and S.-H. Teng. Smoothed analysis of multiobjective optimization. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 681–690, 2009.

[288] D. Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends® in Theoretical Computer Science*, 5(2):73–205, 2010.

[289] A. E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, 92:991–1016, 1984.

[290] A. E. Roth. *Stable Coalition Formation: Aspects of a Dynamic Theory*, pages 228–234. Wuerzberg: Physica-Verlag, 1984.

[291] R. Rubinfeld and A. Shapira. Sublinear time algorithms. *SIAM Journal on Discrete Mathematics*, 25(4):1562–1588, Nov. 2011.

[292] M. Rudelson. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1):60 – 72, 1999.

[293] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *The 4th ACM-SIAM Symposium on Discrete Algorithms*, pages 83–92. SIAM, 1993.

[294] G. Sabidussi. The centrality index of a graph. *Psychometirka*, 31:581–606, 1996.

[295] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.

[296] H. E. Scarf. The core of an N person game. *Econometrica*, 69:35–50, 1967.

[297] A. Schrijver. *Combinatorial Optimization, Volume A*. Number 24 in Algorithms and Combinatorics. Springer, 2003.

[298] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, STOC '86, pages 404–413, 1986.

[299] R. Seidel. Linear programming and convex hulls made easy. In *Proceedings of the 6th Annual Symposium on Computational Geometry*, SCG '90, pages 211–215, 1990.

[300] L. S. Shapley. A value for n-person games. In H. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.

[301] L. S. Shapley. On balanced sets and cores. *Naval Research Logistics*, 14, 1967.

[302] L. S. Shapley. Cores of convex games. *International Journal of Game Theory*, 1(1):11 – 26, 1971.

[303] J. Sherman. Nearly maximum flows in nearly linear time. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 263–269, 2013.

[304] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1-3):21–74, May 2002.

[305] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[306] H. D. Simon and S.-H. Teng. How good is recursive bisection? *SIAM Journal on Scientific Computing*, 18(5):1436–1445, Sept. 1997.

[307] M. Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1st edition, 1996.

[308] L. M. Smith, K. Lerman, C. Garcia-Cardona, A. G. Percus, and R. Ghosh. Spectral clustering with epidemic diffusion. *Physical Review E*, 88(4):042813, Oct. 2013.

[309] R. M. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6:84–85, 1977.

[310] E. Sperner. Neuer Beweis fur die Invarianz der Dimensionszahl und des Gebietes. *Abhandlungen aus dem Mathematischen Seminar Universitat Hamburg*, 6:265–272, 1928.

[311] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th ACM Symposium on the Theory of Computing*, pages 563–568, 2008.

[312] D. A. Spielman and S.-H. Teng. Disk packings and planar separators. In *Proceedings of the 12th Annual Symposium on Computational Geometry*, SCG '96, pages 349–358, 1996.

[313] D. A. Spielman and S.-H. 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*, STOC '04, pages 81–90, 2004.

[314] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[315] D. A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2-3):284–305, mar 2007.

[316] D. A. Spielman and S.-H. Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, Oct. 2009.

[317] D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM Journal on Computing.*, 40(4):981–1025, July 2011.

[318] D. A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing.*, 42(1):1–26, 2013.

[319] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[320] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Wiley, 2nd edition, 2008.

[321] J. Suomela. Survey of local algorithms. *ACM Computing Surveys*, 45(2):24:1–24:40, Feb. 2013.

[322] N. R. Suri and Y. Narahari. Determining the top-k nodes in social networks using the Shapley value. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '08, pages 1509–1512, 2008.

[323] P. L. Szczepański, T. Michalak, and T. Rahwan. A new approach to betweenness centrality based on the Shapley value. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pages 239–246, 2012.

[324] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 1539–1554, 2015.

[325] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: near-optimal time complexity meets practical efficiency. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD, pages 75–86, 2014.

[326] S.-H. Teng. Combinatorial aspects of geometric graphs. *Computational Geometry: Theory and Applications*, 9(4):277–287, Mar. 1998.

[327] S.-H. Teng. Provably good partitioning and load balancing algorithms for parallel adaptive n-body simulation. *SIAM Journal on Scientific Computing*, 19(2):635–656, Mar. 1998.

[328] S.-H. Teng. *Coarsening, Sampling, and Smoothing: Elements of the Multilevel Method*, pages 247–276. Springer, New York, 1999.

[329] S.-H. Teng. The Laplacian Paradigm: Emerging algorithms for massive graphs. In *Proceedings of the 7th Annual Conference on Theory and Applications of Models of Computation*, TAMC'10, pages 2–14, Berlin, Heidelberg, 2010. Springer-Verlag.

[330] S.-H. Teng. Numerical thinking in algorithm design and analysis. In *Computer Science: The Hardware, Software and Heart of It, ed by Blum, K. Edward and Aho, V. Alfred*, pages 349–384. Springer New York, 2011.

[331] S.-H. Teng. Network essence: Pagerank completion and centrality-conforming Markov Chains. In M. Loebl, J. Nesetril, and R. Thomas, editors, *Journey Through Discrete Mathematis: A Tribute to Jiří Matoušek*. 2016.

[332] S.-H. Teng, Q. Lu, M. Eichstaedt, D. Ford, and T. Lehman. Collaborative Web crawling: Information gathering/processing over Internet. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences-Volume 5 - Volume 5*, HICSS '99, page 5044. IEEE Computer Society, 1999.

[333] F. Tohmé and T. Sandholm. Coalition formation processes with belief revision among bounded-rational self-interested agents. *Journal of Logic and Computation*, 9(6):793–815, 1999.

[334] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[335] W. Tutte. How to draw a graph. *Proceedings London Mathematical Society*, 52:743–767, 1963.

[336] H. Tverberg. A generalization of Radon's theorem. *Journal London Mathematical Society*, pages 123–128, 1966.

[337] J. C. Urschel, J. Xu, X. Hu, and L. T. Zikatanov. A cascadic multigrid algorithm for computing the Fiedler vector of graph Laplacians. *Journal of Computational Mathematics*, 33(2):209, 2015.

[338] P. M. Vaidya. An optimal algorithm for the all-nearest-neighbors problem. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, FOCS'86, pages 117–122, 1986.

[339] L. Valiant. Universality consideration in VLSI circuits. *IEEE Transactions on Computers*, 30(2):135–140, 1981.

[340] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, Nov. 1984.

[341] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, STOC '81, pages 263–277, 1981.

[342] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16:264–280, 1971.

[343] S. S. Vempala. *The random projection method*, volume 65 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, Providence, RI, 2004.

[344] N. K. Vishnoi. Lx = b. *Foundations and Trends® in Theoretical Computer Science*, 8(1-2):1 –141, 2013.

[345] K. Voevodski, M.-F. Balcan, H. Röglin, S.-H. Teng, and Y. Xia. Efficient clustering with limited distance information. In *The Conference on Uncertainty in Artificial Intelligence*, UAI'10, pages 632–640, 2010.

[346] K. Voevodski, M.-F. Balcan, H. Röglin, S.-H. Teng, and Y. Xia. Min-sum clustering of protein sequences with limited distance information. In *the International Conference on Similarity-based Pattern Recognition*, pages 192–206. Springer-Verlag, 2011.

[347] K. Voevodski, M.-F. Balcan, H. Röglin, S.-H. Teng, and Y. Xia. Active clustering of biological sequences. *Journal of Machine Learning Research*, 13(1):203–225, Jan. 2012.

[348] K. Voevodski, S.-H. Teng, and Y. Xia. Finding local communities in protein networks. *BMC Bioinformatics*, 10:297, 2009.

[349] C. Wang, W. Chen, and Y. Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, 25(3):545–576, 2012.

[350] J. H. Wilkinson. *The algebraic eigenvalue problem*. Oxford University Press, Inc., New York, NY, USA, 1988.

[351] V. V. Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, STOC '12, pages 887–898. ACM, 2012.

[352] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numerical Linear Algebra with Applications*, 17(6):953–976, 2010.

[353] J. Xu and B. Berger. Fast and accurate algorithms for protein side-chain packing. *Journal of the ACM*, 53(4):533–557, July 2006.

[354] X. Yan, H. Cheng, J. Han, and P. S. Yu. Mining significant graph patterns by leap search. In *the ACM International Conference on Management of Data*, SIGMOD '08, pages 433–444, 2008.

[355] Y. Ye. *Interior Point Algorithms: Theory and Analysis.* John Wiley & Sons, Inc., New York, NY, USA, 1997.

[356] H. P. Young. An axiomatization of Borda's rule. *Journal of Economic Theory*, 9(1):43–52, 1974.

[357] P. S. Yu, J. Han, and C. Faloutsos. *Link Mining: Models, Algorithms, and Applications.* Springer, Incorporated, 1st edition, 2010.

[358] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 1036–1043, 2005.