September 2019

Special Interest Group on Algorithms and Computation Theory

SIGACT news

TABLE OF CONTENTS

Association for Computing Machinery

Advancing Computing as a Science & Profession

The Book Review Column[1]

by Frederic Green

Department of Mathematics and Computer Science
Clark University
Worcester, MA 01610
email: fgreen@clarku.edu

In this column I review the following book:

- **Handbook of Graph Theory, Combinatorial Optimization, and Algorithms**, edited by KT Thulasiraman (Editor-in-Chief), Subramanian Arumugam, Andreas Brandstädt, and Takao Nishizeki. An extensive guide to an extremely important subject. Review by Frederic Green.

As always, please contact me to write a review; choose from among the books listed on the next pages, or, if you are interested in anything not on the list, just send me a note.

[1]

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms

1. *Tractability: Practical approach to Hard Problems*, Edited by Bordeaux, Hamadi, Kohli
2. *Recent progress in the Boolean Domain*, Edited by Bernd Steinbach
3. *Introduction to Property Testing*, by Oded Goldreich.
4. *Algorithmic Aspects of Machine Learning*, by Ankur Moitra.

Miscellaneous Computer Science

1. *Elements of Causal Inference: Foundations and Learning Algorithms*, by Jonas Peters, Dominik Janzing, and Bernhard Schölkopf.
2. *Elements of Parallel Computing*, by Eric Aubanel
3. *CoCo: The colorful history of Tandy's Underdog Computer* by Boisy Pitre and Bill Loguidice
4. *Introduction to Reversible Computing*, by Kalyan S. Perumalla
5. *A Short Course in Computational Geometry and Topology*, by Herbert Edelsbrunner
6. *Partially Observed Markov Decision Processes,* by Vikram Krishnamurthy
7. *Statistical Modeling and Machine Learning for Molecular Biology*, by Alan Moses
8. *The Problem With Software: Why Smart Engineers Write Bad Code,* by Adam Barr.
9. *Language, Cognition, and Computational Models,* Theirry Poibeau and Aline Villavicencio, eds.

Computability, Complexity, Logic

1. *The Foundations of Computability Theory,* by Borut Robič
2. *Applied Logic for Computer Scientists: Computational Deduction and Formal Proofs*, by Mauricio Ayala-Rincón and Flávio L.C. de Moura.
3. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, by Martin Grohe.
4. *Kernelization: Theory of Parameterized Preprocessing,* by Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi.

Cryptography and Security

1. *Cryptography in Constant Parallel Time,* by Benny Appelbaum
2. *Secure Multiparty Computation and Secret Sharing*, Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen
3. *A Cryptography Primer: Secrets and Promises*, by Philip N. Klein

Combinatorics and Graph Theory

1. *Finite Geometry and Combinatorial Applications*, by Simeon Ball
2. *Introduction to Random Graphs*, by Alan Frieze and Michał Karoński
3. *Erdős –Ko–Rado Theorems: Algebraic Approaches*, by Christopher Godsil and Karen Meagher
4. *Combinatorics, Words and Symbolic Dynamics,* Edited by Valérie Berthé and Michel Rigo

Miscellaneous Mathematics

1. *Introduction to Probability*, by David F. Anderson, Timo Seppäläinen, and Benedek Valkó.

Review of[2]
Handbook of Graph Theory, Combinatorial Optimization, and Algorithms
Edited by
Krishnaiyan "KT" Thulasiraman (Editor-in-Chief),
Subramanian Arumugam, Andreas Brandstädt,
and Takao Nishizeki
CRC Press, 2016
1226 pages, Hardcover, $132.00

Review by
Frederic Green fgreen@clarku.edu
Department of Mathematics and Computer Science
Clark University, Worcester, MA

1 Introduction

Some (admittedly controversial) models of physical reality propose that it is, at the fundamental level, based on a certain type of graph, called a spin-network (and this among other models of physical reality based on graphs). Recently, astronomers have verified that the otherwise empty reaches of intergalactic space are laced with wispy tendrils of hot gas that connect galaxies. On a more quotidian level, the machine on which I'm typing this document is constructed out of components that are essentially graphs, those components being connected by electrical edges forming an even larger graph, and this one device in turn being connected via various links to millions of other nodes in a huge global graph called the internet. The notion of a graph encompasses all levels of reality, from the subatomic through everyday life and reaching out to the cosmological scale.

To paraphrase Shakespeare, all the world's a graph, and we merely players in it.

Naturally, it is essential for workers in the fields of mathematics and computer science to understand graphs in many ways, compute their properties, and deploy them in countless applications. Indeed, outside of the three fields named in the title of this book, it is hard to think of any that more completely cover the principal foundational and practical interests of computer science. Even if one tries to argue that the book is "only" about where these three topics intersect (which it isn't!), that too encompasses a remarkably rich field. This is truly the daily bread of both theoretical *and* applied computer science, to say nothing of a central and endlessly fascinating branch of mathematics.

The object of this review is an imposing and massive work, consisting of 44 chapters (written by a total of 49 contributors) distributed over 11 sections. I could not in good faith claim to have actually *read* it! I only received it about a year ago, after all. On the other hand, I had the pleasure of reading some parts of it with care, and skimming through it pretty much cover to cover. And I took notes... see the next section.

2 Summary of Contents

It would be impossible (if not foolhardy) to distill the numerous threads of the book into a small number of words, but I hope that the following summary gives the reader a good sense of what it is about. The 11 sections are as follows:

(I) "Basic Concepts," Ch. 1–3: Definitions and very basic results about graph theory appear in Chapter 1, where the results are simple and common enough to be presented without proof. There is a quick summary of the idea of NP-completeness, a notion that makes many appearances in later chapters (this is graph theory, after all), which sometimes include NP-completeness proofs. The most basic graph algorithms appear in Chapter 2, e.g., minimum spanning trees, shortest path, transitive closure, etc. We now encounter a *lot* of proofs, as we do throughout the rest of the book, but here they all about correctness. The computational complexity of each algorithm is mentioned, and literature cited for the reader to consult for the analysis. Chapter 3 covers depth-first search and related algorithms.

(II) "Flows in Networks," Ch. 4–6: All the surrounding issues, motivating examples, and basic algorithms for max flow are given in Chapter 4. Time complexity analyses of different implementations are also given. Various strategies for augmenting path algorithms are explored, as are other types such as preflow-push algorithms and blocking flow algorithms. Chapter 5 covers minimum cost flow problems (generalizing such problems as shortest path and max flow), giving algorithms following a number of strategies, including Cycle-Cancelling, Successive Shortest-Path, Primal-Dual/Out-of-Kilter algorithms, Network Scaling Algorithms, and the Capacity-Scaling Algorithm. Background notions (e.g., duality) get due consideration. The Minimum Cost Flow problem is further generalized in Chapter 6 to Multicommodity Flows, with a brief discussion of the background and algorithms via (for example) Lagrangian Relaxation and Dantzig-Wolfe Decomposition, both deriving from the method of Lagrange multipliers.

(III) "Algebraic Graph Theory," Ch. 7–11: Graphs really *are* matrices, so it is not surprising that linear algebra plays a prominent role in graph theory, as is surveyed here. The section deals primarily with mathematical foundations rather than algorithms. The intimate connection between graphs and linear algebra, in particular in the elegant duality between cutsets and circuits, which plays a role in electric circuit theory as well as matroid theory, is discussed in Chapter 7. These duality ideas are explored further in the later Chapter 20 (see below) on planarity. The linear algebra connection via incidence, cut, and circuit matrices are further explored in Chapter 8. These tools are used to derive enumerative results such as the number of spanning trees, later applied in Chapter 11 to resistance networks and more general properties of networks such as the Arc-coloring Theorem. (We learn a lot about electrical networks by modeling them as graphs, of course, but conversely, many fundamental properties of graphs can be derived by modeling them as electrical networks.) Chapter 9 studies the adjacency and signal-flow graphs, deriving the Coates and Mason gain formulas. A deep and useful idea is the *spectrum* of a graph (i.e., the spectrum of its associated adjacency matrix), as well as the spectrum of the *Laplacian* (or Kirchoff) matrix, which has led to a number of algorithmic breakthroughs in recent years. Spectra of graphs are studied in Chapter 10, which includes, for example, explicit computations of graph spectra, various fundamental results such as Kirchoff's Matrix-Tree Theorem, and the notion of algebraic connectivity. The Laplacian is revisited and applied in Chapter 30 to problems in graph partitioning.

(IV) "Structural Graph Theory," Ch. 12–19: The basic ideas and results of graph connectivity are set down in Chapter 12, followed by their application to connectivity algorithms in Chapter 13. For example, Menger's Theorem (which implies the max-flow/min-cut theorem) is proved in Chapter 12, and Chapter 13 shows how its variants lead to max flow-based algorithms for vertex and edge connectivities. The problem of augmenting the (edge or vertex) connectivity of a graph is treated in Chapter 14; while this graph-connectivity augmentation is NP-hard, those subproblems which are amenable to efficient solution are studied. The focus is on the mathematical development that is the

basis of algorithms, with extensive pointers to the literature for the algorithms themselves. *Matching* is one of the most important problems in graph theory, for which many aspects of the theory are laid out in Chapter 15 (such as the theorems of Frobenius, Hall, König, and Tutte, decompositions, and permanents to count the number of matchings), followed by the most important matching algorithms in Chapter 16 (e.g., non-bipartite matching according to Edmonds, and the max-flow approach to bipartite matching) and algorithms for the stable marriage problem and some of its variants (Chapter 17). Dominating sets, with special attention to the dominating number of a graph, are introduced in Chapter 18, which contains a proof that computing the dominating number is NP-complete. The final Chapter (19) in this section contains basic results on graph coloring.

(V) "Planar Graphs," Ch. 20–23: Planarity is introduced and studied in Chapter 20, containing proofs of fundamental results such as Euler's formula and Kuratowski's Theorem, as well as the remarkable fact that a graph has a dual iff it is planar. The remaining chapters in this section are very detailed expositions of a number of relevant algorithms. Two techniques for planarity testing are treated. The first is to add one edge at a time to an embedding, and if an edge cannot be added, it isn't planar. This edge-addition planarity testing algorithm is explored at length in Chapter 21. The second is based on a data structure called a PC-tree, which is studied in Chapter 22. Finally, we look into the visual representation of graphs, the subject of graph drawing. This is investigated from the point of view of various styles, e.g., straight-line or rectangular among others, in Chapter 23 (which, not surprisingly, has many of the coolest figures in the book).

(VI) "Interconnection Networks," Ch. 24–26: One reason graphs are of such central importance in computer science and engineering is that they naturally represent many computing systems. So another reason we need to come up with algorithms for graphs is that graphs are the basis of the hardware underlying those algorithms. This is especially true of parallel and distributed algorithms and systems. These chapters deal with the elements of graph theory that relate to certain types of computer networks. Interconnection networks are introduced in Chapter 24, and hypercube networks and quite a number of variants are covered in some depth. Cayley graphs, which arise in group theory, are used in distributed memory parallel architectures. Chapter 25, after a very concise resumé of finite group theory (concise but detailed; you will even find the definition of semi-direct product!), investigates the properties of Cayley graphs, extensions of that notion, and their symmetry properties. The problem of embedding one graph into another is studied in Chapter 26, which considers how to embed both general graphs as well as certain classes of graphs in hypercubes and its variants.

(VII) "Special Graphs," Ch. 27–29: *Program graphs* are instrumental in the optimization phase of compilers, and the relevant algorithms are the subject of Chapter 27. A *perfect graph* is one for which each induced subgraph obeys the property that the number of vertices in a largest clique equals the minimum number of colors need to color the subgraph. Chapter 28 studies a number of classes of perfect graphs for which there are efficient recognition and optimization algorithms. "Tree-structured graphs" are not necessarily trees, but certain classes of graphs have implicit tree structures. For example, as shown in Chapter 29, *chordal graphs* (one type of perfect graph introduced in the previous chapter) are exactly those that have "clique trees." This fact relates the tree structure of chordal graphs to that of hypergraphs. Thus the tree structure of hypergraphs as well as that of chordal graphs receive special attention in Chapter 29. This very substantial chapter includes theoretical developments, algorithms, and also complexity considerations.

(VIII) "Partitioning," Ch. 30: This is a single-chapter section on graph and hypergraph partitioning. Finding

the minimum cut as in Chapter 4 is a simple (and efficiently solvable) form of partitioning. Here the more general k-partitioning problem is addressed. As this problem is in general NP-complete, one must be satisfied with approximations or heuristics. This chapter investigates a variety of approaches, including multiway cut algorithms, submodular optimization-based algorithms, iterative or move-based methods (e.g., the Kernighan-Lin algorithm), and spectral graph theory methods, arising from analysis of the spectrum of the Laplacian matrix, as previously mentioned in connection with Chapter 10. There is also a quick discussion of the application of simulated annealing to this problem.

(IX) "Matroids," Ch. 31–32: Matroids generalize the algebraic notion of linear independence to set systems. The range of their applications extends well beyond graph theory. Chapter 31 is a rigorous and clear introduction to the theory of matroids: definitions by base or rank axioms, duality, and minors. There are also algorithms for such operations as convolution, and matroid union and intersection. "Hybrid analysis" refers to the analysis of electrical networks with both voltages and currents being in the list of unknowns (as opposed to *only* voltages or currents, which can then be determined via Kirchoff's Laws). Choosing variables minimally is related to the principal partitions problem, which can be solved using the matroid union theorem as developed in Chapter 31. These matters are the subject of Chapter 32, "Hybrid Analysis and Combinatorial Optimization."

(X) "Probabilistic Methods, Random Graph Models, and Randomized Algorithms," Ch. 33–35: This section is devoted to these broadly applicable mathematical and algorithmic techniques. The probabilistic method of proof (Chapter 33) is a powerful idea pioneered by Erdős , which uses probability theory to express proofs in terms of bounds on probabilities, rather than the alternative complicated and convoluted counting arguments. It is now used widely in combinatorics, graph theory, and computer science. The concept is introduced in detail after some of the probability theoretic background, and then illustrated by a variety of examples. These include (but are not limited to) bounds relating to cuts and other parameters associated with graphs, and a couple of applications to number theory. The chapter also introduces random graphs, illustrated by a number of results, for example the existence of triangles in a random graph, the concentration of invariants, and connectivity results. Random graph models are then applied to "chemical graphs" in Chapter 34; these are just bounded degree graphs with labels attached to the vertices (thus corresponding to types of atoms in a molecule). The analysis of such graphs is of more general interest, since the problem of determining if two chemical graphs are the same is a type of graph isomorphism problem. Algorithms are given for canonical names of graphs, which facilitates determining which ones are isomorphic. Chapter 35 studies randomized algorithms for graphs, focussing on ZPP-type algorithms, covering interesting examples such as the contraction algorithm for global min-cut, estimating the size of transitive closures, maintaining strongly connected components, and constructing approximate distance oracles.

(XI) "Coping with NP-completeness," Ch. 36–44: Here we confront the fact that most problems having to do with graphs are NP-hard. The emphasis in the section is on approximate solutions to optimization problems. General techniques (e.g., bounding, interval partitioning) are outlined in Chapter 36. Approximation algorithms are then presented for a variety of specific problems: The constrained shortest path problem (CSP) in Chapters 37 (the Hassim-Lorenz-Raz ε-approximation scheme) and Chapter 38 (Lagrangian relaxation techniques), finding disjoint paths (Chapter 39), set cover approximation (Chapter 40), multicommodity flows (Chapter 41), connectivity problems (Chapter 42), and rectilinear Steiner minimum trees, which has many applications in VLSI design (Chapter 43). The book concludes (Chapter 44) with an extensive chapter on a more general topic, fixed-parameter algorithms and complexity. Computational complexity is usually measured in terms of the size of the

input, but for the vast majority of NP-complete problems, problem instances may have many more parameters (e.g., the degree of a given graph, the maximum clause size in an instance of satisfiability, etc.). If certain parameters are fixed and others allowed to vary, the problem may become tractable as a function of the variable parameters; such problems are called *fixed-parameter tractable* (FPT). The concept of parameterized complexity led to new ways of understanding intractability, but also many new, broadly applicable algorithmic techniques, the subject of this chapter. Prominent among these techniques is *kernelization*, which reduces a problem instance to a simpler one in polynomial time in one of the parameters. The "kernel" thus obtained may be thought of as a problem that has already been solved via preprocessing (another term for kernelization), thus allowing the original instance to be solved. Surprisingly, the existence of a kernelization algorithm and fixed-parameter tractability are equivalent. The technique is applied to a couple of problems here, namely Vertex Cover and Max-SAT, and techniques for kernelization lower bounds are studied. Many other FPT-based techniques are discussed (iterative compression, randomized FP algorithms, important separators, and others), and a section on parameterized *intractability* is included as well.

Each chapter ends with a brief summary containing helpful pointers to the literature, and extensive bibliographies.

3 Evaluation and Opinion

This book contains a vast wealth of material. But beyond the sheer volume of its coverage, it also takes diverse approaches. Some of the chapters proceed in a relatively leisurely way, taking time to provide a number of motivating examples, and some applications, before launching into detailed formalism; this was particularly noticeable for much of Section II on network flows. Other chapters (e.g. Chapter 3) present algorithms in their full generality with little ceremony. Still others present research that may not have previously appeared in book form. Whatever the approach, it is done with care never to sacrifice clarity or rigor for any reason.

In short, if you want an *introduction* to some of the topics treated here, you can do that. Or if you want to take *a dive deeply into the details*, you can get that too. There are ample opportunities for both.

Some words on scope: While this volume does concentrate on topics at the confluence of graph theory, combinatorial optimization, and algorithms, in order to present that area coherently, it is necessary to branch out into some of the topics separately to some degree. It should be clear from the foregoing summary that a number of chapters zero in on only one of the three main topics. For example, Chapter 20 focusses on the mathematical aspects of planarity and duality, so it is more of a study of "pure graph theory." Chapter 44 on fixed-parameter tractability focusses on algorithmic issues that apply well outside of graph theory. While most of the chapters deal with the intersection of all three topics, even in its algorithmic aspects, the emphasis is decidedly on the mathematical side, rather than the applied. It was indeed the stated aim of the editors "to emphasize proofs of results and underlying proof techniques..." The authors "give proofs of all theorems unless they [are] too long." In that they have succeeded admirably.

A certain variation in style from chapter to chapter or section to section is desirable in a work of this nature: it is beneficial to see the intuitive as well as the formal side of things, for example. Also, redefining terms when they are needed is helpful, especially to insulate a reader from the need to jump back and forth too much in the book, which is a little unwieldy for that purpose.

On the other hand, there are downsides. While cross-references do appear, the book would benefit by incorporating many more of them throughout the text. Simply citing appropriate earlier or later sections that

relate to a particular topic would go a long way towards "integrating" the material. The nonuniformities referred to above also lead to minor annoyances, e.g., notations that occasionally vary from chapter to chapter. This may be a partial result of the fact that some chapters are edited versions of research papers or chapters that have appeared elsewhere. But fortunately, the result is nevertheless much closer to a truly unified treatment of a broad area than, say, a collection of independent papers.

To sum up, this book gives a lucid, deep, and panoramic view of of graph theory, both broadly conceived and concentrating on its algorithmic and combinatorial optimization aspects. It will be of immense use to anyone with an interest in the area, researcher, teacher, or student, as a reference work or as a resource for self-study. It is a weighty but attractive volume, rich in content and richly illustrated. I highly recommended it!