Foundations and Trends® in
sample
Vol. xx, No xx (xxxx) 1–82

DOI: xxxxxx

now
the essence of knowledge

Pseudorandomness I

Salil P. Vadhan[1]

[1] *Harvard UniversityCambridge, MA02138, USA, salil@eecs.harvard.edu*

Abstract

This is the first volume of a 2-part survey on *pseudorandomness,* the theory of efficiently generating objects that "look random" despite being constructed using little or no randomness. The survey places particular emphasis on the intimate connections that have been discovered between a variety of fundamental "pseudorandom objects" that at first seem very different in nature: expander graphs, randomness extractors, list-decodable error-correcting codes, samplers, and pseudorandom generators. The survey also illustrates the significance the theory of pseudorandomness has for the study of computational complexity, algorithms, cryptography, combinatorics, and communications. The structure of the presentation is meant to be suitable for teaching in a graduate-level course, with exercises accompanying each chapter.

Contents

1

Introduction

1.1 Overview of this Survey

Over the past few decades, *randomization* has become one of the most pervasive paradigms in computer science. Its widespread uses include:

Algorithm Design: For a number of important algorithmic problems, the most efficient algorithms known are randomized. For example:

- PRIMALITY. This was shown to have a randomized polynomial-time algorithm in 1977. It wasn't until 2002 that a deterministic polynomial-time algorithm was discovered. (We will see this algorithm, but not its proof.)
- APPROXIMATE COUNTING. Many approximate counting problems (e.g. counting perfect matchings in a bipartite graph) have randomized polynomial-time algorithms, but the fastest known deterministic algorithms take exponential time.
- UNDIRECTED S-T CONNECTIVITY. This was shown to have a randomized logspace algorithm in 1979. It wasn't until 2005 that a deterministic logspace algorithm was discovered — using tools from the theory of pseudorandomness, as we will see.
- PERFECT MATCHING. This was shown to have a randomized *polylogarithmic*-time parallel algorithm in the late 1970's. Deterministically, we only know polynomial-time algorithms.

Cryptography: Randomization is central to cryptography. Indeed, cryptography is concerned with protecting secrets, and how can something be secret if it is deterministically fixed? For example, we assume that cryptographic keys are chosen at random (e.g. uniformly from the set of n-bit strings). In addition to the keys, it is known that often the cryptographic algorithms themselves (e.g. for encryption) must be randomized to achieve satisfactory notions of security (e.g. that no partial information about the message is leaked).

Combinatorial Constructions: Randomness is often used to prove the *existence* of combinatorial objects with a desired properties. Specifically, if one can show that a randomly chosen object has the property with nonzero probability, then it follows that such an object must, in fact, exist. A famous example due to Erdős is the existence of *Ramsey graphs*: A randomly chosen n-vertex graph has no clique or independent set of size $2\log n$. We will see several other applications of this "Probabilistic Method" in this survey, such as with two important objects mentioned below: expander graphs and error-correcting codes.

Though these *applications* of randomness are interesting and rich topics of study in their own right, they are not the focus of the course. Rather, we ask the following:

Main Question: Can we reduce or even eliminate the use of randomness in these settings?

We have several motivations for doing this.

- Complexity Theory: We are interested in understanding and comparing the power of various kinds of computational resources. Since randomness is such a widely used resource, we want to know how it relates to other resources such as time, space, and parallelism. In particular, we ask: *Can every randomized algorithm be derandomized with only a small loss in efficiency?*
- Using Physical Random Sources: It is unclear whether the real world has physical sources of perfect randomness. We may use sources that seem to have some unpredictability, like the low order bits of a system clock or thermal noise, but these sources will generally have biases and, more problematically, correlations. Thus we ask: What can we do with a source of biased and correlated bits?
- Explicit Constructions: Probabilistic constructions of combinatorial objects often do not provide us with efficient algorithms for using those objects. Indeed, the randomly chosen object often has a description that is exponential in the relevant parameters. Thus, we look for *explicit* constructions — ones that are deterministic and efficient. In addition to their applications, improvements in explicit constructions serve as a measure of our progress in understanding the objects at hand. Indeed, Erdős posed the explicit construction of near-optimal Ramsey graphs as an open problem, and substantial progress on this problem was recently made using the theory of pseudorandomness (namely randomness extractors).
- Unexpected Applications: In addition, the theory of pseudorandomness has turned out to have many applications to problems that seem to have no connection to derandomization. These include data structures, distributed computation (e.g. leader election), circuit lower bounds in complexity theory, reducing interaction in interactive protocols, saving memory in streaming algorithms, and more. We will see some of these applications in this survey (especially the exercises).

The paradigm we will use to study the Main Question is that of *pseudorandomness*: efficiently generating objects that "look random" using little or no randomness.

Specifically, we will study four "pseudorandom" objects:

Pseudorandom generators (PRGs): A PRG is an algorithm that takes as input a short, perfectly random *seed* and then returns a (much longer) sequence of bits that "looks random." That the bits output cannot be perfectly random is clear — the output is determined by the seed and there are far fewer seeds than possible bit sequences. Nevertheless, it is possible for the output to "look random" in a very meaningful and general-purpose sense. Specifically, we will require that no *efficient* algorithm can distinguish the output from those of a truly random sequence. The study of pseudorandomn generators meeting this strong requirement originated in cryptography, where they have numerous applications. In this survey, we will emphasize their role in derandomizing algorithms.

Note that asserting that a function is a PRG is a statement about something that efficient algorithms can't do (in this case, distinguish two sequences). But proving that efficient algorithms cannot compute things is typically out of reach for theoretical computer science; indeed this is why the **P** vs. **NP** question is so hard. Thus, in this course, we will settle for conditional statements. An ideal theorem would be something like: "If $\mathbf{P} \neq \mathbf{NP}$, then pseudorandom generators exist." (The assumptions we make won't exactly be $\mathbf{P} \neq \mathbf{NP}$, but hypotheses of a similar flavor.)

Randomness Extractors: A randomness extractor takes as input a source of biased and correlated bits, and then produces a sequence of almost-uniform bits as output. Their original motivation was the simulation of randomized algorithms with sources of biased and correlated bits, but they have found numerous other applications in theoretical computer science. Ideally, extractors would be deterministic, but as we will see this proves to be impossible for general sources of biased and correlated bits. Nevertheless, we will get close—producing extractors that are only "mildly" probabilistic.

Expander Graphs: Expanders are graphs with two seemingly contradictory properties: they are sparse (e.g. having degree that is a constant, independent of the number of vertices), but also "well-connected" in some precise sense. For example, one might say that the graph cannot be bisected without cutting a large (say, constant) fraction of the edges.

Expander graphs have numerous applications in theoretical computer science. They were originally studied for their use in designing fault-tolerant networks (e.g. for telephone lines), which are networks that maintain good connectivity even when links or nodes fail. But they also have less obvious applications, such as an $O(\log n)$-time algorithm for sorting in parallel.

It is not obvious that expander graphs exist, but in fact it can be shown, via the Probabilistic Method, that a random graph of degree 3 is a "good" expander with high probability. However, many applications of expander graphs need *explicit constructions*, and these proved much harder to find. We will see some explicit constructions in this survey, but they do not always match the bounds given by the probabilistic method (in terms of the degree/expansion tradeoff).

Error-Correcting Codes: Error-correcting codes (ECCs) are tools for communicating over noisy channels. Specifically, they specify a way to encode messages into longer, redundant codewords so that even if the codeword gets somewhat corrupted along the way, it is still possible for the receiver to decode the original message. In his landmark paper that introduced the field of coding theory, Shannon also proved the existence of good error-correcting codes via the probabilistic

method. That is, a random mapping of n-bit messages to $O(n)$-bit codewords is a "good" error-correcting code with high probability. Unfortunately, these probabilistic codes are not feasible to actually use — a random mapping requires an exponentially long description, and we know of no way to decode such a mapping efficiently. Again, explicit constructions are needed.

In this course, we will focus on the problem of *list decoding*. Specifically, we will consider scenarios where the number of corruptions is so large that unique decoding is impossible; at best one can produce a short list that is guaranteed to contain the correct message.

A Unified Theory: Each of the above objects has been the center of a large and beautiful body of research, but until recently these corpora were largely distinct. An exciting development over the past decade has been the realization that all four of these objects are almost *the same* when interpreted appropriately. Their intimate connections will be a major focus of this survey, tying together the variety of constructions and applications that we will see.

The surprise and beauty of these connections has to do with the seemingly different nature of each of these objects. PRGs, by asserting what efficient algorithms cannot do, are objects of complexity theory. Extractors, with their focus on extracting the entropy in a correlated and biased sequence, are information-theoretic objects. Expander graphs are of course combinatorial objects (as defined above), though they can also be interpreted algebraically, as we will see. Error-correcting codes involve a mix of combinatorics, information theory, and algebra. Because of the connections, we obtain new perspectives on each of the objects, and make substantial advances on our understanding of each by translating intuitions and techniques from the study of the others.

1.2 Background Required and Teaching Tips

The presentation assumes a good undergraduate background in the theory of computation, and general mathematical maturity. Specifically, it is assumed that the reader is familiar with basic algorithms and discrete mathematics, e.g. as covered in [CLRS], including some exposure to randomized algorithms; and with basic computational complexity including P, NP, and reductions, e.g. as covered in [Sip2]. Experience with elementary abstract algebra, particularly finite fields, is helpful; recommended texts are [Art, LN].

Most of the material in both volumes is covered in a one-semester graduate course that the author teaches at Harvard University, which consists of 24 lectures of 1.5 hours each. Most of the students in that course take at least one graduate-level course in the theoretical computer science before this one.

The exercises are an important part of the survey, as they include proofs of key facts used in lecture, introduce some concepts that will be used in later chapters, and illustrate applications of the material to other topics. Problems that are particularly challenging or require more creativity than most are marked with a star.

1.3 Notational Conventions

All logarithms are base 2 unless otherwise specified. We denote the set of numbers $\{1, \ldots, n\}$ by $[n]$. We write $\mathbb{N}$ for the set of nonnegative integers (i.e. 0 is a natural number). We write $S \subset T$ to mean that S is a subset of T, and $S \subsetneq T$ for S being a strict subset of T.

Throughout, we consider random variables that can take values in arbitrary discrete sets (not just real-valued random variables). We generally use capital letters, e.g. X, to denote random variables and lowercase letters, e.g. x, to denote specific values. We write $x \stackrel{R}{\leftarrow} X$ to indicate that x is sampled according to X. For a set S, we write $x \stackrel{R}{\leftarrow} S$ to mean that x is selected uniformly at random from S. We use the convention that multiple occurrences of a random variable in an expression refer to the same instantiation, e.g. $\Pr[X = X] = 1$. For an event E, we write $X|_E$ to denote the random variable X conditioned on the event E.

1.4 Chapter Notes and References

General introductions to the theory of pseudorandomness (other than this survey) include [Gol2, Mil2].

Recommended textbooks focused on randomized algorithms are [MU, MR]. The first randomized polynomial-time algorithms for PRIMALITY were discovered by Solovay and Strassen [SS] and Miller and Rabin [Mil1, Rab]; a deterministic polynomial-time algorithm was given by Agrawal, Kayal, and Saxena [AKS1]. The first randomized algorithms for approximate counting were found by Karp and Luby [KLM]; the algorithm for coutning perfect matchings is due to Jerrum, Sinclair, and Vigoda [JSV], building on [Bro, JS]. The randomized logspace algorithm for UNDIRECTED S-T CONNECTIVITY was given by Aleliunas et al. [AKL$^+$]; it was derandomized by Reingold [Rei]. The randomized parallel algorithm for deciding PERFECT MATCHING is due to Lovász [Lov1]; the search version is handled in [KUW] (see also [MVV]).

Recommended textbooks on cryptography are [Gol3, Gol4, KL]. The idea that encryption should be randomized is due to Goldwasser and Micali [GM].

The Probabilistic Method for combinatorial constructions is the subject of the book [AS]. Erdős used this method to prove the existence of Ramsey graphs in [Erd]. Major recent progress on explicit constructions of Ramsey graphs was obtained by Barak, Rao, Shaltiel, and Wigderson [BRSW] via the theory of randomness extractors.

The modern notion of pseudorandom generator was formulated in the works of Blum and Micali [BM] and Yao [Yao], motivated by cryptographic applications. We will spend most of our time on a variant of the Blum–Micali–Yao notion, proposed by Nisan and Wigderson [NW], where the generator is allowed more running time than the algorithms it fools. A detailed treatment of the Blum–Micali–Yao notion can be found in [Gol3].

Surveys on randomness extractors are [NT, Sha1]. The notion of extractor that we will focus on is the one due to Nisan and Zuckerman [NZ].

A detailed survey of expander graphs is [HLW]. The probabilistic construction of expander graphs is due to Pinsker [Pin]. The application of expanders to sorting in parallel is due to Ajtai, Komlós, and Szemerédi [AKS2].

A classic text on coding theory is [MS]. For a modern, CS-oriented treatment, we recommend Sudan's lecture notes [Sud2]. Shannon's paper that gave birth to the field and gave a probabilistic construction of error-correcting codes is [Sha2]. The notion of list decoding was proposed by Elias [Eli] and Wozencraft [Woz], and was reinvigorated in the work of Sudan [Sud1]. Recent progress on list decoding is covered in [Gur].

References

[Adl] L. Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science (Ann Arbor, Mich., 1978)*, pages 75–83. IEEE, Long Beach, Calif., 1978.

[AB] M. Agrawal and S. Biswas. Primality and identity testing via Chinese remaindering. *Journal of the ACM*, 50(4):429–443 (electronic), 2003.

[AKS1] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics. Second Series*, 160(2):781–793, 2004.

[AKS2] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3(1):1–19, 1983.

[AKL$^+$] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (San Juan, Puerto Rico, 1979)*, pages 218–223. IEEE, New York, 1979.

[AMS] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1, part 2):137–147, 1999. Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996).

[AS] N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience [John Wiley & Sons], New York, second edition, 2000. With an appendix on the life and work of Paul Erdős.

[AB] S. Arora and B. Barak. *Complexity Theory: A Modern Approach*. Cambridge University Press, 2008. To appear.

[Art] M. Artin. *Algebra*. Prentice Hall Inc., Englewood Cliffs, NJ, 1991.

[BRSW] B. Barak, A. Rao, R. Shaltiel, and A. Wigderson. 2-source dispersers for sub-polynomial entropy and Ramsey graphs beating the Frankl-Wilson construction. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 671–680, New York, 2006. ACM.

[BR] M. Bellare and J. Rompel. Randomness-Efficient Oblivious Sampling. In *35th Annual Symposium on Foundations of Computer Science*, pages 276–287, Santa Fe, New Mexico, 20–22 Nov. 1994. IEEE.

[BM] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. *SIAM Journal on Computing*, 13(4):850–864, Nov. 1984.

[Bro] A. Z. Broder. How hard is to marry at random? (On the approximation of the permanent). In *STOC*, pages 50–58. ACM, 1986.

[BF] H. Buhrman and L. Fortnow. One-sided two-sided error in probabilistic computation. In *STACS 99 (Trier)*, volume 1563 of *Lecture Notes in Comput. Sci.*, pages 100–109. Springer, Berlin, 1999.

[CW] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

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

[CG] B. Chor and O. Goldreich. On the power of two-point based sampling. *Journal of Complexity*, 5(1):96–106, 1989.

[CLRS] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, second edition, 2001.

[DL] R. A. DeMillo and R. J. Lipton. A Probabilistic Remark on Algebraic Program Testing. *Information Processing Letters*, 7(4):193–195, 1978.

[DS] Z. Dvir and A. Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434 (electronic), 2006/07.

[Eli] P. Elias. *List decoding for noisy channels*. Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Mass., Rep. No. 335, 1957.

[Erd] P. Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53:292–294, 1947.

[EFF] P. Erdős, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51(1-2):79–89, 1985.

[ESY] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984.

[Fil] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *The Annals of Applied Probability*, 1(1):62–87, 1991.

[FKS] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the Association for Computing Machinery*, 31(3):538–544, 1984.

[Gil] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

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

[Gol1] O. Goldreich. A Sample of Samplers - A Computational Perspective on Sampling (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, 4(20), 1997.

[Gol2] O. Goldreich. *Modern cryptography, probabilistic proofs and pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1999.

[Gol3] O. Goldreich. *Foundations of cryptography*. Cambridge University Press, Cambridge, 2001. Basic tools.

[Gol4] O. Goldreich. *Foundations of cryptography. II*. Cambridge University Press, Cambridge, 2004. Basic Applications.

[Gol5] O. Goldreich. On promise problems: a survey. In *Theoretical computer science*, volume 3895 of *Lecture Notes in Comput. Sci.*, pages 254–290. Springer, Berlin, 2006.

[Gol6] O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008. To appear.

[GMW] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity, or All languages in NP have zero-knowledge proof systems. *Journal of the Association for Computing Machinery*, 38(3):691–729, 1991.

[GM] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, Apr. 1984.

[Gur] V. Guruswami. *Algorithmic Results in List Decoding*, volume 2, number 2 of *Foundations and Trends in Theoretical Computer Science*. now publishers, 2006.

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

[Har] N. J. A. Harvey. Algebraic Structures and Algorithms for Matching and Matroid Problems. In *FOCS*, pages 531–542. IEEE Computer Society, 2006.

[Hoe] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[HLW] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the AMS*, 43(4):439–561, 2006.

[IM] K. Iwama and H. Morizumi. An explicit lower bound of $5n - o(n)$ for Boolean circuits. In *Mathematical foundations of computer science 2002*, volume 2420 of *Lecture Notes in Comput. Sci.*, pages 353–364. Springer, Berlin, 2002.

[JS] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989.

[JSV] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697 (electronic), 2004.

[Jof1] A. Joffe. On a sequence of almost deterministic pairwise independent random variables. *Proceedings of the American Mathematical Society*, 29:381–382, 1971.

[Jof2] A. Joffe. On a set of almost deterministic k-independent random variables. *Annals of Probability*, 2(1):161–162, 1974.

[KLNS] J. D. Kahn, N. Linial, N. Nisan, and M. E. Saks. On the cover time of random walks on graphs. *Journal of Theoretical Probability*, 2(1):121–128, 1989.

[KPS] R. Karp, N. Pippenger, and M. Sipser. A time-randomness tradeoff. In *AMS Conference on Probabilistic Computational Complexity*, Durham, New Hampshire, 1985.

[KL] R. M. Karp and R. J. Lipton. Turing machines that take advice. *L'Enseignement Mathématique. Revue Internationale. IIe Série*, 28(3-4):191–209, 1982.

[KLM] R. M. Karp, M. Luby, and N. Madras. Monte Carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 10(3):429–448, 1989.

[KUW] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in Random NC. *Combinatorica*, 6(1):35–48, 1986.

[KL] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007. To appear.

[KS] N. Kayal and N. Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007.

[LR] O. Lachish and R. Raz. Explicit lower bound of $4.5n - o(n)$ for Boolean circuits. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 399–408 (electronic), New York, 2001. ACM.

[Lan] H. O. Lancaster. Pairwise statistical independence. *Annals of Mathematical Statistics*, 36:1313–1317, 1965.

[Lau] C. Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17(4):215–217, 1983.

[Lei] F. T. Leighton. *Introduction to parallel algorithms and architectures*. Morgan Kaufmann, San Mateo, CA, 1992. Arrays, trees, hypercubes.

[LV] D. Lewin and S. Vadhan. Checking polynomial identities over any field: towards a derandomization? In *STOC '98 (Dallas, TX)*, pages 438–447. ACM, New York, 1999.

[LN] R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications*. Cambridge University Press, Cambridge, first edition, 1994.

[Lov1] L. Lovász. On determinants, matchings, and random algorithms. In *FCT*, pages 565–574, 1979.

[Lov2] L. Lovász. *Combinatorial problems and exercises*. AMS Chelsea Publishing, Providence, RI, second edition, 2007.

[Lub1] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, 1986.

[Lub2] M. Luby. Removing randomness in parallel computation without a processor penalty. *J. Comput. System Sci.*, 47(2):250–286, 1993. 29th Annual IEEE Symposium on Foundations of Computer Science (White Plains, NY, 1988).

[LW] M. Luby and A. Wigderson. *Pairwise Independence and Derandomization*, volume 1, number 4 of *Foundations and Trends in Theoretical Computer Science*. now publishers, 2005.

[MS] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland Publishing Co., Amsterdam, 1977. North-Holland Mathematical Library, Vol. 16.

[Mih] M. Mihail. Conductance and Convergence of Markov Chains-A Combinatorial Treatment of Expanders. In *FOCS*, pages 526–531. IEEE, 1989.

[Mil1] G. L. Miller. Riemann's Hypothesis and Tests for Primality. *Journal of Computer and System Sciences*, 13(3):300–317, Dec. 1976.

[Mil2] P. Miltersen. *Handbook of Randomized Computing*, chapter Derandomizing Complexity Classes. Kluwer, 2001.

[MU] M. Mitzenmacher and E. Upfal. *Probability and computing*. Cambridge University Press, Cambridge, 2005. Randomized algorithms and probabilistic analysis.

[MR] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge University Press, Cambridge, 1995.

[MS] M. Mucha and P. Sankowski. Maximum Matchings via Gaussian Elimination. In *FOCS*, pages 248–255. IEEE Computer Society, 2004.

[MVV] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[Mut] S. Muthukrishnan. *Data Streams: Algorithms and Applications*, volume 1, number 2 of *Foundations and Trends in Theoretical Computer Science*. now publishers, 2005.

[Nis] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[NT] N. Nisan and A. Ta-Shma. Extracting Randomness: A Survey and New Constructions. *Journal of Computer and System Sciences*, 58(1):148–173, February 1999.

[NW] N. Nisan and A. Wigderson. Hardness vs Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, Oct. 1994.

[NZ] N. Nisan and D. Zuckerman. Randomness is Linear in Space. *Journal of Computer and System Sciences*, 52(1):43–52, Feb. 1996.

[Pin] M. Pinsker. On the Complexity of a Concentrator. In *7th Annual Teletraffic Conference*, pages 318/1–318/4, Stockholm, 1973.

[Pip] N. Pippenger. On Simultaneous Resource Bounds (Preliminary Version). In *FOCS*, pages 307–311. IEEE, 1979.

[Rab] M. O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12(1):128–138, 1980.

[Rag] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *Journal of Computer and System Sciences*, 37(2):130–143, 1988. Twenty-Seventh Annual IEEE Symposium on the Foundations of Computer Science (Toronto, ON, 1986).

[Ran] D. Randall. Mixing. In *FOCS*, pages 4–. IEEE Computer Society, 2003.

[Rei] O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):Art. 17, 24, 2008.

[RTV] O. Reingold, L. Trevisan, and S. Vadhan. Pseudorandom Walks in Regular Digraphs and the RL vs. L Problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC '06)*, pages 457–466, 21–23 May 2006. Preliminary version as *ECCC* TR05-22, February 2005.

[Ron] D. Ron. Property testing. In *Handbook of randomized computing, Vol. I, II*, volume 9 of *Comb. Optim.*, pages 597–649. Kluwer Acad. Publ., Dordrecht, 2001.

[Rub] R. Rubinfeld. Sublinear time algorithms. In *International Congress of Mathematicians. Vol. III*, pages 1095–1110. Eur. Math. Soc., Zürich, 2006.

[SG] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the Association for Computing Machinery*, 23(3):555–565, 1976.

[Sav] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4:177–192, 1970.

[SSS] J. P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM Journal on Discrete Mathematics*, 8(2):223–250, 1995.

[Sch] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the Association for Computing Machinery*, 27(4):701–717, 1980.

[Sha1] R. Shaltiel. Recent Developments in Extractors. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science*, volume 1: Algorithms and Complexity. World Scientific, 2004.

[Sha2] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Sip1] M. Sipser. A Complexity Theoretic Approach to Randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 330–335, Boston, Massachusetts, 25–27 Apr. 1983.

[Sip2] M. Sipser. *Introduction to the Theory of Computation*. Course Technology, 2nd edition, 2005.

[SS] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6(1):84–85, 1977.

[Spe] J. Spencer. *Ten lectures on the probabilistic method*, volume 64 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 1994.

[Spi] D. A. Spielman. Spectral Graph Theory and its Applications. In *48th Symposium on Foundations of Computer Science (FOCS 2007), 21-23 October 2007, Providence, RI, USA, Proceedings*, pages 29–38, 2007.

[Sud1] M. Sudan. Decoding of Reed Solomon Codes beyond the Error-Correction Bound. *Journal of Complexity*, 13(1):180–193, Mar. 1997.

[Sud2] M. Sudan. Essential Coding Theory (Lecture Notes). `http://people.csail.mit.edu/madhu/FT04/`, 2004.

[Vad] S. Vadhan. Probabilistic Proof Systems, Part I — Interactive & Zero-Knowledge Proofs. In S. Rudich and A. Wigderson, editors, *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*, pages 315–348. American Mathematical Society, 2004.

[WC] M. N. Wegman and J. L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981. Special issue dedicated to Michael Machtey.

[Woz] J. Wozencraft. List decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

[Yao] A. C. Yao. Theory and Applications of Trapdoor Functions (Extended Abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, 3–5 Nov. 1982. IEEE.

[Zip] R. Zippel. Probabilistic algorithms for sparse polynomials. In E. W. Ng, editor, *EUROSAM*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979.