Polynomial-Time Approximation of the Permanent*

Andrew Lyons

Department of Computer Science, Dartmouth College

lyonsam@gmail.com

Abstract

Despite its apparent similarity to the (easily-computable) determinant, it is believed that there is no polynomial-time algorithm for computing the permanent of an arbitrary matrix. In this survey, we review the known approaches for efficiently estimating the permanent and discuss their relative merits and limits. Emphasis is placed on the most successful approach to date, which is based on random sampling via Markov chains. In particular, we review the historical developments that lead to a result of Jerrum, Sinclair, and Vigoda, which states that the permanent of an arbitrary matrix with non-negative entries can be approximated in polynomial time. We describe a number of techniques that were developed for this specific problem and have since been generalized to become standard techniques in the area.

We assume some familiarity with complexity classes (such as P and NP), as well as a rudimentary understanding of Markov chains.

1 Introduction

The *permanent* of an $n \times n$ matrix A is defined as

$$\operatorname{per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)},$$

where S_n denotes the symmetric group on n elements, i.e., permutations of $\{1, \dots, n\}$.

We will be particularly interested in the case when the input A is a 0-1 matrix, as this case captures (nearly) all the difficulty of the problem and has a nice combinatorial characterization. Any such matrix A corresponds to a bipartite graph G with vertex set $U \cup V$, where U and V correspond to the rows and columns of A, respectively, and thus $U = V = \{1, 2, \dots, n\}$. The edge set $E(G)$ contains exactly those edges $\{u, v\}$ such that $a_{uv} = 1$. Recall that a *perfect matching* is a set of edges, no two incident on the same vertex, such that every vertex in G is incident on some edge in the matching. In our context, such collections of edges can be viewed as permutations $\sigma \in S_n$, hence the perfect matchings of G are exactly those monomials of $\operatorname{per}(A)$ that are not annihilated. Thus we see that evaluating the permanent of a 0-1 matrix is equivalent to counting the number of perfect matchings in a bipartite graph.

So how difficult is this quantity to compute?

Counting problems and #P.

Theorem 1 (Valiant's Theorem [30])**.** *The problem of evaluating the permanent of an arbitrary* 0-1 *matrix is* #P*-complete.*

Informally, #P is the class of counting problems associated with decision problems in NP. For example, the #P-complete counting problem #SAT is to determine the number of satisfying assignments for a Boolean formula given in CNF. Clearly, this problem is no easier than determining whether the formula has *any* satisfying assignment, which is the canonical NP-complete problem SAT. It follows that there is no deterministic, polynomial-time algorithm for computing the permanent of a 0-1 matrix unless P = NP.

*Course project for MATH 100: Markov Chain Monte Carlo (Topics in Probability Theory). Instructor: Peter Winkler.

Determinants and the dimer model. Polynomial-time algorithms are known for some special classes of graphs. In particular, motivated by an application in statistical physics known as the *dimer model*, Kasteleyn [18] showed that $\mathrm{per}(A)$ can be computed in $O(n^3)$ time if A is a 0-1 matrix whose associated graph is planar. Kasteleyn's method is one of many that attempt to exploit the close relationship between the permanent and the determinant. Recall that the *determinant* of an $n \times n$ matrix A is defined as

$$\det(A) = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}.$$

The determinant of an arbitrary matrix can be computed in $O(n^3)$ time using methods such as Gaussian elimination. Kasteleyn's idea was to change some of the 1's in A to -1's, thus forming a matrix B with the property that $\mathrm{per}(A) = \det(B)$. It has since been shown [23] that the planar bipartite graphs are part of a larger class of graphs (those that admit a *Pfaffian orientation*) for which the permanent can be computed in this way.

Given Valiant's Theorem above, it should not be surprising that no such reduction to the determinant has been discovered for arbitrary 0-1 matrices. It is believed that there is no way to express the permanent of an arbitrary matrix A as the determinant of some matrix whose size is polynomial in that of A.[1]

2 Approximating the permanent

Consider again the problem #SAT mentioned above. A polynomial-time algorithm for this problem seems far out of reach. Moreover, it is apparent that any algorithm that finds even an *approximate* solution (accurate within any fixed $\epsilon > 0$) to this problem could be adapted to solve the original SAT problem, which is NP-complete. In contrast, though the 0-1 permanent is also #P-complete, a result of Edmonds [10] says that the corresponding decision variant can be solved (and a perfect matching constructed, if one exists) in polynomial time. This property is key, as it allows for the possibility of efficient approximation. We now make this notion more precise. Our hope is to obtain an algorithm that, given an $n \times n$ matrix A and real numbers $\epsilon > 0$, $\delta \in (0, 1)$, can produce an estimate OUT of $per(A)$ such that

$$\Pr\left[(1-\epsilon)\,\mathrm{per}(A) \le \mathrm{OUT} \le (1+\epsilon)\,\mathrm{per}(A)\right] \ge 1 - \delta.$$

Such a scheme is called a *fully-polynomial randomized approximation scheme* (FPRAS) for $\mathrm{per}(A)$ if it runs in time $\mathrm{poly}(n, \epsilon^{-1}, \log \delta^{-1})$. Unfortunately, the following result implies that, under a complexity-theoretic assumption similar to $\mathsf{P} \neq \mathsf{NP}$[2], no such procedure exists.

Theorem 2 (Random-self-reducibility of the permanent [19, 7, 14])**.** *If there exists a FPRAS for the permanent of an arbitrary matrix, then there exists a randomized polynomial-time algorithm for computing the permanent of an arbitrary matrix* exactly*.*

However, it turns out that the restricted case of 0-1 matrices fails to capture the hardness of efficient *approximation* as it did for efficient (exact) evaluation.

Theorem 3 (Jerrum, Sinclair, and Vigoda [14])**.** *There exists a FPRAS for the permanent of an arbitrary matrix with nonnegative entries.*

This celebrated result is the culmination of a line of work on approximating permanents using Markov chain Monte Carlo (MCMC), and relies on a fairly sophisticated method of mixing time analysis. In subsequent sections, we discuss this and earlier approaches based on MCMC. All of these approaches use more or less the same Markov chain, which is fairly simple to define and yet has proved very difficult to analyze. The sophisticated techniques developed for this particular problem have seen wide applicability on other problems.

First, we briefly discuss an approach that is not based on MCMC. This approach has some potential, but has not yet yielded a FPRAS for 0-1 permanents.

[1] Note that this is precisely *Valiant's Hypothesis* [29] $\mathsf{VP} \neq \mathsf{VNP}$, where VP and VNP are algebraic analogues of P and NP.
[2] specifically, the assumption that $\mathsf{P}^{\#\mathsf{P}} \not\subseteq \mathsf{BPP}$

The Godsil-Gutman approach. The Godsil-Gutman approach is a randomized determinant-based approach to approximating the permanent of a 0-1 matrix A. The idea is to construct a matrix B by setting

$$b_{ij} = \begin{cases} a_{ij} & \text{with probability } \frac{1}{2} \\ -a_{ij} & \text{with probability } \frac{1}{2} \end{cases}$$

for each i, j (independently). Then the random variable $b = |\det(B)|^2$ satisfies

$$\mathbb{E}[b] = \operatorname{per}(A),$$

and an estimate of $\operatorname{per}(A)$ is obtained as the average of a collection of samples of b. Unfortunately, the efficiency of this method depends on the variance of b, and the number of such samples required to achieve the desired bounds is $O(1.7^n)$ in this case. A similar technique was used by Karmarkar et al. [17]; their method constructs B by multiplying with random complex numbers and requires $O(1.4^n)$ samples. Later, Chien, Rasmussen, and Sinclair [9] used a further generalization (specifically, quaternions) in a method that requires $O(1.2^n)$ samples. It is not clear whether a variant of the Godsil-Gutman approach will eventually yield a FPRAS for 0-1 permanents; this is the subject of some recent work [1, 8].

3 Counting via almost uniform sampling

In 1986, Broder published a seminal paper [5] in which he outlined a method for efficient approximation of certain 0-1 permanents. Broder's overall approach—based on random sampling from the stationary distribution of a Markov chain—ultimately proved to be the basis of the result by Jerrum, Sinclair, and Vigoda mentioned in the previous section. Importantly, Broder's paper also called attention to those complications that stood in the way of generalizing the framework. We now treat each of these contributions in turn.

Broder's first insight was to reduce the approximate counting counting problem (i.e., the existence of a FPRAS) to (almost) uniform random sampling.[3] Such reductions have since become standard, and much is now known about the close relationship between counting and sampling problems. The key is exploit a property known as *self-reducibility* [24, 25]; rather than giving the (somewhat technical) definition of this property, we illustrate its relevance for the problem under consideration. (Note that this notion is distinct from that of random-self-reducibility discussed above).

Consider the interpretation of the 0-1 permanent as counting perfect matchings in a bipartite graph G. Let $\mathcal{M}_i(G)$ denote the set $\{M \subseteq E(G) : |M| = i\}$, so that the quantity we wish to estimate is $|\mathcal{M}_n(G)|$. As noted above, we may assume (or rather, we can confirm in polynomial time) that $\mathcal{M}_n(G)$ is nonempty, and thus $|U| = |V| = n$. However, $|\mathcal{M}_n(G)|$ could easily be exponential in n, and so it is not at all obvious how $\operatorname{poly}(n)$ samples from the set $\mathcal{M}_n(G)$ could be used to give a reliable estimation of its size. Broder's clever insight was to estimate each of the ratios $\mathcal{M}_k(G)/\mathcal{M}_{k-1}(G)$ by sampling uniformly from the sets $\mathcal{M}_k(G) \cup \mathcal{M}_{k-1}(G)$ for all $k = 1, \ldots, n$, followed by evaluating the telescoping product

$$|\mathcal{M}_n(G)| = \frac{|\mathcal{M}_n(G)|}{|\mathcal{M}_{n-1}(G)|} \times \frac{|\mathcal{M}_{n-1}(G)|}{|\mathcal{M}_{n-2}(G)|} \times \cdots \times \frac{|\mathcal{M}_1(G)|}{|\mathcal{M}_0(G)|} \times |\mathcal{M}_0(G)|,$$

where $\mathcal{M}_0(G)$ denotes the set of empty matchings, of which there are exactly one. In fact, it is enough that the sampling is *close enough* to uniform; we make this notion more precise later in this section. However, (almost) uniform sampling from $\mathcal{M}_k(G) \cup \mathcal{M}_{k-1}(G)$ will lead to a sufficiently accurate estimate of the ratio $\mathcal{M}_k(G)/\mathcal{M}_{k-1}(G)$ only if $\mathcal{M}_{k-1}(G)/\mathcal{M}_k(G)$ is bounded by $\operatorname{poly}(n)$. Unfortunately, simple examples show that the following is true.

Obstruction 1 (Broder's observation [5])**.** *There exist bipartite graphs G for which the ratio $|\mathcal{M}_{n-1}(G)| \,/\, |\mathcal{M}_n(G)|$ is arbitrarily large.*

[3] Jerrum, Valiant, and Vazirani [16] developed contemporaneously (but independently) a similar but far more general framework for relating counting to sampling, describing very general conditions—requiring self-reducibility in particular—under which the two are polynomially equivalent.

Dense graphs. Broder's approach is feasible for a particular class of bipartite graphs. We say that a bipartite graph is *dense* if the degree of every vertex is greater than $n/2$. It can be shown that every dense bipartite graph G satisfies $|\mathcal{M}_{n-1}(G)| / |\mathcal{M}_n(G)| \leq n^2$. However, the standard approach to evaluating $|\mathcal{M}_k(G)| / |\mathcal{M}_{k-1}(G)|$ for $k < n$ would be to evaluate $|\mathcal{M}_n(G')| / |\mathcal{M}_{n-1}(G')|$ for some subgraph G' of G (obtained, for example, by removing some vertices from G). This is a problem, as G' is not guaranteed to be dense. Broder skirted this problem with a clever construction involving *supergraphs* of G, rather than subgraphs. A thorough description of this construction appears in Sinclair's book [26].

Almost uniform sampling. We implicitly assumed in the above discussion that the ratios $\mathcal{M}_k(G)/\mathcal{M}_{k-1}(G)$ were being estimated by drawing a sequence of samples. In other words, we assumed the existence of a procedure that, given the graph G and a number $k \in \{1, \ldots, n\}$, could produce samples drawn uniformly at random from the finite state space $\Omega = \mathcal{M}_k(G) \cup \mathcal{M}_{k-1}(G)$. In practice, however, we will likely not be able to determine exactly the desired uniform distribution π, which assigns an equal $1/|\Omega|$ probability to every element in Ω, as $|\Omega|$ is precisely the quantity we want to compute. Fortunately, we find that it is good enough to sample from a distribution μ on Ω that is close to π in terms of the total variation distance

$$\mathrm{d_{tv}}(\mu, \pi) = \frac{1}{2} \sum_{i \in \Omega} |\mu_i - \pi_i| = \max_{S \subseteq \Omega} \mu_S - \pi_S.$$

It particular, Broder showed that it suffices to construct a procedure that, given an input $\epsilon \in (0, 1]$ in addition to G and k, can generate samples from a distribution μ on Ω such that $\mathrm{d_{tv}}(\mu, \pi) \leq \epsilon$. We call such a procedure a *fully-polynomial almost uniform sampler (FPAUS)* if it runs in time $\mathrm{poly}(n, \log \epsilon^{-1})$.

In summary, we have the following.

Theorem 4 (Broder [5])**.** *Let A be an arbitrary dense 0-1 matrix of size $n \times n$. If there exists a FPAUS for $\mathcal{M}_n(G) \cup \mathcal{M}_{n-1}(G)$, then there exists a FPRAS for $|\mathcal{M}_n(G)|$.*

Because this result hints at the possibility of a FPRAS for dense bipartite graphs, it is natural to ask whether the problem remains #P-complete under this restriction (for if the problem were easily solvable on this subclass, then this result would shed no light on the possibility of a FPRAS for the unrestricted case); Broder showed that this is indeed the case.

4 A FPAUS for dense graphs via MCMC

The crux of Broder's approach is a method for obtaining a FPAUS that is based on Markov chain Monte Carlo. Recall that any ergodic Markov chain on a finite state space Ω is uniquely defined by its $|\Omega| \times |\Omega|$ transition matrix P and will converge to its (unique) stationary distribution π in finite time. The rate at which such a chain converges to π is measured by the *mixing time*

$$\tau(\epsilon) = \max_{i \in \Omega} \tau_i(\epsilon),$$

where

$$\tau_i(\epsilon) = \min\{t \in \mathbb{N} \colon \mathrm{d_{tv}}(P^t_{i\cdot}, \pi) \leq \epsilon \text{ for all } t' \geq t\}.$$

It follows that a Markov chain with state space $\Omega = \mathcal{M}_k(G) \cup \mathcal{M}_{k-1}(G)$ will constitute the desired FPAUS if its mixing time is bounded by $\mathrm{poly}(n, \log \epsilon^{-1})$. Broder constructed such a chain, and provided a proof of rapid mixing that was based on a coupling arguments. However, a fundamental flaw in Broder's proof was discovered in 1987 by Mihail.

Obstruction 2 (Mihail's observation [20, 6, 21])**.** *The natural Markov chains for perfect matchings in dense bipartite graphs (such as that of Broder [5] and the similar one proposed by Jerrum and Sinclair [13]) display a high degree of asymmetry which makes them unwieldy with respect to coupling arguments.*

Jerrum and Sinclair filled this gap the following year.

Theorem 5 (Jerrum and Sinclair [13])**.** *There exists a FPRAS for the number of perfect matchings in an arbitrary dense bipartite graph.*

In the remainder of this section, we describe the proof of this theorem.

Proving rapid mixing for dense bipartite graphs via conductance. Jerrum and Sinclair used a Markov chain (described below) that was almost identical to Broder's, but the techniques they used to prove rapid mixing were entirely new. In a companion paper [28] they introduced a general technique for bounding the mixing time of an ergodic, reversible Markov chain. Recall that a Markov chain is *reversible* if and only if it satisfies the "detailed balance" equation

$$\pi_i P_{ij} = \pi_j P_{ji} = w_{ij}. \tag{1}$$

The symmetry of the detailed balance equation (1) gives rise to a natural notion of an underlying *undirected* graph H with weights $w_{ij} = w_{ji}$ associated with each undirected edge $\{i, j\} \in E(H)$. The *conductance* of an ergodic, reversible Markov chain is defined by

$$\Phi(H) = \min \frac{\sum_{i \in S, j \notin S} w_{ij}}{\sum_{i \in S} \pi_i}$$

where the min is taken over all $S \subset \Omega$ such that $0 < \sum_{i \in S} \pi_i \leq \frac{1}{2}$. The intuition behind conductance similar to that behind similar notions such as graph expansion: the chain will exhibit rapid mixing if there are no significant bottlenecks, i.e., it is difficult for a random walk in H to get stuck in a particular subset S of the vertices. More precisely, we have the following result.

Theorem 6 (Sinclair and Jerrum [28])**.** *If H is the underlying undirected graph of an ergodic, reversible Markov chain in which $\min_{i \in \Omega} P_{ii} \geq \frac{1}{2}$, then*

$$\tau(\epsilon) \leq \frac{2}{\Phi(H)^2}(\ln \pi_{\min}^{-1} + \ln \epsilon^{-1}),$$

where $\pi_{\min} = \min_{i \in \Omega} \pi_i$.

It follows that a bound of the form $\Phi(H) \geq 1/\operatorname{poly}(n)$ suffices to demonstrate rapid mixing in the Markov chain corresponding to H.

Proving bounds on conductance via canonical paths. We now describe the Markov chain used by Jerrum and Sinclair. the transitions are specified as follows (put plainly, we simply pick an edge from G at random and do whatever we can with it without disturbing more than one of the edges in the matching). From any state $M \in \Omega$ pick an edge $\{u, v\} \in E(G)$ uniformly at random and then

(i) If $M \in \mathcal{M}_n(G)$ and $\{u, v\} \in M$, move to state $M' = M - \{u, v\}$;

(ii) If $M \in \mathcal{M}_{n-1}(G)$ and both u and v are unmatched in M, move to state $M' = M + \{u, v\}$;

(iii) If $M \in \mathcal{M}_{n-1}(G)$, u is matched to some w in M and v is unmatched in M, move to $M' = M + \{u, v\} - \{u, w\}$ (and the symmetric analogue when u is unmatched and v is matched);

(iv) In all other cases, do nothing.

Finally, an additional self-loop transition with probability $\frac{1}{2}$ is added for every state in the chain in order to satisfy the conditions of Theorem 6. Jerrum and Sinclair managed to show that the underlying graph H satisfies $\Phi(H) \geq 1/12n^6$ by the use of a method known as *canonical paths*. The idea is to describe a kind of flow on H between every pair i, j of states by defining paths γ_{ij} in H for every such pair. Once these paths are defined, a general upper bound of $3n^4 |\mathcal{M}_{n-1}(G)|$ is obtained on the number of canonical paths traversing any particular edge in H. The interested reader should consult the thorough exposition of this proof in Sinclair's book [26].

5 Extension to general nonnegative matrices

In this final section, we briefly mention the ways in which the abovementioned work of Broder, Jerrum, and Sinclair has bee extended and improved. We saw that the obstruction noted by Mihail, which pointed out the dificulty of proving rapid mixing by coupling arguments, was circumvented by the application of arguments based on conductance. But Broder's obstruction, which necessitated the restriction to dense graphs, remained until the following result was obtained.

Theorem 7 (Jerrum, Sinclair, and Vigoda [14])**.** *There exists a FPRAS for the number of perfect matchings in an arbitrary bipartite graph.*

The algorithm devised for the proof of this theorem runs in $O(n^{10} \log^3 n)$ time; this was later improved to $O(n^7 \log^4 n)$ time [4].

References

[1] V. Arvind and S. Srinivasan. On the hardness of the noncommutative determinant. *Proceedings of the 42nd ACM symposium on Theory of computing*, pp. 677–686. ACM, STOC '10, 2010, doi:10.1145/1806689.1806782. 3

[2] A. Barvinok. Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor. *Random Structures & Algorithms* 14(1):29–61, 1999, doi:10.1002/(SICI)1098-2418(1999010)14:1¡29::AID-RSA2¿3.0.CO;2-X.

[3] A. Barvinok. New permanent estimators via non-commutative determinants. *Arxiv preprint math/0007153*, 2000, arXiv:math/0007153.

[4] I. Bezáková, D. Štefankovič, V. V. Vazirani, and E. Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM Journal on Computing* 37(5):1429–1454, 2008, doi:10.1137/050644033. 6

[5] A. Z. Broder. How hard is it to marry at random? (on the approximation of the permanent). *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pp. 50–58. ACM, STOC '86, 1986, doi:10.1145/12130.12136. 3, 4

[6] A. Z. Broder. Errata to "how hard is to marry at random? (on the approximation of the permanent)". *STOC*, p. 551. ACM, 1988. 4

[7] J.-Y. Cai, A. Pavan, and D. Sivakumar. On the hardness of permanent. *STACS 99*, pp. 90-99. Springer Berlin / Heidelberg, Lecture Notes in Computer Science 1563, 1999, doi:10.1007/3-540-49116-3_8. 2

[8] S. Chien, P. Harsha, A. Sinclair, and S. Srinivasan. Almost settling the hardness of noncommutative determinant. *preprint*, 2011, arXiv:1101.1169. 3

[9] S. Chien, L. Rasmussen, and A. Sinclair. Clifford algebras and approximating the permanent. *Journal of Computer and System Sciences* 67(2):263–290, 2003, doi:10.1016/S0022-0000(03)00010-2. Special Issue on STOC 2002. 3

[10] J. Edmonds. Paths, trees, and flowers. *Canad. J. Math.* 17(3):449–467, 1965, `www.cs.berkeley.edu/~christos/classics/edmonds.ps`. 2

[11] A. Frieze and M. Jerrum. An analysis of a monte carlo algorithm for estimating the permanent. *Combinatorica* 15:67–83, 1995, doi:10.1007/BF01294460.

[12] M. Jerrum. Lecture notes for a course entitled "Counting, sampling and integrating: algorithms and complexity", 2005, `http://homepages.inf.ed.ac.uk/mrj/pubs.html`.

[13] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing* 18(6):1149–1178, 1989, doi:10.1137/0218077. 4

[14] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM* 51:671–697, July 2004, doi:10.1145/1008731.1008738. 2, 6

[15] M. Jerrum and U. Vazirani. A mildly exponential approximation algorithm for the permanent. *Algorithmica* 16:392–401, 1996. 10.1007/BF01940871.

[16] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* 43:169 – 188, 1986, doi:10.1016/0304-3975(86)90174-X. 3

[17] N. Karmarkar, R. Karp, R. Lipton, L. Lovász, and M. Luby. A monte-carlo algorithm for estimating the permanent. *SIAM Journal on Computing* 22(2):284–293, 1993, doi:10.1137/0222021. 3

[18] P. Kasteleyn. Graph theory and crystal physics. *Graph Theory and Theoretical Physics*, pp. 43–110. Academic Press, 1967. 2

[19] R. Lipton. New directions in testing. *Distributed computing and cryptography: proceedings of a DIMACS Workshop, October 4-6, 1989*, pp. 191–202, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 2, 1991. 2

[20] M. Mihail. The approximation of the permanent is still open. Tech. Rep. 8-87, Aiken Comp. Lab., Harvard University, 1987. 4

[21] M. Mihail. On coupling and the approximation of the permanent. *Information Processing Letters* 30(2):91–95, 1989, doi:10.1016/0020-0190(89)90115-4. 4

[22] R. Montenegro. Lecture notes for a course entitled "Convergence of Markov Chains", `http://www.ravimontenegro.com/8843/index.html`.

[23] N. Robertson, P. Seymour, and R. Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Annals of Mathematics* 150:929–975, 1999. 2

[24] C.-P. Schnorr. Optimal algorithms for self-reducible problems. *ICALP*, pp. 322–337, 1976. 3

[25] C. P. Schnorr. On self-transformable combinatorial problems. *Mathematical Programming at Oberwolfach*, pp. 225–243. Springer Berlin Heidelberg, Mathematical Programming Studies 14, 1981, doi:10.1007/BFb0120931. 3

[26] A. Sinclair. *Algorithms for random generation and counting: a Markov chain approach.* Progress in Theoretical Computer Science. Birkhäuser, Boston, 1993. 4, 5

[27] A. Sinclair. Course notes, `http://www.cs.berkeley.edu/~sinclair/cs294/f09.html`. 2009.

[28] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation* 82(1):93–133, 1989, doi:10.1016/0890-5401(89)90067-9. 5

[29] L. G. Valiant. Completeness classes in algebra. *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pp. 249–261. ACM, STOC '79, 1979, doi:10.1145/800135.804419. 2

[30] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science* 8(2):189 – 201, 1979, doi:10.1016/0304-3975(79)90044-6. 1

[31] E. Vigoda. Lecture notes for a course entitled "Markov Chain Monte Carlo Methods", 2003, `http://www.cc.gatech.edu/~vigoda/MCMC_Course/index.html`.