Contents

Chapter 3

Sampling and counting

Accumulated evidence of the kind described in the previous chapter suggests that *exact* counting of combinatorial structures is rarely possible in polynomial time. However, it is in the nature of that evidence[1] that it does not rule out the possibility of *approximate* counting (within arbitrarily small specified relative error). Nor does it rule out the possibility of sampling structures at random from an almost uniform distribution, or even from the precisely uniform distribution (in a suitably defined model of computation), come to that. Indeed these two quests — approximate counting and almost uniform sampling — are intimately related, as we'll see presently.

The aim of this chapter is to illustrate, by means of a concrete example, how almost uniform sampling can be employed for approximate counting, and, after that, how almost uniform sampling can be achieved using Markov chain simulation. But first, let's make precise the various notions we've been talking about informally until now.

3.1 Preliminaries

Consider the problem: given a graph G, return a matching M chosen uniformly at random (u.a.r.) from the set of all matchings in G. In order to discuss sampling problems such as this one we obviously need a model of computation that allows random choices. Less obviously, we also need such a model to discuss approximate counting problems: e.g., given a graph G, compute an estimate of the number of matchings in G that is accurate to within $\pm 10\%$.

[1] Specifically, the property of it described Remark 2.6(d).

A probabilistic Turing machine is a Turing machine T equipped with special coin tossing states. Each coin-tossing state q has two possible successor states q_h and q_t. When T enters state q, it moves on the next step to state q_h with probability $\frac{1}{2}$ and to state q_t with probability $\frac{1}{2}$. Various notions of what it means for a probabilistic Turing machine to decide a predicate or approximate a function (in each case, with high probability) are possible, leading to various randomised complexity classes.

The probabilistic Turing machine is the usual basis for defining randomised complexity classes, but, more pragmatically, we can alternatively take as our model a random access machine (RAM) equipped with coin-tossing instructions, or a simple programming language that incorporates a random choice statement with two outcomes (themselves statements) that are mutually exclusive and each executed with probability $\frac{1}{2}$. All of these possible models are equivalent, modulo polynomial transformations in runtime. So when the phrase "randomised algorithm" is used in this and subsequent chapters, we are usually free to think in terms of any of the above models. However, when specific time bounds are presented (as opposed to general claims that some algorithm is polynomial time) we shall be taking a RAM or conventional programming language view. For a more expansive treatment of these issues, see Papadimitriou's textbook [53, Chaps 2 & 11].

A *randomised approximation scheme* for a counting problem $f : \Sigma^* \to \mathbb{N}$ (e.g., the number of matchings in a graph) is a randomised algorithm that takes as input an instance $x \in \Sigma^*$ (e.g., an encoding of a graph G) and an error tolerance $\varepsilon > 0$, and outputs a number $N \in \mathbb{N}$ (a random variable of the "coin tosses" made by the algorithm) such that, for every instance x,

$$\Pr\left[e^{-\varepsilon} f(x) \le N \le e^{\varepsilon} f(x)\right] \ge \frac{3}{4}. \tag{3.1}$$

We speak of a *fully polynomial randomised approximation scheme*, or *FPRAS*, if the algorithm runs in time bounded by a polynomial in $|x|$ and ε^{-1}.

Remarks 3.1. (a) The number $\frac{3}{4}$ appearing in (3.1) could be replaced by any number in the open interval $(\frac{1}{2}, 1)$.

(b) To first order in ε, the event described in 3.1 is equivalent to $(1 - \varepsilon)f(x) \le N \le (1 + \varepsilon)f(x)$, and this is how the requirement of a "randomised approximation scheme" is more usually specified. However the current definition is equivalent, and has certain technical advantages; specifically, a sequence of approximations of the form $e^{-\varepsilon}\xi_{i+1} \le \xi_i \le e^{\varepsilon}\xi_{i+1}$ compose gracefully.

For two probability distributions π and π' on a countable set Ω, define the *total variation distance* between π and π' to be

$$D_{\mathrm{tvd}}(\pi, \pi') := \frac{1}{2} \sum_{\omega \in \Omega} |\pi(\omega) - \pi'(\omega)| = \max_{A \subseteq \Omega} |\pi(A) - \pi'(A)| . \tag{3.2}$$

A *sampling problem* is specified by a relation $S \subseteq \Sigma^* \times \Sigma^*$ between problem instances x and "solutions" $w \in S(x)$.[2] For example, x might be the encoding of a graph G, and $S(x)$ the set of encodings of all matchings in G. An *almost uniform sampler* for a solution set $S \subseteq \Sigma^* \times \Sigma^*$ (e.g., the set of all matchings in a graph) is a randomised algorithm that takes as input an instance $x \in \Sigma^*$ (e.g., an encoding of a graph G) and an sampling tolerance $\delta > 0$, and outputs a solution $W \in S(x)$ (a random variable of the "coin tosses" made by the algorithm) such that the variation distance between the distribution of W and the uniform distribution on $S(x)$ is at most δ.[3] An almost uniform sampler is *fully polynomial* if it runs in time bounded by a polynomial in x and $\log \delta^{-1}$. We abbreviate "fully-polynomial almost uniform sampler" to FPAUS.

Remarks 3.2. (a) The definitions of FPRAS and FPAUS have obvious parallels. Note however that the dependence of the run-time on the "tolerance" (ε or δ, respectively) is very different: polynomial in ε^{-1} versus $\log \delta^{-1}$ respectively. This difference is deliberate. As we shall see, the relative error in the estimate for $f(x)$ can be improved only at great computational expense, whereas the sampling distribution on $S(x)$ can be made very close to uniform relatively cheaply.

(b) For simplicity, the definitions have be specialised to the case of a uniform distribution on the solution set $S(x)$. However, one could easily generalise the notion of "almost uniform sampler" to general distributions.

The "witness checking predicate" view of the classes NP and #P presented in Chapter 2 carries across smoothly to sampling problems. A witness checking predicate $\chi \subseteq \Sigma^* \times \Sigma^*$ and polynomial p define a sampling problem $S \subseteq \Sigma^* \times \Sigma^*$ via

$$S(x) = \{w \in \Sigma^* : \chi(x, w) \wedge |w| \leq p(|x|)\}, \tag{3.3}$$

[2] We write $S(x)$ for the set $\{w : x \, S \, w\}$ to avoid awkwardness.

[3] If $S(x) = \emptyset$ we allow the almost uniform sampler to return a special undefined symbol $\perp$, otherwise it cannot discharge its obligation.

where particular attention focuses on polynomial-time predicates χ (c.f. (2.1) and (2.2)). If χ is the "Hamilton cycle" checker of Chapter 2, then the related sampling problem $S(x)$ is that of sampling almost uniformly at random a Hamilton cycle in the graph G encoded by x. So we see that each combinatorial structure gives rise to a trio of related problems: decision, counting and sampling. Furthermore, the second of these at least may be considered in exact (FP) and approximate (FPRAS) forms.

Remark 3.3. The distinction between exactly and almost uniform sampling seems less crucial, and, in any case, technical complications arise when one attempts to define exactly uniform sampling: think of the problem that arises when $|S(x)| = 3$ and we are using the probabilistic Turing machine as our model of computation (or refer to Sinclair [57]).

3.2 Reducing approximate counting to almost uniform sampling

Fix a witness-checking predicate χ and consider the associated counting and sampling problems, $f : \Sigma^* \to \mathbb{N}$ and $S \subseteq \Sigma^* \times \Sigma^*$ defined by (2.2) and (3.3), respectively. It is known — under some quite mild condition on χ termed "self-reducibility," which often holds in practice — that the computational complexity of approximating $f(x)$ and sampling almost uniformly from $S(x)$ are closely related. In particular, f admits a FPRAS if and only if S admits an FPAUS. For full details, refer to Jerrum, Valiant and Vazirani [37]. Here we shall explore this relationship in only one direction (FPAUS implies FPRAS) and then only in the context of a specific combinatorial structure, namely matchings in a graph. This reduces the technical complications while retaining the main ideas.

Let $\mathcal{M}(G)$ denote the set of matchings (of all sizes) in a graph G.

Proposition 3.4. *Let G be a graph with n vertices and m edges, where $m \geq 1$ to avoid trivialities. If there is an almost uniform sampler for $\mathcal{M}(G)$ with run-time bounded by $T(n, m, \varepsilon)$, then there is a randomised approximation scheme for $|\mathcal{M}(G)|$ with run-time bounded by $cm^2\varepsilon^{-2}\,T(n, m, \varepsilon/6m)$, for some constant c. In particular, if there is an FPAUS for $\mathcal{M}(G)$ then there is a FPRAS for $|\mathcal{M}(G)|$.*

Proof. Denote the postulated almost uniform sampler by $\mathcal{S}$. The approximation scheme proceeds as follows. Given G with $E(G) = \{e_1, \dots, e_m\}$ (in any order), we consider the graphs $G_i := (V(G), \{e_1, \dots, e_i\})$ for $0 \leq i \leq m$.

Thus, G_{i-1} is obtained from G_i by deleting the edge e_i. The quantity $|\mathcal{M}(G)|$ which we would like to estimate can be expressed as a product

$$|\mathcal{M}(G)| = (\varrho_1 \varrho_2 \dots \varrho_m)^{-1} \tag{3.4}$$

of ratios

$$\varrho_i := \frac{|\mathcal{M}(G_{i-1})|}{|\mathcal{M}(G_i)|}\,.$$

(Here we use the fact that $|\mathcal{M}(G_0)| = 1$.) Observe that $\mathcal{M}(G_{i-1}) \subseteq \mathcal{M}(G_i)$ and that $\mathcal{M}(G_i) \setminus \mathcal{M}(G_{i-1})$ can be mapped injectively into $\mathcal{M}(G_{i-1})$ by sending M to $M \setminus \{e_i\}$. Hence,

$$\frac{1}{2} \le \varrho_i \le 1\,. \tag{3.5}$$

We may assume $0 < \varepsilon \le 1$ and $m \ge 1$. In order to estimate the ϱ_i's, we run our sampler $\mathcal{S}$ on G_i with $\delta = \varepsilon/6m$ and obtain a random matching M_i from $\mathcal{M}(G_i)$. Let Z_i be the indicator variable of the event that M_i is, in fact, in $\mathcal{M}(G_{i-1})$, and set $\mu_i := \mathrm{E}\, Z_i = \Pr[Z_i = 1]$. By choice of δ and the definition of the variation distance,

$$\varrho_i - \frac{\varepsilon}{6m} \le \mu_i \le \varrho_i + \frac{\varepsilon}{6m}\,, \tag{3.6}$$

or, from (3.5),

$$\left(1 - \frac{\varepsilon}{3m}\right)\varrho_i \le \mu_i \le \left(1 + \frac{\varepsilon}{3m}\right)\varrho_i\,; \tag{3.7}$$

so the sample mean of a sufficiently large number s of independent copies[4] $Z_i^{(1)}, \dots, Z_i^{(s)}$ of the random variable Z_i will provide a good estimate for ϱ_i. Specifically, let $s := \lceil 74\varepsilon^{-2}m \rceil \le 75\varepsilon^{-2}m$, and $\overline{Z}_i := s^{-1}\sum_{j=1}^{s} Z_i^{(j)}$.

Note that $\mathrm{Var}\, Z_i = \mathrm{E}[(Z_i - \mu_i)^2] = \Pr[Z_i = 1](1-\mu_i)^2 + \Pr[Z_i = 0]\mu_i^2 = \mu_i(1-\mu_i)$ and that inequalities (3.5) and (3.7) imply $\mu_i \ge 1/3$. Thus, $\mu_i^{-2}\,\mathrm{Var}\, Z_i = \mu_i^{-1} - 1 \le 2$, and hence

$$\frac{\mathrm{Var}\,\overline{Z}_i}{\mu_i^2} \le \frac{2}{s} \le \frac{\varepsilon^2}{37m}\,. \tag{3.8}$$

[4]Obtained from s independent runs of $\mathcal{S}$ on G_i.

As our estimator for $|\mathcal{M}(G)|$, we use the random variable $N := \left(\prod_{i=1}^m \overline{Z}_i\right)^{-1}$. Note that $\mathrm{E}[\,\overline{Z}_1\overline{Z}_2\ldots\overline{Z}_m] = \mu_1\mu_2\ldots\mu_m$, and furthermore

$$\begin{aligned}
\frac{\mathrm{Var}[\,\overline{Z}_1\overline{Z}_2\ldots\overline{Z}_m]}{(\mu_1\mu_2\ldots\mu_m)^2} &= \frac{\mathrm{E}[\,\overline{Z}_1^{\,2}\,\overline{Z}_2^{\,2}\ldots\overline{Z}_m^{\,2}]}{\mu_1^2\mu_2^2\ldots\mu_m^2} - 1 \\
&= \prod_{i=1}^m \frac{\mathrm{E}[\,\overline{Z}_i^{\,2}\,]}{\mu_i^2} - 1 \qquad\qquad \text{since r.v's } \overline{Z}_i \text{ are independent} \\
&= \prod_{i=1}^m \left(1 + \frac{\mathrm{Var}\,\overline{Z}_i}{\mu_i^2}\right) - 1 \\
&\le \left(1 + \frac{\varepsilon^2}{37m}\right)^m - 1 \qquad\qquad \text{by (3.8)} \\
&\le \exp\left(\frac{\varepsilon^2}{37}\right) - 1 \\
&\le \frac{\varepsilon^2}{36}\,,
\end{aligned}$$

since $e^{x/(k+1)} \le 1 + x/k$ for $0 \le x \le 1$ and $k \in \mathbb{N}^+$. Thus, by Chebychev's Inequality,

$$\left(1 - \frac{\varepsilon}{3}\right)\mu_1\mu_2\ldots\mu_m \le \overline{Z}_1\overline{Z}_2\ldots\overline{Z}_m \le \left(1 + \frac{\varepsilon}{3}\right)\mu_1\mu_2\ldots\mu_m, \tag{3.9}$$

with probability at least $1-(\varepsilon/3)^{-2}(\varepsilon^2/36) = \frac{3}{4}$. Since $e^{-x/k} \le 1 - x/(k+1)$ for $0 \le x \le 1$ and $k \in \mathbb{N}^+$, we have the following weakening of inequality (3.9):

$$e^{-\varepsilon/2}\mu_1\mu_2\ldots\mu_m \le \overline{Z}_1\overline{Z}_2\ldots\overline{Z}_m \le e^{\varepsilon/2}\mu_1\mu_2\ldots\mu_m.$$

But from (3.7), using again the fact about the exponential function, we have

$$e^{-\varepsilon/2}\varrho_1\varrho_2\ldots\varrho_m \le \mu_1\mu_2\ldots\mu_m \le e^{\varepsilon/2}\varrho_1\varrho_2\ldots\varrho_m\,,$$

which combined with the previous inequality implies

$$e^{-\varepsilon}\varrho_1\varrho_2\ldots\varrho_m \le \overline{Z}_1\overline{Z}_2\ldots\overline{Z}_m \le e^{\varepsilon}\varrho_1\varrho_2\ldots\varrho_m$$

with probability at least $\frac{3}{4}$. Since $\overline{Z}_1\overline{Z}_2\ldots\overline{Z}_m = N^{-1}$ and $\varrho_1\varrho_2\ldots\varrho_m = |\mathcal{M}(G)|^{-1}$, our estimator N for $|\mathcal{M}(G)|$ satisfies requirement (3.1). Thus the algorithm that computes computes N as above is a FPRAS for $|\mathcal{M}(G)|$.

The run-time of the algorithm is dominated by the number of samples required, which is $sm \le 75\varepsilon^{-2}m^2$, multiplied by the time-per-sample, which is $T(n, m, \varepsilon)$; the claimed time-bound is immediate. □

Exercise 3.5. Prove a result analogous to Proposition 3.4 with (proper vertex) q-colourings of a graph replacing matchings. Assume that the number of colours q is strictly greater than the maximum degree Δ of G. There is no need to repeat all the calculation, which is in fact identical. The key thing is to obtain an inequality akin to (3.5), but for colourings in place of matchings.

In light of the connection between approximate counting and almost uniform sampling, methods for sampling from complex combinatorially defined sets gain additional significance. The most powerful technique known to us is Markov chain simulation.

3.3 Markov chains

We deal exclusively in this section with discrete-time Markov chains on a finite state space Ω. Many of the definitions and claims extend to countable state spaces with only minor complication. In Chapter 6 we shall need to employ Markov chains with continuous state spaces, but the corresponding definitions and basic facts will be left until they are required. See Grimmett and Stirzaker's textbook [32] for a more comprehensive treatment.

A sequence $(X_t \in \Omega)_{t=0}^{\infty}$ of random variables (r.v's) is a *Markov chain* (MC), with state space Ω, if

$$\Pr[X_{t+1} = y \mid X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_0 = x_0] = \Pr[X_{t+1} = y \mid X_t = x_t], \tag{3.10}$$

for all $t \in \mathbb{N}$ and all $x_t, x_{t-1}, \dots, x_0 \in \Omega$. Equation (3.10) encapsulates the *Markovian property* whereby the history of the MC prior to time t is forgotten. We deal only with *(time-) homogeneous* MCs, i.e., ones for which the right-hand side of (3.10) is independent of t. In this case, we may write

$$P(x, y) := \Pr[X_{t+1} = y \mid X_t = x],$$

where P is the *transition matrix* of the MC. The transition matrix P describes single-step transition probabilities; the t-step transition probabilities P^t are given inductively by

$$P^t(x, y) := \begin{cases} I(x, y), & \text{if } t = 0; \\ \sum_{y' \in \Omega} P^{t-1}(x, y')P(y', y), & \text{if } t > 0, \end{cases}$$

where I denotes the identity matrix $I(x, y) := \delta_{xy}$. Thus $P^t(x, y) = \Pr[X_t = y \mid X_0 = x]$.

A *stationary distribution* of a MC with transition matrix P is a probability distribution $\pi : \Omega \to [0,1]$ satisfying

$$\pi(y) = \sum_{x \in \Omega} \pi(x) P(x, y).$$

Thus if X_0 is distributed as π then so is X_1 (and hence so is X_t for all $t \in \mathbb{N}$). A finite MC always has at least one stationary distribution. A MC is *irreducible* if, for all $x, y \in \Omega$, there exists a $t \in \mathbb{N}$ such that $P^t(x, y) > 0$; it is *aperiodic* if $\gcd\{t : P^t(x, x) > 0\} = 1$ for all $x \in \Omega$.[5] A (finite-state) MC is *ergodic* if it is both irreducible and aperiodic.

Theorem 3.6. *An ergodic MC has a unique stationary distribution π; moreover the MC tends to π in the sense that $P^t(x, y) \to \pi(y)$, as $t \to \infty$, for all $x \in \Omega$.*

Informally, an ergodic MC eventually "forgets" its starting state. Computation of the stationary distribution is facilitated by the following little lemma:

Lemma 3.7. *Suppose P is the transition matrix of a MC. If the function $\pi' : \Omega \to [0,1]$ satisfies*

$$\pi'(x) P(x, y) = \pi'(y) P(y, x), \quad \text{for all } x, y \in \Omega, \tag{3.11}$$

and

$$\sum_{x \in \Omega} \pi'(x) = 1,$$

then π' is a stationary distribution of the MC. If the MC is ergodic, then clearly $\pi' = \pi$ is the unique stationary distribution.

Proof. We just need to check that π' is invariant. Suppose X_0 is distributed as π'. Then

$$\Pr[X_1 = y] = \sum_{x \in \Omega} \pi'(x) P(x, y) = \sum_{x \in \Omega} \pi'(y) P(y, x) = \pi'(y).$$

□

[5] In the case of an irreducible MC, it is sufficient to verify the condition $\gcd\{t : P^t(x, x) > 0\} = 1$ for just one state $x \in \Omega$.

Remark 3.8. Condition (3.11) is known as *detailed balance.* A MC for which it holds is said to *time reversible.* Clearly, Lemma 3.7 cannot be applied to non-time-reversible MCs. This is not a problem in practice, since all the MCs we consider are time reversible. In fact, it is difficult in general to determine the stationary distribution of large non-time-reversible MCs, unless there is some special circumstance, for example symmetry, that can be taken into consideration. Furthermore, all the usual methods for constructing MCs with specified stationary distributions produce time-reversible MCs.

Example 3.9. Here is a natural (time homogeneous) MC whose state space is the set $\mathcal{M}(G)$ of all matchings (of all sizes) in a specified graph $G = (V, E)$. The transition matrix of the MC is defined implicitly, by an experimental trial. Suppose the initial state is $X_0 = M \in \mathcal{M}(G)$. The next state X_1 is the result of the following trial:

1. With probability $\frac{1}{2}$ set $X_1 \leftarrow M$ and halt.
2. Otherwise, select $e \in E(G)$ and set $M' \leftarrow M \oplus \{e\}$.[6]
3. If $M' \in \mathcal{M}(G)$ then $X_1 \leftarrow M'$ else $X_1 \leftarrow M$.

Since the MC is time homogeneous, it is enough to describe the first transition; subsequent transitions follow an identical trial. Step 1 may seem a little unnatural, but we shall often include such a looping transition to avoid a certain technical complication. Certainly its presence ensures that the MC is aperiodic. The MC is also irreducible, since it is possible to reach the empty matching from any state by removing edges (and reach any state from the empty matching by adding edges). Thus the MC is ergodic and has a unique stationary distribution.

Exercise 3.10. Demonstrate, using Lemma 3.7, that the stationary distribution of the MC of Example 3.9 is uniform over $\mathcal{M}(G)$.

Exercise 3.10 and Proposition 3.4, taken together, immediately suggest an approach to estimating the number of matchings in a graph. Simulate the MC on $\mathcal{M}(G)$ for T steps, starting at some fixed state X_0, say $X_0 = \emptyset$, and return the final state X_T. If T is sufficiently large, this procedure will satisfy the requirements of an almost uniform sampler for matchings in G. Then the method of Proposition 3.4 may be used to obtain a randomised approximation scheme for the number of matchings $|\mathcal{M}(G)|$. Whether this

[6] The symbol $\oplus$ denotes symmetric difference.

approach is feasible depends crucially on the rate of convergence of the MC to stationarity. We shall prove in Chapter 5 that a modification[7] of the MC described in Example 3.9 does in fact come "close" to stationarity in a polynomial number of steps (in the size of the graph G), hence yielding a FPRAS for the number of matchings in a graph.

[7] In fact, by comparing the original and modified MCs [18], one can show that the MC as presented in Example 3.9 also converges in polynomially many steps.

Bibliography

[1] David Aldous. Random walks on finite groups and rapidly mixing Markov chains. In *Seminar on probability, XVII*, pages 243–297. Springer, Berlin, 1983.

[2] David Aldous and James Fill. Reversible Markov chains and random walks on graphs.
`http://www.stat.berkeley.edu/~aldous/book.html`.

[3] V. S. Anil Kumar and H. Ramesh. Coupling vs. conductance for the Jerrum-Sinclair chain. *Random Structures Algorithms*, 18(1):1–17, 2001.

[4] Catherine Bandle. *Isoperimetric Inequalities and Applications*. Pitman (Advanced Publishing Program), Boston, Mass., 1980.

[5] Edward A. Bender. The asymptotic number of non-negative integer matrices with given row and column sums. *Discrete Math.*, 10:217–223, 1974.

[6] Piotr Berman and Marek Karpinski. On some tighter inapproximability results (extended abstract). In *Automata, languages and programming (Prague, 1999)*, pages 200–209. Springer, Berlin, 1999.

[7] Béla Bollobás. *Modern Graph Theory*. Springer-Verlag, New York, 1998.

[8] Andrei Z. Broder. How hard is it to marry at random? (on the approximation of the permanent). In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, pages 50–58. ACM Press, 1986. Erratum in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 1988, p. 551.

[9] R. L. Brooks. On colouring the nodes of a network. *Proc. Cambridge Philos. Soc.*, 37:194–197, 1941.

[10] Russ Bubley and Martin Dyer. Path coupling: a technique for proving rapid mixing in Markov chains. In *Proceedings of the 38th Symposium on Foundations of Computer Science (FOCS)*, pages 223–231. IEEE Computer Society Press, 1997.

[11] Russ Bubley and Martin Dyer. Path coupling, Dobrushin uniqueness, and approximate counting. Technical Report 97.04, School of Computer Studies, University of Leeds, January 1997.

[12] Russ Bubley and Martin Dyer. Faster random generation of linear extensions. *Discrete Math.*, 201(1-3):81–88, 1999.

[13] Russ Bubley, Martin Dyer, and Catherine Greenhill. Beating the 2Δ bound for approximately counting colourings: a computer-assisted proof of rapid mixing. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1998)*, pages 355–363, New York, 1998. ACM Press.

[14] Russ Bubley, Martin Dyer, and Mark Jerrum. An elementary analysis of a procedure for sampling points in a convex body. *Random Structures Algorithms*, 12(3):213–235, 1998.

[15] Charles J. Colbourn, J. Scott Provan, and Dirk Vertigan. A new approach to solving three combinatorial enumeration problems on planar graphs. *Discrete Appl. Math.*, 60(1-3):119–129, 1995.

[16] Colin Cooper and Alan M. Frieze. Mixing properties of the Swendsen-Wang process on classes of graphs. *Random Structures Algorithms*, 15(3-4):242–261, 1999.

[17] Persi Diaconis. *Group representations in probability and statistics.* Institute of Mathematical Statistics, Hayward, CA, 1988.

[18] Persi Diaconis and Laurent Saloff-Coste. Comparison theorems for reversible Markov chains. *Ann. Appl. Probab.*, 3(3):696–730, 1993.

[19] Persi Diaconis and Daniel Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Appl. Probab.*, 1(1):36–61, 1991.

[20] Alexander Dinghas. Über eine Klasse superadditiver Mengenfunktionale von Brunn-Minkowski-Lustenikschem typus. *Math. Zeitschr.*, 68:111–125, 1957.

[21] Martin Dyer, Alan Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. In *Proceedings of the 40th Symposium on Foundations of Computer Science (FOCS)*, pages 210–217. IEEE Computer Society Press, 1999.

[22] Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. Assoc. Comput. Mach.*, 38(1):1–17, 1991.

[23] Martin Dyer and Catherine Greenhill. A more rapidly mixing Markov chain for graph colorings. *Random Structures Algorithms*, 13(3-4):285–317, 1998.

[24] Martin Dyer and Catherine Greenhill. On Markov chains for independent sets. *J. Algorithms*, 35(1):17–49, 2000.

[25] H. G. Eggleston. *Convexity*. Cambridge University Press, New York, 1958.

[26] Tomás Feder and Milena Mihail. Balanced matroids. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 26–38. ACM Press, 1992.

[27] Uriel Feige and Carsten Lund. On the hardness of computing the permanent of random matrices. *Comput. Complexity*, 6(2):101–132, 1996/97.

[28] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM J. Comput.*, 22(5):994–1005, 1993.

[29] Michael R. Garey and David S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco, Calif., 1979.

[30] Leslie Ann Goldberg. Computation in permutation groups: counting and randomly sampling orbits. In *Surveys in combinatorics, 2001 (Sussex)*, pages 109–143. Cambridge Univ. Press, Cambridge, 2001.

[31] Oded Goldreich. *Introduction to Complexity Theory*. Lecture Notes Series of the Electronic Colloquium on Computational Complexity. `http://www.eccc.uni-trier.de/eccc/`, 1999.

[32] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. The Clarendon Press Oxford University Press, New York, second edition, 1992.

[33] Mark Jerrum. Computational Pólya theory. In *Surveys in combinatorics, 1995 (Stirling)*, pages 103–118. Cambridge Univ. Press, Cambridge, 1995.

[34] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.*, 22(5):1087–1116, 1993.

[35] Mark Jerrum and Alistair Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 482–520. PWS, 1996.

[36] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Electronic Colloquium on Computational Complexity*, TR00-079, 2000.

[37] Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.*, 43(2-3):169–188, 1986.

[38] Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures Algorithms*, 11(1):1–50, 1997.

[39] Richard M. Karp, Michael Luby, and Neal Madras. Monte Carlo approximation algorithms for enumeration problems. *J. Algorithms*, 10(3):429–448, 1989.

[40] P. W. Kasteleyn. Graph theory and crystal physics. In Frank Harary, editor, *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.

[41] Torgny Lindvall. *Lectures on the coupling method.* John Wiley & Sons Inc., New York, 1992. A Wiley-Interscience Publication.

[42] Torgny Lindvall and L. C. G. Rogers. Coupling of multidimensional diffusions by reflection. *Ann. Probab.*, 14(3):860–872, 1986.

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

[44] László Lovász and M. D. Plummer. *Matching Theory.* North-Holland, 1986.

[45] Michael Luby and Eric Vigoda. Approximately counting up to four (extended abstract). In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 682–687. ACM Press, 1997.

[46] Michael Luby and Eric Vigoda. Fast convergence of the Glauber dynamics for sampling independent sets. *Random Structures Algorithms*, 15(3-4):229–241, 1999.

[47] Neal Madras and Dana Randall. Markov chain decomposition for convergence rate analysis. To appear in the *Annals of Applied Probability.*

[48] Meena Mahajan and V. Vinay. A combinatorial algorithm for the determinant. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 730–738. ACM/SIAM, 1997.

[49] Lisa McShine. Random sampling of labeled tournaments. *Electron. J. Combin.*, 7(1):Research Paper 8, 9 pp. (electronic), 2000.

[50] Milena Mihail. Conductance and convergence of Markov chains: a combinatorial treatment of expanders. In *Proceedings of the 30th Symposium on Foundations of Computer Science (FOCS)*, pages 526–531. IEEE Computer Society Press, 1989.

[51] Ben Morris and Alistair Sinclair. Random walks on truncated cubes and sampling 0-1 knapsack solutions. In *Proceedings of the 40th Symposium on Foundations of Computer Science (FOCS)*, pages 230–240. IEEE Computer Society Press, 1999.

[52] J. R. Norris. *Markov chains.* Cambridge University Press, Cambridge, 1997.

[53] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley Publishing Company, Reading, MA, 1994.

[54] Neil Robertson, P. D. Seymour, and Robin Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Ann. of Math. (2)*, 150(3):929–975, 1999.

[55] Janos Simon. On the difference between one and many. In *Automata, Languages and Programming (Fourth Colloq., Univ. Turku, Turku, 1977), Lecture Notes in Computer Science, Vol. 52*, pages 480–491. Springer, Berlin, 1977.

[56] Alistair Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combin. Probab. Comput.*, 1(4):351–370, 1992.

[57] Alistair Sinclair. *Algorithms for Random Generation and Counting: a Markov Chain Approach.* Birkhäuser Boston Inc., Boston, MA, 1993.

[58] P. Tetali and S. Vempala. Random sampling of Euler tours. *Algorithmica*, 30(3):376–385, 2001.

[59] Seinosuke Toda and Mitsunori Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 21(2):316–328, 1992.

[60] W. T. Tutte. *Graph Theory*, volume 21 of *Encyclopedia of Mathematics: Combinatorics.* Addison-Wesley, 1984.

[61] L. G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th annual ACM Symposium on Theory of Computing (STOC)*, pages 249–261. ACM, 1979.

[62] L. G. Valiant. The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8(2):189–201, 1979.

[63] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoret. Comput. Sci.*, 47(1):85–93, 1986.

[64] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

[65] J. H. van Lint and R. M. Wilson. *A course in combinatorics.* Cambridge University Press, Cambridge, 1992.

[66] Eric Vigoda. Improved bounds for sampling colorings. *J. Math. Phys.*, 41(3):1555–1569, 2000.