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 $\pm 10\%$.

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

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

statements) that are mutually exclusive and each executed with probability $\frac{1}{2}$. All of these possible models are equivalent, modulo polynomial transformations in run-time. 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 [67, 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

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

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.

[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.

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

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

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 [72]).

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 an FPRAS if and only if S admits an FPAUS. For full details, refer to Jerrum, Valiant and Vazirani [49]. 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 an FPRAS for $|\mathcal{M}(G)|$.

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

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

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

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

As our estimator for $|\mathcal{M}(G)|$, we use the random variable

$$N := \left(\prod_{i=1}^{m} \overline{Z}_i \right)^{-1}.$$

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

Note that $\mathbb{E}[\,\overline{Z}_1\overline{Z}_2\ldots\overline{Z}_m] = \mu_1\mu_2\ldots\mu_m$, and furthermore

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

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 an 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 [39] 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

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

for all $t \in \mathbb{N}$ and all $x_t, x_{t-1}, \ldots, x_0 \in \Omega$. 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 an 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. An 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 an MC. If the function $\pi' : \Omega \to [0, 1]$ satisfies*

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

and

[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 $x \in \Omega$.

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

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

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

□

Remark 3.8. Condition (3.11) is known as *detailed balance*. An 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

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

this 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 an FPRAS for the number of matchings in a graph.

[7] In fact, by comparing the original and modified MCs [22], one can show that the MC as presented in Example 3.9 also converges in polynomially many steps.