CHAPTER

12

THE MARKOV CHAIN MONTE CARLO METHOD: AN APPROACH TO APPROXIMATE COUNTING AND INTEGRATION

Mark Jerrum **Alistair Sinclair**

In the area of statistical physics, Monte Carlo algorithms based on Markov chain simulation have been in use for many years. The validity of these algorithms depends crucially on the rate of convergence to equilibrium of the Markov chain being simulated. Unfortunately, the classical theory of stochastic processes hardly touches on the sort of non-asymptotic analysis required in this application. As a consequence, it had previously not been possible to make useful, mathematically rigorous statements about the quality of the estimates obtained.

Within the last ten years, analytical tools have been devised with the aim of correcting this deficiency. As well as permitting the analysis of Monte Carlo algorithms for classical problems in statistical physics, the introduction of these tools has spurred the development of new approximation algorithms for a wider class of problems in combinatorial enumeration and optimization. The "Markov chain Monte Carlo" method has been applied to a variety of such problems, and often provides the only known efficient (i.e., polynomial time) solution technique.

INTRODUCTION 12.1

This chapter differs from the others in being concerned more with problems of counting and integration, and correspondingly less with optimization. The problems we address still tend to be complete, but now for the complexity class of counting problems known as #*P*, rather than for the more familiar class *NP* of decision problems. It also differs from most of the others in being centred around a general paradigm for designing approximation algorithms, rather than around a specific problem domain. We shall refer to this paradigm as the "Markov chain Monte Carlo method." It has been widely used for many years in several application areas, most notably in computational physics and combinatorial optimization. However, these algorithms have been almost entirely heuristic in nature, in the sense that no rigorous guarantees could be given for the quality of the approximate solutions they produced. Only relatively recently have analytical tools been developed that allow Markov chain Monte Carlo algorithms to be placed on a firm foundation with precise performance guarantees. This has led to an upsurge of interest in this area in computer science, and in the development of the first provably efficient approximation algorithms for several fundamental computational problems. This chapter aims to describe these new tools, and give the reader a flavor of the most significant applications.

The Markov chain Monte Carlo method provides an algorithm for the following general computational task. Let Ω be a very large (but finite) set of combinatorial structures (such as the set of possible configurations of a physical system, or the set of feasible solutions to a combinatorial optimization problem), and let π be a probability distribution on Ω. The task is to sample an element of Ω at random according to the distribution π.

In addition to their inherent interest, combinatorial sampling problems of this kind have many computational applications. The most notable of these are the following:

I. *Approximate counting*: i.e., estimate the cardinality of Ω. A natural generalization is *discrete integration*, where the goal is to estimate a weighted sum of the form $\sum_{x\in\Omega} w(x)$, where w is a positive function defined on Ω.

II. *Statistical physics*: here Ω is the set of configurations of a statistical mechanical system, and π is a natural probability distribution on Ω (such as the Gibbs distribution), in which the probability of a configuration is related to its energy. The task is to sample configurations according to π, in order to examine properties of a "typical" configuration and to estimate the expectations of certain natural random variables (such as the mean energy of a configuration). Computations of this kind are typically known as "Monte Carlo experiments."

III. *Combinatorial optimization*: here Ω is the set of feasible solutions to an optimization problem, and π is a distribution that assigns, in some natural way, higher weight to solutions with a better objective function value. Sampling from π thus favors better solutions. An example of this approach is the popular optimization heuristic known as "simulated annealing."

In all the above applications, more or less routine statistical procedures are used to infer the desired computational information from a sequence of independent random samples from the distribution π. (This point will be illustrated by examples later in the chapter.) In algorithms of this kind, therefore, it is the sampling itself which presents the major challenge.

The Markov chain Monte Carlo method solves the sampling problem as follows. We construct a Markov chain having state space Ω and stationary distribution π. The Markov chain is designed to be *ergodic*, i.e., the probability distribution over Ω converges asymptotically to π, regardless of the initial state. Moreover, its transitions correspond to simple random perturbations of structures in Ω, and hence are simple to simulate. Now we may sample from π as follows: starting from an arbitrary state in Ω, simulate the Markov chain for some number, T, of steps, and output the final state. The ergodicity means that, by taking T large enough, we can ensure that the distribution of the output state is arbitrarily close to the desired distribution π.

In most applications it is not hard to construct a Markov chain having the above properties. What is not at all obvious, however, is how to choose the number of simulation steps T, which is the crucial factor in the running time of any algorithm that uses the chain. Of course, if the algorithm is to be efficient, then T must be very much smaller than the size of Ω; equivalently, we require that the Markov chain be close to its stationary distribution after taking a very short random walk through Ω. Loosely, we shall call a Markov chain having this property "rapidly mixing," and the number of steps required for the distribution to become close to π the "mixing time" of the chain.

In heuristic applications of the Markov chain Monte Carlo method, T is usually chosen by empirical observation of the Markov chain, or by an appeal to combinatorial or physical intuition. This means that no precise claim can be made about the distribution of the samples, so no performance guarantee can be given for the associated approximation algorithms. This observation holds for almost all existing Monte Carlo experiments in physics, and for almost all applications of simulated annealing in combinatorial optimization. It is a considerable challenge for theoretical computer science to analyze the mixing time in such applications, and hence to place these algorithms on a firm foundation.

Unfortunately, the classical theory of stochastic processes hardly touches upon the sort of non-asymptotic analysis required in this situation. In recent years, however, novel analytical tools have been developed that allow the mixing time of Markov chains of this kind to be determined quite precisely. This in turn has led to the first rigorous analysis of the running time of various approximation algorithms based on the Markov chain Monte Carlo method, as well as to the design of entirely new algorithms of this type. This chapter aims to present some of these analytical tools, and to describe their most important algorithmic applications.

The remainder of the chapter is organized as follows. Section 12.2 illustrates how the Markov chain Monte Carlo method can be applied to a combinatorial problem that is very simple to state, namely the problem of counting the number of solutions to an instance of the Knapsack problem. Section 12.3 describes two tools for bounding the mixing time of Markov chains that have proved successful in a number of applications (though not as yet in the case of the Knapsack solution counting problem). An illustration of how these tools might be applied is provided by a toy example, which is a radically simplified version of the Knapsack problem. Section 12.4 introduces a more substantial and better motivated application drawn from the field of statistical physics, namely, estimating the partition function of a monomer-dimer system. This computational problem includes, as a special case, approximately counting matchings of all sizes in a graph. Section 12.5 then catalogues various other problems to which the Markov chain Monte Carlo method has been successfully applied. The concluding Section 12.6 formulates the simulated annealing heuristic as an instance of the Markov chain Monte Carlo method, and indicates how the techniques described in Sections 12.3 and 12.4 can, in certain cases, give rigorous results on the performance of the heuristic.

12.2 AN ILLUSTRATIVE EXAMPLE

To introduce and motivate the Markov chain Monte Carlo method, consider the following problem: given $a = (a_0, \dots, a_{n-1}) \in \mathbb{N}^n$ and $b \in \mathbb{N}$, estimate the number N of 0,1-vectors $x \in \{0,1\}^n$ satisfying the inequality $a \cdot x = \sum_{i=0}^{n-1} a_i x_i \leq b$. If the vector a gives the sizes of n items to be packed into a knapsack of capacity b, the quantity to be estimated can be interpreted as the number of combinations of items that can be fitted into the knapsack, which we shall refer to as "Knapsack solutions." Although this problem is perhaps not of pressing practical importance, it does provide a convenient demonstration of the method. No efficient deterministic algorithm is known for accurately counting Knapsack solutions and there is convincing complexity-theoretic evidence that none exists. In this regard at least, the chosen example is more realistic than the familiar classical demonstration of the Monte Carlo method, which involves estimating π by casting a needle onto a ruled surface [Usp37].

The nature of the "convincing evidence" mentioned above is that the problem of counting Knapsack solutions is complete for Valiant's complexity class #*P* [GJ79, Val79b] with respect to polynomial-time Turing reductions. The class #*P* is the counting analogue of the more familiar class *NP* of decision problems. A #*P*-complete problem is computationally equivalent (via polynomial-time Turing reductions) to computing the number of satisfying assignments of a boolean formula in CNF, or the number of accepting computations of a polynomial-time nondeterministic Turing machine. Obviously, computing the number of accepting computations is at least as hard as deciding whether an accepting computation exists, so #*P* certainly contains *NP*. Less obviously, as Toda [Tod89] has demonstrated, #*P* also essentially contains the entire Meyer-Stockmeyer polynomial-time hierarchy. Thus, in structural terms, and maybe in fact, a #*P*-complete problem is computationally even harder than an *NP*-complete one [Jer94].

A classical Monte Carlo approach to solving the Knapsack problem would be based on an estimator of the following type. Select uniformly at random (u.a.r.) a vector $x \in \{0,1\}^n$ from the corners of the n-dimensional boolean hypercube; if $a \cdot x \leq b$ then return 2^n, otherwise return 0. The outcome of this experiment is a random variable whose expectation is precisely N, the value we are required to estimate. In principle, we need only perform sufficiently many trials and take the mean of the results to obtain a reliable approximation to N within any desired accuracy. In practice, the method fails badly, as we can see by taking $a = (1, \ldots, 1)$ and $b = n/3$. Note that, with these values, the expected number of trials before the first non-zero outcome is exponential in n. Thus, a sequence of trials of "reasonable" length will typically yield a mean of 0, even though the actual number of Knapsack solutions is exponentially large. Clearly, the variance of the estimator is far too large for it to be of any practical value.

Before considering other, potentially better approaches, we should pause to consider what distinguishes a good algorithm from a bad one. In the theoretical computer science tradition, we consider an efficient algorithm to be one that terminates in a number of steps that is bounded by a polynomial in the length of the input. More formally, suppose $f : \Sigma^* \to \mathbb{N}$ is a function mapping problem instances (encoded as words over some convenient alphabet Σ) to natural numbers. For example, in the case of the Knapsack problem, f might map (encodings of) the pair $a \in \mathbb{N}^n$ and $b \in \mathbb{N}$ to the number of solutions of $a \cdot x \leq b$ in the set $x \in \{0, 1\}^n$. It should be clear that any combinatorial enumeration problem can be cast in this framework. A *randomized approximation scheme* for f is a randomized algorithm that takes as input a word (instance) $x \in \Sigma^n$ and $\varepsilon > 0$, and produces as output a number Y (a random variable) such that[1]

$$\Pr\big((1-\varepsilon)f(x) \leq Y \leq (1+\varepsilon)f(x)\big) \geq \tfrac{3}{4}. \tag{12.1}$$

A randomized approximation scheme is said to be *fully polynomial* [KL83] if it runs in time polynomial in n (the input length) and ε^{-1}. We shall abbreviate the rather unwieldy phrase "Fully Polynomial Randomized Approximation Scheme" to FPRAS.

The above provides a clear-cut definition of an "efficient approximation algorithm" that has at least a certain degree of intuitive appeal. The naive Monte Carlo algorithm described earlier is not efficient in the FPRAS sense, which is reassuring. On the other hand, it is certainly debatable whether an algorithm with running time n^{10} constitutes an efficient solution in anything other than a theoretical sense. In this chapter, we always use the FPRAS as our notion of efficient approximation algorithm; while this has the advantage of providing us with clear goals, it is obvious that in practical applications some more demanding notion of "efficient approximation" would be necessary.

Returning to the Knapsack problem, we might try applying the Markov chain Monte Carlo method as follows. Consider the Markov chain $\mathfrak{M}_{\mathrm{Knap}}$ with state space $\Omega = \{x \in \{0, 1\}^n : a \cdot x \leq b\}$, i.e., the set of all Knapsack solutions, and transitions from each state $x = (x_0, \ldots, x_{n-1}) \in \Omega$ defined by the following rule:

I. with probability $\frac{1}{2}$ let $y = x$; otherwise,

II. select i u.a.r. from the range $0 \leq i \leq n-1$ and let $y' = (x_0, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_{n-1})$;

III. if $a \cdot y' \leq b$, then let $y = y'$, else let $y = x$;

the new state is y. Informally, the process $\mathfrak{M}_{\mathrm{Knap}}$ may be interpreted as a random walk (with stationary moves) on the boolean hypercube, truncated by the hyperplane $a \cdot x = b$.

---

[1]There is no significance in the constant $\frac{3}{4}$ appearing in the definition, beyond its lying strictly between $\frac{1}{2}$ and 1. Any success probability greater than $\frac{1}{2}$ may be boosted to $1 - \delta$ for any desired $\delta > 0$ by performing a small number of trials and taking the median of the results; the number of trials required is $O(\ln \delta^{-1})$ [JVV86].

The Markov chain $\mathfrak{M}_{\text{Knap}}$ is ergodic, since all pairs of states intercommunicate via the state $(0, \dots, 0)$, and the presence of loops ensures aperiodicity; it is readily checked that the stationary distribution is uniform over Ω. This observation immediately suggests a procedure for selecting Knapsack solutions almost u.a.r.: starting in state $(0, \dots, 0)$, simulate $\mathfrak{M}_{\text{Knap}}$ for sufficiently many steps that the distribution over states is "close" to uniform, then return as result the current state. Of course, sampling from Ω is not quite the same as estimating the size of Ω (which is our goal), but the second task can be related to the first using a simple trick, which we now describe.[2]

We keep the vector a fixed, but allow the bound b to vary, writing $\Omega(b)$ and $\mathfrak{M}_{\text{Knap}}(b)$ to make explicit the dependence of the Markov chain on b. Assume without loss of generality that $a_0 \le a_1 \le \cdots \le a_{n-1}$, and define $b_0 = 0$ and $b_i = \min\left\{b, \sum_{j=0}^{i-1} a_j\right\}$, for $1 \le i \le n$. It may easily be verified that $|\Omega(b_{i-1})| \le |\Omega(b_i)| \le (n+1)|\Omega(b_{i-1})|$, for $1 \le i \le n$, the key observation being that any element of $\Omega(b_i)$ may be converted into an element of $\Omega(b_{i-1})$ by changing the rightmost 1 to a 0. Now write

$$|\Omega(b)| = |\Omega(b_n)| = \frac{|\Omega(b_n)|}{|\Omega(b_{n-1})|} \times \frac{|\Omega(b_{n-1})|}{|\Omega(b_{n-2})|} \times \cdots \times \frac{|\Omega(b_1)|}{|\Omega(b_0)|} \times |\Omega(b_0)|, \tag{12.2}$$

where, of course, $|\Omega(b_0)| = 1$. The reciprocals $\rho_i = |\Omega(b_{i-1})|/|\Omega(b_i)|$ of each of the ratios appearing in (12.2) may be estimated by sampling almost uniformly from $\Omega(b_i)$ using the Markov chain $\mathfrak{M}_{\text{Knap}}(b_i)$, and computing the fraction of the samples that lie within $\Omega(b_{i-1})$.

Consider the random variable associated with a single trial — i.e., one run of the Markov chain $\mathfrak{M}_{\text{Knap}}(b_i)$ — that is defined to be 1 if the final state is a member of $\Omega(b_{i-1})$, and 0 otherwise. If we were able to simulate $\mathfrak{M}_{\text{Knap}}(b_i)$ "to infinity," the expectation of this random variable would be precisely ρ_i. In reality, we must terminate the simulation at some point, thereby introducing a small though definite bias that ought to be accounted for. To avoid obscuring the main ideas, let us ignore this technical complication for the time being; details of this kind will be attended to when we address a more realistic example in Section 12.4. With the simplifying assumption of zero bias, the expectation of an individual trial is ρ_i, and its variance, since it is a 0,1-variable, is $\rho_i(1-\rho_i)$. Suppose we perform $t = 17\varepsilon^{-2}n^2$ trials, and let $\overline{X}_i$ denote the sample mean. In analyzing the efficiency of Monte Carlo estimators, the quantity to focus on is the ratio of the variance of the estimator to the square of its expectation; in this instance we have

$$\frac{\operatorname{Var}\overline{X}_i}{\rho_i^2} = \frac{1-\rho_i}{t\rho_i} \le \frac{n}{t} = \frac{\varepsilon^2}{17n},$$

where the inequality follows from earlier-noted bound $\rho_i = |\Omega(b_{i-1})|/|\Omega(b_i)| \ge (n+1)^{-1}$.

Suppose the above process is repeated for each of the n ratios in equation (12.2), and denote by Z the random variable $Z = \overline{X}_n\overline{X}_{n-1}\dots\overline{X}_1$ which is the product of the various sample means. Then, since the random variables $\overline{X}_i$ are independent, the expectation

[2]For a more detailed discussion of the problem of inferring information from observations of a Markov chain, see [Ald87, Gill93, Kah94].

of Z is $\mathrm{E}\,Z = \rho_n \rho_{n-1} \ldots \rho_1 = |\Omega(b)|^{-1}$, and

$$\frac{\operatorname{Var} Z}{(\mathrm{E}\,Z)^2} = \prod_{i=1}^{n} \left[1 + \frac{\operatorname{Var} \overline{X}_i}{\rho_i^2} \right] - 1 \le \left[1 + \frac{\varepsilon^2}{17n} \right]^n - 1 \le \frac{\varepsilon^2}{16},$$

assuming $\varepsilon \le 1$. By Chebyshev's inequality, this implies that

$$\Pr\big((1-\varepsilon/2)|\Omega(b)|^{-1} \le Z \le (1+\varepsilon/2)|\Omega(b)|^{-1}\big) \ge \tfrac{3}{4},$$

so the random variable $Y = Z^{-1}$ satisfies (12.1), i.e., it yields a randomized approximation scheme for the number of Knapsack solutions. The idea of expressing the quantity to be estimated as a product of small factors in the style of (12.2) and then estimating each of the factors by separate Monte Carlo experiments, is one that has repeatedly proved useful in this area, since it provides a general tool for reducing approximate counting to sampling.

Observe that the total number of trials (Markov chain simulations) used is $nt = 17\varepsilon^{-2}n^3$, which is polynomial in n and ε^{-1}. The method described above is therefore an FPRAS for the number of Knapsack solutions, provided the Markov chain $\mathfrak{M}_{\mathrm{Knap}}$ is "rapidly mixing," that is to say, is close to stationarity after a number of steps that is polynomial in n. This is a non-trivial condition, since the size of the state space Ω is exponential in n. Given the relative simplicity of the Markov chain $\mathfrak{M}_{\mathrm{Knap}}$, it is humbling that the question of whether $\mathfrak{M}_{\mathrm{Knap}}$ is rapidly mixing is even now unresolved. The wider question of whether there exists an FPRAS of any kind for the Knapsack problem is also unresolved, though the Markov chain simulation approach sketched above seems to offer the best hope. Using it, Dyer et al. [DFKKPV93] were able to obtain a randomized approximation scheme for the number of Knapsack solutions whose running time is $\varepsilon^{-2}\exp\big(\mathrm{O}(\sqrt{n}\,(\log n)^{5/2})\big)$, and this is asymptotically the fastest known.

OPEN PROBLEM 12.1 Is the Markov chain $\mathfrak{M}_{\mathrm{knap}}$ rapidly mixing (i.e., is its mixing time bounded by a polynomial in the dimension n — see next section) for all choices of the bound b and item sizes a?

12.3 TWO TECHNIQUES FOR BOUNDING THE MIXING TIME

It will be clear from Section 12.2 that successful application of the Markov chain Monte Carlo method rests on obtaining good bounds on the time taken for a Markov chain to become close to stationarity.

There are a number of ways of quantifying "closeness" to stationarity, but they are all essentially equivalent in this application. Let $\mathfrak{M}$ be an ergodic Markov chain on state space Ω with transition probabilities $P : \Omega^2 \to [0,1]$. Let $x \in \Omega$ be an arbitrary state, and denote by $P^t(x,\cdot)$ the distribution of the state at time t given that x is the initial state. Denote by π the stationary distribution of $\mathfrak{M}$. Then the *variation distance* at time t with respect to the initial state x is defined to be

$$\Delta_x(t) = \max_{S \subseteq \Omega} \left| P^t(x,S) - \pi(S) \right| = \tfrac{1}{2} \sum_{y \in \Omega} \left| P^t(x,y) - \pi(y) \right|.$$

Note that the variation distance provides a uniform bound, over all events $S \subseteq \Omega$, of the difference in probabilities of occurrence of event S under the stationary and t-step distributions. The rate of convergence of $\mathfrak{M}$ to stationarity may then be measured by the function

$$\tau_x(\varepsilon) = \min\{t : \Delta_x(t') \le \varepsilon \text{ for all } t' \ge t\},$$

which we shall refer to as the "mixing time" of the Markov chain.

The classical approach to bounding $\tau_x(\varepsilon)$ is via a "coupling" argument. This approach is very successful in the context of highly symmetric Markov chains (e.g., those associated with card shuffling [Ald81, Dia88]), but seems difficult to apply to the kind of "irregular" Markov chains that arise in the analysis of Monte Carlo algorithms. Two exceptions are the analyses of Aldous [Ald90] and Broder [Bro89] for a Markov chain on spanning trees of a graph, and of Matthews [Mat91] for a Markov chain related to linear extensions of a partial order. A glance at the latter paper will give an impression of the technical complexities that can arise.[3]

We should point out that the coupling method has very recently shown signs of staging a comeback. Jerrum [Jer95] has presented a simple application to sampling vertex colorings of a low-degree graph. Propp and Wilson [PW95] have some novel and attractive thoughts on applying coupling when the state space of the Markov chain has a natural lattice structure; their ideas are encouraging, and provide one of the ingredients in Luby, Randall, and Sinclair's [LRS95] analysis of a Markov chain on dimer coverings of certain planar (geometric) lattice graphs. Also, Bubley, Dyer, and Jerrum [BDJ96] have applied coupling to demonstrate rapid mixing of a certain random walk in a convex body, a situation we return to in Section 12.5.2. Finally, coupling has been used in a Markov chain approach to protocol testing by Mihail and Papadimitriou [MP94]. Despite this activity, it is not yet clear how far the coupling method can be pushed in the analysis of complex Markov chains.

In this section we consider two recently proposed alternatives to coupling, which tend to give weaker bounds but which are applicable in a wider range of situations. Historically [Sin93, SJ89], these two methods were not separate, but were developed together in a composite approach to bounding $\tau_x(\varepsilon)$; however, for practical purposes it is better to view them now as distinct approaches. We describe the "canonical path" argument first, and complete the section with a treatment of the "conductance" argument. For further discussion of these approaches, and various refinements of them, see, e.g., [DS91, Sin92, DSC93, Kah95].

We shall assume throughout the rest of the section that $\mathfrak{M}$ is *reversible*, that is to say, satisfies the *detailed balance* condition:

$$Q(x,y) = \pi(x)P(x,y) = \pi(y)P(y,x), \quad \text{for all } x, y \in \Omega;$$

furthermore, we assume the loop probabilities $P(x,x)$ are at least $\frac{1}{2}$ for all $x \in \Omega$. Since

[3]For a more direct approach to this problem, using a conductance argument as described below, see [KK90].

the Markov chain $\mathfrak{M}$ is a constructed one, it is not at all difficult to arrange that these two conditions are met.

12.3.1 CANONICAL PATHS

To describe the canonical path argument, we view $\mathfrak{M}$ as an undirected graph with vertex set Ω and edge set $E = \big\{\{x, y\} \in \Omega^{(2)} : Q(x, y) > 0\big\}$; this makes sense because of the reversibility condition. For each (ordered) pair $(x, y) \in \Omega^2$, we specify a canonical path γ_{xy} from x to y in the graph (Ω, E); the canonical path γ_{xy} corresponds to a sequence of legal transitions in $\mathfrak{M}$ that leads from initial state x to final state y. Denote by $\Gamma = \{\gamma_{xy} : x, y \in \Omega\}$ the set of all canonical paths. For the method to yield good bounds, it is important to choose a set of paths Γ that avoids the creation of "hot spots:" edges of the graph that carry a particularly heavy burden of canonical paths. The degree to which an even loading has been achieved is measured by the quantity

$$\bar{\rho} = \bar{\rho}(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)|\gamma_{xy}|,$$

where the maximum is over oriented edges e of (Ω, E), and $|\gamma_{xy}|$ denotes the length of the path γ_{xy}.

Intuitively, we might expect a Markov chain to be rapidly mixing if it contains no "bottlenecks," i.e., if it admits a choice of paths Γ for which $\bar{\rho}(\Gamma)$ is not too large. This intuition is formalized in the following result from Sinclair [Sin92], which is a slight modification of a theorem of Diaconis and Stroock [DS91].

PROPOSITION 12.1 Let $\mathfrak{M}$ be a finite, reversible, ergodic Markov chain with loop probabilities $P(x, x) \geq \frac{1}{2}$ for all states x. Let Γ be a set of canonical paths with maximum edge loading $\bar{\rho} = \bar{\rho}(\Gamma)$. Then the mixing time of $\mathfrak{M}$ satisfies $\tau_x(\varepsilon) \leq \bar{\rho}(\ln \pi(x)^{-1} + \ln \varepsilon^{-1})$, for any choice of initial state x.[4]

Proof. Combine Proposition 1 of [Sin92] and Theorem 5 of [Sin92]. ■

We demonstrate the canonical path method by applying it to a radically simplified version of the Knapsack Markov chain from Section 12.2. Instead of a random walk on the truncated boolean hypercube, we consider a random walk on the the full hypercube. This can be viewed as the degenerate case of the Knapsack Markov chain which obtains when $\sum_i a_i \leq b$, i.e., the knapsack is large enough to contain all items simultaneously.

Let $x = (x_0, x_1, \ldots, x_{n-1})$ and $y = (y_0, y_1, \ldots, y_{n-1})$ be arbitrary states in $\Omega = \{0, 1\}^n$. The canonical path γ_{xy} from x to y is composed of n edges, 0 to $n-1$, where edge i is simply $\big((y_0, \ldots, y_{i-1}, x_i, x_{i+1}, \ldots x_{n-1}), (y_0, \ldots, y_{i-1}, y_i, x_{i+1}, \ldots x_{n-1})\big)$, i.e., we flip the value of the ith bit from x_i to y_i. Note that some of the edges may be loops (if $x_i = y_i$). To compute $\bar{\rho}$, fix attention on a particular (oriented) edge

$$e = (w, w') = \big((w_0, \ldots, w_i, \ldots w_{n-1}), (w_0, \ldots, w_i', \ldots w_{n-1})\big),$$

and consider the number of canonical paths γ_{xy} that include e. The number of possible

[4] This Proposition also has a suitably stated converse; see Theorem 8 of [Sin92].

choices for x is 2^i, as the final $n-i$ positions are determined by $x_j = w_j$, for $j \geq i$, and by a similar argument the number of possible choices for y is 2^{n-i-1}. Thus, the total number of canonical paths using a particular edge e is 2^{n-1}; furthermore, $Q(e) = \pi(w)P(w,w') \geq 2^{-n}(2n)^{-1}$, and the length of every canonical path is exactly n. Plugging all these bounds into the definition of $\bar{\rho}$ yields $\bar{\rho} \leq n^2$. Thus, by Proposition 12.1, the mixing time for the random walk on the boolean hypercube is $\tau_x(\varepsilon) \leq n^2\big((\ln 2)n + \ln \varepsilon^{-1}\big)$. We call this Markov chain "rapidly mixing" because its mixing time grows only polynomially with the input size n (even though the size of the state space is exponential in n). The above bound is some way off the exact answer [Dia88], which is $\tau_x(\varepsilon) = \mathrm{O}\big(n(\ln n + \ln \varepsilon^{-1})\big)$, and the slackness we see here is typical of the method.

On reviewing the canonical path argument, we perceive what appears to be a major weakness. In order to compute the key quantity $\bar{\rho}$, we needed in turn to compute quantities such as $Q(e)$ that depend crucially on the size of the state space Ω. In the hypercube example this does not present a problem, but in more interesting examples we do not know the size of the state space: indeed, our ultimate goal will often be to estimate this very quantity. Fortunately, it is possible to finesse this obstacle by implicit counting using a carefully constructed injective map. The idea will be illustrated by application to the hypercube example.

Let edge $e = (w, w')$ be as before, and denote by $\mathrm{cp}(e) = \{(x,y) : \gamma_{xy} \ni e\}$ the set of all (endpoints of) canonical paths that use edge e. Define the map $\eta_e : \mathrm{cp}(e) \to \Omega$ as follows: if $(x,y) = \big((x_0,\dots,x_{n-1}),(y_0,\dots,y_{n-1})\big) \in \mathrm{cp}(e)$ then

$$\eta_e(x,y) = (u_0,\dots,u_{n-1}) = (x_0,\dots,x_{i-1},w_i,y_{i+1},\dots,y_{n-1}).$$

The crucial feature of the map η_e is that it is injective. To see this, observe that x and y may be unambiguously recovered from $(u_0,\dots,u_{n-1}) = \eta_e(x,y)$ through the explicit expressions

$$x = (u_0,\dots,u_{i-1},w_i,w_{i+1},\dots,w_{n-1})$$

and

$$y = (w_0,\dots,w_{i-1},w_i',u_{i+1},\dots,u_{n-1}).$$

Using the injective map η_e it is possible to evaluate $\bar{\rho}$ without recourse to explicit counting. Noting[5] that $\pi(x)\pi(y) = \pi(w)\pi(\eta_e(x,y))$, we have

$$\begin{aligned}\frac{1}{Q(e)}\sum_{\gamma_{xy}\ni e}\pi(x)\pi(y)|\gamma_{xy}| &= \frac{1}{\pi(w)P(w,w')}\sum_{\gamma_{xy}\ni e}\pi(w)\pi(\eta_e(x,y))\,|\gamma_{xy}| \\ &= \frac{n}{P(w,w')}\sum_{\gamma_{xy}\ni e}\pi(\eta_e(x,y)) \leq \frac{n}{P(w,w')} \leq 2n^2,\end{aligned}$$

where the penultimate inequality follows from the facts that η_e is injective, and that π is a probability distribution. Since the above argument is valid uniformly over the choice of e, we deduce $\bar{\rho} \leq 2n^2$. The factor of 2 as compared with the direct argument was lost to slight redundancy in the encoding: the map η_e was not quite a bijection.

[5]This is a trivial observation when the stationary distribution is uniform, as it is here, but it is sometimes possible, by judicious choice of η_e, to contrive such an identity even when the stationary distribution is non-uniform. See Section 12.4 for an example.

12.3.2 CONDUCTANCE

As advertised earlier, we now consider an alternative "conductance" approach to bounding $\tau_x(\varepsilon)$, which has proved useful in situations where the Markov chain can be given a geometric interpretation [DFK91]. The *conductance* [SJ89] of Markov chain $\mathfrak{M}$ is defined by

$$\Phi = \Phi(\mathfrak{M}) = \min_{\substack{S\subset\Omega \\ 0<\pi(S)\le 1/2}} \frac{Q(S,\overline{S})}{\pi(S)}, \tag{12.3}$$

where $Q(S,\overline{S})$ denotes the sum of $Q(x,y)$ over edges $\{x,y\}\in E$ with $x\in S$ and $y\in\overline{S}=\Omega-S$. The conductance may be viewed as a weighted version of edge expansion of the graph (Ω,E) associated with $\mathfrak{M}$. Alternatively, the quotient appearing in (12.3) can be interpreted as the conditional probability that the chain in equilibrium escapes from the subset S of the state space in one step, given that it is initially in S; thus, Φ measures the readiness of the chain to escape from any small enough region of the state space, and hence to make rapid progress towards equilibrium. This intuitive connection can be given a precise quantitative form as follows. (See [Ald87, Alon86, AM85, Che70, LS88] for related results.)

PROPOSITION 12.2 Let $\mathfrak{M}$ be a finite, reversible, ergodic Markov chain with loop probabilities $P(x,x)\ge\frac{1}{2}$ for all states x. Let Φ be the conductance of $\mathfrak{M}$ as defined in (12.3). Then the mixing time of $\mathfrak{M}$ satisfies $\tau_x(\varepsilon)\le 2\Phi^{-2}(\ln\pi(x)^{-1}+\ln\varepsilon^{-1})$, for any choice of initial state x.

Proof. Combine Proposition 1 of [Sin92] and Theorem 2 of [Sin92]. ∎

From Proposition 12.2 it will be apparent that good lower bounds on conductance translate to good upper bounds on the mixing time $\tau_x(\varepsilon)$. As we shall see presently, it is possible to bound the conductance of the random walk on the hypercube by considering the geometry of the hypercube and applying an "isoperimetric inequality."

For $x\in\Omega=\{0,1\}^n$ and $S\subseteq\Omega$, define

$$C(x)=\left\{\xi=(\xi_0,\dots,\xi_{n-1}) : |\xi_i-x_i|\le\tfrac{1}{2},\ \text{for all } i\right\},$$

and $C(S)=\bigcup_{x\in S}C(x)$. Observe that the mapping C provides a geometric interpretation of each set S of states as a body in n-dimensional space, and that within this interpretation the entire state space Ω is a hypercube $K=C(\Omega)$ of side 2. Each possible transition from a state in S to a state in $\overline{S}$ contributes one unit of area (i.e., $(n-1)$-dimensional volume) to $\partial C(S)-\partial K$, where ∂ denotes boundary, and each transition occurs with probability $\frac{1}{2n}$; thus,

$$Q(S,\overline{S})=\frac{1}{2n|\Omega|}\operatorname{vol}_{n-1}(\partial C(S)-\partial K), \tag{12.4}$$

where vol_d denotes d-dimensional volume.

Intuitively, if $\operatorname{vol}_n C(S)$ is large (but less than $\frac{1}{2}\operatorname{vol}_n K$), then $\partial C(S)-\partial K$ must also be large. It is this kind of intuition that is captured and formalized in an isoperimetric inequality. Rather than working with the Euclidean norm and using a classical isoperimetric inequality, it is advantageous in this instance to work with the l_∞-norm $\|\xi\|_\infty = \max\{|\xi_0|, \dots, |\xi_{n-1}|\}$ and its dual the l_1-norm $\|\xi\|_\infty^* = \|\xi\|_1 = |\xi_0| + \cdots + |\xi_{n-1}|$, and invoke a very refined isoperimetric inequality due to Dyer and Frieze [DF91], which holds for arbitrary norms.

Observe that $\mathrm{vol}_n C(S) = |S|$, $\mathrm{vol}_n K = 2^n$, and $\mathrm{diam}\, K = 2$, where diam denotes diameter in the l_∞-norm. From Theorem 3 of [DF91], taking F to be identically 1, we have, for $|S| \le \frac{1}{2}|\Omega|$,

$$\frac{\mathrm{vol}_n C(S)}{\mathrm{vol}_{n-1}(\partial C(S) - \partial K)} \le \tfrac{1}{2} \mathrm{diam}\, K;$$

it follows immediately that $\mathrm{vol}_{n-1}(\partial C(S) - \partial K) \ge |S|$. Combining this inequality with equation (12.4) yields

$$Q(S, \overline{S}) \ge \frac{|S|}{2n|\Omega|} = \frac{\pi(S)}{2n}.$$

From the definition of conductance, $\Phi \ge \frac{1}{2n}$, and hence, by Proposition 12.2, $\tau_x(\varepsilon) \le 8n^2\big((\ln 2)n + \ln \varepsilon^{-1}\big)$. It will be seen that for this example the two bounds obtained using the conductance and canonical paths arguments differ by just a small constant factor.

12.4 A MORE COMPLEX EXAMPLE: MONOMER-DIMER SYSTEMS

In this section we describe a significant computational problem to which the Markov chain Monte Carlo method has been successfully applied to yield an efficient approximation algorithm, or FPRAS. (This is in contrast to the Knapsack problem discussed in Section 12.2, which is still open.) Moreover, the Markov chain Monte Carlo method is to date the *only* approach that yields a provably efficient algorithm for this problem. This application will illustrate the full power of the analysis techniques described in the previous section. Our presentation is an improved version of one we originally gave in [JS89, Sin93].

The problem in question is a classical one from statistical physics, known as the *monomer-dimer problem*. In a monomer-dimer system, the vertices of a finite undirected graph $G = (V, E)$ are covered by a non-overlapping arrangement, or *configuration* of monomers (molecules occupying one site, or vertex of G) and dimers (molecules occupying two vertices that are neighbors in G). Typically, G is a regular lattice in some fixed number of dimensions. Three-dimensional systems occur classically in the theory of mixtures of molecules of different sizes [Gugg52] and in the cell-cluster theory of the liquid state [CdBS55]; in two dimensions, the system is used to model the adsorption of diatomic molecules on a crystal surface [Rob35]. For a more detailed account of the history and significance of monomer-dimer systems, the reader is referred to the seminal paper of Heilmann and Lieb [HL72] and the references given there.

It is convenient to identify monomer-dimer configurations with matchings in the graph G; a *matching* in G is a subset $M \subseteq E$ such that no two edges in M share an endpoint. Thus, a matching of cardinality k, or a k-*matching*, corresponds precisely to a monomer-dimer configuration with k dimers and $2(n-k)$ monomers, where $2n = |V|$ is the number of vertices in G.[6] To each matching M, a *weight* $w(M) = \lambda^{|M|}$ is assigned, where λ is a positive real parameter that reflects the contribution of a dimer to the energy of the system. The *partition function* of the system is defined as

$$Z(\lambda) \equiv Z_G(\lambda) = \sum_M w(M) = \sum_{k=0}^{n} m_k \lambda^k, \tag{12.5}$$

where $m_k \equiv m_k(G)$ is the number of k-matchings in G (or equivalently, the number of monomer-dimer configurations with k dimers). For a physical interpretation of (12.5), see [HL72].[7]

The partition function is a central quantity in statistical physics, and captures essentially everything one needs to know about the thermodynamics of the system, including quantities such as the free energy and the specific heat, and the location of phase transitions. With this in mind, in the remainder of this section we will develop an algorithm for computing Z_G at an arbitrary point $\lambda \geq 0$. We should also point out that $Z_G(\lambda)$ is of independent combinatorial interest, being nothing other than the generating function for matchings, or *matching polynomial* of G [LP86]. Thus, for example, $Z_G(1)$ enumerates all matchings in G, and the coefficient m_k enumerates matchings of cardinality k. We shall have more to say about these connections in Section 12.5.1.

Our starting point is the observation that no feasible method is known for computing Z *exactly* for general monomer-dimer systems; indeed, for any fixed value of $\lambda > 0$, the problem of computing $Z_G(\lambda)$ exactly for a given graph G is complete for the class #P of enumeration problems, which, as we explained in Section 12.2, may be regarded as convincing evidence that no polynomial time exact algorithm can exist for this problem [Val79b].[8] It is therefore pertinent to ask whether there exists an FPRAS for this problem. In this context, by an FPRAS we mean an algorithm which, given a pair (G, λ), and a parameter $\varepsilon > 0$, outputs a number Y such that

$$\Pr\big((1-\varepsilon)Z_G(\lambda) \leq Y \leq (1+\varepsilon)Z_G(\lambda)\big) \geq \tfrac{3}{4},$$

and runs in time polynomial in n and $\lambda' = \max\{1, \lambda\}$.[9]

[6] The assumption that the number of vertices in G is even is inessential and is made for notational convenience.

[7] More generally, there may be a weight λ_e associated with each edge $e \in E$, and the weight of M is then $w(M) = \prod_{e \in M} \lambda_e$. The algorithm we present here extends in a straightforward fashion to this more general setting.

[8] An efficient algorithm *does* exist for computing the leading coefficient m_n exactly, provided the graph G is planar. This quantity has an interpretation as the partition function of a system of *hard dimers*, in which no monomers are permitted. This algorithm, due independently to Fisher, Kasteleyn, and Temperley [Fish61, Kast61, TF61] in 1961, is a landmark achievement in the design of combinatorial algorithms. Unfortunately, it does not seem to extend either to non-planar graphs or to other coefficients.

[9] By analogy with the definition given in Section 12.2, this assumes that the edge weight λ is presented in unary. Thus, if the running time of the algorithm is to be polynomial in the size of the system, n, then the edge weight λ must be polynomially bounded in n. This is not a severe restriction in practice when computing the partition function.

For a given graph G, we will construct an FPRAS for Z_G by Monte Carlo simulation of a suitable Markov chain $\mathfrak{M}_{\text{match}}(\lambda)$, parameterized on the edge weight λ. The state space, Ω, is the set of all matchings in G, and the transitions are constructed so that the chain is ergodic with stationary distribution π_λ given by

$$\pi_\lambda(M) = \frac{\lambda^{|M|}}{Z(\lambda)}. \tag{12.6}$$

(Since G is fixed from now on, we drop the subscript from Z.) In other words, the stationary probability of each matching (monomer-dimer configuration) is proportional to its weight in the partition function (12.5). The Markov chain $\mathfrak{M}_{\text{match}}(\lambda)$, if simulated for sufficiently many steps, provides a method of sampling matchings from the distribution π_λ.

Distributions of this form are natural in statistical physics and are usually referred to as *canonical* or *Gibbs* distributions. Note that an alternative interpretation of the partition function is as the normalizing factor in this distribution. Sampling from this distribution at various values of λ has many applications, such as estimating the expectation of certain natural quantities associated with a configuration (e.g., the mean number of monomers, or the mean distance between a pair of monomers in a dense configuration of dimers). As we shall see shortly, it also allows one to approximate the partition function itself.

It is not hard to construct a Markov chain $\mathfrak{M}_{\text{match}}(\lambda)$ with the right asymptotic properties. Consider the chain in which transitions from any matching M are made according to the following rule:

I. with probability $\frac{1}{2}$ let $M' = M$; otherwise,

II. select an edge $e = \{u, v\} \in E$ u.a.r. and set

$$M' = \begin{cases} M - e & \text{if } e \in M; \\ M + e & \text{if both } u \text{ and } v \text{ are unmatched in } M; \\ M + e - e' & \text{if exactly one of } u \text{ and } v \text{ is matched in } M \\ & \qquad \text{and } e' \text{ is the matching edge;} \\ M & \text{otherwise;} \end{cases}$$

III. go to M' with probability $\min\{1, \pi_\lambda(M')/\pi_\lambda(M)\}$.

It is helpful to view this chain as follows. There is an underlying graph defined on the set of matchings Ω in which the neighbors of matching M are all matchings M' that differ from M via one of the following local perturbations: an edge is removed from M (a *type 1* transition); an edge is added to M (a *type 2* transition); or a new edge is exchanged with an edge in M (a *type 0* transition). Transitions from M are made by first selecting a neighbor M' u.a.r., and then actually making, or *accepting* the transition with probability $\min\{1, \pi_\lambda(M')/\pi_\lambda(M)\}$. Note that the ratio appearing in this expression is easy to compute: it is just λ^{-1}, λ or 1 respectively, according to the type of the transition.

As the reader may easily verify, this acceptance probability is constructed so that the transition probabilities $P(M, M')$ satisfy the detailed balance condition

$$Q(M, M') = \pi_\lambda(M) P(M, M') = \pi_\lambda(M') P(M', M), \quad \text{for all } M, M' \in \Omega,$$

i.e., $\mathfrak{M}_{\text{match}}(\lambda)$ is reversible with respect to the distribution π_λ. This fact, together with the observation that $\mathfrak{M}_{\text{match}}(\lambda)$ is irreducible (i.e., all states communicate, for example via the empty matching) and aperiodic (by step 1, the self-loop probabilities $P(M, M)$

are all non-zero), ensures that $\mathfrak{M}_{\text{match}}(\lambda)$ is ergodic with stationary distribution π_λ, as required.[10]

Having constructed a family of Markov chains with stationary distribution π_λ, our next task is to explain how samples from this distribution can be used to obtain a reliable statistical estimate of $Z(\lambda)$ at a specified point $\lambda = \widehat{\lambda} \geq 0$. Our strategy is to express $Z(\widehat{\lambda})$ as the product

$$Z(\widehat{\lambda}) = \frac{Z(\lambda_r)}{Z(\lambda_{r-1})} \times \frac{Z(\lambda_{r-1})}{Z(\lambda_{r-2})} \times \cdots \frac{Z(\lambda_2)}{Z(\lambda_1)} \times \frac{Z(\lambda_1)}{Z(\lambda_0)} \times Z(\lambda_0), \tag{12.7}$$

where $0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda_{r-1} < \lambda_r = \widehat{\lambda}$ is a suitably chosen sequence of values. Note that $Z(\lambda_0) = Z(0) = 1$. We will then estimate each factor $Z(\lambda_i)/Z(\lambda_{i-1})$ in this product by sampling from the distribution π_{λ_i}. This approach is analogous to that described in Section 12.2 for the Knapsack problem (see Equation (12.2)). For reasons that will become clear shortly, we will use the sequence of values $\lambda_1 = |E|^{-1}$ and $\lambda_i = (1+\frac{1}{n})^{i-1}\lambda_1$ for $1 \leq i < r$. The length r of the sequence is taken to be minimal such that $(1+\frac{1}{n})^{r-1}\lambda_1 \geq \widehat{\lambda}$, so we have the bound

$$r \leq \left\lceil 2n\left(\ln\widehat{\lambda} + \ln|E|\right)\right\rceil + 1. \tag{12.8}$$

To estimate the ratio $Z(\lambda_i)/Z(\lambda_{i-1})$, we will express it, or rather its reciprocal, as the expectation of a suitable random variable. Specifically, define the random variable $f_i(M) = \left(\frac{\lambda_{i-1}}{\lambda_i}\right)^{|M|}$, where M is a matching chosen from the distribution π_{λ_i}. Then we have

$$\mathrm{E}\, f_i = \sum_M \left(\frac{\lambda_{i-1}}{\lambda_i}\right)^{|M|} \frac{\lambda_i^{|M|}}{Z(\lambda_i)} = \frac{1}{Z(\lambda_i)} \sum_M \lambda_{i-1}^{|M|} = \frac{Z(\lambda_{i-1})}{Z(\lambda_i)}.$$

Thus, the ratio $\rho_i = Z(\lambda_{i-1})/Z(\lambda_i)$ can be estimated by sampling matchings from the distribution π_{λ_i} and computing the sample mean of f_i. Following (12.7), our estimator of $Z(\widehat{\lambda})$ will be the product of the reciprocals of these estimated ratios. Summarizing this discussion, our algorithm can be written down as follows:

ALGORITHM $\mathcal{A}$

Step 1: Compute the sequence $\lambda_1 = |E|^{-1}$ and $\lambda_i = \left(1+\frac{1}{n}\right)^{i-1}\lambda_1$ for $1 \leq i < r$, where r is the least integer such that $\left(1+\frac{1}{n}\right)^{r-1}\lambda_1 \geq \widehat{\lambda}$. Set $\lambda_0 = 0$ and $\lambda_r = \widehat{\lambda}$.

Step 2: For each value $\lambda = \lambda_1, \lambda_2, \ldots, \lambda_r$ in turn, compute an estimate X_i of the ratio ρ_i as follows:

- ***(a)*** by performing S independent simulations of the Markov chain $\mathfrak{M}_{\text{match}}(\lambda_i)$, each of length T_i, obtain an independent sample of size S from (close to) the distribution π_{λ_i};

[10]The device of performing random walk on a connected graph with acceptance probabilities of this form is well known in Monte Carlo physics under the name of the "Metropolis process" [Met53]. Clearly, it can be used to achieve any desired stationary distribution π for which the ratio $\pi(u)/\pi(v)$ for neighbors u, v can be computed easily. It is also the standard mechanism used in combinatorial optimization by simulated annealing: see Section 12.6.

(b) let X_i be the sample mean of the quantity $\left(\frac{\lambda_{i-1}}{\lambda_i}\right)^{|M|}$.

Step 3: Output the product $Y = \prod_{i=1}^{r} X_i^{-1}$.

To complete the description of the algorithm, we need to specify the sample size S in Step 2, and the number of simulation steps T_i required for each sample. Our goal is to show that, with suitable values for these quantities, Algorithm $\mathcal{A}$ is an FPRAS for $Z(\lambda)$.

The issue of the sample size S is straightforward. Using elementary statistical calculations, we can show the following:

PROPOSITION 12.3 In Algorithm $\mathcal{A}$, suppose the sample size S in Step 2 is $S = \lceil 130e\varepsilon^{-2}r\rceil$, and that the simulation length T_i is large enough that the variation distance of $\mathfrak{M}_{\text{match}}(\lambda_i)$ from its stationary distribution π_{λ_i} is at most $\varepsilon/5er$. Then the output random variable Y satisfies

$$\Pr\big((1-\varepsilon)Z(\widehat{\lambda}) \le Y \le (1+\varepsilon)Z(\widehat{\lambda})\big) \ge \tfrac{3}{4}.$$

Since r is a relatively small quantity (essentially linear in n: see (12.8)), this result means that a modest sample size at each stage suffices to ensure a good final estimate Y, provided of course that the samples come from a distribution that is close enough to π_{λ_i}.

It is in determining the number of simulation steps, T_i, required to achieve this that the meat of the analysis lies: of course, this is tantamount to investigating the mixing time of the Markov chain $\mathfrak{M}_{\text{match}}(\lambda_i)$. Our main task in this section will be to show:

PROPOSITION 12.4 The mixing time of the Markov chain $\mathfrak{M}_{\text{match}}(\lambda)$ satisfies

$$\tau_X(\varepsilon) \le 4|E|n\lambda'\big(n(\ln n + \ln\lambda') + \ln\varepsilon^{-1}\big).$$

The proof of this result will make use of the full power of the machinery introduced in Section 12.3. Note that Proposition 12.4 is a very strong statement: it says that we can sample from (close to) the complex distribution π_λ over the exponentially large space of matchings in G, by performing a Markov chain simulation of length only a low-degree polynomial in the size of G.[11]

According to Proposition 12.3, we require a variation distance of $\varepsilon/5er$, so Proposition 12.4 tells us that it suffices to take

$$T_i = \big\lceil 4|E|n\lambda_i'\big(n(\ln n + \ln\lambda_i') + \ln(5er/\varepsilon)\big)\big\rceil. \tag{12.9}$$

This concludes our specification of the Algorithm $\mathcal{A}$.

Before proceeding to prove the above statements, let us convince ourselves that together they imply that Algorithm $\mathcal{A}$ is an FPRAS for $Z(\lambda)$. First of all, Proposition 12.3 ensures that the output of Algorithm $\mathcal{A}$ satisfies the requirements of an FPRAS for Z. It remains only to verify that the running time is bounded by a polynomial in n, $\widehat{\lambda}'$, and ε^{-1}. Evidently, the running time is dominated by the number of Markov chain simulations

[11] Incidentally, we should point out that Proposition 12.4 immediately tells us that we can sample monomer-dimer configurations from the canonical distribution π_λ, in time polynomial in n and λ'. This is in itself an interesting result, and allows estimation of the expectation of many quantities associated with monomer-dimer configurations.

steps, which is $\sum_{i=1}^{r} ST_i$; since T_i increases with i, this is at most rST_r. Substituting the upper bound for r from (12.8), and values for S from Proposition 12.3 and T_r from (12.9), we see that the overall running time of Algorithm $\mathcal{A}$ is bounded by[12]

$$\mathrm{O}\big(n^4 |E| \widehat{\lambda}' (\ln n \widehat{\lambda}')^3 \varepsilon^{-2}\big),$$

which grows only polynomially with n, $\widehat{\lambda}'$ and ε^{-1}. We have therefore proved

THEOREM 12.1 Algorithm $\mathcal{A}$ is an FPRAS for the partition function of an arbitrary monomer-dimer system.

We return now to prove Proposition 12.3 and Proposition 12.4. The first of these can be dispensed with quickly. It rests on the standard observation that the sample size S required at each value $\lambda = \lambda_i$ to ensure that our final estimate is good with high probability depends on the *variances* of the random variables f_i, or more precisely on the quantities $(\mathrm{Var}\, f_i)/(\mathrm{E}\, f_i)^2$. Intuitively, if these quantities are not too large, a small sample will suffice. Since f_i takes values in the range [0, 1], it is clear that $\mathrm{Var}\, f_i \le \mathrm{E}\, f_i = \rho_i$, so that $(\mathrm{Var}\, f_i)/(\mathrm{E}\, f_i)^2 \le \rho_i^{-1}$. Now, from the definition of Z and λ_i we have for $1 \le i \le r$,

$$\rho_i^{-1} = \frac{Z(\lambda_i)}{Z(\lambda_{i-1})} = \frac{\sum_k m_k \lambda_i^k}{\sum_k m_k \lambda_{i-1}^k} \le \left(\frac{\lambda_i}{\lambda_{i-1}}\right)^n \le \left(1 + \frac{1}{n}\right)^n \le \mathrm{e}. \tag{12.10}$$

Also, it is easy to see (using the fact that matchings are subsets of E) that $Z\big(|E|^{-1}\big) \le \mathrm{e}$, so (12.10) holds for $i = 0$ also. Thus, we have $(\mathrm{Var}\, f_i)/(\mathrm{E}\, f_i)^2 \le \mathrm{e}$ for all i. This explains our choice of values for the λ_i.

Armed with this bound on the variances of the f_i, one can prove Proposition 12.3 by a routine statistical calculation. The details are unedifying and are deferred to the Appendix.

We turn now to the more challenging question of proving Proposition 12.4. Our strategy will be to carefully choose a collection of canonical paths $\Gamma = \{\gamma_{XY} : X, Y \in \Omega\}$ in the Markov chain $\mathfrak{M}_{\mathrm{match}}(\lambda)$ for which the "bottleneck" measure $\bar{\rho}(\Gamma)$ of Section 12.3 is small. We can then appeal to Proposition 12.1 to bound the mixing time. Specifically, we shall show that our paths satisfy

$$\bar{\rho}(\Gamma) \le 4|E|n\lambda'. \tag{12.11}$$

Since the number of matchings in G is certainly bounded above by $(2n)!$, the stationary probability $\pi_\lambda(X)$ of any matching X is bounded below by $\pi_\lambda(X) \ge 1/(2n)!\lambda'^n$. Using (12.11) and the fact that $\ln n! \le n \ln n$, the bound on the mixing time in Proposition 12.4 can now be read off Proposition 12.1.

It remains for us to find a set of canonical paths Γ satisfying (12.11). For a pair of matchings X, Y in G, we define the canonical path γ_{XY} as follows. Consider the symmetric difference $X \oplus Y$. A moment's reflection should convince the reader that this consists of a disjoint collection of paths in G (some of which may be closed cycles),

[12] In deriving the O-expression, we have assumed w.l.o.g. that $T_r = \mathrm{O}\big(|E|n^2\widehat{\lambda}' \ln n\widehat{\lambda}'\big)$. This follows from (12.9) with the additional assumption that $\ln \varepsilon^{-1} = \mathrm{O}(n \ln n)$. This latter assumption is justified since the problem can always be solved exactly by exhaustive enumeration in time $\mathrm{O}(n(2n)!)$, which is $\mathrm{O}(\varepsilon^{-2})$ if $\ln \varepsilon^{-1}$ exceeds the above bound.

each of which has edges that belong alternately to X and to Y. Now suppose that we have fixed some arbitrary ordering on all simple paths in G, and designated in each of them a so-called "start vertex," which is arbitrary if the path is a closed cycle but must be an endpoint otherwise. This ordering induces a unique ordering $P_1, P_2, \ldots, P_m$ on the paths appearing in $X \oplus Y$. The canonical path from X to Y involves "unwinding" each of the P_i in turn as follows. There are two cases to consider:

(i) *P_i is not a cycle.* Let P_i consist of the sequence $(v_0, v_1, \ldots, v_l)$ of vertices, with v_0 the start vertex. If $(v_0, v_1) \in Y$, perform a sequence of type 0 transitions replacing (v_{2j+1}, v_{2j+2}) by (v_{2j}, v_{2j+1}) for $j = 0, 1, \ldots,$ and finish with a single type 2 transition if l is odd. If on the other hand $(v_0, v_1) \in X$, begin with a type 1 transition removing (v_0, v_1) and proceed as before for the reduced path $(v_1, \ldots, v_l)$.

(ii) *P_i is a cycle.* Let P_i consist of the sequence $(v_0, v_1, \ldots, v_{2l+1})$ of vertices, where $l \geq 1$, v_0 is the start vertex, and $(v_{2j}, v_{2j+1}) \in X$ for $0 \leq j \leq l$, the remaining edges belonging to Y. Then the unwinding begins with a type 1 transition to remove (v_0, v_1). We are left with an open path O with endpoints v_0, v_1, one of which must be the start vertex of O. Suppose v_k, $k \in \{0, 1\}$, is *not* the start vertex. Then we unwind O as in (i) above but treating v_k as the start vertex. This trick serves to distinguish paths from cycles, as will prove convenient shortly.

This concludes our definition of the family of canonical paths Γ. Figure 12.1 will help the reader picture a typical transition t on a canonical path from X to Y. The path P_i (which happens to be a cycle) is the one currently being unwound; the paths $P_1, \ldots, P_{i-1}$ to the left have already been processed, while the ones $P_{i+1}, \ldots, P_m$ are yet to be dealt with.

We now proceed to bound the "bottleneck" measure $\bar{\rho}(\Gamma)$ for these paths, using the injective mapping technology introduced in Section 12.3. Let t be an arbitrary edge in the Markov chain, i.e., a transition from M to $M' \neq M$, and let $\mathrm{cp}(t) = \{(X, Y) : \gamma_{XY} \ni t\}$ denote the set of all canonical paths that use t. (We use the notation t in place of e here to avoid confusion with edges of G.) Just as in Section 12.3, we shall obtain a bound on the total weight of all paths that pass through t by defining an injective mapping $\eta_t : \mathrm{cp}(t) \to \Omega$. By analogy with the hypercube example in Section 12.3, what we would like to do is to set $\eta_t(X, Y) = X \oplus Y \oplus (M \cup M')$; the intuition for this is that $\eta_t(X, Y)$ should agree with X on paths that have already been unwound, and with Y on paths that have not yet been unwound (just as $\eta_e(x, y)$ agreed with x on positions $1, \ldots, i-1$ and with y on positions $i+1, \ldots, n-1$). However, there is a minor complication concerning the path that we are currently processing: in order to ensure that $\eta_t(X, Y)$ is indeed a matching, we may — as we shall see — have to remove from it the edge of X adjacent to the start vertex of the path currently being unwound: we shall call this edge e_{XYt}. This leads us to the following definition of the mapping η_t:

$$\eta_t(X, Y) = \begin{cases} X \oplus Y \oplus (M \cup M') - e_{XYt}, & \text{if } t \text{ is type 0 and the} \\ & \quad \text{current path is a cycle;} \\ X \oplus Y \oplus (M \cup M'), & \text{otherwise.} \end{cases}$$

Figure 12.2 illustrates the encoding $\eta_t(X, Y)$ that would result from the transition t on the canonical path sketched in Figure 12.1.

Let us check that $\eta_t(X, Y)$ is always a matching. To see this, consider the set of edges $A = X \oplus Y \oplus (M \cup M')$, and suppose that some vertex, u say, has degree two in A. (Since $A \subseteq X \cup Y$, no vertex degree can exceed two.) Then A contains edges $\{u, v_1\}, \{u, v_2\}$ for

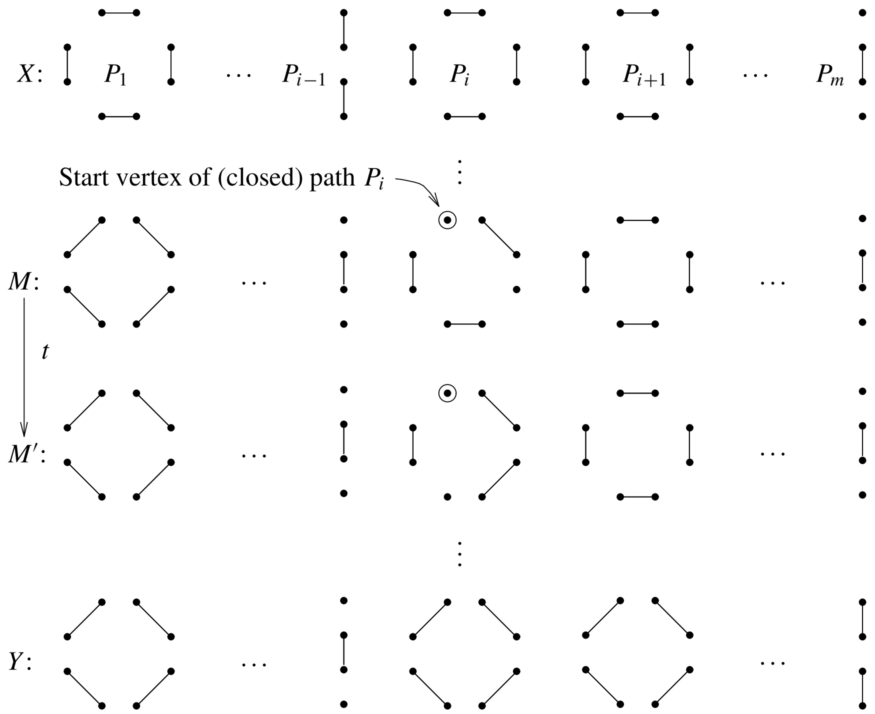

FIGURE 12.1

A transition t in the canonical path from X to Y.

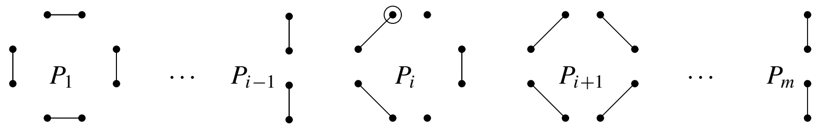

FIGURE 12.2

The corresponding encoding $\eta_t(X, Y)$.

distinct vertices v_1, v_2, and since $A \subseteq X \cup Y$, one of these edges must belong to X and the other to Y. Hence, both edges belong to $X \oplus Y$, which means that neither can belong to $M \cup M'$. Following the form of $M \cup M'$ along the canonical path, however, it is clear that there can be at most one such vertex u; moreover, this happens precisely when the current path is a cycle, u is its start vertex, and t is type 0. Our definition of η_t removes one of the edges adjacent to u in this case, so all vertices in $\eta_t(X, Y)$ have degree at most one, i.e., $\eta_t(X, Y)$ is indeed a matching.

We now have to check that η_t is injective. It is immediate from the definition of η_t

that the symmetric difference $X \oplus Y$ can be recovered from $\eta_t(X, Y)$ using the relation

$$X \oplus Y = \begin{cases} \eta_t(X,Y) \oplus (M \cup M') + e_{XYt}, & \text{if } t \text{ is type 0 and the current path is a cycle;} \\ \eta_t(X,Y) \oplus (M \cup M'), & \text{otherwise.} \end{cases}$$

Note that, once we have formed the set $\eta_t(X, Y) \oplus (M \cup M')$, it will be apparent whether the current path is a cycle from the sense of unwinding. (Note that e_{XYt} is the unique edge that forms a cycle when added to the path.) Given $X \oplus Y$, we can at once infer the sequence of paths $P_1, P_2, \dots, P_m$ that have to be unwound along the canonical path from X to Y, and the transition t tells us which of these, P_i say, is the path currently being unwound. The partition of $X \oplus Y$ into X and Y is now straightforward: X has the same parity as $\eta_t(X, Y)$ on paths $P_1, \dots, P_{i-1}$, and the same parity as M on paths $P_{i+1}, \dots, P_m$. Finally, the reconstruction of X and Y is completed by noting that $X \cap Y = M - (X \oplus Y)$, which is immediate from the definition of the paths. Hence, X and Y can be uniquely recovered from $\eta_t(X, Y)$, so η_t is injective.

We are almost done. However, the fact that η_t is injective is not sufficient in this case because, in contrast to the hypercube example, the stationary distribution π_λ is highly non-uniform. What we require in addition is that η_t be "weight-preserving," in the sense that $Q(t)\pi_\lambda(\eta_t(X, Y)) \approx \pi_\lambda(X)\pi_\lambda(Y)$. More precisely, we will show in a moment that

$$\pi_\lambda(X)\pi_\lambda(Y) \le 2|E|\lambda'^2 Q(t)\pi_\lambda(\eta_t(X,Y)). \tag{12.12}$$

First, let us see why we need a bound of this form in order to estimate $\bar{\rho}$. We have

$$\begin{aligned} \frac{1}{Q(t)} \sum_{\gamma_{XY} \ni t} \pi_\lambda(X)\pi_\lambda(Y)|\gamma_{XY}| &\le 2|E|\lambda'^2 \sum_{\gamma_{XY} \ni t} \pi_\lambda(\eta_t(X,Y))\,|\gamma_{XY}| \\ &\le 4|E|n\lambda'^2 \sum_{\gamma_{XY} \ni t} \pi_\lambda(\eta_t(X,Y)) \\ &\le 4|E|n\lambda'^2, \end{aligned} \tag{12.13}$$

where the second inequality follows from the fact that the length of any canonical path is bounded by $2n$, and the last inequality from the facts that η_t is injective and π_λ is a probability distribution.

It remains for us to prove inequality (12.12). Before we do so, it is helpful to notice that $Q(t) = (2|E|)^{-1} \min\{\pi_\lambda(M), \pi_\lambda(M')\}$, as may easily be verified from the definition of $\mathfrak{M}_{\text{match}}(\lambda)$. We now distinguish four cases:

(i) *t is a type 1 transition.* Suppose $M' = M - e$. Then $\eta_t(X, Y) = X \oplus Y \oplus M$, so, viewed as multisets, $M \cup \eta_t(X, Y)$ and $X \cup Y$ are identical. Hence, we have

$$\begin{aligned} \pi_\lambda(X)\pi_\lambda(Y) &= \pi_\lambda(M)\pi_\lambda(\eta_t(X,Y)) \\ &= \frac{2|E|Q(t)}{\min\{\pi_\lambda(M), \pi_\lambda(M')\}} \times \pi_\lambda(M)\pi_\lambda(\eta_t(X,Y)) \\ &= 2|E|Q(t)\max\{1, \pi_\lambda(M)/\pi_\lambda(M')\}\pi_\lambda(M)\pi_\lambda(\eta_t(X,Y)) \\ &\le 2|E|\lambda' Q(t)\pi_\lambda(\eta_t(X,Y)), \end{aligned}$$

from which (12.12) follows.

(ii) *t is a type 2 transition*. This is handled by a symmetrical argument to (i) above, with the roles of M and M' interchanged.

(iii) *t is a type 0 transition and the current path is a cycle*. Suppose $M' = M + e - e'$, and consider the multiset $M \cup \eta_t(X,Y)$. Then $\eta_t(X,Y) = X \oplus Y \oplus (M+e) - e_{XYt}$, so the multiset $M \cup \eta_t(X,Y)$ differs from $X \cup Y$ only in that e and e_{XYt} are missing from it. Thus, we have

$$\begin{aligned}\pi_\lambda(X)\pi_\lambda(Y) &\le \lambda'^2 \pi_\lambda(M)\pi_\lambda(\eta_t(X,Y))\\ &= 2|E|\lambda'^2 Q(t)\pi_\lambda(\eta_t(X,Y)),\end{aligned}$$

since in this case $\pi_\lambda(M) = \pi_\lambda(M')$, and so $Q(t) = (2|E|)^{-1}\pi_\lambda(M)$. Therefore, (12.12) is again satisfied.

(iv) *t is a type 0 transition and the current path is not a cycle*. This is identical with (iii) above, except that the edge e_{XYt} does not appear in the analysis. Accordingly, the bound is

$$\pi_\lambda(X)\pi_\lambda(Y) \le 2|E|\lambda' Q(t)\pi_\lambda(\eta_t(X,Y)).$$

This concludes our proof of (12.12). We may now deduce from (12.13), that $\bar{\rho}(\Gamma) \le 4|E|n\lambda'^2$. However, one additional observation will allow us to improve the bound to $\bar{\rho}(\Gamma) \le 4|E|n\lambda'$, which is what we claimed in (12.11). Looking at the above case analysis we see that, in all cases except case (iii), (12.12), and hence (12.13), actually hold with λ'^2 replaced by λ'. But in case (iii) we can argue that $\eta_t(X,Y)$ must have such a restricted form that $\sum_{\gamma_{XY}\ni t}\pi_\lambda(\eta_t(X,Y))$ is bounded above by λ'^{-1}. Using this fact in the final inequality in (12.13), we get the improved upper bound of $4|E|n\lambda'$ in this case, and hence in all cases. This will complete our verification of the bound (12.11) on $\bar{\rho}(\Gamma)$.

To justify the above claim, note that $\eta_t(X,Y)$ has at least two unmatched vertices, namely the start vertex of the current cycle and the vertex that is common to both e and e'. Moreover, in $\eta_t(X,Y) \oplus M$ these vertices are linked by an alternating path that starts and ends with an edge of M. So we may associate with each matching $\eta_t(X,Y)$ another matching, say $\eta_t'(X,Y)$, obtained by augmenting $\eta_t(X,Y)$ along this path. But this operation is uniquely reversible, so all matchings $\eta_t'(X,Y)$ created in this way are distinct. Moreover, $\pi_\lambda(\eta_t(X,Y)) = \lambda\pi_\lambda(\eta_t(X,Y))$. Hence we have $\sum \pi_\lambda(\eta_t(X,Y)) = \lambda^{-1}\sum \pi_\lambda(\eta_t'(X,Y)) \le \lambda^{-1}$, so $\sum \pi_\lambda(\eta_t(X,Y)) \le \lambda'^{-1}$ as claimed.

MORE APPLICATIONS 12.5

In this section we review some further applications of the techniques described in Section 12.3 to problems in combinatorial enumeration and integration. In each case, as with the monomer-dimer problem of Section 12.4, the Markov chain Monte Carlo method provides the only known basis for an efficient algorithm in the FPRAS sense.

12.5.1 THE PERMANENT

Historically, the first major application of the methods of Section 12.3 was to the approximation of the permanent function. The *permanent* of an $n \times n$ integer matrix $A = (a_{ij} : 0 \leq i, j \leq n-1)$ is defined by

$$\operatorname{per} A = \sum_{\pi} \prod_{i=0}^{n-1} a_{i,\pi(i)},$$

where the sum is over all permutations π of $[n] = \{0, \ldots, n-1\}$. For convenience, we take A to be a 0,1-matrix, in which case the permanent of A has a simple combinatorial interpretation: namely, $\operatorname{per} A$ is equal to the number of perfect matchings (1-factors) in the bipartite graph $G = (V_1, V_2, E)$, where $V_1 = V_2 = [n]$, and $(i, j) \in E$ iff $a_{ij} = 1$. Valiant [Val79a] demonstrated that evaluating the permanent of a 0,1-matrix is complete for the class #*P*; thus, just as in the case of the monomer-dimer partition function, we cannot expect to find an algorithm that solves the problem exactly in polynomial time.[13] Interest has therefore centered on finding computationally feasible approximation algorithms.

It turns out that the Markov chain Monte Carlo method can be used to construct such an algorithm (in the FPRAS sense) for almost all instances of this problem. To state the result precisely, we will use the perfect matching formulation. Let $G = (V_1, V_2, E)$ be a bipartite graph with $|V_1| = |V_2| = n$. A special role will be played in the result by the number of *near-perfect* matchings in G, i.e., matchings with exactly two unmatched vertices. Following the notation of the previous section, let us write $m_k = m_k(G)$ for the number of k-matchings in G. Then the number of perfect matchings is m_n, and the number of near-perfect matchings is m_{n-1}. Jerrum and Sinclair [JS89] showed that there exists a randomized approximation scheme for the number of perfect matchings m_n whose running time is polynomial in n, ε^{-1} and the ratio m_{n-1}/m_n.

Note that this algorithm is not in general an FPRAS, since there exist $(n+n)$-vertex graphs G for which the ratio m_{n-1}/m_n is exponential in n. However, it turns out that these examples are wildly atypical in the sense that the probability that a randomly selected G on $n+n$ vertices violates the inequality $m_{n-1}/m_n \leq 4n$ tends to 0 as $n \to \infty$.[14] Thus, the above algorithm constitutes an FPRAS for almost all graphs; moreover, the condition that the ratio m_{n-1}/m_n be bounded by a specified polynomial in n can be tested for an arbitrary graph in polynomial time [JS89]. It is also known [Bro86] that *every* sufficiently dense graph (specifically, those in which every vertex has degree at least $\frac{1}{2}n$) satisfies $m_{n-1}/m_n = \mathrm{O}(n^2)$. Moreover, it has recently been shown by Kenyon, Randall, and Sinclair [KRS96] that the ratio m_{n-1}/m_n is guaranteed to be small for a wide class of homogeneous graphs G, including the important case of geometric lattice graphs in any number of dimensions. We should also point out that, although the above description has been couched in terms of matchings in bipartite graphs because of the connection with the permanent, everything extends to general $2n$-vertex graphs.

[13] In contrast, as is well known, the *determinant* of an $n \times n$ matrix can be evaluated in $\mathrm{O}(n^3)$ arithmetic operations using Gaussian elimination.

[14] For more refined results along these lines, see Frieze [Friez89] or Motwani [Mot89].

It was Broder [Bro86, Mih89a] who first proposed a Markov chain Monte Carlo approach to approximating the permanent via Markov chain simulation. His idea was to sample perfect matchings in a bipartite graph G almost u.a.r. by simulating a Markov chain whose states are perfect and near-perfect matchings in G; then, using a reduction similar in spirit to the one described in Section 12.2 for the Knapsack problem, the number of perfect matchings could be counted. Broder's Markov chain was first proved to be rapidly mixing (under the above condition on G) by Jerrum and Sinclair [JS89], using a canonical paths argument as in Section 12.3.

An alternative, more natural approximation algorithm for the permanent follows quite painlessly from our results about the monomer-dimer problem derived in the previous section. Note that m_n is precisely the leading coefficient of the partition function $Z_G(\lambda)$ of the monomer-dimer system associated with G (see (12.5)). In the previous section, we saw how to sample matchings in G from the distribution

$$\pi_\lambda(M) = \frac{\lambda^{|M|}}{Z_G(\lambda)} = \frac{\lambda^{|M|}}{\sum_{k=0}^{n} m_k \lambda^k} \tag{12.14}$$

for any desired $\lambda > 0$, in time polynomial in n and $\lambda' = \max\{\lambda, 1\}$, by Monte Carlo simulation of the Markov chain $\mathfrak{M}_{\mathrm{match}}(\lambda)$. We also saw how this fact can be used to compute $Z_G(\lambda)$ to good accuracy in time polynomial in n and λ'. Suppose then that we have computed a good estimate $\widehat{Z}_G(\lambda)$ of $Z_G(\lambda)$. Then we can get a good estimator for m_n by sampling matchings from the distribution π_λ and computing the proportion, X, of the sample that are perfect matchings; since $\mathrm{E}\,X = m_n\lambda^n / Z_G(\lambda)$, our estimator is $Y = X\lambda^{-n}\widehat{Z}_G(\lambda)$.

The sample size required to ensure a good estimate depends on the variance of a single sample, or more precisely on the quantity $(\mathrm{E}\,X)^{-1}$. Clearly, by making λ large enough, we can make this quantity, and hence the sample size, small: this corresponds to placing very large weight on the perfect matchings, so that their proportion can be estimated well by random sampling. How large does λ have to be? This analysis is eased by the beautiful fact that the sequence $m_0, m_1, \ldots, m_n$ is *log-concave*, i.e., $m_{k-1}m_{k+1} \le m_k^2$ for $k = 1, 2, \ldots, n-1$. (This is well known [HL72]; a direct combinatorial proof may be found in [JS89].) As a consequence, it follows that $m_{k-1}/m_k \le m_{n-1}/m_n$ for all k, and hence that $m_k/m_n \le (m_{n-1}/m_n)^{n-k}$. This means that, if we take $\lambda \ge m_{n-1}/m_n$, we get

$$\mathrm{E}\,X = \frac{m_n\lambda^n}{Z_G(\lambda)} = \frac{m_n\lambda^n}{\sum_{k=0}^{n} m_k\lambda^k} \ge \frac{1}{n+1}, \tag{12.15}$$

which implies that the sample size required grows only linearly with n. Thus, it is enough to take λ about as large as the ratio m_{n-1}/m_n. Since the time required to generate a single sample grows linearly with λ (see Proposition 12.4), the running time of the overall algorithm is polynomial in n, ε^{-1} and the ratio m_{n-1}/m_n, as claimed.

OPEN PROBLEM 12.2 Is there an FPRAS for the permanent of a general 0,1 matrix? Note that this problem is not phrased as a question about the mixing time of a specific Markov chain, and certainly the chain $\mathfrak{M}_{\mathrm{match}}(\lambda)$ described here is not directly applicable: as we have seen, it seems to be useful only when the ratio m_{n-1}/m_n for the associated bipartite graph is polynomially bounded. However, the Markov chain Monte Carlo method seems to offer the best hope for a positive resolution of this question. Essentially, the issue is whether the Markov chain $\mathfrak{M}_{\mathrm{match}}(\lambda)$ can be suitably adapted to provide a general solution, or perhaps used as a "black box" following some ingenious preprocessing of the input matrix. (This latter idea has been used in a weaker way by Jerrum and Vazirani [JV92] to obtain a randomized approximation scheme for the general 0,1 permanent whose running time, while still not polynomial, is asymptotically significantly faster than that of more naïve methods.)

We conclude our discussion of the permanent by mentioning some extensions. First of all, it is not hard to see, again using the log-concavity property, that the above technique can be extended to approximate the entire sequence (m_k), or equivalently all the coefficients of the monomer-dimer partition function [JS89]. The running time per coefficient is no worse than for m_n. Secondly, many other approximate enumeration (and sampling) problems can be reduced to enumeration of perfect matchings; examples include counting Hamiltonian cycles in dense or random graphs (Dyer, Frieze, and Jerrum [DFJ94], Frieze and Suen [FS92]), counting graphs with given degree sequence (Jerrum and Sinclair [JS90a], Jerrum, McKay, and Sinclair [JMS92]), and counting Eulerian orientations of an undirected graph (Mihail and Winkler [MW91]).

12.5.2 VOLUME OF CONVEX BODIES

A problem that has attracted much attention in the context of the Markov chain Monte Carlo method is that of estimating the volume of a convex body in high-dimensional space. Computing the volume of a polytope in $n = 3$ dimensions is not a computationally demanding task, but the effort required rises dramatically as the number n of dimensions increases. This empirical observation is supported by a result of Dyer and Frieze [DF88] to the effect that evaluating the volume of a polytope *exactly* is #*P*-hard.

In contrast, by applying the Markov chain Monte Carlo method, Dyer, Frieze, and Kannan [DFK91] were able to construct an FPRAS for the volume of a convex body in Euclidean space of *arbitrary* dimension. The convex body K in question is presented to the algorithm using a very general mechanism called a *membership oracle*: given a point x, the membership oracle simply reveals whether or not $x \in K$. Other ways of specifying the body K — for example as a list of vertices or $(n-1)$-dimensional facets — can be recast in the oracle formulation. The algorithm must also be provided with a guarantee in the form of two balls, one contained in K and of non-zero radius, and the other containing K. This seemingly technical condition is essential, for without such a guarantee the task is hopeless.

There are several difficult technical points in the construction and analysis of the volume approximation algorithm of Dyer et al., but, at a high enough level of abstraction, the method is quite simple to describe. The idea is to divide space into n-dimensional (hyper)cubes of side δ, and to perform a random walk on the cubes that lie within the body K. Suppose the random walk is at cube C at time t. A cube C' that is orthogonally adjacent to C is selected uniformly at random; if $C' \in K$ then the walk moves to C', otherwise it stays at C. It is easy to check that the walk (or something close to it) is ergodic, and that the stationary distribution is uniform on cubes in K. The cube size δ is selected so as to provide an adequate approximation to K, while permitting the random walk to "explore" the state space within a reasonable time. Rapid mixing (i.e., in time polynomial in n) is proved via the conductance argument of Section 12.3, by considering the geometry of the state space of the random walk and applying classical isoperimetric inequalities.

Once the sampling problem has been solved, the volume of K can be computed by the technique of Section 12.2. Let $B_0 \subset B_1 \subset \cdots \subset B_m$ be a sequence of concentric balls chosen so that $B_0 \subseteq K \subseteq B_m$ and the volume of B_i exceeds that of B_{i-1} by (say) a factor of 2. Consider the sequence of convex bodies

$$B_0 = K \cap B_0 \subseteq K \cap B_1 \subseteq \cdots \subseteq K \cap B_m = K. \tag{12.16}$$

The volume of the first is known, while the ratios of volumes of successive bodies can be estimated by Monte Carlo sampling using simulation of the random walk described earlier. Random sampling is effective in this context because the volumes of adjacent bodies in sequence (12.16) differ by a factor of at most 2. By multiplying the estimates for the various ratios, the volume of the final body $K \cap B_m = K$ may be computed to any desired degree of approximation.

Although there are many situations in which a source of random bits seems to aid computation, the current example is particularly interesting in that randomness is of *provable* value. It has been shown by Elekes [Elek86] that a deterministic algorithm that is restricted to a subexponential number of oracle calls is unable to obtain a good (say, to within a ratio of 2) approximation to the volume of a convex body.

The close relationship of volume estimation to (approximate) multi-dimensional integration has provided strong practical impetus to research in this area. Since the appearance of the original paper of Dyer et al., much effort has gone into extending the algorithm to a wider class of problems, and into reducing its running time, which, though polynomial in n, is still rather high in practical terms. Applegate and Kannan [AK91] have generalized the algorithm to the integration of log-concave functions over convex regions in arbitrary dimensional space, while Dyer and Frieze [DF91], and Lovász and Simonovits [LS93] have devised many improvements that have successively reduced the time complexity of the algorithm. The success of the latter pursuit may be judged from the dramatic improvement in the dependence of the time-complexity on the dimension n: from $\mathrm{O}(n^{27})$ for the original algorithm of Dyer et al., to $\widetilde{\mathrm{O}}(n^7)$ as claimed recently by Kannan, Lovász, and Simonovits [KLS94a].[15] Some of the ideas that have led to these improvements are sketched below; for more detail the reader is referred to Kannan's survey article [Kan94], and the references therein.

One source of inefficiency in the early approach was that the random walk in K could, in principle, get stuck for long periods near "sharp corners" of K. Indeed, in the first algorithm, Dyer et al. found it necessary to "round off" the corners of K before simulating the random walk. Applegate and Kannan obtained a substantial improvement in efficiency by providing the random walk with a fuzzy boundary. Rather than estimating the volume of K directly, their version of the algorithm estimates the integral of a function F that takes the value 1 on K, and decays to 0 gracefully outside K. The random walk on cubes is modified so that its stationary distribution is approximately proportional

[15] The $\widetilde{\mathrm{O}}(\,)$ notation hides not merely constants, but also arbitrary powers of $\log n$. Kannan et al.'s algorithm requires just $\widetilde{\mathrm{O}}(n^5)$ oracle calls, but the cost of effecting a single step of their random walk may be as high as $\mathrm{O}(n^2)$.

to the function F. As we saw in Section 12.4, in the context of the matching Markov chain $\mathfrak{M}_{\text{match}}(\lambda)$, this end is easily achieved by using a Metropolis-style rule to determine transition probabilities. Provided F decays sufficiently rapidly outside K, the integral of F over the whole of $\mathbb{R}^n$ will be a close approximation to the volume of K.

Another strategy that has been employed in the pursuit of efficiency is to attempt to reduce the length m of sequence (12.16), which amounts to arranging for the extreme balls B_0 and B_m to be as close as possible in volume. In the earlier papers, the body K is subjected to a linear transformation that allows the transformed convex body to be sandwiched between balls whose radii differ by a factor $O(n^{3/2})$. By contenting themselves with a less demanding notion of "approximate sandwiching," Kannan, Lovász, and Simonovits [KLS94b] have recently reduced this factor to $O(\sqrt{n})$, which is best possible. Observe that this improvement in the sandwiching ratio reduces the length of sequence (12.16) roughly by a factor n.

Finally, much thought has gone into potentially more efficient random walks for sampling from within K. This is an attractive line of inquiry, as the original "cubes walk," which only ever makes short steps, intuitively seems rather inefficient. Lovász and Simonovits [LS93] consider instead a "ball walk" with continuous state space, which operates as follows. Suppose $x \in K$ is the position of the walk at time t, and denote by $B(x, \delta)$ the ball with centre x and radius δ. The probability density of the position of the walk at time $t + 1$, conditional on its position at time t being x, is uniform over the region $K \cap B(x, \delta)$, and zero outside. The parameter δ is chosen to exploit the trade-off discussed briefly in the context of the cubes walk. The conductance argument can be extended to the continuous case without essential change. The ball walk saves a factor n in the number of oracle calls; unfortunately, as the moves of the random walk are now more complex than before, there is no saving in net time complexity (i.e., excluding oracle calls).

An interesting problem related to volume estimation is that of approximately counting contingency tables: given $m + n$ positive integers $r_1, \ldots, r_m$ and $c_1, \ldots, c_n$, compute an approximation to the number of $m \times n$ non-negative integer matrices with row-sums $r_1, \ldots, r_m$ and column-sums $c_1, \ldots, c_n$. This problem arises in the interpretation of the results of certain kinds of statistical experiment; see, for example, Diaconis and Efron [DE85].

It is easy to see that the contingency tables with given row- and column-sums are in 1-1 correspondence with integer lattice points contained in an appropriately defined polytope of dimension $nm - n - m$. We might hope that a sufficiently uniform distribution on lattice points could be obtained by sampling from the (continuous) convex polytope and rounding to a nearby lattice point. Dyer, Kannan, and Mount [DKM95] show that this can be done, provided that the row- and column-sums are sufficiently large; specifically, that each sum is at least $(n + m)nm$. The case of small row- and column-sums remains open. There is no hope of an FPRAS for unrestricted 3-dimensional contingency tables (unless $NP = RP$), as Irving and Jerrum [IJ94] have shown that deciding feasibility (i.e, whether there is at least one realization of the contingency table) is NP-complete in 3-dimensions, even when the row- column- and file-sums are all either 0 or 1.

OPEN PROBLEM 12.3 An elegant direct approach to sampling contingency tables has been proposed by Diaconis. Consider the Markov chain $\mathfrak{M}_{\text{CT}}$, whose state space is the set of all matrices with specified row and column sums, and whose transition probabilities are defined as follows. Let the current state (matrix) be $A = (a_{ij})$. Select a pair of rows (i, i') with $i \neq i'$, and a pair of columns (j, j') with $j \neq j'$, both u.a.r. Form a new matrix A' from A by incrementing by one the array elements $a_{ij}, a_{i'j'}$, and decrementing by one the elements $a_{ij'}, a_{i'j}$. Note that A' has the same row- and column-sums as A. If A' is non-negative then we accept it as the next state; otherwise the chain remains at state A. It is easy to verify that $\mathfrak{M}_{\mathrm{CT}}$ is ergodic and reversible with uniform stationary distribution. Moreover, it appears to work well in practice as a uniform sampling procedure for contingency tables. However, its mixing time is not known to be bounded by any polynomial in the size of the input. (For obvious reasons, we must assume that the row- and column-sums are expressed in unary notation when defining the input size.)

12.5.3 STATISTICAL PHYSICS

We have already seen, in Section 12.4, a detailed example of the use of the Markov chain Monte Carlo method in statistical physics. It was in fact in this area that the first computational use of the technique was made, and today Markov chain simulations related to physical systems account for vast quantities of CPU time on high performance machines. These methods, while often ingenious, are hardly ever statistically rigorous, so the numerical results obtained from them have to be treated with some degree of caution. One of the most exciting applications of the analytical techniques presented here is the potential they open up for the rigorous quantification of these methods. In this subsection, we sketch the progress that has been made in this direction to date.

The most intensively studied model in statistical physics is the *Ising model*, introduced in the 1920s by Lenz and Ising as a means of understanding the phenomenon of ferromagnetism. An instance of the Ising model is specified by giving a set of *n sites*, a set of *interaction energies* V_{ij} for each unordered pair of sites i, j, a *magnetic field intensity* B, and an *inverse temperature* β. A *configuration* of the system defined by these parameters is one of the 2^n possible assignments σ of ± 1 *spins* to each site. The energy of a configuration σ is given by the *Hamiltonian* $H(\sigma)$, defined by

$$H(\sigma) = -\sum_{\{i,j\}} V_{ij}\sigma_i\sigma_j - B\sum_k \sigma_k.$$

The more interesting part of $H(\sigma)$ is the first sum, which consists of a contribution from each pair of sites. The contribution from the pair i, j is dependent on the interaction energy V_{ij}, and whether the spins at i and j are equal or unequal. The second sum has a contribution from each site i whose sign depends on the sign of the spin at i. In physically realistic applications, the sites are arranged in a regular fashion in 2- or 3-dimensional space, and V_{ij} is non-zero only for "adjacent" sites. From a computational point of view, this special structure seems difficult to exploit. For more detail on this and other models in statistical physics, viewed from a computational perspective, consult the survey by Welsh [Wel90].

A central problem in the theory is evaluating the *partition function* $Z = \sum_\sigma \exp(-\beta H(\sigma))$, where the sum is over all possible configurations σ. This is analogous to the monomer-dimer partition function in Section 12.4, which is also a weighted sum over configurations. The significance of Z is that it is the normalizing factor in the *Gibbs distribution*, which assigns probability $\exp(-\beta H(\sigma))/Z$ to each state (configuration) σ in the steady state. Other problems relate to the evaluation of the expectation of certain random variables of σ, when σ is sampled according to the Gibbs distribution: the *mean magnetic moment* and *mean energy* are two such.

When the interaction energies are unconstrained (this corresponds to a so-called *spin glass*) the partition function is hard even to approximate [JS93], so we restrict attention to the important *ferromagnetic* case, where $V_{ij} \geq 0$ for all pairs $\{i, j\}$ of sites. Even here, *exact* computation of the partition function is #*P*-complete [JS93], so it is again natural to ask whether an FPRAS exists. Jerrum and Sinclair [JS93] answered this question in the affirmative, and in addition presented an FPRAS for the mean magnetic moment and mean energy. Applying the Markov chain Monte Carlo method to the Ising model required an additional twist, as the "natural" random walk on configurations, in which two configurations are adjacent if they differ in just one spin, is *not* rapidly mixing.[16] The twist is to simulate an apparently unrelated Markov chain on a different set of configurations — based on edges rather than vertices — which happens to have essentially the same partition function as the Ising model proper. Using the canonical paths argument, it can be shown that the new, edge-based Markov chain is rapidly mixing. The twist just described is one factor that makes this application one of the most intricate so far devised.

In addition to the Ising model and monomer-dimer systems, other models in statistical physics that have been solved in the FPRAS sense are the *six-point ice model* [MW91] and the *self-avoiding walk model* for linear polymers [BS85, RS94]. The former problem is again connected with matchings in a graph, but rather remotely, and a fair amount of work is required to establish and verify the connection [MW91]. The latter makes use of a Markov chain that is much simpler in structure to those considered here [BS85], and whose analysis requires a far less sophisticated application of the canonical paths approach. The analysis in fact relies on a famous conjecture regarding the behavior of self-avoiding walks: the resulting algorithm is somewhat novel in that it either outputs reliable numerical answers, or produces a counterexample to the conjecture [RS94].

12.5.4 MATROID BASES: AN OPEN PROBLEM

A particularly appealing open problem in this area, and one that would be very rich in terms of consequences, is to determine useful bounds on the mixing time of the *basis-exchange* Markov chain for a general matroid. (A matroid is an algebraic structure that provides an abstract treatment of the concept of linear independence.) The states of this Markov chain are the bases (maximum independent sets) of a given matroid, and a transition is available from base B to base B' if the symmetric difference of B and B' consists of precisely two elements of the ground set. All transition probabilities are equal, so the chain is ergodic and reversible with uniform stationary distribution.

[16] A more elaborate random walk on spin configurations proposed by Swendsen and Wang [SW87] *may* be rapidly mixing, but nothing rigorous is known.

A concrete example is provided by the *graphic matroid* associated with an undirected graph G. In this case, the bases are spanning trees of G, and a transition from a given tree T is effected by adding a single edge (selected u.a.r.) to T, thus creating a cycle, and then breaking the cycle by deleting one of its edges (selected u.a.r.). The basis-exchange Markov chain is known to be rapidly mixing for graphic matroids, and, somewhat more generally, for matroids satisfying a certain "balance condition" (see Feder and Mihail [FM92]). A proof of rapid mixing in the general case would imply the existence of an FPRAS for a number of important problems in combinatorial enumeration, all of which are #P-complete, including counting connected spanning subgraphs of a graph (network reliability), forests of given size in a graph, and independent subsets of vectors in a set of n-vectors over GF(2).

12.6 THE METROPOLIS ALGORITHM AND SIMULATED ANNEALING

We conclude this survey with a rather different application of the Markov chain Monte Carlo method. Like the applications we have discussed so far, Markov chain simulation will again be used to sample from a large combinatorial set according to some desired probability distribution. However, whereas up to now we have used this random sampling to estimate the expectations of suitably defined random variables over the set, we will now use it to optimize a function. This is the key ingredient of several randomized search heuristics in combinatorial optimization, the most celebrated of which is known as simulated annealing.

As usual, let Ω be a large combinatorial set, which we think of now as the set of feasible solutions to some optimization problem. Let $f : \Omega \to \mathbb{R}^+$ be an objective function defined on Ω; our goal is to find a solution $x \in \Omega$ for which the value $f(x)$ is maximum (or, symmetrically, minimum). As an illustrative example, let us take the maximum cut problem. Here Ω is the set of partitions of the vertices of a given undirected graph $G = (V, E)$ into two sets S and $\overline{S} = V - S$. Our goal is to find a partition that maximizes the number of edges between S and $\overline{S}$.

Here is a very general approach to problems of this kind. First, we define a connected, undirected graph H on vertex set Ω: this graph is often referred to as a *neighborhood structure*. Typically, the neighbors of a solution $x \in \Omega$ are close to x under some measure of distance that is natural to the combinatorial structures in question: for example, in the maximum cut problem, the neighbors of a particular partition $(S, \overline{S})$ might be all partitions of the form $(S-s, \overline{S}+s)$ and $(S+t, \overline{S}-t)$ obtained by moving one element across the partition. Next we construct a Markov chain in the form of a biased random walk on the graph H of a special form. Let $d(x)$ denote the degree of vertex x in H, and let D be an upper bound on the maximum degree. Then transitions from any state $x \in \Omega$ are made as follows:

I. with probability $\frac{1}{2}$ let $y = x$; otherwise,

II. select $y \in \Omega$ according to the distribution

$$\Pr(y) = \begin{cases} \frac{1}{D} & \text{if } y \text{ is a neighbor of } x; \\ 1 - \frac{d(x)}{D} & \text{if } y = x; \\ 0 & \text{otherwise;} \end{cases}$$

III. go to y with probability $\min\{1, \alpha^{f(y)-f(x)}\}$.

Here $\alpha \geq 1$ is a fixed parameter whose role will become clear shortly. We shall refer to this Markov chain as $\mathfrak{MC}(\alpha)$. Note that $\mathfrak{MC}(\alpha)$ always accepts transitions to neighbors with better values of f, but rejects transitions to poorer neighbors with a probability that depends on α.[17]

Let us observe some general properties of this Markov chain. First, since H is connected, the chain is irreducible, and since all self-loop probabilities are non-zero it is aperiodic; hence it is ergodic. Now define

$$\pi_\alpha(x) = \frac{\alpha^{f(x)}}{Z(\alpha)}, \qquad \text{for } x \in \Omega, \tag{12.17}$$

where $Z(\alpha)$ is a normalizing factor to make π_α a probability distribution. Then it is an easy matter to check that the chain is reversible with respect to π_α, i.e., the transition probabilities $P(x, y)$ satisfy the detailed balance condition

$$\pi_\alpha(x)P(x, y) = \pi_\alpha(y)P(y, x), \quad \text{for all } x, y \in \Omega.$$

All this implies that the Markov chain converges to the stationary distribution π_α. A Markov chain of this form is known as a *Metropolis process*, in honor of one of its inventors [Met53].

Now let us examine the stationary distribution more closely. From (12.17) it is clear that, for any value of $\alpha \geq 1$, π_α is a monotonically increasing function of $f(x)$. Hence it favors better solutions. Moreover, the effect of this bias increases with α: as $\alpha \to \infty$, the distribution becomes more sharply peaked around optimal solutions. At the other extreme, when $\alpha = 1$ the distribution is uniform over Ω.

Our optimization algorithm is now immediate: simply simulate the Markov chain $\mathfrak{MC}(\alpha)$ for some number, T, of steps, starting from an arbitrary initial solution, and output the best solution seen during the simulation. We shall refer to this algorithm as the *Metropolis algorithm at* α. How should we choose the parameter α? For sufficiently large T, we can view the algorithm as essentially sampling from the stationary distribution π_α. If we want to be reasonably sure of finding a good solution, we want to make α small so that π_α is well concentrated. On the other hand, intuitively, as α increases the chain becomes less mobile and more likely to get stuck in local optima: indeed, in the limit as $\alpha \to \infty$, $\mathfrak{MC}(\alpha)$ simply becomes a very naïve "randomized greedy" algorithm. This tradeoff suggests that we should use an intermediate value of α.

To precisely quantify the performance of the Metropolis algorithm at a given value of α, we would need to analyze the expected hitting time from the initial solution to the set of optimal (or near-optimal) solutions. However, we can get an upper bound on the time taken to find a good solution by analyzing the mixing time. Certainly, if $\mathfrak{MC}(\alpha)$ is close to stationarity after T steps, then the probability that we find a good solution is at

[17] In the case where we wish to minimise f, everything we say carries over with α replaced by α^{-1}.

least the weight of such solutions in the stationary distribution π_α. We shall illustrate this approach by adapting the matching example of Section 12.4, for which we have already developed all the necessary technology.

Consider the classical optimization problem of finding a matching of maximum cardinality in a graph. Thus Ω is the set of all matchings in a graph $G = (V, E)$, and we are trying to maximize the function $f : \Omega \to \mathbb{R}$ given by $f(M) = |M|$. It is well known that this problem can be solved in polynomial time, but the algorithm for non-bipartite graphs is far from trivial [Edm65]. We shall show that the much simpler Metropolis algorithm solves the problem for most graphs, and finds a good approximate solution for all graphs, with high probability in polynomial time. The key to the algorithm's success is a carefully chosen value of the parameter α.

We have in fact already defined a suitable Metropolis process for the maximum matching problem: it is the Markov chain $\mathfrak{M}_{\text{match}}(\lambda)$ from Section 12.4. A glance at the definition of this chain reveals that it is a Metropolis process whose neighborhood structure is defined by edge additions, deletions, and exchanges, and with $D = |E|$ and $\alpha = \lambda$. We saw in Section 12.4 that $\mathfrak{M}_{\text{match}}(\lambda)$ gets very close to its stationary distribution, π_λ, in time polynomial in λ and the number of vertices in G.

Let us first consider the case of $2n$-vertex graphs G for which the ratio m_{n-1}/m_n is polynomially bounded, i.e., $m_{n-1}/m_n \le q(n)$ for some fixed polynomial q.[18] (Of course, for such graphs maximum matchings are perfect matchings.) As we have seen in Section 12.5.1, this actually covers almost all graphs, as well as several interesting special families such as dense graphs. We also saw in Section 12.5.1 that, if we take $\lambda = q(n) \ge m_{n-1}/m_n$, then the weight of perfect matchings in the stationary distribution π_λ is at least $\frac{1}{n+1}$ (see equation (12.15)). Hence, by running the Metropolis algorithm $O(n)$ times (or, alternatively, by increasing λ by a constant factor), we can be almost certain of finding a perfect matching. The running time for each run is polynomial in n and $\lambda = q(n)$, and hence polynomial in n. The same result holds more generally for graphs with a maximum matching of size k_0, provided that m_{k_0-1}/m_{k_0} is polynomially bounded.

The above analysis breaks down for arbitrary graphs because the value of λ required to find a maximum matching could be very large. However, for arbitrary graphs, we can prove the weaker result that the Metropolis algorithm will find an *approximately* maximum matching in polynomial time. Let G be an arbitrary graph, and suppose we wish to find a matching in G of size at least $k = \lceil (1-\varepsilon)k_0 \rceil$, where k_0 is the size of a maximum matching in G and $\varepsilon \in (0, 1)$. We claim that, if we run the Metropolis algorithm for a polynomial number of steps with $\lambda = |E|^{(1-\varepsilon)/\varepsilon}$, then with probability at least $\frac{1}{n+1}$ we will find such a matching. (Note, however, that the running time is exponential in the accuracy parameter ε^{-1}.) Once again, the success probability can be boosted by repeated trials, or by increasing λ by a small constant factor.

To justify the above claim, we use the log-concavity property of matchings and the fact that $m_{k_0} \ge 1$ to deduce that

$$m_{k-1} = m_{k_0} \prod_{j=k}^{k_0} \frac{m_{j-1}}{m_j} \ge \left(\frac{m_{k-1}}{m_k}\right)^{k_0-k+1}. \tag{12.18}$$

But since j-matchings in G are subsets of E of size j, there is also the crude upper bound

[18] Recall that m_k denotes the number of k-matchings in G.

$m_{k-1} \leq |E|^{k-1}$. Hence, from (12.18) we conclude that

$$\frac{m_{k-1}}{m_k} \leq |E|^{(1-\varepsilon)/\varepsilon} = \lambda.$$

Now we use log-concavity again to argue that, for $0 \leq i < k$, we have $m_i/m_k \leq (m_{k-1}/m_k)^{k-i} \leq \lambda^{k-i}$. It follows that the weight of i-matchings in the stationary distribution π_λ is bounded above by the weight of the k-matchings. Hence, the probability of being at a matching of size k or more is at least $\frac{1}{n+1}$, as we claimed.

Rigorous results like this about the performance of the Metropolis algorithm on non-trivial optimization problems are few and far between. The above result on approximating maximum matchings was obtained via a more complex argument by Sasaki and Hajek [SH88], who also show that this result is best possible in the sense that the Metropolis algorithm cannot be expected to find a truly maximum matching in arbitrary graphs in polynomial time, even if the algorithm is allowed to vary the parameter α in an arbitrarily complicated fashion. Negative results of a similar flavor for other problems can be found in [Sas91] and [Jer92]. Jerrum and Sorkin [JS93] prove a positive result for the graph bisection problem analogous to the one above for finding a maximum matching in random graphs: they show that, for almost every input graph in a suitable random graph model, the Metropolis algorithm run at a carefully chosen value of the parameter α will find a minimum bisection of the graph in polynomial time with high probability. Their approach is different from the one presented here, in that they argue directly about the hitting time rather than analyzing the mixing time as we have done. Finally, a recent paper of Kannan, Mount, and Tayur [KMT94] shows how the Metropolis algorithm can be used to efficiently find approximate solutions to a class of convex programming problems.

We close with a brief discussion of the popular optimization heuristic known as simulated annealing, first proposed in [KGV83]. This heuristic is widely used in combinatorial optimization: for a comprehensive survey of experimental results, see for example [JAMS88, JAMS91]. Essentially, the idea is to simulate the Metropolis process while at the same time varying the parameter α according to a heuristic scheme. Thus, a simulated annealing algorithm is specified by a Metropolis process $\mathfrak{MC}(\alpha)$, together with an increasing function $\alpha : \mathbb{N} \to [1, \infty)$. At time t, the process makes a transition according to $\mathfrak{MC}(\alpha(t))$; we can therefore view it as a *time-inhomogeneous* Markov chain on Ω. After some specified number of steps, the algorithm terminates and returns the best solution encountered so far.

The function α is usually referred to as a *cooling schedule*, in accordance with the interpretation of α^{-1} as a "temperature." Increasing α thus corresponds to decreasing temperature, or cooling. The term "simulated annealing" derives from the analogy with the physical annealing process, in which a substance such as glass is heated to a high temperature and then gradually cooled, thereby "freezing" into a state whose energy is locally minimum. If the cooling is done sufficiently slowly, this state will tend to be a *global* energy minimum, corresponding to maximum strength of the solid.

This more complex process is even harder to analyze than the Metropolis algorithm itself. Since the Markov chain is not time-homogeneous, even the question of asymptotic convergence is non-trivial. Holley and Stroock [HS88] proved the existence of a cooling schedule that guarantees convergence to a global optimum: however, the schedule is so slow that the time taken to converge is comparable with the size of Ω, which makes the algorithm uncompetitive with naïve exhaustive search. It remains an outstanding open problem to exhibit a natural example in which simulated annealing with any non-trivial cooling schedule provably outperforms the Metropolis algorithm at a carefully chosen fixed value of α.

Acknowledgments Mark Jerrum was supported in part by a Nuffield Foundation Science Research Fellowship, Grant GR/F 90363 of the UK Science and Engineering Research Council, and EU Esprit Working Group No. 7097, "RAND". Alistair Sinclair was supported in part by NSF Grant CCR-9505448 and a UC Berkeley Faculty Research Grant.

REFERENCES

[Ald81] D. Aldous. Random walks on finite groups and rapidly mixing Markov chains, *Séminaire de Probabilités XVII*, Springer Lecture Notes in Mathematics 986, 1981/82, 243–297.

[Ald82] D. Aldous. Some inequalities for reversible Markov chains, *Journal of the London Mathematical Society*, 25(2):564–576, 1982.

[Ald87] D. Aldous. On the Markov chain simulation method for uniform combinatorial distributions and simulated annealing, *Probability in the Engineering and Informational Sciences*, 1:33–46, 1987.

[Ald90] D. Aldous. The random walk construction for spanning trees and uniform labeled trees, *SIAM Journal on Discrete Mathematics*, 3:450–465, 1990.

[AD86] D. Aldous and P. Diaconis. Shuffling cards and stopping times, *American Mathematical Monthly*, 93:333–348, 1986.

[Alon86] N. Alon. Eigenvalues and expanders, *Combinatorica*, 6:83–96, 1986.

[AM85] N. Alon and V.D. Milman. λ_1, isoperimetric inequalities for graphs and superconcentrators, *Journal of Combinatorial Theory Series B*, 38:73–88, 1985.

[AK91] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, 156–163, 1991.

[BS85] A. Berretti and A.D. Sokal. New Monte Carlo method for the self-avoiding walk, *Journal of Statistical Physics*, 40:483–531, 1985.

[Bro86] A.Z. Broder. How hard is it to marry at random? (On the approximation of the permanent), *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, ACM Press, 50–58, 1986. Erratum in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, p. 551, 1988.

[Bro89] A.Z. Broder. Generating random spanning trees, *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, 442–447, 1989.

[BDJ96] R. Bubley, M. Dyer, and M. Jerrum. *A new approach to polynomial-time random walks for volume computation* (preprint), 1996.

[Che70] J. Cheeger. A lower bound for the smallest eigenvalue for the Laplacian, *Problems in Analysis* (R.C. Gunning, ed.), Princeton University Press, Princeton NJ, 1970, 195–199.

[CdBS55] E.G.D. Cohen, J. de Boer, and Z.W. Salsburg. A cell-cluster theory for the liquid state II, *Physica*, XXI:137–147, 1955.

[DLMV88] P. Dagum, M. Luby, M. Mihail, and U. V. Vazirani. Polytopes, permanents and graphs with large factors, *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, 412–421, 1988.

[Dia88] P. Diaconis. *Group representations in probability and statistics*, Lecture Notes Monograph Series Vol. 11, Institute of Mathematical Statistics, Hayward, CA, 1988.

[DE85] P. Diaconis and B. Efron. Testing for independence in a two-way table, *Annals of Statistics*, 13:845–913, 1985.

[DSC93] P. Diaconis and L. Saloff-Coste. Comparison techniques for reversible Markov chains, *Annals of Applied Probability*, 3:696–730, 1993.

[DS91] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains, *Annals of Applied Probability*, 1:36–61, 1991.

[DF88] M.E. Dyer and A.M. Frieze. On the complexity of computing the volume of a polyhedron, *SIAM Journal on Computing*, 17:967–975, 1988.

[DF91] M. Dyer and A. Frieze. Computing the volume of convex bodies: a case where randomness provably helps, *Probabilistic Combinatorics and its Applications*, Proceedings of AMS Symposia in Applied Mathematics, 44:123–170, 1991.

[DFJ94] M. Dyer, A. Frieze, and M. Jerrum. Approximately counting Hamilton cycles in dense graphs, *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, 336–343, 1994. Full version to appear in *SIAM Journal on Computing*.

[DFK91] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies, *Journal of the ACM*, 38:1–17, 1991.

[DFKKPV93] M. Dyer, A. Frieze, R. Kannan, A. Kapoor, L. Perkovic, and U. Vazirani. A mildly exponential time algorithm for approximating the number of solutions to a multidimensional knapsack problem, *Combinatorics, Probability and Computing*, 2:271–284, 1993.

[DKM95] M. Dyer, R. Kannan, and J. Mount. *Sampling contingency tables* (preprint), 1995.

[Edm65] J. Edmonds. Paths, trees and flowers, *Canadian Journal of Mathematics*, 17:449–467, 1965.

[Elek86] G. Elekes. A geometric inequality and the complexity of computing volume, *Discrete and Computational Geometry*, 1:289–292, 1986.

[FM92] T. Feder and M. Mihail. Balanced matroids, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, 26–38, 1992.

[Fish61] M.E. Fisher. Statistical mechanics of dimers on a plane lattice, *Physics Review*, 124:1664–1672, 1961.

[Friez89] A.M. Frieze. *A note on computing random permanents* (unpublished manuscript), 1989.

[FS92] A. Frieze and S. Suen. Counting the number of Hamiltonian cycles in random digraphs, *Random Structures and algorithms*, 3:235–241, 1992.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979, p. 176.

[Gill93] D. Gillman. A Chernoff bound for random walks on expander graphs, *Proceedings of the 34th Annual IEEE Conference on Foundations of Computer Science*, 680–691, 1993.

[Gugg52] E.A. Guggenheim. *Mixtures*, Clarendon Press, Oxford, 1952.

[HL72] O.J. Heilmann and E.H. Lieb. Theory of monomer-dimer systems, *Communications in Mathematical Physics*, 25:190–232, 1972.

[HS88] R. Holley and D.W. Stroock. Simulated annealing via Sobolev inequalities, *Communications in Mathematical Physics*, 115:553–569, 1988.

[IJ94] R.W. Irving and M.R. Jerrum. 3-D statistical data security problems, *SIAM Journal on Computation*, 23:170-184, 1994.

[Jer87] M.R. Jerrum. Two-dimensional monomer-dimer systems are computationally intractable, *Journal of Statistical Physics*, 48:121–134, 1987. Erratum in *Journal of Statistical Physics*, 59:1087–1088, 1990.

[Jer92] M.R. Jerrum. Large cliques elude the Metropolis process, *Random Structures and Algorithms*, 3:347–359, 1992.

[Jer93b] M. Jerrum. Uniform sampling modulo a group of symmetries using Markov chain simulation, *Expanding Graphs*, DIMACS Series in Discrete Mathematics and Computer Science 10 (J. Friedman, ed.), American Mathematical Society, 1993, 37–47.

[Jer94] M. Jerrum. The computational complexity of counting, *Proceedings of the International Congress of Mathematicians, Zürich 1994*, Birkhäuser, Basel, 1995, 1407–1416.

[Jer95] M. Jerrum. A very simple algorithm for estimating the number of k-colourings of a low-degree graph, *Random Structures and Algorithms*, 7:157–165, 1995.

[JMS92] M. Jerrum, B. McKay, and A. Sinclair. When is a graphical sequence stable? *Random Graphs* 2 (A. Frieze and T. Łuczak, eds), Wiley, 1992, 101–115.

[JS89] M.R. Jerrum and A.J. Sinclair. Approximating the permanent, *SIAM Journal on Computing*, 18:1149–1178, 1989.

[JS90a] M.R. Jerrum and A.J. Sinclair. Fast uniform generation of regular graphs, *Theoretical Computer Science*, 73:91–100, 1990.

[JS93] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model, *SIAM Journal on Computing*, 22:1087–1116, 1993.

[JS94] M. Jerrum and G.B. Sorkin. Simulated annealing for graph bisection, *Proceedings of the 34th Annual IEEE Conference on Foundations of Computer Science*, Computer Society Press, 94–103, 1993.

[JVV86] 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.

[JV92] M. Jerrum and U. Vazirani. A mildly exponential approximation algorithm for the permanent, *Proceedings of the 33rd Annual IEEE Conference on Foundations of Computer Science*, Computer Society Press, 320–326, 1992.

[JAMS88] D.S. Johnson, C.R. Aragon, L.A. McGeogh, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; Part I, graph partitioning, *Operations Research*, 37:865–892, 1988.

[JAMS91] D.S. Johnson, C.R. Aragon, L.A. McGeogh, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning, *Operations Research*, 39:378–406, 1991.

[Kah94] N. Kahale. *Large deviation bounds for Markov chains*, DIMACS Technical Report 94-39, June 1994. To appear in *Combinatorics, Probability and Computing*.

[Kah95] N. Kahale. *A semidefinite bound for mixing rates of Markov chains*, DIMACS Technical Report 95-41, September 1995.

[Kan94] R. Kannan. Markov chains and polynomial time algorithms. *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 656–671, 1994.

[KLS94a] R. Kannan, L. Lovász, and M. Simonovits. *Random walks and a faster algorithm for convex sets* (manuscript).

[KLS94b] R. Kannan, L. Lovász, and M. Simonovits. Isoperimetric problems for convex sets and a localization lemma, *Discrete and Computational Geometry*, 13:541–559, 1995.

[KMT94] R. Kannan, J. Mount, and S. Tayur. A randomized algorithm to optimize over certain convex sets, *Mathematics of Operations Research*, 20:529–550, 1995.

[KL83] R.M. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems, *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science*, 56–64, 1983.

[KK90] A. Karzanov and L. Khachiyan. *On the conductance of order Markov chains*, Technical Report DCS 268, Rutgers University, June 1990.

[Kast61] P.W. Kasteleyn. The statistics of dimers on a lattice I: The number of dimer arrangements on a quadratic lattice, *Physica*, 27:1209–1225, 1961.

[KRS96] C. Kenyon, D. Randall, and A. Sinclair. Approximating the number of monomer-dimer coverings of a lattice, *Journal of Statistical Physics*, 83:637–659, 1996.

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing, *Science*, 220:671–680, 1983.

[LS88] G.F. Lawler and A.D. Sokal. Bounds on the L^2 spectrum for Markov chains and Markov processes: a generalization of Cheeger's inequality, *Transactions of the American Mathematical Society*, 309:557–580, 1988.

[LP86] L. Lovász and M.D. Plummer. *Matching Theory*, North-Holland, Amsterdam, 1986.

[LS93] L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm, *Random Structures and Algorithms*, 4:359–412, 1993.

[LRS95] M. Luby, D. Randall, and A. Sinclair. Markov chain algorithms for planar lattice structures, *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, 150–159, 1995.

[Mat91] P. Matthews. Generating random linear extensions of a partial order, *The Annals of Probability*, 19:1367–1392, 1991.

[Met53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculation by fast computing machines, *Journal of Chemical Physics*, 21:1087–1092, 1953.

[Mih89a] M. Mihail. On coupling and the approximation of the permanent, *Information Processing Letters*, 30:91–95, 1989.

[Mih89b] M. Mihail. Conductance and convergence of Markov chains: a combinatorial treatment of expanders, *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, 526–531, 1989.

[MP94] M. Mihail and C.H. Papadimitriou. On the random walk method for protocol testing, *Proceedings of the 6th International Conference on Computer Aided Verification*, Springer Lecture Notes in Computer Science 818, 1994, 132–141.

[MW91] M. Mihail and P. Winkler. On the number of Eulerian orientations of a graph, *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, 138–145, 1992.

[Mot89] R. Motwani. Expanding graphs and the average-case analysis of algorithms for matchings and related problems, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, ACM Press, 550–561, 1989.

[PW95] J. Propp and D. Wilson. *Exact sampling with coupled Markov chains and applications to statistical mechanics* (preprint), 1995. To appear in *Random Structures & Algorithms*, 1996.

[RS94] D. Randall and A.J. Sinclair. Testable algorithms for self-avoiding walks, *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, ACM Press, 593–602, 1994.

[Rob35] J.K. Roberts. Some properties of adsorbed films of oxygen on tungsten, *Proceedings of the Royal Society of London A*, 152:464–480, 1935.

[Sas91] G.H. Sasaki. The effect of the density of states on the Metropolis algorithm, *Information Processing Letters*, 37:159–163, 1991.

[SH88] G.H. Sasaki and B. Hajek. The time complexity of maximum matching by simulated annealing, *Journal of the ACM*, 35:387–403, 1988.

[Sin92] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow, *Combinatorics, Probability and Computing*, 1:351–370, 1992.

[Sin93] A.J. Sinclair. *Randomised algorithms for counting and generating combinatorial structures*, Advances in Theoretical Computer Science, Birkhäuser, Boston, 1993.

[SJ89] A.J. Sinclair and M.R. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains, *Information and Computation*, 82:93–133, 1989.

[SW87] R.H. Swendsen and J-S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations, *Physical Review Letters*, 58:86–88, 1987.

[TF61] H.N.V. Temperley and M.E. Fisher. Dimer problem in statistical mechanics—an exact result, *Philosophical Magazine*, 6:1061–1063, 1961.

[Tod89] S. Toda. On the computational power of PP and $\oplus$P, *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, Computer Society Press, 514–519, 1989.

[Usp37] J.V. Uspensky. *Introduction to mathematical probability*, McGraw Hill, 1937.

[Val79a] L.G. Valiant. The complexity of computing the permanent, *Theoretical Computer Science*, 8:189–201, 1979.

[Val79b] L.G. Valiant. The complexity of enumeration and reliability problems, *SIAM Journal on Computing*, 8:410–421, 1979.

[Wel90] D.J.A. Welsh. The computational complexity of some classical problems from statistical physics, *Disorder in Physical Systems*, Oxford University Press, 1990, 307–321.

APPENDIX

Proof of Proposition 12.3. The proof essentially hinges on the bound $(\operatorname{Var} f_i)/(\operatorname{E} f_i)^2 \leq \mathrm{e}$, which we established for the random variable f_i. However, this random variable is defined with respect to the distribution π_{λ_i}, whereas our samples come from a distribution $\widehat{\pi}_{\lambda_i}$ obtained from a finite-length simulation of the Markov chain, whose variation distance from π_{λ_i} satisfies

$$\|\widehat{\pi}_{\lambda_i} - \pi_{\lambda_i}\| \leq \frac{\varepsilon}{5\mathrm{e}r}. \tag{A.1}$$

We shall therefore work with the random variable $\widehat{f_i}$, defined analogously to f_i except that the matching M is selected from the distribution $\widehat{\pi}_{\lambda_i}$ rather than π_{λ_i}. Since $\widehat{f_i}$ takes values in $(0, 1]$, its expectation $\operatorname{E}\widehat{f_i} = \widehat{\rho_i}$ clearly satisfies $|\widehat{\rho_i} - \rho_i| \leq \varepsilon/5\mathrm{e}r$, which by (12.10) implies

$$\left(1 - \frac{\varepsilon}{5r}\right)\rho_i \leq \widehat{\rho_i} \leq \left(1 + \frac{\varepsilon}{5r}\right)\rho_i. \tag{A.2}$$

Moreover, again using (12.10), the variance of $\widehat{f_i}$ satisfies

$$(\operatorname{Var}\widehat{f_i})/(\operatorname{E}\widehat{f_i})^2 \leq \widehat{\rho_i}^{\,-1} \leq 2\rho_i^{-1} \leq 2\mathrm{e}, \tag{A.3}$$

where we have also used (A.2) crudely to deduce that $\widehat{\rho_i} \geq \frac{1}{2}\rho_i$.

We can now compute the sample size needed to ensure a good final estimate. Let $X_i^{(1)}, \ldots, X_i^{(S)}$ be a sequence of S independent copies of the random variable $\widehat{f_i}$ obtained by sampling S matchings from the distribution $\widehat{\pi}_{\lambda_i}$, and let $\overline{X}_i = S^{-1}\sum_{j=1}^{S} X_i^{(j)}$ be the sample mean. Clearly, $\operatorname{E}\overline{X}_i = \operatorname{E}\widehat{f_i} = \widehat{\rho_i}$, and $\operatorname{Var}\overline{X}_i = S^{-1}\operatorname{Var}\widehat{f_i}$. Our estimator of $\rho = Z(\widehat{\lambda})^{-1}$ is the random variable $X = \prod_{i=1}^{r}\overline{X}_i$. The expectation of this estimator is $\operatorname{E}X = \prod_{i=1}^{r}\widehat{\rho_i} = \widehat{\rho}$, which by (A.2) satisfies

$$\left(1 - \frac{\varepsilon}{4}\right)\rho \leq \widehat{\rho} \leq \left(1 + \frac{\varepsilon}{4}\right)\rho. \tag{A.4}$$

Also, by (A.3), the variance satisfies

$$\begin{aligned}
\frac{\operatorname{Var}X}{(\operatorname{E}X)^2} &= \prod_{i=1}^{r}\left(1 + \frac{\operatorname{Var}\overline{X}_i}{(\operatorname{E}\overline{X}_i)^2}\right) - 1 \\
&\leq \left(1 + \frac{2\mathrm{e}}{S}\right)^r - 1 \\
&\leq \exp(2\mathrm{e}r/S) - 1 \\
&\leq \varepsilon^2/64,
\end{aligned}$$

provided we choose the sample size $S = \lceil 130\mathrm{e}\varepsilon^{-2}r \rceil$. (Here we are using the fact that $\exp(x/65) \leq 1 + x/64$ for $0 \leq x \leq 1$.) Now Chebyshev's inequality applied to X yields

$$\Pr(|X - \widehat{\rho}| > (\varepsilon/4)\widehat{\rho}) \leq \frac{16}{\varepsilon^2}\frac{\operatorname{Var}X}{(\operatorname{E}X)^2} \leq \frac{1}{4},$$

so we have, with probability at least $\frac{3}{4}$,

$$\left(1-\frac{\varepsilon}{4}\right)\widehat{\rho} \le X \le \left(1+\frac{\varepsilon}{4}\right)\widehat{\rho}. \tag{A.5}$$

Combining (A.4) and (A.5) we see that, with probability at least $\frac{3}{4}$, $Y = X^{-1}$ lies within ratio $1 \pm \varepsilon$ of $\rho^{-1} = Z(\widehat{\lambda})$, which completes the proof. ■