MIXING

DANA RANDALL*

Abstract

In this tutorial, we introduce the notion of a Markov chain and explore how it can be used for sampling from a large set of configurations. Our primary focus will be determining how quickly a Markov chain "mixes," or converges to its stationary distribution, as this is the key factor in the running time. We provide an overview of several techniques used to establish good bounds on the mixing time. Examples will be chosen from applications in statistical physics, although the methods are much more general.

1 Introduction

Markov chain Monte Carlo is ubiquitous across scientific disciplines as a computational means for studying large, complicated sets. The idea is to simulate a random walk that moves among configurations in the large set. Even though each configuration might only lead to a small set of nearest neighbors, eventually the Markov chain underlying the random walk will converge to a useful distribution over the entire space of configurations.

The mathematical foundations underlying the design of these algorithms can be found in probability theory. The field of stochastic processes gives conditions prescribing when a Markov chain will converge to a unique stationary distribution and how to determine what that distribution is. Designing a Markov chain that converges quickly to the desired distribution provides a useful tool for sampling.

Over the last 15 years there has been a flurry of activity leading to breakthroughs in our understanding of how to bound the convergence rate of Markov chains and ultimately design efficient sampling algorithms. This tutorial is intended to give an introduction into some of the key ideas underlying these results; for further details we refer the reader to the references provided, especially surveys [18, 22, 23, 35].

Two of the most notable success stories based on this method are estimating the volume of a convex body [10] and estimating the permanent of a matrix [20]. The exact versions of both of these problems are $\#P-Complete$, the complexity class of hard counting problems. In a seminal paper, Valiant defined the class $\#P$ in the context of counting the number of perfect matchings in a graph; this is precisely the problem of calculating the permanent of the adjacency matrix of a bipartite graph [38]. The solutions to the approximate version rely heavily on randomly sampling, in the first case, points in the convex body, and in the second, perfect matchings of the graph. The intimate connection between random sampling and approximate counting was established by Jerrum, Valiant, and Vazirani for a wide class of problems known as *self-reducible*. We refer the reader to [21] and [36] for details underlying this very important connection. In this tutorial, we concentrate solely on the sampling aspects of Monte Carlo experiments, foregoing the many beautiful applications of sampling, including approximate counting.

1.1 The basics of sampling

Markov chains are useful when we have a finite set of configurations Ω from which we would like to sample. The idea behind designing a Markov chain is first to connect the state space so that each configuration has a small number of nearest neighbors. Then, starting at some arbitrary start configuration $x_0 \in \Omega$, the Markov chain defines a random walk along the edges of this graph, walking from one configuration in Ω to another.

Let us consider, for example, how to design a chain for randomly sampling independent sets of some finite graph. One simple way to connect the state space is to allow transitions between independent sets I and I' that have Hamming distance $\varphi(I, I') = 1$. The Markov chain starts at some initial state I_0, say the empty set. During each step of the simulation, it proceeds by choosing a vertex v uniformly from V; if $v \in I_t$, then we remove it and let $I_{t+1} = I \setminus \{v\}$, whereas if $v \notin I_t$ we can add it and set $I_{t+1} = I \cup \{v\}$, provided this leads to a valid independent set. In the case that we cannot add v because a neighbor is present, then we do nothing and set $I_{t+1} = I_t$. The transition probabilities of this chain are

$$P(I, I') = \begin{cases} 1/n, & \text{if } \varphi(I, I') = 1, \\ 0, & \text{if } \varphi(I, I') > 1, \\ 1 - \sum_{J \neq I} P(I, J), & \text{if } I = I'. \end{cases}$$

*College of Computing, Georgia Institute of Technology, Atlanta GA. Email: randall@cc.gatech.edu. Supported in part by NSF CCR-0105639 and an Alfred P. Sloan research fellowship. Part of this work was done at the focussed research group on discrete probability at BIRS.

The following definition and lemmas formalize why this is a good candidate chain.

Definition. A Markov chain is *ergodic* if it is

1. *irreducible*, i.e., $\forall x, y \in \Omega$, there is a t such that $P^t(x, y) > 0$, and
2. *aperiodic*, i.e., $\forall x, y \in \Omega$, g.c.d. $\{t : P^t(x, y) > 0\} = 1$.

We can see that our chain on independent sets is irreducible because we can get from any configuration to the the empty independent set by successively removing vertices. Moreover, the chain is aperiodic because of the self-loop probabilities that occur whenever we choose a vertex that cannot be added without violating the independence requirement.

For chains that are not aperiodic, self-loops can be added at every vertex at a small constant cost to the running time of the algorithm. A chain that has self-loop probabilities of $1/2$ everywhere is called a *lazy chain*. Ergodicity is a useful minimum requirement for defining a useful Markov chain.

Lemma 1.1 *Any finite, ergodic Markov chain converges to a unique stationary distribution π, i.e., for all $x, y \in \Omega$, we have that $\lim_{t\to\infty} P^t(x, y) = \pi(y)$.*

To reason about the limiting probability distribution, known as the *stationary distribution*, we rely the *detailed balance condition* given in the following lemma.

Lemma 1.2 *Let M be an ergodic Markov chain on a finite state state space Ω with transition probabilities $P(\cdot, \cdot)$. If $\pi' : \Omega \to [0, 1]$ is any function satisfying the detailed balance condition:*

$$\pi(x)P(x, y) = \pi(y)P(y, x),$$

and if it also satisfies $\sum_{x\in\Omega} \pi'(x) = 1$, then π' is the unique stationary distribution of M.

Any chain that satisfies detailed balance for some π' is called *time-reversible*.

For the independent set chain, the transition probabilities are symmetric, i.e., $P(I, I') = P(I', I)$ for all $I, I' \in \Omega$. If follows from lemma 1.2 and the ergodicity of the chain that the stationary distribution must be uniform.

Two questions immediately present themselves. First, how do we modify this chain in order to sample from a more complicated distribution? Second, how long do we have to simulate the walk before we can trust that our samples are chosen from very close to the stationary distribution? The celebrated Metropolis algorithm gives a standard way of approaching the first of these questions, and is just based on a careful consideration of lemma 1.2. The second question is the subject of the remaining part of this paper and requires much more sensitive consideration.

1.2 The Metropolis algorithm

The Metropolis-Hastings algorithm [29] is a remarkably simple, yet tremendously robust idea that is the starting point for anyone interested in sampling. It tells us how to assign the transition probabilities of any Markov chain so that so it will converge to any distribution. In 2000, it was selected as one of the top 10 algorithms by *Computing in Science and Engineering* [3].

The Metropolis Algorithm

Starting at x, repeat:

1. Pick a neighbor y of x in Ω uniformly with probability $1/2\Delta$, where Δ is the maximum degree in the graph G.
2. Move to y with probability $\min\left(1, \frac{\pi(y)}{\pi(x)}\right)$.
3. With all remaining probability stay at x.

Using detailed balance, it is easy to verify that if the state space is connected, then π must be the stationary distribution.

Returning to our example of independent sets, let us now assume that we wish to sample from the weighted distribution

$$\pi(I) = \frac{\lambda^{|I|}}{Z},$$

where $Z = \sum_{I'\in\Omega} \lambda^{|I'|}$ is the normalizing constant. We justify why this is a natural weighting in section 5. As before, we connect pairs of independent sets if they have Hamming distance 1. Let I and I' be two such sets, where $v \notin I$ and $I' = I \cup \{v\}$, for some vertex v. Since $|I'| = |I| + 1$, the stationary probabilities satisfy $\pi(I') = \lambda\pi(I)$.

The Metropolis algorithm says that we should define the transitions so that

$$P(I, I') = \frac{1}{2n} \min(1, \lambda),$$

while

$$P(I', I) = \frac{1}{2n} \min\left(1, \lambda^{-1}\right).$$

Notice that the normalizing constant Z drops out of the equation! This is quite fortuitous since we typically do not have any direct way of calculating it. Considering each of the cases where $\lambda > 1$ or $\lambda \leq 1$, we see that P and

π always satisfies detailed balance. This means that using these modified transition probabilities will allow us to converge on the correct stationary distribution.

This leaves our main question: *How long do we have to simulate the chain in order to get a good sample?*

1.3 The mixing time

The time a Markov chain takes to converge to its stationary distribution, known as the *mixing time* of the chain, is measured in terms of the variation distance between the distribution at time t and the stationary distribution. For a comparison of rates of convergence based on different measures of distance see [2, 23].

Definition. Letting $P^t(x,y)$ denote the t-step probability of going from x to y, the *total variation distance* at time t is

$$\|P^t, \pi\|_{tv} = \max_{x \in \Omega} \frac{1}{2} \sum_{y \in \Omega} |P^t(x,y) - \pi(y)|.$$

This is just the L_1 norm, with the $1/2$ introduced so that the distance is always at most 1. We now have

Definition. For $\varepsilon > 0$, the *mixing time* $\tau(\varepsilon)$ is

$$\tau(\varepsilon) = \min\{t : \|P^{t'}, \pi\|_{tv} \leq \varepsilon, \forall t' \geq t\}.$$

We say a Markov chain is *rapidly mixing* if the mixing time is bounded above by a polynomial in n and $\log \epsilon^{-1}$, where n is the size of each configuration in the state space.

It is well-known from probability theory that the eigenvalue gap of the transition matrix of the Markov chain provides a good bound on the mixing rate of a chain. We let $Gap(P) = 1 - |\lambda_1|$ denote the spectral gap, where $\lambda_0, \lambda_1, \ldots, \lambda_{|\Omega|-1}$ are the eigenvalues of the transition matrix P and $1 = \lambda_0 > |\lambda_1| \geq |\lambda_i|$ for all $i \geq 2$. The following result relates the spectral gap with the mixing time of the chain (see, e.g., [36]):

Theorem 1.3 *Let $\pi_* = \min_{x \in \Omega} \pi(x)$. For all $\varepsilon > 0$ we have*

(a) $\tau(\varepsilon) \leq \frac{1}{1-|\lambda_1|} \log(\frac{1}{\pi_* \varepsilon})$.

(b) $\tau(\varepsilon) \geq \frac{|\lambda_1|}{2(1-|\lambda_1|)} \log(\frac{1}{2\varepsilon})$.

Notice that the lazy chain with self-loop probabilities of 1/2 everywhere has only non-negative eigenvalues; this follows from the fact that the eigenvalues $\{\widehat{\lambda}_i\}$ of the lazy chain will satisfy $\widehat{\lambda}_i = (1 + \lambda_i)/2$ and $|\lambda_i| \leq 1$ for all i.

While this view of mixing is extremely useful for card shuffling applications and walks on symmetric groups, it tends to be less useful for the more complicated state spaces that arise in computer science. In particular, for most algorithmic applications the size of the state space is exponentially large and we typically do not have a compact, mathematical representation for the adjacency matrix, so it is far too difficult to determine the eigenvalues of the transition matrix. We are therefore left with the challenging task of finding sophisticated, indirect methods to establish the efficiency of our chains.

2 Coupling

One of the most popular methods for bounding mixing times is coupling, both because of its elegance and its simplicity. This was first introduced to computer science in the context of sampling spanning trees [4], and has since seen many more applications.

Definition. A *coupling* is a Markov chain on $\Omega \times \Omega$ defining a stochastic process $(X_t, Y_t)_{t=0}^{\infty}$ with the properties:

1. Each of the processes X_t and Y_t is a faithful copy of $\mathcal{M}$ (given initial states $X_0 = x$ and $Y_0 = y$).
2. If $X_t = Y_t$, then $X_{t+1} = Y_{t+1}$.

Condition 1 ensures that each process, viewed in isolation, is just simulating the original chain – yet the coupling updates them simultaneously so that they will tend to coalesce, or move closer together, according to some notion of distance. Once the pair of configurations agree, condition 2 guarantees they agree from that time forward. The coupling (or expected coalescence) time can provide a good bound on the mixing time of $\mathcal{M}$ if it is a carefully chosen coupling.

Definition. For initial states x, y let

$$T^{x,y} = \min\{t : X_t = Y_t \mid X_0 = x, Y_0 = y\},$$

and define the *coupling time* to be $T = \max_{x,y} \mathrm{E} T^{x,y}$.

The following result relates the mixing time and the coupling time (see, e.g., [1]).

Theorem 2.1 $\tau(\epsilon) \leq \lceil T \mathrm{e} \ln \epsilon^{-1} \rceil$.

We will consider a toy example of choosing a random vertex in the n-dimensional hypercube, $\Omega = \{0,1\}^n$. A natural Markov chain performs a simple random walk along the edges of the hypercube by iteratively flipping a random bit. However, we consider instead the lazy version of the chain because the hypercube is bipartite and will not be aperiodic unless we add self-loop probabilities.

The lazy chain also turns out to be more conducive to a coupling argument.

$\underline{MC_{cube}}$

Starting at the vertex $X_0 = (0, ..., 0)$, repeat:

1. Pick $(i, b) \in \{1, ..., n\} \times \{0, 1\}$.
2. Let X_{t+1} be X_t with the ith bit changed to b.

Letting $\varphi(\cdot, \cdot)$ be the Hamming distance, the transition matrix of this chain is

$$P(X, Y) = \begin{cases} 1/2n, & \text{if } \varphi(X, Y) = 1, \\ 1/2, & \text{if X=Y,} \\ 0, & \text{otherwise,} \end{cases}$$

It is easy to check that this chain is ergodic and symmetric, hence the stationary distribution is uniform.

To couple, we start with any two vertices X_0 and Y_0 on the hypercube and update them simultaneously by choosing the same pair (i, b). It is straightforward to see that the two configurations will coalesce as soon as we have updated each bit at least once. By the coupon collector's theorem this takes $O(n \ln n)$ steps, in expectation. Appealing to theorem 2.1, we have a bound on the mixing time.

Theorem 2.2 *The Markov chain on MC_{cube} has mixing time $\tau(\epsilon) = O(n \ln(n\epsilon^{-1}))$.*

The logarithmic dependence on ϵ^{-1} is typical for mixing rates, and the $O(n \ln n)$ is optimal.

In general, coupling proofs are a little more complicated because typically the distance between configurations can also increase, whereas on the hypercube it only decreases or remains unchanged. The strategy in this case is to consider the random walk on the random variable representing the distance. If we show that distance is decreasing in expectation, then we have a drift towards zero that allows us to prove a chain is rapidly mixing. Typically as the distance approaches zero there are fewer moves that will decrease the distance, so the coupon collecting theorem suggests we should expect an $O(n \ln n)$ coupling time. In section 2.1 we will see a more realistic example in the context of sampling k-colorings that also achieves this bound. The proof will use the more refined method of *path coupling*, although it can be easily replicated using a direct coupling argument.

2.1 Path coupling

While coupling is potentially a powerful technique, it is often prohibitively cumbersome to measure the expected change in distance between two arbitrary configurations. The method of *path coupling*, introduced by Bubley and Dyer, greatly simplifies this approach by showing that we really need only consider pairs of configurations that are close [5]. Since the configurations will agree in most positions, measuring the expected change in distance becomes much more palatable. Every path coupling argument can also be made directly using coupling, but usually this would require much more work.

The idea behind path coupling is to consider a small set $U \subseteq \Omega \times \Omega$ of pairs of configurations that are "close" according to some distance metric φ. For now we can think of the pairs of configurations with Hamming distance 1. Suppose that we have shown that the expected change in distance is decreasing for all of the pairs in U. To now reason about arbitrary configurations $X, Y \in \Omega$, we define a shortest path $z_0 = X,\ z_1, ...,\ z_r = Y$ of length r from X to Y, where $(z_i, z_{i+1}) \in U$ for all $0 \leq i < r$. If we define U correctly, then $\varphi(X, Y) = \sum_{i=0}^{r-1} \varphi(z_i, z_{i+1})$. If this is the case, we are done: by linearity of expectation, the expected change in distance between X and Y is the sum of the expected change between the pairs (z_i, z_{i+1}), and each of these has been shown to be at most zero. Of course, after the update there might be a shorter path between the new configurations X' and Y', but this just causes the new distance to be even smaller.

The following version of the path coupling theorem is convenient.

Theorem 2.3 (Dyer and Greenhill [12]) *Let φ be an integer valued metric defined on $\Omega \times \Omega$ which takes values in $\{0, \ldots, B\}$. Let U be a subset of $\Omega \times \Omega$ such that for all $(x_t, y_t) \in \Omega \times \Omega$ there exists a path $x_t = z_0, z_1, \ldots, z_r = y_t$ between x_t and y_t such that $(z_i, z_{i+1}) \in U$ for $0 \leq i < r$ and*

$$\sum_{i=0}^{r-1} \varphi(z_i, z_{i+1}) = \varphi(x_t, y_t).$$

Let $\mathcal{M}$ be a Markov chain on Ω with transition matrix P. Consider any random function $f : \Omega \to \Omega$ such that $Pr[f(x) = y] = P(x,y)$ for all $x, y \in \Omega$, and define a coupling of the Markov chain by $(x_t, y_t) \to (x_{t+1}, y_{t+1}) = (f(x_t), f(y_t))$.

1. *If there exists $\beta < 1$ such that*

$$\mathrm{E}[\varphi(x_{t+1}, y_{t+1})] \leq \beta\varphi(x_t, y_t),$$

for all $(x_t, y_t) \in U$, then the mixing time satisfies

$$\tau(\epsilon) \leq \frac{\ln(B\epsilon^{-1})}{1 - \beta}.$$

2. *If $\beta = 1$ (so $\mathrm{E}[\Delta\varphi(x_t, y_t)] \leq 0$, for all $x_t, y_t \in U$), let $\alpha > 0$ satisfy $Pr[\varphi(x_{t+1}, y_{t+1}) \neq \varphi(x_t, y_t)] \geq$*

α for all t such that $x_t \neq y_t$. The mixing time of $\mathcal{M}$ then satisfies

$$\tau(\epsilon) \leq \left\lceil \frac{\mathrm{e}B^2}{\alpha} \right\rceil \lceil \ln \epsilon^{-1} \rceil.$$

We now demonstrate the technique of path coupling on a Markov chain MC_{col} for sampling k-colorings of a graph G. This algorithm, based on local moves, is known as *Glauber dynamics*.

$\underline{MC_{col}}$:

Starting at t_0, repeat t times:

1. With probability 1/2 do nothing.
2. Pick $(v, c) \in V \times \{1, ..., k\}$.
3. If v can be recolored with color c, recolor it; otherwise do nothing.

We can easily verify that this Markov chain converges to the uniform distribution over k-colorings. To couple, we start with two arbitrary colorings X_0 and Y_0. Our first attempt at a reasonable coupling suggests that we should choose the same pair (v, c) to update each of X_t and Y_t at every step. This coupling is enough to demonstrate that MC_{col} is rapidly mixing when the number of colors is large enough, although an even better coupling is presented in [15].

Theorem 2.4 *The Markov chain on k-colorings MC_{col} has mixing time $\tau(\epsilon) = O(n \ln(n\epsilon^{-1}))$ on any n-vertex graph with maximum degree d whenever $k \geq 3d + 1$.*

Proof. Let x_0 and y_0 be two starting configurations. To couple, we choose $(v, c) \in v \times \{1, ..., k\}$ uniformly at each time t. We then update each of x_t and y_t by recoloring vertex v color c, if possible, thus defining x_{t+1} and y_{t+1}.

To apply path coupling, let $\varphi : \Omega \times \Omega \to \mathbf{Z}$ be a metric defined by the minimum length path connected configurations at Hamming distance 1. In other words, for any $x, y \in \Omega$, let $x = z_0, z_1, ..., z_\ell = y$ be the shortest path such that z_i and z_{i+1} are colorings that differ at a single vertex, and set $\varphi(x, y) = \ell$. When the number of colors used is much larger than the largest degree of G, it is a simple exercise to verify that $\varphi(x, y) \leq B = 2n$, for any x and y, where $n = |V|$.

Let U be the set of pairs of colorings at distance 1. To apply theorem 2.3, we need to consider $\mathrm{E}[\Delta\varphi(r, s)]$ for any $(r, s) \in U$. Suppose w is the vertex that is colored differently in r and s. We consider three cases:

- Case 1: $\underline{W = v}$: If $w = v$, then any color c not currently used for the neighbors of w in r and s will be a move accepted by both processes, and therefore r and s will agree in the next step. There are at least $k - d$ such colors. If, on the other hand, c agrees with a color used by at least one of the neighbors of w, then the move will be rejected by both processes. Together this tells us that if $w = v$, then $E[\Delta\varphi(x_t, y_t)] \leq -\frac{k-d}{kn}$.
- Case 2: $\underline{(w, v) \in E}$: If w is a neighbor of v, then the distance between r and s will remain unchanged unless c is the color of v in either r or s. In each of these two cases, the move will be accepted by at most one of the two processes; it can of course be rejected by both, in which case r and s remain unchanged and at distance 1. If it is accepted by one process, r or s, then $E[\Delta\varphi(r, s)] \leq \frac{2}{kn}$. This bound holds for each of the d choices of w.
- Case 3: $\underline{w \neq v \text{ and } (w, v) \notin E}$: If w has distance at least 2 from v in the graph G, then any proposed move will be accepted by both processes in the coupling, or rejected by both processes. In either case the expected change in the distance is 0.

Putting these pieces together, we find

$$\mathrm{E}[\Delta\varphi(r, s)] \leq \frac{1}{kn}(-(k - d) + 2d) = \frac{3d - k}{kn}.$$

This gives us the bound

$$\tau(\epsilon) \leq \frac{\ln n\epsilon^{-1}}{1 - \frac{1}{kn}},$$

which lets us conclude $O(n \ln(n\epsilon^{-1}))$ mixing. □

2.2 Extensions

While coupling is a very attractive approach to bounding mixing times when there exists a distance metric that contracts during every step of the coupled chain, this is of course a lot to expect. There have been several observations that have allowed us to get more mileage out of this tantalizingly simple method. Some of the key concepts are briefly described here, but we refer the interested reader to the references provided for more detailed analysis.

<u>Choosing the right coupling:</u> Jerrum noticed that using a smarter coupling allows us to show M_{col} in fact mixes rapidly when $k > 2d$ [15]. The improvement stems from a careful look at the moves that potentially increase the distance between r and s, i.e., when the vertex we are updating is a neighbor of the vertex where the two colorings differ. The improved coupling pairs the choices (v, c) and (v, c') that are blocked because of the difference in colors at w, i.e., c in r and c' in s. This modification halves the number of potentially bad moves; see [15] for details.

Interestingly, this $2d$ bound was also found by Salas and Sokal in the context of "uniqueness of the Gibbs measure," a fundamental concept in the study of phase transitions of physical systems [33]. These two results were the first that suggested a fundamental relationship between phase transitions in physics and rapid vs. slow mixing of locally defined Markov chains.

Changing the Markov chain: Luby, Randall and Sinclair extended the utility of the coupling method by noticing that when changing the coupling or the distance metric is not enough, it might be better to simply modify the Markov chain itself [24]. In other words, by settling for a related Markov chain that converges to the same stationary distribution, it may be possible that a simple coupling can be used effectively to establish efficiency. They apply this idea in the context of of planar lattice problems, including colorings and matchings on simply connected regions in the 2-dimensional grid. The modified Markov chains include additional moves that update several sites in a configuration at once.

Subsequently, Randall and Tetali showed that the rapid mixing of the modified chains used by [24] implies rapid mixing of the original Glauber dynamics by a comparison theorem. We define the comparison method in section 4.1.

Macromoves: The latest trend in coupling is based on "macromoves" comprised of many steps of the coupled chain. The main idea is that if we wait until most of the sites in the lattice have been updated at least once, we can most likely avoid worst-case pairs of configurations that give pessimistic bounds on the coupling time. This was used in [9, 11, 30] where the chains were analyzed after an initial "burn-in" period that helped randomize the configurations before each step of the coupling.

Vigoda and Hayes make a substantial improvement by extending the macromove concept to the actual coupling phase. Instead of looking at the effects of coupling over a single step of the coupled chain, they couple over macromoves using a so-called "non-Markovian coupling." Using this idea, they achieve a best-possible bound of $O(n \log n)$ mixing of the Glauber dynamics for k-coloring when $k \geq (1+\epsilon)\Delta$ on graphs with girth ≥ 9 and degree $\geq c \log n$. Notice that for smaller k it might not even be possible to find a k-coloring. This improvement can be found in these proceedings [14].

3 Canonical paths and flows

In contrast to coupling, which localizes our analysis of a Markov chain to its behavior on pairs of configurations, the method of canonical paths and flows captures the global behavior of the chain. It demonstrates that slow (exponential time) mixing is characterized by a bottleneck, i.e., an exponentially small cut in the state space. It is not surprising that this is sufficient for slow mixing, since we can see that it will take exponential time to move from one side of the cut to the other; what is surprising is that it is also a necessary condition. To show that a chain is rapidly mixing, then, it is enough to show that there is no small cut. Equivalently, we can also show that there is sufficiently high flow across every cut. The rich method of canonical paths lets us argue this for an arbitrary cut.

3.1 Min cut

The conductance, introduced by Jerrum and Sinclair, is a reasonably good measure of the mixing rate of a chain [17]. For any set $S \subset \Omega$, let

$$\varphi_S = \frac{\sum_{x\in S, y\notin S} Q(x,y)}{\pi(S)},$$

where $Q(x,y) = \pi(x)P(x,y)$ is regarded as the "capacity" of the edge (x,y) and $\pi(S) = \sum_{x\in S}\pi(x)$ is the weight of the cutset. Note that by detailed balance $Q(x,y) = Q(y,x)$. We now define the *conductance* as

$$\Phi = \min_{S:\pi(S)\leq 1/2} \Phi_S.$$

It is clear that if a Markov chain has low conductance, then there is a bad cut in the state space that will cause a bottleneck in the mixing time. The following theorem establishes the converse as well.

Theorem 3.1 (Jerrum and Sinclair [17]) *For any Markov chain with conductance Φ and an eigenvalue gap $Gap(P) = 1 - |\lambda_1|$, we have*

$$\frac{\Phi^2}{2} \leq Gap(P) \leq 2\Phi.$$

Together with theorem 1.3 relating the spectral gap and the mixing time, this tells us that a Markov chain is rapidly mixing provided the conductance is not too small.

3.2 Max flow

It will be convenient to reformulate the idea of conductance in terms of flows, which also allows us to get a slightly sharper bound on the mixing rate. First, think of the graph

G with vertex set Ω and edges along all nonzero transitions, i.e., those (x, y) such that $P(x, y) > 0$. Since the conductance is just a minimum cut in G, we can naturally reinterpret it as a maximum flow along the edges.

For each ordered pair of distinct vertices $x, y \in \Omega$, we define a *canonical path* γ_{xy} in the graph G from x to y. Then, for any such collection of paths $\Gamma = \{\gamma_{xy} : x, y \in \Omega, x \neq y\}$, define the *congestion*

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

We can think of each path from x to y as carrying $\pi(x)\pi(y)$ units of flow. The congestion ρ then measures the maximum ratio of the total load routed through any edge e to its capacity $Q(e)$. Low congestion implies the absence of bottlenecks in the graph and we have just seen that this characterizes fast mixing. Let $\overline{\rho} = \min_\Gamma \rho(\Gamma)\ell(\Gamma)$, where $\ell(\Gamma)$ is the maximum length of a path in Γ.

Theorem 3.2 (Sinclair [34]) *For an ergodic, reversible Markov chain with stationary distribution π and self-loop probabilities $P(y, y) \geq \frac{1}{2}$ for all states $y \in X$, we have*

$$\tau_x(\epsilon) \leq \overline{\rho}\Big(\log \pi(x)^{-1} + \log \epsilon^{-1}\Big). \qquad \square$$

To demonstrate how to use this technique, we revisit the toy example of sampling a random vertex in a hypercube using MC_{cube} defined in section 2. We now need to establish paths $\gamma(x, y)$ between any pair of configurations x and y using edges of the Markov chain. The obvious choice is to walk through the bits of x and successively flip the bit whenever it differs in x and y. After at most n steps we will have visited all of the bits in x and we will reach y.

To determine the congestion $\rho(\Gamma)$, we consider an arbitrary edge $e = (u, v)$ on the hypercube. To bound $\rho(\Gamma)$, we first have to consider $\sum_{\rho(x,y) \ni e} \pi(x)\pi(y)$. Since the stationary distribution is uniform, all paths will be carrying the same load, so we can just count the number of paths that use (u, v).

Suppose that u and v differ only in the ith bit. How many paths can be routed through this edge? It is easy to see that the first $i + 1$ bits of v must agree with the end of the path y since we have already adjusted these as we flip the bits. On the other hand, we have not yet adjusted the final $n - i$ bits of v, so these must agree with x. Summing up, we have that $x = (x_1, ..., x_{i-1}, u_i, v_{i+1}, v_{i+1}, ..., v_n)$ and $y = (v_1, v_2, ..., v_i, y_{i+1}, ..., y_n)$ There are 2^{n-1} ways to assign the bits $x_1, ..., x_{i-1}, y_{i+1}, ..., y_n$, so this is exactly the number of paths that use the edge e. Hence

$$\sum_{\gamma_{xy} \ni e} \pi(x)\pi(y) = 2^{n-1}(2^{-2n}) = 2^{-(n+1)}.$$

We can also see that on this simple chain, $Q(e) = \pi(u)P(u, v) = \frac{1}{2^n}\frac{1}{2n}$ for every edge e. We can conclude that

$$\rho(\Gamma) = n2^{n+1} \cdot 2^{-(n+1)} = n.$$

Finally, since every path γ has length at most n, theorem 3.2 tells us

$$\tau(\epsilon) \leq n^2(n \ln 2 + ln\epsilon^{-1}).$$

Notice that for this example flows give a weaker bound than we were able to attain using coupling. The simplicity of the hypercube example and the relatively weak bound should not mislead you – for many important applications flows provide the best bounds to date. In the next section, for example, we explain how the ideas laid out here in the context of the hypercube can be extended to sampling matchings in a graph.

3.3 Extensions

We give a brief overview of some more interesting applications of flows and canonical paths to further demonstrate their significance.

Complementary paths: One of the first applications of canonical paths and flows was to analyze a Markov chain used to sample the set of matchings in a graph [17]. Given a constant $\mu > 0$, we will be interested in sampling from the distribution

$$\pi(M) = \frac{\mu^{|M|}}{Z},$$

where $Z = \sum_{M' \in \Omega} \mu^{|M'|}$ is the normalizing constant. The Markov chain MC_{match} updates a matching M at each step by choosing an edge $e = (u, v) \in E$ uniformly. Then, if $e \in M$, we remove it with probability $\min(1, \mu^{-1})$. If $e \notin M$ and both u and v are unmatched in M, we add it with probability $\min(1, \mu)$. If exactly one endpoint u or v is matched using an edge e', we remove e' and add e instead. Finally, if both u and v are matched, we do nothing. It is easy to verify that this Metropolis chain converges to the desired distribution.

This problem is much more interesting than the hypercube example for several reasons. First, the distribution is no longer uniform so we have to be careful to measure the amount of flow along each path. Second, we must be careful in this case how we define the paths to make sure that we always have a valid matching. Nonetheless, the analysis we set up for the hypercube is the main mechanism that is used here once the proper canonical paths are chosen.

Let x and y be any two matchings in G. If we take the symmetric difference $x \oplus y$, we will find a collection of alternating cycles and paths. We will order them in some fixed manner. To define the path from x to y, we take the first component and alternate removing edges from x and adding edges from y until the component is "processed." Then we move on to the next components, in order, and process them similarly. Like the hypercube example, at any intermediate point along this path, the components that we have already processed will agree with y and the components that we have not yet processed will agree with x. But how many paths actually pass through some particular edge (u, v)?

Here is an ingenious insight that lets us sketch the idea behind the more sophisticated use of the paths argument. Let us simultaneously consider a path that starts at y and ends at x. Notice that since $x \oplus y = y \oplus x$, this *complementary path* is working through the exact same set of alternating cycles and paths and in the same order. After roughly the same number of steps it took to pass through the edge (u, v), our complementary path will pass through an edge $e' = (u', v')$. However, on this edge the components we have already processed will agree with x and those we have not yet processed will agree with y. Intuitively, that means that from (u, v) and (u', v') we should be able to reconstruct x and y. Of course this assumes that we know the cycle structure of $x \oplus y$. But $u \oplus u'$ also tells us this information!

Our final concern should be making sure that we do not route too many paths that have large weight through edges with very small capacity. It turns out that the total number of edges in u and u' will always be very close to the number of edges in x and y. This is enough to get a polynomial bound on the congestion, and therefore the mixing time of the chain. These ideas are formalized in [17, 36].

Balanced flows: One variant of the method of canonical paths and flows is to allow the flow between x and y to be distributed along many paths to avoid congestion. Morris and Sinclair use a carefully chosen set of paths to analyze a chain on the set of walks on a hypercube truncated by a hyperplane and sampling from the set of feasible solutions to a 0/1 knapsack problem [31]. The paths from x to y they use are based on *balanced, almost uniform permutations* describing the order in which the bits of x are modified so as to agree with y. The simple one-path method where we fix the bits in some predetermined order will not work because we cannot guarantee that we stay in the state space now that the hypercube is truncated. Balanced almost uniform permutations spread the flow among a set of paths that are forced to stay within the state space. See [31] for definitions and details.

4 Auxiliary methods

When direct methods such as coupling and flows fail to provide good bounds on the mixing time of a Markov chain, indirect methods have proven quite useful. They allow us to analyze related chains instead and then infer the fast mixing of the chain in which we are interested from the fast mixing of the related chains. These theorems are most easily stated in terms of the spectral gap, so we refer back to theorem 1.3 relating the spectral gap and the mixing time.

4.1 Comparison

The comparison method tells us ways in which we can slightly modify one of these Markov chains without qualitatively changing the mixing time. This will also allow us to add additional transition edges or to amplify some of the transition probabilities, which can be quite useful.

Let $\widetilde{P}$ and P two reversible Markov chains on the same state space Ω with the same stationary distribution π. The comparison method (see [8] and [32]) allows us to relate the mixing times of these two chains. The idea is that the mixing time, $\tau_{\widetilde{P}}(\varepsilon)$, of $\widetilde{P}$ is known (or bounded) and we desire to obtain a bound for the mixing time, $\tau_P(\varepsilon)$, of P.

Let $E(P) = \{(x, y) : P(x, y) > 0\}$ and $E(\widetilde{P}) = \{(x, y) : \widetilde{P}(x, y) > 0\}$ denote the sets of edges of the two chains, viewed as directed graphs. For each x, y with $\widetilde{P}(x, y) > 0$, define a *path* γ_{xy} using a sequence of states $x = x_0, x_1, \ldots, x_k = y$ with $P(x_i, x_{i+1}) > 0$, and let $|\gamma_{xy}|$ denote the length of the path. Let $\Gamma(z, w) = \{(x, y) \in E(\widetilde{P}) : (z, w) \in \gamma_{xy}\}$ be the set of paths that use the transition (z, w) of P. Finally, define

$$A = \max_{(z,w) \in E(P)} \left\{ \frac{1}{\pi(z) P(z, w)} \sum_{\Gamma(z,w)} |\gamma_{xy}| \pi(x) \widetilde{P}(x, y) \right\}.$$

Theorem 4.1 (Diaconis and Saloff-Coste [8]) *With the above notation, we have* $Gap(P) \geq \frac{1}{A} Gap(\widetilde{P})$

Randall and Tetali show how to use the comparison method to get bounds on several natural Markov chains, including matchings on lattices and triangulations of convex point sets. See [32] for the details of these examples.

4.2 Decomposition

Madras and Randall introduced the *decomposition* method as a top down approach to analyzing mixing times [26]. Decomposition allows the state space to be broken down into pieces, and relates the mixing time of the original chain to the mixing times of the restricted Markov chains, which are forced to stay within each of the pieces, and a measure of the flow between these sets [26]. This method allows us to reduce the mixing of a complicated chain to the problem of bounding the mixing times of several much simpler chains. In addition, it allows us to attempt a hybrid approach towards analyzing the smaller pieces, perhaps using coupling to bound the restricted chains and canonical paths to bound the flow between the pieces. The version presented here is due to Martin and Randall and is based on a disjoint partition of the state space [27].

Suppose that the state space is partitioned into m disjoint pieces $\Omega_1, \ldots, \Omega_m$. For each $i = 1, \ldots, m$, define $P_i = P\{\Omega_i\}$ as the restriction of P to Ω_i which rejects moves that leave Ω_i. In particular, the restriction to A_i is a Markov chain, $\mathcal{M}_i$, where the transition matrix $P\{A_i\})$ is defined as follows: If $x \neq y$ and $x, y \in A_i$ then $P\{A_i\}(x,y) = P(x,y)$; if $x \in A_i$ then $P\{A_i\}(x,x) = 1 - \sum_{y \in A_i, y \neq x} P\{A_i\}(x,y)$. Let π_i be the normalized restriction of π to Ω_i, i.e., $\pi_i(A) = \frac{\pi(A \cap \Omega_i)}{b_i}$ where $b_i = \pi(\Omega_i)$.

Define $\overline{P}$ to be the following aggregated transition matrix on the state space $\{1, \ldots, m\}$:

$$\overline{P}(i,j) = \frac{1}{b_i} \sum_{\substack{x \in \Omega_i, \\ y \in \Omega_j}} \pi(x) P(x,y).$$

Theorem 4.2 (Martin and Randall [27]) *Let P_{Ω_i} and $\overline{P}$ be as above. Then*

$$Gap(P) \geq \frac{1}{2} Gap(\overline{P}) \min_{i=1,\ldots,m} Gap(P_i).$$

Decomposition has played a central role in several applications, e.g., [7, 26, 27].

5 Commonly studied models

Many of the combinatorial models that arise in the context of sampling fall under a common umbrella. Here we present a unifying framework that draws parallels between these models as they arise in computer science and statistical physics. The intimate connections with physics have provided a bilateral dialogue that has helped shape the design of good sampling algorithms, the methods used to analyze these algorithms, and even the intuition for when a Markov chain should be fast or slow. We start by restating several familiar models in the context of generating functions.

Independent sets: Let G be any graph and let Ω be the set of independent sets in G. We can think of each independent set as a map from V to $\{0,1\}$, where $f(v) = 1$ if v is in the independent set and $f(v) = 0$ otherwise. In addition, we can assign weights X_0 and X_1 to control the desirability of having a vertex in or out of an independent set, and define a weight

$$w(I) = \prod_{v \in V} X_{f(v)}.$$

Notice that when $X_0 = 1$ and X_1 is an integer, this corresponds to having X_1 particles at each vertex as candidates for the independent set. If we choose to sample independent sets according to this weight function, we get the following probability measure on Ω:

$$\pi(I) = \frac{w(I)}{\sum_{I' \in \Omega} w(I')}.$$

Letting $\lambda = X_1$, we find

$$\pi(I) = \frac{\lambda^{|I|}}{\sum_{I' \in \Omega} \lambda^{|I'|}}. \tag{2}$$

When λ is large we favor dense independent sets and when λ is small we favor sparse ones.

Colorings: We represent the set of k-colorings of a graph G using similar notation. Let $f : V \to \{1, ..., k\}$ and let Ω be the set of proper k-colorings of G. Let $X_0, ..., X_k$ be weights associated with each of the colors. For any $C \in \Omega$, let

$$w(C) = \prod_{v \in V} X_{f(v)} = \prod_{i=1}^{k} X_i^{c_i},$$

where c_i is the number of vertices in C that are colored with color i. When $X_i = 1$ for all i, this is the uniform distribution over proper k-colorings.

Matchings: Let $f : E \to \{0,1\}$ and let Ω be the set of matchings on a graph G of any size. As before, we let X_0 and X_1 be weights. Then for any $M \in \Omega$,

$$w(M) = \prod_{e \in E} X_{f(e)} = \mu^{|M|},$$

if we let $X_0 = 1$ and let $\mu = X_1$. When μ is integral, we see that $w(M)$ weights matchings as though G were a multigraph with μ parallel edges replacing each true edge. We find

$$\pi(M) = \frac{\mu^{|M|}}{\sum_{M' \in \Omega} \mu^{|M'|}}.$$

Pairwise influence models: The final model we consider is a generalization of the more familiar problem instances just mentioned and is based on pairwise interactions. For any n-vertex graph $G = (V, E)$, we let

Statistical Physics	Computer Science
monomer-dimer coverings	matchings
dimer coverings	perfect matchings
hard core lattice gas model	independent sets
spin	bit
ground states	highest probability configurations
ground states of the Potts model	vertex colorings
partition function	normalizing constant
connectivity	degree
activity or fugacity	vertex weight
interaction	edge weight
ferromagnetism	positive correlation
antiferromagnetism	negative correlation
mean-field	K_n
the Bethe lattice	the complete regular tree
polynomial mixing	rapid mixing
rapid mixing	$O(n \log n)$ mixing

Table 1: A lexicon of terms

$\Omega = \{1, ..., q\}^n$ where $f : V \to \{1, \dots q\}$ assigns a value from the set $\{1, ..., q\}$ to each vertex in the graph. We define a symmetric set of weights $\{X_{i,j} = X_{j,i}\}$ for each pair $i, j \in \{1, \dots, q\}$ and weight each configuration $\sigma \in \Omega$ by

$$w(\sigma) = \prod_{u,v:(u,v)\in E} X_{f(u),f(v)}.$$

Again,

$$\pi(\sigma) = \frac{w(\sigma)}{\sum_{\tau\in\Omega} w(\tau)}.$$

By adjusting the values for $X_{i,j}$ we can favor configurations such that the values on the endpoints of edges tend to agree, or disagree, and we can favor which assignments to vertices are preferred over all. For example, letting $X_{i,j} = 1$ for all $i \neq j$ and letting $X_{i,j} = 0$ whenever $i = j$, the probability distribution arising from the pairwise influence model is precisely the uniform distribution on the set of proper q-colorings.

A unifying framework: A minor change in notation lets us connect these problem instances to models well studied in statistical physics. This simple observation has allowed combinatorial and computational work on these models to flourish.

In statistical physics, models are defined to represent simple physical systems. Just like a spring relaxing, systems tend to favor configurations that minimize energy, and this preference is controlled by temperature. The energy function on the space of configurations is determined by a so-called *Hamiltonian* $H(\sigma)$. Since we are trying to minimize energy, we weigh configurations by

$$w(\sigma) = e^{-\beta H(\sigma)},$$

where $\beta = 1/T$ is inverse temperature. Thus, for low values of β the differences between the energy of configurations are dampened, while at large β these differences are magnified. The likelihood of each configuration is then

$$\pi(\sigma) = \frac{w(\sigma)}{Z},$$

where $Z = \sum_\tau w(\tau)$ is the normalizing constant known as the *partition function*. This is known as the *Gibbs (or Boltzmann) distribution.* Taking derivatives of the generating function Z (or $\ln Z$) with respect to the appropriate variables allows us to calculate many of the interesting thermodynamic properties of the system, such as the specific heat and the free energy.

For example, if we let our state space be the set of independent sets of a graph, then we let the Hamiltonian be

$$H(I) = -\sum_{v\in V} \delta_{v\in I} = -|I|,$$

where δ is the Kronecker delta that takes on value 1 if $v \in I$ and 0 otherwise. The probability distribution is given by

$$\pi(I) = e^{-\beta H(\sigma)}/Z,$$

where Z is the partition function. Setting $\lambda = e^{\beta}$, we see that this is precisely the same distribution given in equation 2. Note that the minus sign in the Hamiltonian that is immediately cancelled by the exponent of e is merely suggestive of the fact that we are trying to favor configurations of minimum energy. This model is known as the *hard core lattice gas model* in statistical physics under the premise that gas particles occupy area and two particles cannot be too close to each other. This model has what is known as a *hard constraint* because the probability of two particles occupying neighboring sites is zero.

Another common model from statistical physics is the *Ising model*, which is an example of a model with a *soft constraint* – certain configurations are given very small weight, but all configurations occur with positive probability. Given a graph G on n vertices, our state space is defined by the 2^n ways of assigning *spins* $+1$ or -1 to each of the vertices. The Hamiltonian is defined so as to favor configurations which tend to have equal spins on the endpoints of its edges. Hence, for $\sigma \in \{\pm 1\}^n$ we have

$$H(\sigma) = -\sum_{(u,v)\in E} \sigma_u \sigma_v = |D(\sigma)| - |A(\sigma)|$$

where $D(\sigma)$ is the number of edges $(u,v) \in E$ such that $\sigma_u \neq \sigma_v$ and $A(\sigma)$ is the number of edges (u,v) such that $\sigma_u = \sigma_v$. Then the Gibbs distribution is

$$\begin{aligned}\pi(\sigma) &= \frac{e^{-\beta H(\sigma)}}{Z} \\ &= \frac{e^{\beta(|E|)} e^{-2\beta|A(\sigma)|}}{\sum_{\tau\in\{0,1\}^n} e^{\beta(|E|)} e^{-2\beta|A(\tau)|}} \\ &= \frac{e^{-2\beta|A(\sigma)|}}{\sum_{\tau\in\{0,1\}^n} e^{-2\beta|A(\tau)|}}\end{aligned}$$

Notice that this is just a special case of the pairwise influence model with $q = 2$, where we set $X_{00} = X_{11} = e^{-2\beta}$ and $X_{01} = X_{10} = 1$. See [6] for a very nice introduction to the Ising model.

The pairwise influence model specializes to another interesting physics model for larger q. Taking $X_{i,j} = 1$ whenever $i \neq j$ and $X_{i,j} = \lambda$ whenever $i = j$ gives us a more general system known as the *Potts model*. When $\lambda > 1$ we have the *ferromagnetic* case where positive correlations on edges are rewarded, and when $\lambda < 1$ we are in the *antiferromagnetic* case where negative correlations are.

Table 1 explains some of the common terms that are used in statistical physics, occasionally taking some liberties with the translations.

References

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

[2] D. Aldous and J. Fill. Reversible Markov chains and Random Walks on Graphs. In preparation, 2003.

[3] I. Beichl and F. Sullivan. The Metropolis Algorithm. *Computing in Science and Engineering*, **2**: 65–69, 2000.

[4] A. Broder. Generating random spanning trees. em Proc. 30th IEEE Symp. on Foundations of Computer Science, 442–447, 1989.

[5] R. Bubley and M. Dyer. Faster random generation of linear extensions. *Discrete Mathematics*, **201**:81–88, 1999.

[6] B.A. Cipra. An introduction to the Ising model. *American Mathematical Monthly* **94**, pp. 937–959, 1987.

[7] C. Cooper, M.E. Dyer, A.M. Frieze and R. Rue. Mixing Properties of the Swendsen-Wang Process on the Complete Graph and Narrow Grids. *J. Math. Phys.* **41**, 1499–1527, 2000.

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

[9] M.E. Dyer and A. Frieze. Randomly colouring graphs with lower bounds on girth and maximum degree. *Proc. 42nd Symp. on Foundations of Computer Science,* 579–587, 2001.

[10] M.E. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of a convex body. *Proc. 24th ACM Symp. on Theory of Computing*, 26–38, 1992.

[11] M.E. Dyer, L.A. Goldberg, C. Greenhill, M.R. Jerrum, and M. Mitzenmacher. An extension of path coupling and its application to the Glauber dynamics for graph colorings. *SIAM Journal on Computing*, **30**: 1962–1975, 2001.

[12] M. Dyer and C. Greenhill. A more rapidly mixing Markov chain for graph colorings. *Random Structures and Algorithms*, **13**:285–317, 1998.

[13] W. Feller. *An introduction to probability theory and its applications, Volume I.* Wiley, New York, 1968.

[14] T.P. Hayes and E. Vigoda. A non-Markovian coupling for randomly sampling colorings. *Proc. 44th Symp. on Foundations of Computing*, 2003.

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

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

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

[18] M.R. Jerrum and A.J. Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. In *Approximation Algorithms for NP-Hard Problems,* D.S. Hochbaum, ed., PwS Publishing, Boston: 482–520, 1997.

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

[20] M.R. Jerrum, A.J. Sinclair and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Proc. 33rd ACM Symp. on Theory of Computing*, 712-721, 2001.

[21] 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.

[22] R. Kannan. Markov chains and polynomial time algorithms. *Proc. 35th IEEE Symp. on Foundations of Computer Science,* 656–671, 1994.

[23] L. Lovsz and P. Winkler. Mixing times. *Microsurveys in Discrete Probability*, D. Aldous and J. Propp, eds., *DIMACS Series in Discrete Math. and Theoretical Computer Science,* **41**: 85-134, 1998.

[24] M. Luby, D. Randall, and A.J. Sinclair. Markov Chains for Planar Lattice Structures. *SIAM Journal on Computing*, **31**: 167–192, 2001.

[25] M. Luby and E. Vigoda. Fast convergence of the Glauber dynamics for sampling independent sets. *Random Structures and Algorithms*, 229–241, 1999.

[26] N. Madras and D. Randall. Markov chain decomposition for convergence rate analysis. *Annals of Applied Probability*, **12**: 581–606, 2002.

[27] R.A. Martin and D. Randall. Sampling adsorbing staircase walks using a new Markov chain decomposition method. *Proceedings of the 41st IEEE Symp. on Foundations of Computer Science*, 492–502, 2000.

[28] R.A. Martin and D. Randall. Disjoint decomposition with applications to sampling circuits in some Cayley graphs. Preprint, 2003.

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

[30] M. Molloy. The Glauber dynamics on colorings of a graph with high girth and maximum degree. *Proc. 34th ACM Symp. on Theory of Computing,* 91–98, 2002.

[31] B. Morris and A. Sinclair. Random walks on truncated cubes and sampling 0/1 knapsack solutions. *Proc. 40th IEEE Symp. on Foundations of computer Science,* 230–240, 1999.

[32] D. Randall and P. Tetali. Analyzing Glauber dynamics by comparison of Markov chains. *Journal of Mathematical Physics*, **41**:1598–1615, 2000.

[33] J. Salas and A. Sokal. Absense of phase transitions for antiferromagnetic Potts models via the Dobrushin uniqueness theorem. *Journal of Statistical Physics,* **86**: 551–579, 1997.

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

[35] A.J. Sinclair. Convergence rates for Monte Carlo experiments. *Numerical Methods for Polymeric Systems,* S.G. Whittington, ed., *IMA Volumes in Mathematics and its Applications,* 1-18, 1997.

[36] A.J. Sinclair. *Algorithms for random generation & counting: a Markov chain approach.* Birkhäuser, Boston, 1993.

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

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