

Rapidly Mixing Markov Chains with Applications in Computer Science and Physics

Monte Carlo algorithms often depend on Markov chains to sample from very large data sets. A key ingredient in the design of an efficient Markov chain is determining rigorous bounds on how quickly the chain “mixes,” or converges, to its stationary distribution. This survey provides an overview of several useful techniques.

Markov chain Monte Carlo algorithms are ubiquitous across scientific disciplines as a computational means for studying large, complicated data sets. The idea is to simulate a random walk that moves among configurations in a large set. Even though each configuration might have only 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 past 15 years, a flurry of activity has led to breakthroughs in our understanding of how to provide rigorous bounds on the convergence rates of Markov chains and ultimately design provably

efficient sampling algorithms. This tutorial is intended to give an introduction to some of the key ideas underlying these results; for additional information, there are many useful surveys.¹⁻⁴

Two of the most notable success stories based on these rigorous methods are estimating the volume of a convex body⁵ and estimating the permanent of a matrix.^{1,6,7} 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;⁸ this is precisely the problem of calculating the permanent of the adjacency matrix of a bipartite graph. 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. Jerrum, Valiant, and Vazirani established the intimate connection between random sampling and approximate counting for a wide class of problems known as *self-reducible*.⁹ In this introduction to Markov chains, we will concentrate solely on the sampling aspects of Monte Carlo experiments, forgoing the many beautiful applications of sampling, including approximate counting.

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

1521-9615/06/\$20.00 © 2006 IEEE
Copublished by the IEEE CS and the AIP

DANA RANDALL
Georgia Institute of Technology

is first to connect the state space so that each configuration has a small number of nearest neighbors. Then, starting at some arbitrary initial 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. A familiar example of this is opening a brand-new deck of cards, so the initial configuration is completely ordered, and then performing a series of random shuffles until the deck is well mixed. Another example is a tourist wandering aimlessly in an unknown city, making random decisions about how to proceed at each intersection.

The sampling problems considered here are motivated by statistical physics and have simple combinatorial descriptions. Let us consider, for example, how to design a chain for randomly sampling independent sets of some finite graph. An independent set on a graph $G = (V, E)$ is a subset of vertices $I \subseteq V$ such that no two vertices in I are connected by an edge in E . When the graph is a region of the two-dimensional Cartesian lattice, an independent set is a configuration of the hard-core lattice gas model from physics in which molecules (vertices) have non-negligible mass that precludes any two from occupying adjacent sites in the lattice.

One simple way to connect the state space to define a Markov chain is to allow transitions between independent sets I and I' that have Hamming distance $\varphi(I, I') \leq 1$ (that is, the number of vertices in which the two independent sets differ is at most 1). The Markov chain starts at some initial state I_0 —say, the empty set—and we denote the independent set at time t by I_t . During each step of the simulation, the chain proceeds by choosing a vertex v uniformly from V ; if $v \in I_t$, then we remove it and let $I_{t+1} = I_t \setminus \{v\}$, whereas if $v \notin I_t$, we can add it and set $I_{t+1} = I_t \cup \{v\}$, provided this leads to a valid independent set. In the case that we can't 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

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

Chains like this, which make only local changes during each move of the chain, are commonly referred to as *Glauber dynamics*.¹⁰ Notice that $P^t(x, y)$ is then the probability of being at state y after t steps of the chain if we start at x . The following definition and lemmas formalize why this is a good candidate chain.

Definition 1: A Markov chain is *ergodic* if it is

1. *irreducible*, that is, for all $x, y \in \Omega$, there is a t such that $P^t(x, y) > 0$, and
2. *aperiodic*, that is, for all $x, y \in \Omega$, $\gcd\{t: P^t(x, y) > 0\} = 1$, where \gcd is the greatest common divider.

We can see that our chain on independent sets is irreducible because we can get from any configuration to the empty independent set by successively removing vertices, and hence we can move between any two arbitrary configurations. Moreover, the chain is aperiodic because most configurations have self-loop probabilities—that is, moves that keep the configuration unchanged (these occur whenever vertices can't be added without violating the independence requirement).

The sampling problems considered here are motivated by statistical physics and have simple combinatorial descriptions.

For chains that aren't aperiodic, self-loops can be added at every vertex at a small constant cost to the algorithm's running time. Such 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 Markov chain.

Lemma 1: Any finite, ergodic Markov chain converges to a unique stationary distribution π —that is, for all $x, y \in \Omega$, we have that $\lim_{t \rightarrow \infty} P^t(x, y) = \pi(y)$.

To reason about the limiting probability distribution, known as the *stationary distribution*, we rely on the *detailed balance condition* given in Lemma 2. Any chain that satisfies detailed balance for some π is called *time-reversible*.

Lemma 2: Let M be an ergodic Markov chain on a finite-state state space Ω with transition probabilities $P(\cdot, \cdot)$. If $\pi: \Omega \rightarrow [0, 1]$ 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 .

For the independent set chain, the transition probabilities are symmetric—that is, $P(I, I') = P(I', I)$ for all $I, I' \in \Omega$. It follows from Lemma 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 to sample from a more complicated distribution? Second, for how long do we have to simulate the walk before we can trust that our samples are chosen from a distribution that is very close to the stationary distribution? The celebrated Metropolis algorithm gives a standard way of approaching the first of these questions, and is simply based on a careful consideration of Lemma 2. The second question is the subject of the remaining part of this article and requires much more sensitive consideration.

The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm¹¹—a remarkably simple, yet tremendously robust idea—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 chosen distribution. In 2000, *Computing in Science & Engineering* selected it as one of the 10 most important algorithms.¹²

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 number of neighbors of any vertex.
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's easy to verify that if the state space is connected, then π must be the stationary distribution. Often, an additional factor of $1/2$ is added to step 2 to ensure that the chain is aperiodic at a cost of slowing the chain slightly. This is known as the *lazy chain*.

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

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

where $Z = \sum_{I \in \Omega} \lambda^{|I|}$ is the normalizing constant. We'll justify why this is a natural weighting toward the end of the article. 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 . Because $|I'| = |I| + 1$, 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

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

Notice that the normalizing constant Z drops out of the equation! This is quite fortuitous because we typically don't have a 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 the chain to converge to the correct stationary distribution.

This leaves our main question: how long do we have to simulate the chain to get a provably good sample?

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 *total variation distance* between the distribution at time t and the stationary distribution. A comparison of rates of convergence based on different measures of distance appears elsewhere.^{3,13}

Definition 2: 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 3: For $\epsilon > 0$, the mixing time $\tau(\epsilon)$ is

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

We say a Markov chain is *rapidly mixing* if the mixing time is bounded 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 spectral gap of the Markov chain's transition matrix provides a good bound on the mixing rate of a chain. In particular, if we let $\lambda_0, \lambda_1, \dots, \lambda_{|\Omega|-1}$ be the eigenvalues of the transition matrix P , where

$1 = \lambda_0 > |\lambda_1| \geq |\lambda_i|$ for all $i \geq 2$, then the *spectral gap* is $\text{Gap}(P) = 1 - |\lambda_1|$. Theorem 1 relates the spectral gap with the chain's mixing time.⁹

Theorem 1: Let $\pi_* = \min_{x \in \Omega} \pi(x)$. For all $\varepsilon > 0$, we have

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

and

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

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 $\{\lambda_i\}$ of the lazy chain will satisfy $\lambda_i = (1 + \lambda_i)/2$ and $|\lambda_i| \leq 1$ for all i .

Although 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—we typically don't have a compact, mathematical representation for the adjacency matrix, so it's 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.

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 trees¹⁴ and has since seen many more applications.

Definition 4: 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 means that each process, viewed in isolation, is just simulating the probabilities as prescribed by 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 agrees, 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 5: For initial states x, y , let

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

and let $ET^{x,y}$ denote its expectation. Then we define the coupling time to be $T = \max_{x,y} ET^{x,y}$.

Theorem 2 relates the mixing time and the coupling time.¹⁵

Theorem 2: The mixing time satisfies $\tau(\varepsilon) \leq \lceil T\varepsilon \ln e^{-1} \rceil$, where e is the base of the natural logarithm.

One of the most popular methods for bounding mixing times is coupling, both because of its elegance and its simplicity.

Let's consider a toy problem of choosing a random vertex in the n -dimensional hypercube, $\Omega = \{0, 1\}^n$. A natural Markov chain MC_{cube} performs a simple random walk along the edges of the hypercube by iteratively flipping a randomly chosen bit. However, we consider instead the so-called "lazy" version of the chain, which keeps the configuration unchanged with probability 1/2 at each move of the chain. We need to add some self-loop probabilities because the hypercube is bipartite and won't be aperiodic unless we add self-loop probabilities. The lazy chain also turns out to be more conducive to a coupling argument.

MC_{cube}

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

1. Pick $(i, b) \in \{1, \dots, n\} \times \{0, 1\}$.
2. Let X_{t+1} be X_t with the i th 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's 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 update each bit at least once. By the coupon collector's theorem, this takes $O(n \ln n)$ steps, in expectation. Appealing to Theorem 3, we have a bound on the mixing time. The logarithmic dependence on e^{-1} is typical for mixing rates and the $O(n \ln n)$ is optimal.

Theorem 3: The Markov chain MC_{cube} has mixing time $\tau(e) = O(n \ln(ne^{-1}))$.

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 performed by the random variable representing the distance. If we show that distance is merely decreasing in expectation, then we have a drift toward zero that still allows us to prove a chain is rapidly mixing. Typically, as the distance approaches zero, fewer moves will decrease the distance, so the coupon-collecting theorem suggests we should again expect an $O(n \ln n)$ coupling time. Next, we'll see a more realistic example in the context of sampling k -colorings that 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.

Path Coupling

Although 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 according to some metric on the state space.¹⁶ Often the metric used is the Hamming distance measuring the number of places in which two configurations differ. Because close configurations tend to 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 we've shown that the expected change in distance is decreasing for all the pairs in U . To now reason about arbitrary configurations $X, Y \in \Omega$, we define a shortest path $z_0 = X, z_1, \dots, z_r = Y$ of length r from X to Y , where

$(z_i, z_{i+1}) \in U$ for all $0 \leq i < r$. If U is defined appropriately, then $\varphi(X, Y) = \sum_{i=0}^{r-1} \varphi(z_i, z_{i+1})$, and we're finished: by linearity of expectation, the expected change in distance between X and Y along this path 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 formalizes the intuition.

Theorem 4: (Dyer and Greenhill¹⁷) Let φ be an integer-valued metric defined on $\Omega \times \Omega$, which takes values in $\{0, \dots, B\}$. 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 $x_t = z_0, z_1, \dots, z_r = y_t$ 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 $f: \Omega \rightarrow \Omega$ such that $\Pr[f(x) = y] = P(x, y)$ for all $x, y \in \Omega$, and define a coupling of the Markov chain by $(x_t, y_t) \rightarrow (x_{t+1}, y_{t+1}) = (f(x_t), f(y_t))$.

1. If there exists $\beta < 1$ such that $E[\varphi(x_{t+1}, y_{t+1})] \leq \beta \varphi(x_t, y_t)$, for all $(x_t, y_t) \in U$, then the mixing time satisfies

$$\tau(e) \leq \frac{\ln(Be^{-1})}{1 - \beta}.$$

2. If $\beta = 1$ (that is, $E[\Delta\varphi(x_t, y_t)] \leq 0$, for all $x_t, y_t \in U$), let $\alpha > 0$ satisfy $\Pr[\varphi(x_{t+1}, y_{t+1}) \neq \varphi(x_t, y_t)] \geq \alpha$ for all t such that $x_t \neq y_t$. The mixing time of \mathcal{M} then satisfies

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

We now demonstrate the technique of path coupling on a Markov chain MC_{col} for sampling k -colorings of a graph G . This local chain is another example of Glauber dynamics.

MC_{col}

Starting at t_0 , repeat t times:

1. With probability $1/2$, do nothing.
2. Pick $(v, c) \in V \times \{1, \dots, k\}$.
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 has been used.¹⁵ Theorem 5 demonstrates the simplicity of the technique.

Theorem 5: The Markov chain MC_{col} on k -colorings has mixing time $\tau(e) = O(n \ln(ne^{-1}))$ 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 $(v, c) \in v \times \{1, \dots, k\}$ 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 $\varphi: \Omega \times \Omega \rightarrow \mathbb{Z}$ 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 $x = z_0, z_1, \dots, z_\ell = y$ 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 4, we need to consider $E[\Delta\varphi(r, s)]$ for any $(r, s) \in U$. Suppose w is the vertex that is colored differently in r and s . Let's consider three cases:

- Case 1: $w = v$. If $w = v$, then any color c not currently used by any of the neighbors of w in r and s will be a move accepted by both processes, in which case r and s will agree in the next step. At least $k - d$ such colors exist. If, on the other hand, c agrees with a color used by at least one of the neighbors of w in the two configurations, then both processes will reject the move. Together, this tells us that if $w = v$, then $E[\Delta\varphi(x_t, y_t)] \leq -(k - d)/kn$.
- Case 2: $(w, v) \in E$. 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 and the distance can increase; it can, of course, be rejected by both, in which case r and s remain unchanged and at distance 1. There are at most two bad choices for c , so $E[\Delta\varphi(r, s)] \leq 2/kn$. This

bound holds for each of the d neighbors of w .

- Case 3: $w \neq v$ and $(w, v) \in E$. 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

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

which is negative when $k \geq 3d + 1$. This gives us the bound

$$\tau(e) \leq \frac{\ln ne^{-1}}{1 - \frac{1}{kn}},$$

hence, Theorem 4 gives us $O(n \ln(ne^{-1}))$ mixing time. **QED**

Although coupling is a very attractive technique when there exists a distance metric that contracts in expectation during every step of the coupled chain, this is, of course, a lot to expect. Several observations have allowed us to get more mileage out of this tantalizingly simple method. Jerrum, for example, noticed that using a smarter coupling allows us to show that MC_{col} in fact mixes rapidly when $k > 2d$.¹⁸ The improvement stems from a careful look at the moves that potentially increase the distance between r and s ; this is precisely when the vertex we are updating is a neighbor of the vertex in which the two colorings differ. The improved coupling pairs the choices (v, c) and (v, c') that are blocked because of the difference in the color of w in r and s . This modification halves the number of potentially bad moves. Alternatively, we can sometimes change the distance function or even the Markov chain to facilitate a coupling proof. Luby, Randall, and Sinclair modified a natural Markov chain to sample tilings of certain lattice regions and three-colorings of the two-dimensional grid.¹⁹

More recent extensions of coupling include analyzing "macromoves" consisting of several steps of the Markov 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.²⁰⁻²³

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 chain's global behavior. It demonstrates that slow (exponential time) mixing is characterized by a "bottleneck," which is a set of edges that together have small stationary probability and whose removal disconnects the state space into two sets of exponentially larger size (also known as a "small cut"). It is not surprising that the presence of a bottleneck implies slow mixing because 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's the only obstacle. To show that a chain is rapidly mixing, then, it's enough to show that there is no small cut. The rich method of canonical paths provides a tool to argue this for an arbitrary cut.

Min Cut

The conductance, introduced by Sinclair and Jerrum, is a reasonably good measure of the mixing rate of a chain.²⁴ For any set $S \subset \Omega$, let

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

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 cut set. Note that by detailed balance $Q(x, y) = Q(y, x)$. We now define the conductance as

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

If a Markov chain has low conductance, then a small cut in the state space will cause a bottleneck in the mixing time. Theorem 6 establishes the converse as well.

Theorem 6: For any Markov chain with conductance Φ , we have

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

where $\text{Gap}(P) = 1 - |\lambda_1|$ is the spectral gap.²⁴

Together with Theorem 1 relating the gap and the mixing time, this tells us that a Markov chain is rapidly mixing provided the conductance isn't too small.

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 transitions (x, y) , such that $P(x, y) > 0$. Because the conductance is just a minimum cut in g , we can naturally reinterpret it as a maximum flow

along the edges, appealing to the duality between min cut and max flow.

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

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

We can think of each path from x to y as carrying $\pi(x)\pi(y)$ units of flow. The congestion ρ 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've just seen that this characterizes fast mixing.

Let $\bar{\rho} = \min_{\Gamma} \rho(\Gamma)\ell(\Gamma)$, where $\ell(\Gamma)$ is the maximum length of a path in Γ .

Theorem 7: (Sinclair²⁵) For an ergodic, reversible Markov chain with stationary distribution π and self-loop probabilities $P(y, y) \geq 1/2$ for all states $y \in X$, we have

$$\tau_x(e) \leq \bar{\rho}(\log \pi(x)^{-1} + \log e^{-1}).$$

To demonstrate how to use this technique, we revisit the toy example of sampling a random vertex in a hypercube using the chain MC_{cube} defined earlier. We now need to establish paths $\gamma(x, y)$ between any pair of vertices x and y using edges of the hypercube. The obvious choice is to walk through the bits of x and successively flip any bit that 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 $\rho(\Gamma)$ defined in Equation 1, we consider an arbitrary edge $e = (u, v)$ on the hypercube. To bound $\rho(\Gamma)$, we first have to consider $\sum_{\rho(x, y) \ni e} \pi(x)\pi(y)$. Because the stationary distribution is uniform, all paths will be carrying the same load, so we just have to count the number of paths that use (u, v) .

Suppose that $(u, v) \in E$ and u and v differ in the i th bit. How many paths can be routed through this edge? It's easy to see that the first $i + 1$ bits of v must agree with the end of the path y because we've already adjusted these as we flip the bits. On the other hand, we haven't yet adjusted the final $n - i$ bits of v , so these must agree with x . Summing up, we have that $x = (x_1, \dots, x_{i-1}, u_i, v_{i+1}, v_{i+1}, \dots, v_n)$ and $y = (v_1, v_2, \dots, v_i, y_{i+1}, \dots, y_n)$. There are 2^{n-1} ways to assign the bits $x_1, \dots, x_{i-1}, y_{i+1}, \dots, y_n$, 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, $Q(e) = \pi(u)P(u, v) = 1/2^n \cdot 1/2n$ for every edge e . Hence,

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

Finally, because every path γ has length at most n , Theorem 7 tells us

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

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 be misleading—for many important applications flows provide the best bounds to date. In the next section, for example, we see how the ideas laid out here in the context of the hypercube can be extended to sampling matchings in a graph.

Complementary Paths

One of the first applications of canonical paths and flows was analyzing a Markov chain used to sample the set of matchings in a graph.²⁴ 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's easy to verify that this Metropolis chain converges to the desired distribution.

This problem is much more challenging than the hypercube example for several reasons. First, the distribution isn't uniform, so we now 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 or we won't be in state space. Nonetheless, the analysis we set up for the hypercube is the main mechanism 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 find a collection of alternating cycles and paths in G . We will order them in some fixed manner. To define the

canonical 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 what is the total flow 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's simultaneously consider a path that starts at y and ends at x . Notice that because $x \oplus y = y \oplus x$, 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 reach the edge (u, v) , our complementary path will reach an edge $e' = (u', v')$. However, on this edge, the components we've already processed will agree with x and those we have not yet processed will agree with y . Intuitively, this 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 don't 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 elsewhere.^{9,24}

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. We merely outline the general techniques here, and refer the interested reader to the cited papers for details of the theorems.

Comparison

The comparison method of Diaconis and Saloff-Coste²⁶ tells us ways in which we can slightly modify a Markov chain without qualitatively changing the mixing time. This will also allow us to add ad-

Table 1. A lexicon of terms.

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	Complete graph K_n
Bethe lattice	Complete regular tree
Polynomial mixing	Rapid mixing
Rapid mixing	$O(n \log n)$ mixing

ditional transition edges or to amplify some of the transition probabilities, which has proven to be a useful tool.

Let \tilde{P} and P represent two reversible Markov chains on the same state space Ω with the same stationary distribution π . The comparison method allows us to relate the mixing times of these two chains. The idea is that the mixing time, $\tau_{\tilde{P}}(\epsilon)$, of \tilde{P} is known (or bounded) and we desire to obtain a bound for the mixing time, $\tau_P(\epsilon)$, of P . For each pair of configurations x, y such that $\tilde{P}(x, y) > 0$, we construct a canonical path from x to y using moves in P . This allows us to relate the congestion of the two chains.

This approach has been used to get good bounds on several natural Markov chains arising in computer science, such as matchings on lattices and triangulations of convex point sets.²⁷

Decomposition

Madras and Randall²⁸ introduced the *decomposition* method as a top-down approach to analyzing mixing times. Decomposition allows the state space to be broken down into pieces; it relates the mixing time of the original chain to the mixing times of restricted Markov chains, each of which is forced to stay within one of the pieces, and a measure of the flow between these sets. This method tells us how to reduce the problem of bounding the mixing of a complicated chain to that of bounding the mixing times of several much simpler chains. In addition, it allows us to attempt a hybrid approach toward

analyzing the smaller pieces, perhaps using coupling to bound the restricted chains and canonical paths and thus bound the flow between the pieces.

Martin and Randall²⁹ presented a related decomposition theorem based on a disjoint partition of the state space that seems more conducive to applications, and this has been further generalized by Jerrum et al.³⁰ Decomposition has played a central role in several applications in statistical physics and computing.^{28,29,31,32}

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

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 $\lambda \in \mathbf{R}^+$ 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 $f: V \rightarrow \{1, \dots, k\}$, let Ω be the set of proper k -colorings of G , and let X_0, \dots, 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 colored with color i . When $X_i = 1$ for all i , this is the uniform distribution over proper k -colorings.

Matchings

Let $f: E \rightarrow \{0, 1\}$ 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 $M \in \Omega$,

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

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 problem instances just mentioned and is based on pairwise interactions. For any n -vertex graph $G = (V, E)$, we let $\Omega = \{1, \dots, q\}^n$, where $f: V \rightarrow \{1, \dots, q\}$ assigns a value from the set $\{1, \dots, q\}$ to each vertex in the graph. We define a symmetric set of weights $\{X_{ij} = X_{ji}\}$ 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_{ij} we can favor configurations such that the values on the endpoints of edges tend to agree or disagree. For ex-

ample, letting $X_{ij} = 1$ for all $i \neq j$ and letting $X_{ij} = 0$ 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 $H(\sigma)$. Because we’re trying to minimize energy, we assign configurations weights

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

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

$$\pi(\sigma) = w(\sigma)/Z,$$

where $Z = \sum_{\tau} w(\tau)$ is the normalizing constant known as the partition function. This distribution π 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 specific heat and free energy.

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

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

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

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

where Z is the partition function. Setting $\lambda = e^\beta$, we see that this is precisely the same distribution

given earlier 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're 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 can't be very close. 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, 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. In the ferromagnetic Ising model, the Hamiltonian is defined so as to favor configurations that tend to have equal spins on the endpoints of its edges; the antiferromagnetic Ising model favors configurations with unequal spins on the endpoints of its edges.

In the ferromagnetic case, 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

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

where $Z = \sum_{\tau \in \{0,1\}^n} e^{-\beta H(\tau)}$. Rewriting this expression, we have


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

Notice that this is a special case of the pairwise influence model with $q = 2$, $X_{00} = X_{11} = e^{-2\beta}$ and $X_{01} = X_{10} = 1$.

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 favoring positive correlations on edges, and

when $\lambda < 1$, we have the antiferromagnetic case favoring negative correlations.

The physics interpretation for each of these familiar models reveals deeper insights into their structure. For each, we expect fast mixing at high temperature and slow mixing at low temperature. For instance, consider the independent set model on any bipartite graph. At sufficiently high temperature, independent sets will tend to be sparse, and we can use a coupling argument to show that Glauber dynamics (the local chain) is rapidly mixing. At sufficiently low temperature, independent sets will be dense and will lie predominantly on either side of the bipartition, revealing an exponentially small cut in the state space.

In addition to helping us appreciate the dichotomy underlying the behavior of Markov chains based on local updates, the physical interpretation can also aid our design of better sampling algorithms. It's known, for example, that locally defined chains are not efficient for sampling Ising configurations at low temperature, so this suggests that we should look for alternate algorithms based on nonlocal chains. Jerrum and Sinclair found such an algorithm for estimating the partition function of any Ising model at any temperature,³⁴ and Randall and Wilson showed that this can always be used to sample Ising configurations at any temperature, even when there is an external magnetic field.³⁵ These results demonstrate that the interplay between statistical physics and computer science can bring to light new ideas in the world of sampling, and suggest many promising directions for future research. 

Acknowledgments

An earlier version of this article appeared in the *Proceedings of the 44th IEEE Symposium on the Foundations of Computer Science* (IEEE CS Press, 2003, pp. 4–15). This work was supported in part by the US National Science Foundation under grants CCR-0515105 and DMS-0505505.

References

1. M.R. Jerrum and A.J. Sinclair, "The Markov Chain Monte Carlo Method: An Approach to Approximate Counting and Integration," *Approximation Algorithms for NP-Hard Problems*, D.S. Hochbaum, ed., PWS Publishing, 1997, pp. 482–520.
2. R. Kannan, "Markov Chains and Polynomial Time Algorithms," *Proc. 35th IEEE Symp. Foundations of Computer Science*, IEEE CS Press, 1994, pp. 656–671.
3. L. Lovász and P. Winkler, "Mixing Times," *Microsurveys in Discrete Probability*, vol. 41, DIMACS Series in Discrete Mathematical and Theoretical Computer Science, D. Aldous and J. Propp, eds., Am. Mathematical Soc., 1998, pp. 85–134.

4. 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, Am. Mathematical Soc., 1997, pp. 1–18.
5. 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. Theory of Computing*, ACM Press, 1992, pp. 26–38.
6. A.J. Sinclair, *Algorithms for Random Generation and Counting: A Markov Chain Approach*, Birkhauser, 1993.
7. 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. Theory of Computing*, ACM Press, 2001, pp. 712–721.
8. L.G. Valiant "The Complexity of Computing the Permanent," *Theoretical Computer Science*, vol. 8, 1979, pp. 189–201.
9. M.R. Jerrum, L.G. Valiant, and V.V. Vazirani, "Random Generation of Combinatorial Structures from a Uniform Distribution," *Theoretical Computer Science*, vol. 43, 1986, pp. 169–188.
10. R. Glauber, "Time Dependent Statistics of the Ising Model," *J. Mathematical Physics*, vol. 4, 1963, pp. 294–307.
11. N. Metropolis et al., "Equation of State Calculations by Fast Computing Machines," *J. Chemical Physics*, vol. 21, 1953, pp. 1087–1092.
12. I. Beichl and F. Sullivan, "The Metropolis Algorithm," *Computing in Science & Eng.*, vol. 2, no. 1, 2000, pp. 65–69.
13. D. Aldous and J. Fill, "Reversible Markov Chains and Random Walks on Graphs," in preparation, 2005; www.stat.berkeley.edu/users/aldous/RWG/book.html.
14. A. Broder, "Generating Random Spanning Trees," *Proc. 30th IEEE Symp. Foundations of Computer Science*, IEEE CS Press, 1989, pp. 442–447.
15. D. Aldous, "Random Walks on Finite Groups and Rapidly Mixing Markov Chains," *Seminaire de Probabilités XVII*, Lecture Notes in Mathematics, vol. 986, 1981/82, pp. 243–297.
16. R. Bubley and M. Dyer, "Faster Random Generation of Linear Extensions," *Discrete Mathematics*, vol. 201, 1999, pp. 81–88.
17. M. Dyer and C. Greenhill, "A More Rapidly Mixing Markov Chain for Graph Colorings," *Random Structures and Algorithms*, vol. 13, 1998, pp. 285–317.
18. M.R. Jerrum, "A Very Simple Algorithm for Estimating the Number of k -Colorings of a Low-Degree Graph," *Random Structures and Algorithms*, vol. 7, 1995, pp. 157–165.
19. M. Luby, D. Randall, and A.J. Sinclair, "Markov Chains for Planar Lattice Structures," *SIAM J. Computing*, vol. 31, 2001, pp. 167–192.
20. M.E. Dyer and A. Frieze, "Randomly Colouring Graphs with Lower Bounds on Girth and Maximum Degree," *Proc. 42nd Symp. Foundations of Computer Science*, 2001, pp. 579–587.
21. M.E. Dyer et al., "An Extension of Path Coupling and Its Application to the Glauber Dynamics for Graph Colorings," *SIAM J. Computing*, vol. 30, 2001, pp. 1962–1975.
22. M. Molloy, "The Glauber Dynamics on Colorings of a Graph with High Girth and Maximum Degree," *Proc. 34th ACM Symp. Theory of Computing*, ACM Press, 2002, pp. 91–98.
23. T.P. Hayes and E. Vigoda, "A Non-Markovian Coupling for Randomly Sampling Colorings," *Proc. 44th IEEE Symp. Foundations of Computing*, IEEE Press, 2003.
24. A.J. Sinclair and M.R. Jerrum, "Approximate Counting, Uniform Generation and Rapidly Mixing Markov Chains," *Information and Computation*, vol. 82, 1989, pp. 93–133.
25. A.J. Sinclair, "Improved Bounds for Mixing Rates of Markov Chains and Multicommodity Flow," *Combinatorics, Probability and Computing*, vol. 1, 1992, pp. 351–370.
26. P. Diaconis and L. Saloff-Coste, "Comparison Theorems for Reversible Markov Chains," *Annals of Applied Probability*, vol. 3, 1993, pp. 696–730.
27. D. Randall and P. Tetali, "Analyzing Glauber Dynamics by Comparison of Markov Chains," *J. Mathematical Physics*, vol. 41, 2000, pp. 1598–1615.
28. N. Madras and D. Randall, "Markov Chain Decomposition for Convergence Rate Analysis," *Annals of Applied Probability*, vol. 12, 2002, pp. 581–606.
29. R.A. Martin and D. Randall, "Sampling Adsorbing Staircase Walks Using a New Markov Chain Decomposition Method," *Proc. 41st IEEE Symp. Foundations of Computer Science*, IEEE CS Press, 2000, pp. 492–502.
30. M.R. Jerrum et al., "Elementary Bounds on Poincaré and Log-Sobolev Constants for Decomposable Markov Chains," *Annals of Applied Probability*, vol. 14, 2004, pp. 1741–1765.
31. C. Cooper et al., "Mixing Properties of the Swendsen-Wang Process on the Complete Graph and Narrow Grids," *J. Mathematical Physics*, vol. 41, 2000, pp. 1499–1527.
32. D. Randall, "Decomposition Methods and Sampling Circuits in the Cartesian Lattice," *Proc. 5th Symp. Mathematical Foundations of Computer Science*, LNCS 1256, J. Sgall, A. Pultr, and P. Kolman, eds., Springer-Verlag, 2001, pp. 72–84.
33. M.R. Jerrum and A.J. Sinclair, "Polynomial-Time Approximation Algorithms for the Ising Model," *SIAM J. Computing*, vol. 22, 1993, pp. 1087–1116.
34. D. Randall and D.B. Wilson, "Sampling Spin Configurations of an Ising System," *Proc. 10th ACM/SIAM Symp. Discrete Algorithms*, ACM Press, 1999, pp. 959–960.

Dana Randall is an associate professor in the College of Computing and an adjunct professor in the School of Mathematics, Georgia Institute of Technology. Her research interests include theoretical computer science, randomized algorithms, combinatorics, and statistical physics. Randall has a PhD in computer science from the University of California, Berkeley. She has been the recipient of an NSF career award and an Alfred E. Sloan fellowship. Contact her at randall@cc.gatech.edu.