PERFECT SAMPLING USING BOUNDING CHAINS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Mark Lawrence Huber

May 1999

PERFECT SAMPLING USING BOUNDING CHAINS

Mark Lawrence Huber, Ph.D.

Cornell University 1999

In Monte Carlo simulation, samples are drawn from a distribution to estimate properties of the distribution that are too difficult to compute analytically. This has applications in numerous fields, including optimization, statistics, statistical mechanics, genetics, and the design of approximation algorithms.

In the Monte Carlo Markov chain method, a Markov chain is constructed which has the target distribution as its stationary distribution. After running the Markov chain "long enough", the distribution of the final state will be close to the stationary distribution of the chain. Unfortunately, for most Markov chains, the time needed to converge to the stationary distribution (the mixing time) is completely unknown.

Here we develop several new techniques for dealing with unknown mixing times. First we introduce the idea of a bounding chain, which delivers a wealth of information about the chain. Once a bounding chain is created for a particular chain, it is possible to empirically estimate the mixing time of the chain. Using ideas such as coupling from the past and the Fill-Murdoch-Rosenthal algorithm, bounding chains can also become the basis of perfect sampling algorithms. Unlike traditional Monte

Carlo Markov chain methods, these algorithms draw samples which are exactly distributed according to the stationary distribution.

We develop bounding chains for several Markov chains of practical interest, chains from statistical mechanics like the Swendsen-Wang chain for the Ising model, the Dyer-Greenhill chain for the discrete hard core gas model, and the continuous Widom-Rowlinson mixture model with more than three components in the mixture. We also give techniques for sampling from weighted permutations which have applications in database access and nonparametric statistical tests. In addition, we present here bounding chains for a variety of Markov chains of theoretical interest, such as the k coloring chain, the sink free orientation chain, and the antiferromagnetic Potts model with more than three colors. Finally we develop new Markov chains (and bounding chains) for the continuous hard core gas model and the Widom-Rowlinson model which are provably faster in practice.

Biographical Sketch

Mark Lawrence Huber was born in 1972 in the quaint town of Austin, Minnesota, home of the Hormel corporation. Sensing the eventual rise to power of Jesse "The Body" Ventura, Mark's family moved out of the state, living in Yukon, OK, Stillwater, OK, Ames, IA, Corvallis, OR, and Hays, KA before setting in La Grande, OR, where he graduated high school in 1990. Mark then moved south to sunny Claremont, California and Harvey Mudd College, where he completed a Bachelor of Science in Mathematics in 1994. Missing the snow of his native land, Mark headed to Cornell and the School of Operations Research and Industrial Engineering. Upon completion of his Ph.D., Mark will spend the next two years at Stanford University as a National Science Foundation Postdoctoral Fellow in the Mathematical Sciences.

To my grandmother, whose love has always been an inspiration.

Acknowledgements

First of all, I would like to thank my advisor, David Shmoys, who has lasted stoically through my research ups and downs, my eccentric grammar, and my complete lack of organizational skills. He has been a good friend and a great advisor, and I hope to work with him again in the future.

Other professors have also had a great impact on my Cornell experience. The remaining members of my thesis committee, Sid Resnick, Jon Kleinberg, and Rick Durrett, were always willing to lend a hand. I'd also like to thank Persi Diaconis, who first introduced me to the beauty and elegance of rapidly mixing Markov chain theory, a world of problems and techniques that I hadn't known existed.

This thesis would not be here without the tireless efforts of Nathan Edwards, whose computer expertise, LaTeX prowess, and willingness to listen to my litany of problems is second to none.

Financially, this work was supported through numerous sources. An Office of Naval Research Fellowship carried me through the early years, with ONR grants N00014-96-1-00500, NSF grants CCR-9307391, CCR-9700029, DMS-9505155, and ASSERT grant N0001197-1-0881 finishing the job.

I would also like to thank my roommates during my stay at Cornell, Bas de Blank, and Chris Papadopoulos, both of whom had to suffer with my policies (or lack thereof) regarding the proper storage of all manner of items.

The friends I have made here are too many to list, however, I would like to specifically thank my graduate class, Greta Pangborn, Nathan Edwards, Ed Chan, Paulo Zanjacomo, Semyon Kruglyak, and Fabian Chudak. A guy couldn't ask for a better group of people to commiserate with. Thanks go out to Stephen Gulyas who always had a spot on his intermural softball teams for me, and who was always willing to try to turn my tennis swing back into a baseball swing come springtime.

Finally, I would like to thank all of my family from the Midwest to the Pacific Northwest, whose support has always been unconditional. Some great events have happened while I've been at Cornell, and I'm glad to have been able to be there for them.

Table of Contents

List of Tables

List of Figures

Chapter 1

The Need for Markov chains

> How dare we speak of the laws of chance? Is not chance the antithesis of all law?
>
> *-Bertrand Russell, Calcul des probabilités*

Modeling a roulette wheel is quite a bit simpler with probability theory than with Newtonian mechanics. While it is theoretically possible to observe the initial angular momentum imparted to the wheel and ball, followed by an equally intense investigation into the frictional properties that cause the ball to land on red instead of black, a probabilistic approach captures the phenomenon in an elegant manner, and yields results that are both accurate and insightful as regards long term predictions over whether the player will return home poorer than before.

Analysis of gambling systems provided an initial impetus to build a theory of probability, but today many systems whose interactions are too complex to model exactly are modeled probabilistically. Often this provides enormous gains both in

the simplicity of the model and in the ability to analyze various properties of the system.

Today's models have evolved into probability distributions that are quite easy to describe, but which require the use of sophisticated arguments to analyze directly. A way to avoid use of these complicated techniques is to use a simulation approach. When using this technique, no attempt is made to analyze properties of the distributions directly, instead samples are drawn from the distributions, and then statistical estimates are formed for properties of interest. Once the ability to generate random samples is given, a host of statistical methods can be brought to bear on the problem. The accuracy and application of these statistical estimates has been extensively studied. The question that remains is how exactly does one draw these random samples in an efficient manner? The answer for many applications is to use Monte Carlo Markov chain methods. This work introduces a new method in this area called bounding chains, an idea that often can lead to theoretical and experimental insight into a problem.

In this first chapter we describe the Markov chain method, starting with basic definitions and facts concerning Markov chains and laying out the notation that will be used throughout this dissertation. The next chapter presents the discrete models and distributions to which we will be applying our methods, after which we describe the Markov chains developed previously for these problems. In the fourth chapter we introduce our primary tool, bounding chains. Bounding chains provide a basis for algorithms for experimentally determining the time needed to draw approximate random samples using a Markov chain, as well as forming the

foundation for algorithms that draw perfectly random samples. After introducing bounding chains, we show how this approach may be utilized for each of the discrete models discussed earlier. The following chapter explores the difference between approximate and perfect sampling, with an exposition of one technique for perfect sampling, the coupling from the past method of Propp and Wilson [41]. After that, we present another perfect sampling technique, an algorithm of Murdoch and Rosenthal that builds on earlier work of Fill [24], as well as the first analysis of its running time. Finally we turn to continuous models, and explore how the successful techniques from the discrete case may be used to construct and analyze Markov chains for infinite state space processes.

For many probability distributions of interest, we present the first algorithms for perfect sampling: the hard core gas model using the Dyer-Greenhill chain, the heat bath chain for sink-free orientations of a graph, the heat bath chain for k colorings of a graph, the antiferromagnetic Potts model, (these last two were independently discovered though not fully analyzed in [18]), the move ahead one chain for database access, the continuous Widom-Rowlinson mixture model with at least 3 elements in the mixture, the antivoter model, and the Swendsen-Wang chain for the Ising model. For the continuous hard core gas model we develop new bounds on the mixing time of the chain. In addition, we develop new Markov chains for the continuous hard core gas model, the repulsive area interaction model and the Widom-Rowlinson model which have provably stronger bounds on the mixing time than were previously known. The work on perfect sampling and mixing times primarily emerges from an understanding of bounding chains, a simple but extraordinarily powerful tool. The

improved chains come from a generalization and application of a simple "swapping" move of Broder to a wide variety of chains.

1.1 Monte Carlo Markov chain methods

Anyone who has ever shuffled a deck of cards has used a Monte Carlo Markov chain (MCMC) method. The goal of any MCMC algorithm is to draw a random sample from a specific probability distribution. In the case of shuffling cards, the distribution is the uniform distribution over all permutations of the cards.

A Markov chain is a stochastic process possessing the forgetfulness property. Informally, this means that the next state of the chain depends only on the value of the previous state and random choices made during that time step. It does not depend on the value of any prior state. This property makes simulation of a Markov chain very easy. The user need only compute a random function of the current state without regard to the values of prior states.

For the first five chapters of this dissertation, we will be dealing with Markov chains which have a finite state space, and so the definitions we give here will apply to this discrete case. In chapter 8 we expand our scope to include continuous state spaces, and we present a more general treatment of Markov chains in that chapter.

1.1.1 Markov chains

Let Ω be a finite set which we will call the *sample space* or *state space*. Let $X = (\ldots, X_{-1}, X_0, X_1, \ldots)$ be a stochastic process with values chosen from Ω. Let

$\sigma(\ldots, X_{-1}, X_0, X_1, \ldots, X_i)$ be the σ-algebra generated by $\ldots, X_{i-1}, X_i$.

Definition 1.1 *The stochastic process $X = (\ldots, X_{-1}, X_0, X_1, \ldots)$ on Ω is said to be a* Markov chain *if*

$$P(X_{i+1} = j | \sigma(\ldots, X_{i-1}, X_i)) = P(X_{i+1} = j | X_i).$$

We will use a matrix P to denote the probability of moving from state i to state j at a given time. More precisely, let $P(i, j) = P(X_{t+1} = j | X_t = i)$.

Definition 1.2 *An $|\Omega| \times |\Omega|$ matrix P is a* transition matrix *for a Markov chain if*

$$P(i, j) = P(X_{t+1} = j | X_t = i)$$

for all i and j in Ω.

From the definition of P this fact follows immediately.

Fact 1.1 *Suppose that the random variable X_t has distribution p. Then X_{t+1} has distribution pP.*

An easy induction argument together with the above fact yields the following.

Fact 1.2 *Let $P^k(i, j)$ denote the i, j element of matrix P^k. Then*

$$P^k(i, j) = P(X_{t+k} | X_t = i).$$

Often it is desirable that the process X be able to move over every state of the Markov chain.

Definition 1.3 *A Markov chain is* connected *or* irreducible *if for all i, j in Ω there exists a time $t \leq |\Omega|$ such that*

$$P^t(i, j) > 0.$$

Consider a random walk on a graph where the states are nodes of a bipartite graph and at each time step the state changes to a random neighbor of the node. Then at all even times the state will be in one bipartition, and at odd times it will be in the other. We say that such a Markov chain has period 2. More generally, we have the following definition.

Definition 1.4 *Suppose that we have an irreducible Markov chain, and that Ω is partitioned into k sets $\mathcal{E} = \{E_0, \ldots, E_{k-1}\}$. If for all $i = 0, \ldots, m-1$ and all $x, y \in E_i$, $\sum_{y \in E_j} P(x, y) = 1$ where $j = i + 1 \mod k$, then $\mathcal{E}$ forms a* k-cycle *in the Markov chain. The* period *of a Markov chain is the largest value of k for which a k-cycle exists. If $k = 1$, the Markov chain is said to be* aperiodic.

Together, the properties of aperiodicity and irreducibility have important implications for a Markov chain.

Definition 1.5 *A Markov chain is* ergodic *if it is both connected and aperiodic.*

We have seen that if X_t has distribution p, then X_{t+1} will have distribution pP. If in fact, $pP = p$, then X_{t+1} will have the same distribution as X_t. Induction can be used to show that $X_{t'}$ will all be identically distributed for all $t' > t$.

Definition 1.6 *If $\pi P = \pi$, then π is a* stationary distribution *of the Markov chain.*

Throughout this work, the symbol π will be used to denote the stationary distribution of some Markov chain, though often we will describe π before we describe the Markov chain for which it is the stationary distribution.

Ergodic Markov chains are useful for the following reason [10].

Theorem 1.1 *An ergodic Markov chain has a unique stationary distribution.*

Intuitively, a Markov chain step represents a moving of probability flow along edges (i, j) such that $P(i, j) > 0$. The stationary distribution is the probability distribution such that probability flow into each node is exactly balanced by the probability flow out of each node. This is a "general balance" condition that flow in equals flow out. A more restrictive condition would be to require that flow across an edge in one direction is exactly balanced by the flow across the edge in the opposite direction.

Definition 1.7 *A Markov chain is* reversible *or is said to satisfy the* detailed balance condition *if for all i, j in Ω,*

$$\pi(i)P(i, j) = \pi(j)P(j, i).$$

If $\pi(i)P(i, j) = \pi(j)P(j, i)$ then we say that the Markov chain is reversible with respect to the distribution π (and in fact π will be a stationary distribution of the chain).

A Markov chain satisfies the detailed balance condition if at stationarity, the probability flow along each edge (i, j) is the same as the probability flow along each edge (j, i). The following shows that detailed balance is a stronger condition than general balance.

Fact 1.3 *If a Markov chain is reversible with respect to the distribution p, then p is a stationary distribution for the chain. If the Markov chain is connected, this p is the unique stationary distribution for the chain.*

The names "reversibility" and "detailed balance" appear to have nothing whatsoever to do with one another. In chapter 6 we go into more detail about why this condition is also called reversibility.

1.1.2 Going the distance

Our ultimate goal is to sample from a target distribution. Given an algorithm which generates from some probability distribution, we need a method for determining when our algorithmic output is close to our desired distribution. That is, we need a metric on distributions.

We will use two common measures of distance, the total variation distance, and the separation.

Definition 1.8 *The total variation distance between a pair of distributions p and q is denoted $||p, q||_{TV}$, and defined as*

$$||p, q||_{TV} = \sup_{A \subset \Omega} |p(A) - q(A)|.$$

The following fact about total variation distance will come in handy later.

Fact 1.4 *For discrete state spaces Ω,*

$$||p, q||_{TV} = \sum_{x \in |\Omega|} \frac{1}{2} |p(x) - q(x)|.$$

The separation between a distribution and π is defined as follows.

Definition 1.9 *The stationary distance between p and π is denoted $||p, \pi||_S$, and defined as*

$$||p, \pi||_S = \sup_{A \subset \Omega | \pi(A) > 0} \frac{p(A) - \pi(A)}{\pi(A)} = \sup_{A \subset \Omega} \left(1 - \frac{\pi(A)}{p(A)}\right).$$

Since $\pi(A) \leq 1$ for all $A \subset \Omega$, clearly $||p, \pi||_S \geq ||p, \pi||_{TV}$, and this is a stronger way of measuring how far p is from stationarity.

These two means of quantifying how close the current state is to stationarity have some advantageous theoretical properties, and are by far the most commonly seen in practice.

The heart of the Monte Carlo Markov chain method is the idea that if a chain is run for a long time from any starting distribution, then it will move towards a stationary distribution of the chain. Armed with our metrics, we may now make statements about limits of distributions, and make this concept precise.

Theorem 1.2 *Suppose we have an aperiodic Markov chain. Then $\lim_{t \to \infty} pP^t$ will be stationary, and if the Markov chain is ergodic, then*

$$\lim_{t \to \infty} pP^t = \pi,$$

where π is the unique stationary distribution of the chain.

Once we know that pP^t is converging to π, the next logical question is, how fast is it converging? We measure this by the mixing time of the chain. A fact will make this definition easier.

Fact 1.5 *Given an ergodic Markov chain with stationary distribution π, then we have that $||pP^t - \pi||_{TV}$ is a monotonically decreasing function of t. Put another way, $||pP^{t'} - \pi||_{TV} \leq ||pP^t - \pi||_{TV}$ for all $t' > t$.*

Let δ_x denote the distribution that puts probability 1 on state x and 0 elsewhere, and set $P_x^t = \delta_x P_x^t$. This is the distribution of a process that was begun in state x and run for t time steps.

Definition 1.10 *Let $\tau_{TV}(x, \epsilon)$ be the smallest time such that $||P_x^t - \pi||_{TV} < \epsilon$. Let*

$$\tau_{TV}(\epsilon) = \max_{x \in \Omega} \tau_{TV}(x, \epsilon).$$

Define $\tau_S(x, \epsilon)$ and $\tau_S(\epsilon)$ in the same fashion, using the stationary distance.

Once we know the mixing time of a chain, describing the basic Monte Carlo Markov chain method is easy.

Monte Carlo Markov chain (MCMC) method

Input: $\epsilon, (\Omega, P)$

Set $X_0 = x$ for some $x \in \Omega$.
For $i = 1$ to $t_x(\epsilon)$
  Take one step on the Markov chain (Ω, P) from X_i
  Set X_{i+1} to be the output of this step
Output $X_{t_x(\epsilon)}$

Figure 1.1: General Monte Carlo Markov chain method

The only thing preventing this from being an actual algorithm is lack of knowledge about the value of $\tau_x(\epsilon)$. While many heuristics exists for determining this

value, we will concern ourselves in the following chapters with developing upper bounds for $\tau(\epsilon)$ which are always accurate.

Definition 1.11 *When* $\tau(\epsilon)$ *is polynomial in* $\ln(\Omega)$ *and* $\ln(1/\epsilon)$*, we say that the chain is rapidly mixing.*

In this work we will develop methods for showing when chains are rapidly mixing, and ways to deal with them when they are not.

Chapter 2

The Discrete Models

> The laws of history are as absolute as the laws of physics, and if the probabilities of error are greater, it is only because history does not deal with as many humans as physics does atoms.
> *-Isaac Asimov, Foundation and Empire*

The Monte Carlo Markov chain method is applicable to an amazing range of problems. Anywhere one wishes to obtain estimates of statistics for a probabilistic model, often MCMC is the only reasonable approach for even approximating the answer to a problem.

One of the richest sources of problems in this area comes from statistical mechanics. In this area, substances are modeled as random samples drawn from probability distributions. These distributions have particular values for physical parameters such as energy. Any real substance contains on the order of 10^{23} particles, so central limit theorems definitely apply, and statistics for a model are often highly concen-

trated about a single value. Naturally, in evaluating the usefulness of a particular model, the question arises of what is the average of a particular statistic over a distribution. For special cases, this question may be answered analytically, but often the distribution is too complex to allow for a direct approach.

What is perhaps surprising is that many of these models from statistical physics have counterparts of interest in theoretical computer science. Evaluation of most statistics of interest in these models are examples of $\sharp P$-complete problems, and often simply being able to generate a random sample from these problems cannot be done efficiently unless $RP = NP$. These statistical models were introduced (in many cases) decades before the corresponding graph theoretical problem was shown to be NP-hard.

Recall that a problem is in NP if it is a decision problem where a certificate that the answer is true may be checked in polynomial time. Problems in this class include determining whether a boolean expression is satisfiable or whether or not a graph has a proper 3 coloring. Optimization versions of problems in NP include the traveling salesman problem and integer programming.

The class $\sharp P$ is the set of problems where the goal is to count the number of accepting certificates to a problem in NP. For example, consider once more the problem in NP of finding an assignment of variables which leads to a given Boolean expression being true. The corresponding problem in $\sharp P$ is to *count* the number of assignments which lead to the expression being true. Clearly a problem in $\sharp P$ is more difficult than one in NP, since if we know how many assignments are true, then we certainly know if at least one assignement is true.

Given the difficulty of solving a $\sharp P$ problem exactly, the logical question is, when can we develop algorithms which approximate the true answer? For most of these problems, the ability to sample from the distribution of interest immediately yields such an approximation algorithm [26].

The statistical physics models we consider are interesting for their ability to model and predict real world substances, and also for their theoretical properties. Often these models exhibit a phenomenon known as a phase transition, where a small change in the parameter of the model leads to an enormous change in the macroscopic properties of the distribution. Phase transitions are often linked to the speed at which a Markov chain simulation runs. Roughly speaking, on one side of the phase transition a chain may be rapidly mixing, but on the other side it may converge very slowly.

This behavior is similar to that of the *NP*-complete problems related to these models. It is well known that for many problems approximating an answer to an optimization problem to within a certain constant or higher may be possible in polynomial time, while any improvement in that constant leads to a *NP*-complete problem.

For all the models we consider, the sample space will be the colorings of a graph, that is, $\Omega = C^V$ where V is the vertex set of a graph (V, E) and $C = \{1, \ldots, Q\}$ is a set of Q colors. This contains a wide variety of problems, from generating a random permutation (where $\Omega \subset V^V$) to mixture models of gases. Often the graphs considered in these model are simple lattices in 2 or 3 dimensions, although they will be defined for arbitrary graphs. Throughout this work we will use n to refer to

the number of nodes in the graph, and m to refer to the number of edges.

Of course, not all of the models we will consider come from statistical physics. We also present problems arising from database searches and the numerical evaluation of statistical tests, as well as several models of interest for their theoretical properties.

2.1 The Ising model

Perhaps the most venerated model in statistical physics is the Ising model. (This is sometimes called the Lenz-Ising model since Lenz first proposed it to Ising, who was his student at the time.) The model is quite simple, yet contains within it the phase transition behavior mentioned earlier.

The model was first introduced as a model of magnetism. More recently it has found use as a model of alloys, and for Quantum Chromodynamics computations. The idea is simple. Our color set consists of two colors $C = \{-1, 1\}$. Following the use of the Ising model as a model of magnetism, we shall refer to nodes colored 1 as spin up and nodes colored -1 as spin down.

A configuration x consists of an assignation of colors to each of the nodes of the graph (V, E). The Hamiltonian of a configuration $H(x)$ is set to be

$$H(x) = -\sum_{(i,j)\in E} \alpha_{(i,j)} x(i)x(j).$$

The α variables measure the strength of interaction across a particular edge. Although the methods we will discuss can deal with the case of arbitrary α, for simplicity we will assume that every $\alpha_{(i,j)}$ is 1.

The Ising model is a probability distribution on the set of configurations. The probability of selecting configuration x is

$$\pi(x) = \frac{\exp\{-JH(x)/(kT)\} + \sum_v B_v x(v)}{Z_T}.$$

Here J is either 1 (for ferromagnetism) or -1 (for antiferromagnetism), B is a parameter that measures the external magnetic field, k is Boltzman's constant, T is the temperature of the model, and Z_T is the value which makes π a probability distribution, i.e., the normalizing constant. Often, Z_T is referred to as the partition function.

It will be helpful to gain some intuition about how the parameters interact. Suppose we are dealing with the ferromagnetic Ising model ($J = 1$). Note that $-H(x)$ is large when the values of $x(i)$ and $x(j)$ for an edge (i, j) are the same. Hence a configuration where the endpoints of edges receive the same color is more likely than a configuration where they are different. Physically, this means that the spins tend to line up, making for a stronger magnet.

If $J = -1$ (antiferromagnetism) then the highest probability states are ones where the endpoints of edges are colored differently. Here the Hamiltonian is largest when spins do not align.

The value B_v measures the presence of an external magnetic field that biases the configuration towards either spin up or spin down at the node v. While the techniques that we use can be modified to incorporate a nonzero B, for simplicity we will always take B to be 0.

The temperature measures how free the spins are to fight their natural ferro-

magnetic or antiferromagnetic tendencies. When the temperature is very high, $\pi(x)$ is roughly $1/Z_T$ regardless of the value of $H(x)$. In this case, all the configurations are equally likely, and each individual node is almost as likely to be spin up as spin down.

When T is very small, only states with very large $H(x)$ have significant probability of occurring. For instance, in the ferromagnetic state with high probability most of the states will be pointing in the same direction. This means that the state tends to exhibit long range behavior, where the spin of two nodes on the opposite sides of the graph are highly correlated. The distribution π changes smoothly with T, and so there is a point where these long range correlations start to appear. Roughly speaking, this is the idea behind phase transitions.

Technically, phase transitions involve discontinuities in properties of the graph, and so truly only occur in graphs with an infinite number of nodes. However, even for relatively modest size graphs, the presence of a phase transition will result in large changes in the properties of a graph with small changes in a parameter such as T. One of the reasons for the primacy of the Ising model in statistical simulations is the fact that this simple model exhibits a phase transition for graphs such as the 2 dimensional lattice. Phase transitions are of course prevalent throughout the physical world, for instance, the process of ice turning to water is a phase transition.

The constant k is Bolzmann's constant, and arises out of the statistical mechanical justification for the Hamiltonian in the Ising model. Note that x is a two coloring, and therefore defines a cut of the graph. Let $C(x)$ denote the number of edges which cross the cut (this is the unweighted value of the cut). Then as we have

defined it, $H(x) = 2C(x) - m$, where m is the number of edges in our graph. The $\exp(m/(kT))$ term in the exponential is a constant, so we may introduce a scaled temperature T' such that

$$\pi = \frac{\exp\{-JC(x)/T'\}}{Z'_T}.$$

2.1.1 The antiferromagnetic Ising model and MAX CUT

There exists a very close relation between the antiferromagnetic Ising model and the problem MAX CUT, which is the problem of finding the largest cut in an arbitrary graph. This problem is known to be *NP*-complete, so if this problem could be solved in polynomial time, then every problem in *NP* could be solved in polynomial time. This makes it quite unlikely that an efficient solution to this problem will be found.

The probability of generating a particular cut from the antiferromagnetic Ising model will be $\exp\{C(x)/T\}/Z_T$. When $T = \ln 2/n$, $\pi(x) = (2^n)^{C(x)}/Z_T$. Let the configuration x_{max} be the coloring associated with the maximum cut in the graph. There are only $2^n - 1$ other colorings of the graph, hence the total weight of cuts which are of smaller size than the maximum is at most

$$2^n \cdot (2^n)C(x_{max}) - 1/Z_T \leq (2^n)^C(x_{max})/Z_T = \pi(x_{max}).$$

Hence with probability 1/2, a random sample drawn from this distribution will find the maximum cut in the graph. In fact, when the graph is regular (all degrees of the graph are the same) we may do much better.

Let Δ denote the degree of each node in the graph. We shall show that when $T = O(1/\Delta)$, an algorithm for sampling from the Ising model leads to a constant

factor approximation algorithm for MAX CUT.

Definition 2.1 *Given a maximization problem with optimal value OPT, and a polynomial time algorithm which produces solutions with value ALG, we say that it is a ρ-approximation algorithm if $ALG/OPT \geq \rho$. Similarly, if OPT is the optimal solution for a minimization problem, we have a ρ-approximation algorithm if $ALG/OPT \leq \rho$.*

Theorem 2.1 *Suppose that we have a graph of bounded degree Δ and an efficient means for sampling from the Ising model on arbitrary graphs of maximum degree Δ for some temperature T. Let $\rho = 1 - \frac{4T \ln 2}{\Delta}$. Then if $\rho > 0$, we have a randomized ρ-approximation algorithm for MAX CUT.*

Proof: Let $A(\rho)$ denote the set of configurations x such that $C(x) \leq \rho C(x_{max})$. Then

$$\begin{aligned}
\pi(A) &= \sum_{x \in A} \frac{\exp\{C(x)/T\}}{Z_T} \\
&\leq \sum_{x \in A} \frac{\exp\{\rho C(x_{max})/T\}}{Z_T} \\
&\leq \pi(x_{max})|A|e^{-(1-\rho)C(x_{max})/T} \\
&\leq \pi(x_{max})e^{n \ln 2 - (1-\rho)C(x_{max})/T}
\end{aligned}$$

In order to have $\pi(x_{max}) \geq \pi(A)$, we need $n \ln 2 - (1-\rho)C(x_{max})/T \leq 0$, or $T \leq (1-\rho)C(x_{max})/(n \ln 2)$.

Since there are $n\Delta/2$ edges, we know that a maximum cut contains at least $n\Delta/4$ edges. Therefore, if

$$T \leq \frac{(1-\rho)n\Delta}{4n\ln 2} = \frac{(1-\rho)\Delta}{4\ln 2},$$

then we have a randomized ρ-approximation to MAX CUT.

It turns out that (even for bounded degree graphs) solving MAX CUT is not only *NP*-complete, but *APX*-complete. Loosely speaking, *APX* is the set of optimization problems such that there exists a bound ρ' such that if a better than ρ-approximation algorithm exists, then $P = NP$. For us this means (that unless $RP = NP$), there exists a constant α such that no polynomial time algorithm exists for sampling from the Ising model when $T < \alpha\Delta$. It is not surprising, then, that in chapter 5 we shall analyze an algorithm of Häggström and Nelander [17] and show that it runs in polynomial time when $T \geq 0.5\Delta$.

This all indicates the difficulty of even sampling from the antiferromagnetic Ising model. The ferromagnetic Ising model is a different story, and in fact a polynomial time algorithm exists [27] [42] that generates samples drawn approximately from the distribution of the Ising model for arbitrary graphs.

2.2 The Potts Model

In the Ising model we had spin up and spin down, but we live in a three dimensional world. It is easy to consider a model with spin up, down, right, left, into the plane and out of the plane. In general, instead of using two colors we now use Q colors.

This is the Potts model, and it is an important generalization of the Ising model [39].

Let x be a coloring of the nodes using colors from $C = \{0, 1, \ldots, Q-1\}$. Then $C(x)$ now refers to the number of edges which cross the Q way cut determined by x. As in the Ising model, we let

$$\pi(x) = \frac{\exp\{-JC(x)/T\}}{Z_T},$$

where $J = 1$ for ferromagnetism and $J = -1$ for antiferromagnetism.

As with the Ising model, the antiferromagnetic Potts model is NP difficult to sample from at arbitrary temperatures, with the reduction being to MAX Q CUT. Unlike the Ising case, however, no method for sampling from the ferromagnetic Potts model is known to run quickly at all temperatures. We will give a partial answer to this question by analyzing two chains for the Potts model, single site update and Swendsen-Wang.

2.3 The hard core gas model

In the hard core gas model, we again two color a graph. However, here coloring a node 1 means that a gas molecule occupies that node, while the color 0 means that it is empty. These molecules take up a fixed amount of space, the core of the molecule. These cores are "hard" and so are not allowed to intersect. In our model, this means that no two adjacent nodes contain gas molecules. One way to enforce this condition mathematically given a coloring x is to require that $x(i)x(j) = 0$ for all edges (i, j). Let $n(x) = \sum_v x(v)$. If we think of the configuration x as defining the locations of a set of gas molecules, then $n(x)$ represents the number of molecules

in the configuration. The distribution over this configurations is

$$\pi(x) = \frac{\lambda^{n(x)}}{Z_\lambda},$$

where λ is a parameter known as the activity or fugacity.

This type of set, where no two nodes which are adjacent are both in the set, is known as an independent set. Just as the Ising model is a close relative of MAX CUT, the hard core gas model is closely linked to the problem of finding the maximum independent set of a graph, which is an *NP*-complete problem. When $\lambda \geq n$, then with probability at least $1/n$ the largest independent set in the graph is chosen. In fact, Dyer Frieze, and Jerrum showed [11] by reduction to a different NP-complete problem that it there does not exist an efficient means for sampling from this distribution when $\lambda = 1$ and $\Delta = 25$ unless $RP = NP$, even when the graph is restricted to be bipartite.

2.4 The Widom-Rowlinson mixture model

Related to the hard core gas model is the Widom-Rowlinson mixture model [49]. In this section we consider a discrete version of the model [31]. Suppose we have Q different types of substances in a mixture. Particles of substance i are allowed to be close to one another. However, two adjacent sites cannot be occupied by two different substances. Mathematically, $C = \{0, 1, \ldots, Q\}$, with the color 0 indicating that a site is empty and color i indicating that a particle of substance i occupies the site. We require that for all edges (i, j) of the graph, either $X(i) = X(j)$ or $X(i)X(j) = 0$.

Häggström and Nelander [18] gave an perfect sampling algorithm for this discrete model when $Q \geq 3$. Later we will present an improved algorithm for which sharper running time bounds may be shown. In addition, in chapter 8 we will construct an perfect sampling algorithm for the continuous case as well.

2.5 Q colorings of a graph

As the temperature T drops in the antiferromagnetic Potts model, more weight is given to those states where the size of the cut is the entire graph. In other words, the highest weight is given to colorings where the endpoints of an edge have different colors.

Definition 2.2 *A Q coloring of the nodes of a graph is* proper *if the endpoints of each edge in the graph have different colors.*

One way of defining the distribution when $T = 0$ is to give equal weight to a proper Q coloring of the graph, and 0 weight if the endpoints of an edge are given the same color.

Jerrum [25] constructed a Markov chain for sampling uniformly from the colorings of a graph when $Q \geq 2\Delta$. It is NP-hard to determine if there is a coloring when $Q = \Delta$, and for $Q = \Delta + 1$ the chain of Jerrum is not connected. However, for $\Delta + 1 < Q < 2\Delta$ very little is known about the behavior of the chain. In chapter 5 we give an perfect sampling algorithm for this problem (also independently given in [18]).

This problem of sampling from the Q colorings of a graph also provides an easy illustration for how a random sampling algorithm can be turned into an approximate counting algorithm. The problem of counting the number of Q colorings of a graph is a $\sharp P$-complete problem. Suppose that $k \geq 2\Delta$, so we know from Jerrum's work that we have an approximate sampling algorithm. The work in [26] gives an algorithm for creating an perfect sampling algorithm that runs in $O(mT_{RS})$ time, where T_{RS} is the time needed to take a random sample. We present here an algorithm that runs in $O(nT_{RS})$ time.

2.6 Sink Free Orientations of a graph

Given an undirected graph, a sink free orientation is an assignment of directions to edges such that no edge has outdegree 0. The problem of computing the number of sink free orientations of a graph is $\sharp P$-hard [6], and is also a special case of evaluating the Tutte polynomial of a graph, making this problem of theoretical interest. As with the other problems we consider, this one may be formulated as a coloring.

Suppose that we have an edge $e = \{i, j\}$ where $i < j$. Then if we have in our coloring $x(e) = 1$, the edge is oriented (i, j), but if $x(e) = -1$ the edge is oriented (j, i). To be sink free, we require that $\sum_{j<i}(-x(\{i,j\}) - 1) + \sum_{j>i}(x(\{i,j\}) - 1) > 0$ for all nodes i.

2.7 Hypercube slices

A hypercube may be thought of as having state space $\{0,1\}^n$ where n is the dimension of the cube. Edges exist between any two points in the state which differ by at most one coordinate (so that changes only occur parallel to the coordinate axis). Each possible state is considered equally likely.

If the k colorings of a graph may be considered the zero temperature limit of the Potts model, then the hypercube chain may be thought of as the infinite temperature limit. Here each possible configuration is equally likely, and the coloring of a site (coordinate) is completely independent of the other coordinates.

A restriction of the hypercube is used to study instances where we wish the magnetization is the Ising model to remain constant [50]. Recall that for the Ising model, the magnetization is the number of nodes colored 1. In the hypercube slice model, we restrict the set of allowable configuration. Let L and U be integers with $L \leq U$. Then the state space of the hypercube slice model is those points in $\{0,1\}^n$ satisfying $L \leq \sum_i x(i) \leq U$. If L is close to U, then the magnetization will be roughly constant.

2.8 Applying the Monte Carlo method

All the models discussed above are probability distributions, and the question of interest is how to sample efficiently from these distributions. For decades, construction of Markov chains has been possible for these problems and others by using some basic techniques. We next explore the most successful of these techniques.

Chapter 3

Building Markov chains

> On two occasions I have been asked [by members of Parliament], 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?' I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.
>
> *-Charles Babbage*

Perhaps the earliest Monte Carlo algorithm (outside of astrology) is a method for finding the value of the numerical constant π dating back to the 1600's. This method involved flipping a toothpick onto ruled paper and measuring the number of times the toothpick crossed the lines, and was used by some to while away the time [5]. Even at that time, far better methods existed for approximating π, as this method is agonizingly slow.

With a computer, pseudorandom numbers can be generated by the millions, and used to drive a Markov chain towards its stationary distribution. As Babbage

notes, however, the "right figures" must first be put into the machine. In our case, that means that we must be able to construct a Markov chain whose stationary distribution is the same as the distribution given by our probabilistic models.

Broadly, these chains fall into two classes, local update chains and nonlocal update chains. Recall that for all of our problems, $\Omega = C^V$ consists of the colorings of a graph. Local update algorithms take advantage of this structure by updating the colors on only one or two nodes at a time. Nonlocal update algorithms are often far less intuitive than their local counterparts, but many times can avoid regions where local chains are not rapidly mixing by changing the color of many nodes simultaneously.

3.1 Conditioning chains

The reason why we cannot sample directly from these distributions π is that they are defined as $\pi(x) = w(x)/Z$ where $w(x)$ is an easily computable weight function, and Z is an unknown normalizing constant that is often very difficult to compute. Conditioning will allow us to eliminate the need to know Z when taking a step.

As an example, consider the hard core gas model, where the desired distribution is $\pi(x) = \lambda^{n(x)}/Z_\lambda$. Here the weight function is trivial to compute, but finding Z_λ is $\sharp P$ hard [33]. Suppose that someone else has generated a random sample and sent it to us, however, the color at node v is missing. All of the other color data came through just fine. We now seek to create a new random sample from π conditioned on the values of the colors at all nodes other than node v.

If one of the neighbors of v is colored 1 (that is, included in the independent set), then we did not need the missing data since v must be colored 0. However, if all the neighbors of v are colored 0, then two configurations are possible. We wish to choose random $X(v)$ given $X(V \setminus \{v\})$. Note that $n(X)$ when $X(v) = 1$ is exactly 1 more than $n(X)$ when $X(v) = 0$. Let x_1 denote the configuration where v is colored 1 and x_0 denote the configuration where v is colored 0. Then

$$\begin{aligned} P[X = x_1 | X(V \setminus \{v\}) = x(V \setminus \{v\})] &= \frac{P(X = x_1)}{P(X(V \setminus \{v\}))} \\ &= \frac{P(X = x_1)}{P(X = x_1) + P(X = x_0)} \\ &= \frac{\lambda^{n(x_1)}/Z}{\lambda^{n(x_1)}/Z + \lambda^{n(x_0)}/Z} \\ &= \frac{\lambda}{\lambda + 1} \end{aligned}$$

Similarly, it is easy to show that

$$P(X = x_0 | X(V \setminus \{v\}) = \frac{1}{\lambda + 1}.$$

A wonderful thing has occurred, in that Z_λ was canceled out in the computation. This is not an accident, but a general feature of conditioning arguments.

More generally, suppose that we know the value of a configuration on all but a small set of nodes V_U (here the U in the subscript stands for unknown). Then $x(V_U)$ is the coloring of V_U, and $x(V \setminus V_U)$ is the coloring on the remaining nodes, so that $(x(V_U), x(V \setminus V_U))$ describes a complete configuration on V. Given part of a configuration $x(V \setminus V_U)$, we can randomly extend it to a complete configuration by using

$$
\begin{aligned}
P[X(V) = x(V) | X(V \setminus V_U) = x(V \setminus V_U)] &= \frac{w((x(V_U), x(V \setminus V_U)))/Z}{P(X(V \setminus V_U = x(V \setminus V_U)))} \\
&= \frac{w((x(V_U), x(V \setminus V_U)))}{\sum_{x(V_U)} w((x(V_U), x(V \setminus V_U)))}. \\
&= \frac{w((x(V_U), x(V \setminus V_U)))/Z}{\sum_{x(V_U)} w((x(V_U), x(V \setminus V_U)))/Z}. \\
&= \frac{w((x(V_U), x(V \setminus V_U)))}{\sum_{x(V_U)} w((x(V_U), x(V \setminus V_U)))}.
\end{aligned}
$$

Again, the normalizing constant Z has dropped completely out of the picture. The general conditioning chain, then, picks some subset of nodes at random, discards their value, and then replaces the colors on the subset by randomly extending the remaining coloring.

3.1.1 The Heat Bath chain

The heat bath chain is a special case of conditioning chains where the probability distribution on V_U does not depend upon the current state of the chain. It works like this. A set V_U is chosen at random and the colors on those nodes are discarded. The colors are then replaced randomly by choosing colors for V_U from π conditioned on the values of the colors at all of the other nodes. At every step we use the same distribution on subsets of V for choosing our random V_U. Often V_U is a randomly chosen dimension/node of the state space.

To describe choosing random numbers, we use $a \in_U A$ to denote the act of choosing an element of A uniformly from that set. We will use $a \in_R A$ to denote

choosing a from A at random according to an arbitrary distribution. Because we are choosing V_U and only updating that portion, often the value of $x(V \setminus V_U)$ is clear from context. Therefore, for notational convenience, we will use $w(x(V_U))$ to denote $w((x(V_U), x(V \setminus V_U)))$.

The general heat bath Markov chain

Input: $X_t \in C^V$
w a weight function on $\Omega = C^V$
p a distribution on subsets of V

Set $X = X_t$
Choose $V_U \in_R 2^V$ according to distribution p
Choose $X(V_U) \in_R C^{V_U}$ according to
$P(X(V_U) = x(V_U)) = \frac{w(x(V_U)))}{\sum_{x'(V_U)} w(x'(V_U))}$
Set $X_{t+1} = X$

Figure 3.1: The general heat bath Markov chain

Reversibility allows us to show that this chain has the desired stationary distribution $\pi = w/Z$. Two states x_1 and x_2 are connected if their colors differ on a set v_U which has positive probability of being selected ($p(v_U) > 0$). There can be more than one set v_U for which this is true. Let $\mathcal{V}_U$ denote the set of subsets of V that contain all the nodes on which x_1 and x_2 have different colors. We now show that the heat bath chain is reversible. Let $x_1 \to x_2$ denote the event that configuration moves from state x_1 to x_2 at one step of the Markov chain.

$$\begin{aligned} \pi(x_1)P(x_1 \to x_2) &= \pi(x_1) \sum_{v_U} p(v_U) P(x_1 \to x_2 | V_U = v_U) \\ &= \frac{w(x_1)}{Z} \sum_{v_U} p(v_U) P(x_1 \to x_2 | V_U = v_U) \end{aligned}$$

Single site hard core heat bath chain

Set $X = X_t$
Choose a vertex v uniformly at random from V
Choose U uniformly from $[0, 1]$
If $U \leq \frac{1}{1+\lambda}$ or a neighbor of v is colored 1
  Set $X(v) = 0$
Else
  Set $X(v) = 1$
Set $X_{t+1} = X$

Figure 3.2: Single site hard core heat bath chain

$$\begin{aligned}
&= \frac{w(x_1)}{Z} \sum_{v_U \in \mathcal{V}_U} p(v_U) \frac{w(x_2)}{\sum_{x:x(V\setminus V_U)} = x_2(V \setminus V_U) w(x)} \\
&= \frac{w(x_2)}{Z} \sum_{v_U \in \mathcal{V}_U} p(v_U) \frac{w(x_1)}{\sum_{x:x(V\setminus V_U)} = x_1(V \setminus V_U) w(x)} \\
&= \frac{w(x_2)}{Z} \sum_{v_U} p(v_U) P(x_2 \to x_1 | V_U = v_U) \\
&= \pi(x_2) P(x_2, x_1)
\end{aligned}$$

In the single site update algorithm for the hard core gas model, V_U is chosen to be a single vertex v with probability $1/n$. The probability of setting $X(v) = 1$ and $X(v) = 0$ are exactly those computed in the previous section, so we know that this chain has the correct distribution.

Suppose that the graph is bipartite with $V = V_L \cup V_R$, where each edge has an endpoint in each of V_L and V_R. Then the neighbors of any node in V_L lie only in V_R, and neighbors of V_R all lie in V_L. This allows us to switch all of the values of V_L or V_R simultaneously. This is definitely nonlocal, since the average number of

Nonlocal hard core heat bath chain

Set $X \leftarrow X_t$
Choose V_U to be V_L or V_R each with probability $1/2$
For each vertex v in V_U
  Choose U uniformly from $[0, 1]$
  If $U \leq \frac{1}{1+\lambda}$ or a neighbor of v is colored 1
    Set $X(v) \leftarrow 0$
  Else
    Set $X(v) \leftarrow 1$
Set $X_{t+1} \leftarrow X$

Figure 3.3: Nonlocal hard core heat bath chain

nodes which can be affected in a move is $n/2$.

3.2 Metropolis-Hastings

Metropolis-Hastings chains [35] take a rate approach rather than a conditioning approach. This approach works best when all of the possible colorings for V_U are roughly equal in value.

In the heat bath chain, we chose V_U, threw away the colors on those nodes, and then replaced them according to the conditional probability. In a Metropolis-Hastings type algorithm, we attempt to change the values on V_U and sometimes reject the change if it would lead to a state with lower weight.

Again reversibility is used to show that this chain has the desired stationary distribution. As before, suppose that x_1 and x_2 are two states with positive weight where the set of subsets $\mathcal{V}_U$ containing all nodes with different colors has positive

The general Metropolis-Hastings Markov chain

Input: $X_t \in C^V$
w a weight function on $\Omega = C^V$
p a distribution on subsets of V

Set $X \leftarrow X_t$
Choose V_U at random according to p
Choose $x(V_U)$ uniformly at random from C^{V_U}
If $w((x(V_U), X(V \setminus V_U)) \geq w(X)$
  Set $X(V_U) \leftarrow x(V_U)$
Else
  Choose U uniformly at random from $[0, 1]$
  If $U \leq \frac{w(x(V_U))}{w(X)}$
    Set $X(V_U) \leftarrow x(V_U)$
Set $X_{t+1} \leftarrow X$

Figure 3.4: The general Metropolis-Hastings Markov chain

weight in p. Then without loss of generality, let $w(x_1) \leq w(x_2)$.

$$\begin{aligned} \pi(x_1)P(x_1 \to x_2) &= \pi(x_1) \sum_{v_U \in \mathcal{V}_U} p(v_U) P(x_1 \to x_2 | V_U = v_U) \\ &= \frac{w(x_1)}{Z} \sum_{v_U} p(v_U) \frac{1}{|C|^{|v_U|}} \end{aligned}$$

and

$$\begin{aligned} \pi(x_2)P(x_2 \to x_1) &= \pi(x_2) \sum_{v_U} p(v_U) P(x_2 \to x_1 | V_U = v_U) \\ &= \frac{w(x_2)}{Z} \sum_{v_U} p(v_U) \frac{w(x_1)/w(x_2)}{|C|^{|v_U|}} \\ &= \frac{w(x_1)}{Z} \sum_{v_U} p(v_U) \frac{1}{|C|^{|v_U|}} \end{aligned}$$

so they are equal and the Metropolis-Hastings chain is reversible with the correct distribution.

Applying this to the specific example of the hard core gas model, we have the following. When the color chosen for v is 1 and no neighbors have color 1 already,

Single site hard core Metropolis-Hastings chain

Set $X \leftarrow X_t$
Choose a vertex v uniformly at random from V
Choose c uniformly from $\{0, 1\}$
If $c = 1$ and v has no neighbors colored 1
  Set $X(v) \leftarrow c$
Else
  Choose U uniformly at random from $[0, 1]$
  If $U \leq \frac{1}{\lambda}$
    Set $X(v) \leftarrow c$
Set $X_{t+1} \leftarrow X$

Figure 3.5: Single site hard core Metropolis-Hastings chain

then setting v to 1 raises the weight of the configuration, so we always proceed. If, however, the color chosen for v is 0 then the weight might drop from $\lambda^{n(x)}$ to $\lambda^{n(x)-1}$. The smaller over the larger of these weights is $1/\lambda$, so with this probability we switch a 1 to 0.

As with the heat bath, when the graph is bipartite these ideas may be used to construct a nonlocal algorithm as well.

3.3 The acceptance rejection heat bath chain

An idea which will come in handy later in a particular implementation of the heat bath chain we will call the acceptance rejection heat bath chain. The transition probabilities are exactly the same as for the heat bath chain, the difference lies in how a generate a Markov chain step. In words, what we do is having selected the

The general acceptance rejection heat bath Markov chain

Input: $X_t \in C^V$
w a weight function on $\Omega = C^V$
p a distribution on subsets of V

Set $X \leftarrow X_t$
Choose $V_U \in_R 2^V$ according to distribution p
Set $M \geq \max\{w(x(V_U))\}$
Repeat
Choose $X(V_U) \in_U C^{V_U}$
Choose $W \in_U [0,1]$
Until $W \leq \frac{w(x(V_U))}{M}$
Set $X_{t+1} \leftarrow X$

Figure 3.6: The general acceptance rejection heat bath Markov chain

portion of the chain to change, we then select a coloring for that portion uniformly at random. We then test a uniform against the weight of that coloring (normalized against M, an upper bound on the weight). If the test accepts, we accept the value, and if it rejects, we choose another coloring and try again.

Although the form is similar, this chain is not the Metropolis-Hastings chain! It is simply another formulation of the heat bath chain, as shown by the following theorem. This theorem is not new, and is a staple of courses in stochastic processes.

Theorem 3.1 *The probability that the node set V_U is given coloring $x(V_U)$ by the acceptance rejection heat bath chain is*

$$P(X(V_U) = x(V_U)) = \frac{w(x(V_U))}{\sum_{x'(V_U)} w(x'(V_U))}.$$

Proof: We do a first step analysis, computing the probability that $X(V_U) = x(V_U)$ and we rejected the color chosen in the first step plus the probability that $X(V_U) = x(V_U)$ and we accept the color in the first step. Let ACCEPT denote the event that we accept the first step, and REJECT denote the event that we reject the first step.

$$\begin{aligned} P(X(V_U) = x(V_U)) &= P(X(V_U) = x(V_U), \text{ACCEPT}) \\ &\quad + P(X(V_U) = x(V_U), \text{REJECT}) \\ &= \frac{1}{Q} \cdot \frac{w(x(V_U))}{M} \\ &\quad + P(X(V_U) = x(V_U) | \text{REJECT}) P(\text{REJECT}) \end{aligned}$$

When the first step rejects, we start over, so

$$P(X(V_U) = x(V_U) | \text{REJECT}) = P(X(V_U) = x_U),$$

and

$$\begin{aligned} P(X(V_U) = x(V_U)) &= \frac{1}{Q} \cdot \frac{w(x(V_U))}{M} \\ &\quad + P(X(V_U) = x(V_U)) P(\text{REJECT}) \\ P(X(V_U) = x(V_U))[1 - P(\text{REJECT})] &= \frac{1}{Q} \cdot \frac{w(x(V_U))}{M} \\ P(X(V_U) = x(V_U)) &= \frac{1}{P(\text{ACCEPT})} \cdot \frac{1}{Q} \cdot \frac{w(x(V_U))}{M} \end{aligned}$$

$$= \frac{1}{\frac{1}{Q}\sum_{x'(V_U)} w(x'(V_U))/M} \frac{1}{Q} \cdot \frac{w(x(V_U))}{M}$$
$$= \frac{w(x(V_U))}{\sum_{x'(V_U)} w(x'(V_U))}$$

which completes the proof. □

As an example, we now present the single site heat bath chain for the hard core gas model as an acceptance rejection chain (the version here is for when $\lambda \geq 1$).

Acceptance rejection single site hard core heat bath chain

Set $X \leftarrow X_t$
Choose a vertex v uniformly at random from V
If a neighbor of v has color 1
  Set $X(v) = 0$
Else
  Repeat
    Choose $c \in_U \{0, 1\}$
    Choose $U \in_U [0, 1]$
  Until $c = 1$ or ($c = 0$ and $U \leq 1/\lambda$)
  Set $X(v) \leftarrow c$
Set $X_{t+1} \leftarrow X$

Figure 3.7: Acceptance rejection single site hard core heat bath chain

3.4 The Swap Move

Broder first introduced what we will call the "swap move" for a chain for generating matchings of a graph. Basically, the idea is that if the color of exactly one neighbor prevents a chosen site from being colored according to the heat bath distribution, then change the color of the neighbor to accommodate the color of the chosen node.

Broder applied this move to construct a chain for sampling from the set of perfect matchings of a graph (used in approximating the permanent of a 0-1 matrix). This chain was later shown by Jerrum and Sinclair [27] to be rapidly mixing.

Dyer and Greenhill applied this technique to the hard core gas model, thereby improving the ability to analyze the chain. We will apply this move to several chains, including the continuous hard core gas model, and the discrete and continuous Widom-Rowlinson mixture models.

Suppose we are using the single site heat bath chain for the hard core gas model. When v has any neighbors colored 1, we are unable to turn v to 1. Suppose however, that exactly one neighbor of v is colored 1, with the rest colored 0. Then a valid move (in that it stays in the state space) would be to swap the color of v with its neighbor with some probability p_{swap}. This chain is presented in figure 3.8

Proving that this chain preserves the stationary distribution may be accomplished via a direct application of reversibility. The new moves are symmetric, so that if x and y are two configurations reached by a swap move then

$$P(x,y) = P(y,x) = p_{swap}\frac{\lambda}{n(1+\lambda)}$$

and $\pi(x) = \pi(y)$. Therefore, clearly $\pi(x)P(x,y) = \pi(y)P(y,x)$ and these moves are symmetric. Since in the old chain there was no probability of moving from x to y, adding these moves preserves the stationary distribution.

Later, we will show how this swap move improves the performance of chains for the continuous hard core gas model and Widom-Rowlinson mixture model.

Dyer and Greenhill hard core chain step

Set $X \leftarrow X_t$
Choose a vertex v uniformly at random from V
Choose U uniformly from $[0, 1]$
Case 1: v has no neighbors colored 1 in X_t, then
  If $U \leq \frac{\lambda}{1+\lambda}$
    Set $X(v) \leftarrow 1$
  Else
    Set $X(v) \leftarrow 0$
Case 2: v has exactly 1 neighbor w colored 1
  If $U \leq p_{swap}\frac{\lambda}{1+\lambda}$
    Set $X(v) \leftarrow 1$, $X(w) \leftarrow 0$
  Else
    Set $X(v) \leftarrow 0$
Set $X_{t+1} \leftarrow X$

Figure 3.8: Dyer and Greenhill hard core chain step

3.5 The Ising and Potts models

We present here the single site heat bath update chain for the Ising and Potts models. The Metropolis-Hastings chain is constructed in a similar fashion.

For a vertex v, let $b_v(c)$ denote the number of neighbors of v which have color c. (The b stands for blocking, as these colors tend to block v from receiving color c in antiferromagnetic models.)

As with many local update chains, the fact that we are picking vertices at random indicates that we must run for at least $n \ln n$ steps before we have any reasonable chance of modifying all the nodes. A nice feature of this chain is that when the temperature T is large enough, then $O(n \ln n)$ steps suffice for this chain to mix.

Single site Potts heat bath chain

Set $x \leftarrow X_t$
Choose a vertex v uniformly at random from V
Choose $c \in_R C$ according to $\frac{\exp(-Jb_v(c)/T)}{\sum_{c' \in C} \exp(-Jb_v(c')/T)}$
Set $x(v) \leftarrow c$
Set $X_{t+1} \leftarrow X$

Figure 3.9: Single site Potts heat bath chain

3.5.1 Antiferromagnetic Potts model at zero temperature

Recall that the problem of uniformly sampling from the Q colorings of a graph may be thought of as the zero temperature limit of the antiferromagnetic Potts model. The heat bath chain for this problem is straightforward, since all proper colorings of the graph have the same weight. We begin by making our notion of blocking more precise.

Definition 3.1 *A color c* blocks *or* neighbors *a node v if a node adjacent to v has color c. A color c which is not blocking for v is* nonblocking.

Single site Q coloring heat bath chain

Set $x \leftarrow X_t$
Choose a vertex v uniformly at random from V
Choose c uniformly from the set of nonblocking colors for v
Set $x(v) = c$
Set $X_{t+1} = x$

Figure 3.10: Single site Q coloring heat bath chain

3.5.2 Swendsen-Wang

Swendsen-Wang is a nonlocal chain for the ferromagnetic Ising and Potts models which utilizes the random cluster viewpoint. It has the advantage of being provably faster than the single site update model under low temperature conditions. Although this model is known not to be rapidly mixing for all temperatures [15], it is in widespread use as a means for generating samples from the Potts model.

The Swendsen-Wang procedure has two phases. In phase 1, the coloring of the graph is used to divide the nodes. An edge is placed between two nodes if the endpoints of those nodes receive the same color. The connected components of this graph are connected sets of nodes of the original graph that have the same color. Each of the edges of this graph are randomly (and independently) removed with probability $1-p$, where $p = 1 - \exp(-1/T)$ is high when the temperature is high and close to 0 when the temperature is low. Once this has been accomplished, the number of connected components will be even higher.

In phase II, the remaining connected components are each assigned a color uniformly and independently from $C = \{0, \ldots, Q-1\}$. All the nodes in that component are assigned this color. For a subset of edges A let $\mathcal{C}(A)$ denote the set of connected components of A, and let C_v denote the lowest numbered node in a connected component. This is written algorithmically in figure 3.11.

Swendsen and Wang [48] both introduced this chain and showed that it has the correct stationary distribution. We will present an analysis of the mixing time of this chain, a result similar to that proved recently by Cooper and Frieze [8].

Swendsen-Wang Step

Set $x \leftarrow X_t$
Let $A \leftarrow \{\{v, w\} \in E : x(v) = x(w)\}$
For each edge $e \in E$ set $U(e) \in_U [0, 1]$
For each node $v \in V$ set $c(v) \in_U \{0, \ldots, Q-1\}$
For each edge $e \in A$
  If $U(e) < 1 - p$
    Set $A \leftarrow A \setminus \{e\}$
For all $C \in \mathcal{C}_A$
  Set $X(w) \leftarrow c(C_v)$ for all $w \in C$
Set $X_{t+1} \leftarrow x$

Figure 3.11: Swendsen-Wang chain

3.6 Sink Free Orientations

As noted in the previous chapter, the problem of generating a sink free orientation of a graph can be seen as have state space $\Omega = \{-1, 1\}^E$ where the two colors for each edge refers to the two possible orientations of that edge. A heat bath chain for this problem picks an edge at random and then randomly picks an orientation that does not create a sink.

Single edge heat bath sink free orientations chain

Set $x \leftarrow X_t$
Choose $e \in_U E$
Choose c uniformly from the set of orientations that
  do not create a sink in the graph
Set $x(e) \leftarrow c$
Set $X_{t+1} \leftarrow x$

Figure 3.12: Single edge heat bath sink free orientations chain

3.7 Widom-Rowlinson

We present several chains for the Widom-Rowlinson model. First we consider the local heat bath chain of [18], then a discrete version of a nonlocal chain for continuous Widom-Rowlinson due to Häggström, van Lieshout, and Møller.

Recall that in the Widom-Rowlinson model nodes are either assigned a color from $\{1, \ldots, Q\}$ which indicates the type of particle occupying the site, or a 0 indicating that that site is unoccupied. If c_i is the number of sites occupied by the ith particle type, then the distribution is $\lambda_1^{c_1} \cdots \lambda_Q^{c_Q}/Z$.

The heat bath chain chooses a node uniformly and then changes the color of that node conditioned on the remaining nodes.

Single site heat bath discrete Widom-Rowlinson chain

Set $x \leftarrow X_t$
Choose $v \in_U V$
Case 1: All neighbors of v are 0
  Choose $c \in_R \{0, \ldots, Q\}$ so that $P(c = 0) = 1/(1 + \sum_i \lambda_i)$
    and $P(c = i) = \lambda_i/(1 + \sum_i \lambda_i)$ for all $1 \leq i \leq Q$
Case 2: All the neighbors of v are either i or 0
  Choose $c \in_R \{0, \ldots, Q\}$ so that $P(c = 0) = 1/(1 + \lambda_i)$
    and $P(c = i) = \lambda_i/(1 + \lambda_i)$
Case 3: Two neighbors of v have different positive colors
  Set $c \leftarrow 0$
Set $x(v) \leftarrow c$
Set $X_{t+1} \leftarrow x$

Figure 3.13: Single site heat bath discrete Widom-Rowlinson chain

The nonlocal chain takes a more direct approach. At each stage all the points of a chosen color are removed from the chain. Then new points of that color are

put back in the chain according to the stationary distribution conditioned on the positions of the remaining colors. In the next chapter we will see that these two

Nonlocal conditioning discrete Widom-Rowlinson chain

Set $x \leftarrow X_t$
Choose $i \in_U \{1, \ldots, Q\}$
For all v such that all neighbors of v are either 0 or i
  Choose $c \in_R \{0, i\}$ so that $P(c = 0) = 1/(1 + \lambda_i)$
    and $P(c = i) = \lambda_i/(1 + \lambda_i)$
  Set $x(v) \leftarrow c$
Set $X_{t+1} \leftarrow x$

Figure 3.14: Nonlocal conditioning discrete Widom-Rowlinson chain

chains have roughly the same performance.

A birth death type process looks at Widom-Rowlinson from a different point of view. Let x_i be the set of nodes colored i in a configuration x. Then the birth death approach says that at each time step, a point of color i is "born" at a specific node with probability $\lambda_i/[n(1 + \sum_i \lambda_i)]$. A particular point "dies" with probability $1/[n(1 + \sum_i \lambda_i)]$, by which we mean the point is removed from x_i by setting the color of the point to 0. Algorithmically, the process is described as follows. Say that a color i is blocked at node v if either v or a neighbor of v has a positive color different from i. Reversibility is easily seen to be satisfied with this chain, which has a slightly worse performance bound than the single site heat bath chain. We introduce it here because it is easy to add a "swap" type move which will increase the range of λ_i over which we may prove that the chain is rapidly mixing.

In the vanilla birth death chain, if a color is blocked when it tries to be born, it

Birth death discrete Widom-Rowlinson chain

Set $x \leftarrow X_t$
Choose $U \in_U [0,1]$
Set $p_0 \leftarrow 1/(1+\sum_i \lambda_i)$
For all $0 < i < Q$
  Set $p_i \leftarrow \lambda_i/(1+\sum_i \lambda_i)$
Choose $i \in_R \{0,\ldots,Q\}$ according to p
Choose $v \in_U V$
If $i = 0$
  Set $x(v) \leftarrow 0$
Else
  If color i is not blocked for node v
    Set $x(v) = i$
Set $X_{t+1} \leftarrow x$

Figure 3.15: Birth death discrete Widom-Rowlinson chain

simply fails to be born. If however, a point is blocked by only a single point in v and the neighbors of v, our swap move will remove the blocking point, and introduce our point in its place.

3.8 The Antivoter model

The antivoter model is one of two models we will consider where the chain itself is part of the model. Throughout this work, the discussions and results for the antivoter model is joint work with Gesine Reinhert.

Naturally, the antivoter model is closely related to the voter model. Suppose that we have $C = \{0,1\}$. At each step of the voter model, a vertex is chosen at random, and the color of the vertex is changed to be the same as a randomly chosen

Birth death swapping discrete Widom-Rowlinson chain

Set $x \leftarrow X_t$
Choose $U \in_U [0,1]$
Set $p_0 \leftarrow 1/(1+\sum_i \lambda_i)$
For all $0 < i < Q$
  Set $p_i \leftarrow \lambda_i/(1+\sum_i \lambda_i)$
Choose $i \in_R \{0,1,\ldots,Q\}$ according to p
Choose $v \in_U V$
If $i = 0$
  Set $x(v) \leftarrow 0$
Else
  Case 1: color i is not blocked for node v
    Set $x(v) \leftarrow i$
  Case 2: color i blocked by exactly one node $w \in \{v \cup$ neighbors of $v\}$
    If $U < p_{swap}$
      Set $x(v) \leftarrow i$
      Set $x(w) \leftarrow 0$
Set $X_{t+1} \leftarrow x$

Figure 3.16: Birth death swapping discrete Widom-Rowlinson chain

neighbor. (This may be used to model forest fires and the spread of infectious agents.) This chain has two absorbing states where the colors of all of the nodes are the same. We will call such a state unanimous.

In the antivoter model, again a vertex is chosen uniformly at random, but now the color of the vertex is changed to the opposite of a randomly chosen neighbor. Unlike the voter model, the antivoter model (on graphs that are nonbipartite) state space is connected (except for the unanimous states, which once left are never reached again) and has a unique stationary distribution on the states [1]. However, this chain is not reversible. Given a state where v is surrounded by nodes colored 1, then v can move to being colored 0, but cannot move back to being colored 1.

The Antivoter model chain step

Set $x \leftarrow X_t$
Choose a vertex v uniformly at random from V
Choose a neighbor w of v uniformly at random
Set $x(v) \leftarrow 1 - x(w)$
Set $X_{t+1} \leftarrow X$

Figure 3.17: The Antivoter model chain step

Therefore $P(x, y) > 0$ but $P(y, x) = 0$ for some x and y, and the chain cannot be reversible. This is the only chain we will consider in this dissertation which is not reversible.

3.9 The List Update Problem

The list update problem is the second problem we will consider where the chain itself is part of the problem. Throughout this work, the discussions and results for the list update problem is joint work with James Fill.

Suppose that we have a list of n items arranged in order $\mu(1), \mu(2), \ldots, \mu(n)$ (in other words, μ is a permutation). Requests come in for items, which must be served by starting at the beginning of the list and moving inwards until the item is found. Therefore it takes $\mu(i)$ time to locate item i.

When an item is found, we are allowed to bring it forward, that is, move it to any position in the list prior to $\mu(i)$ without cost. In addition, we may transpose any two adjacent items in the list at cost 1.

Given this situation, there are several strategies one might consider for rear-

ranging the items based upon requests. For instance, in the move to front (MTF) method, when an item is selected it is moved to the front of the list. Sleator and Tarjan [47] showed that no matter what the sequence of requests, this protocol is worse than the optimal solution where all requests are known ahead of time by at most a factor of 2.

This is a worst case analysis of the problem. Another approach is to use average case analysis, where the set of requests is a stochastic process, and the goal is to find the procedure which minimizes the expected costs of serving the process.

Consider the move ahead one (MA1) protocol. In this method, instead of the selected item moving to the front, it is instead placed ahead 1 position. Unlike the MTF method, this method has no nontrivial guarantee on the value of the solution it delivers. Suppose the list is ordered $1, 2, \ldots, n$, and the set of requests is $n, n-1, n, n-1, \ldots$. Then the MA1 chain will always require n time to service a request, when the optimal (offline) solution is to move $n-1$ to the first position and n to the second and hold them there, resulting in an average cost per query of $3/2$.

The worst case for MA1 is horribly bad, but does the average case do better? First, we must define what we mean by an "average input". This is most often done for the list update problem by considering the input as a stream of independently identically distributed requests. The probability of requesting i at each step is p_i.

With this random set of requests, the list becomes a Markov chain, with random requests altering the chain based on whether we use MTF, MA1, or some other rule.

MA1 list update chain

Choose $i \in_R \{1, \ldots n\}$ where probability of choosing i is p_i
If $\mu(i) > 1$
  Let $a \leftarrow \mu(i)$
  Let j be the item such that $\mu(j) = \mu(i) - 1$
  Swap i and j (set $\mu(j) \leftarrow a$, $\mu(i) \leftarrow a - 1$)

Figure 3.18: MA1 list update chain

MTF list update chain

Request $i \in_R \{1, \ldots, n\}$ with the probability of choosing i is p_i
If $\mu(i) > 1$
  Let $a \leftarrow \mu(i)$
  For all j such that $\mu(j) \leq a$
    Set $\mu(j) \leftarrow \mu(j) + 1$
  Set $\mu(i) = 1$

Figure 3.19: MTF list update chain

To compare the asymptotic behavior of these chains, we compute the stationary distribution of the lists given the input distribution. The asymptotic cost is the expected cost of accessing an item from a stationary permutation. Rivest [45] showed that under this scheme, the $MA1$ chain has lower expected cost at stationarity than the MTF method. Of course, the best ordering under this probability distribution is $1, 2, 3, \ldots, n$ if $p_1 \geq p_2 \geq \cdots \geq n$. However, we assume that we do not have knowledge of the p_i, and we only see the set of random requests.

We will use reversibility to derive the stationary distribution of the $MA1$ chain. For the MA1 list update chain, consider the distribution

$$\pi_{MA1}(\mu) = p_1^{n-\mu(1)} p_2^{n-\mu(2)} \ldots p_n^{n-\mu(n)} / Z.$$

Say that the edge connecting the permutations μ and ν works by switching adjacent items i and j.

$$\begin{aligned}
\pi_{MA1}(\mu)P(\mu,\nu) &= \frac{p_1^{n-\mu(1)}p_2^{n-\mu(2)}\cdots p_n^{n-\mu(n)}}{Z}p_i \\
&= \frac{\left(\prod_{k\neq i,k\neq j} p_k^{n-\mu(k)}\right) p_i^{n-\mu(i)} p_j^{n-[\mu(i)-1]} p_i}{Z} \\
&= \frac{\left(\prod_{k\neq i,k\neq j} p_k^{n-\mu(k)}\right) p_i^{n-[\mu(i)-1]} p_j^{n-\mu(i)} p_j}{Z} \\
&= \pi_{MA1}(\nu)P(\nu,\mu)
\end{aligned}$$

Therefore π_{MA1} is reversible with respect to the MA1 chain. Moreover, this chain is easily shown to be ergodic, so as our notation suggests, π_{MA1} is the unique stationary distribution of the MA1 chain.

The MTF chain is not reversible, since if an item moves from the back to the front, there is no corresponding reverse move which places it in the back again. However, it is easy to see what the probability is that when μ is stationarity $\mu(i) < \mu(j)$. Consider the process $\ldots, \mu_0, \ldots$ which is stationary. With probability 1, sometime before time 0 either i or j was selected. Then $\mu_0(i) < \mu_0(j)$ if at the last choice of i or j, i was selected. Similarly, if j was selected at the last choice of j or i, then $\mu_0(j) < \mu_0(i)$. Therefore, conditioned on the fact that the last choice was either i or j, the probability that i was chosen is just

$$P(\mu_0(i) < \mu_0(j)) = \frac{p_i}{p_i + p_j}.$$

Now suppose that we wish to compute the expected time needed to access an item of μ_0. The time needed to access an item is 1 plus the number of items which

proceed it, so

$$\begin{aligned} E[\text{access } i] &= 1 + \sum_{j \neq i} E[1_{\mu_0(j) < \mu_0(i)}] \\ &= 1 + \sum_{j \neq i} P(\mu_0(j) < \mu_0(i)) \\ &= 1 + \sum_{j \neq i} \frac{p_j}{p_i + p_j} \end{aligned}$$

Each item i is chosen to be accessed with probability p_i, therefore

$$\begin{aligned} E[\text{access}] &= \sum_i p_i \left[1 + \sum_{j \neq i} \frac{p_j}{p_i + p_j} \right] \\ &= 1 + 2 \sum_{1 \leq j \leq i \leq n} \frac{p_j p_i}{p_i + p_j} \end{aligned}$$

a result shown in [30], [19], [7], and [34].

Now lets upper bound $P(\mu_0(i) < \mu_0(j))$ for the MA1 chain. Suppose that we know the positions of all items other than i or j. Denote the smaller of the two remaining positions a, and the larger b. Then there are two possibilities left for i and j, one with i at position a and j at position b, with relative weight $p_i^{n-a} p_j^{n-b}$. The other possibility is that j is at position a and i is at position j, with relative weight $p_i^b p_j^a$. Therefore, the probability that $i < j$ given the position of all items other than i or j is

$$\frac{p_i^{n-a} p_j^{n-b}}{p_i^{n-b} p_j^{n-a} + p_i^{n-a} p_j^{n-b}} = \frac{p_i^{b-a} p_j^0}{p_i^0 p_j^{b-a} + p_i^{b-a} p_j^0}.$$

Now $b - a \geq 1$, so this probability is at most $p_i/(p_j + p_i)$. In fact, many states of positive weight of $b - a > 1$, in which case the probability that i comes before j is higher, at least $p_i^2/(p_j^2 + p_i^2)$. (Note this second expression may be rewritten as

$p_i/[p_j(p_j/p_i) + p_i)]$. The p_j/p_i term in the denominator is less than 1, and so this second fraction is greater than $p_i/(p_j + p_i)$.

Hence the sum over j not equal to i of $E[1_{\mu_0(j)<\mu_0(i)}]$ is lower for the MA1 chain than for the MTF chain, and asymptotically it is guaranteed to perform faster.

Two questions remain. First, asymptotic efficiency is useless if the chain takes too long to converge to its stationary distribution. Actually, we do not need to measure mixing time here, but we do need some way of measuring how quickly the average access time converges to the stationary average access time.

Adjacent transposition for MA1 distribution chain

Set $x \leftarrow X_t$
Choose $v \in_U \{2, 3, \ldots, n\}$
Set $i \leftarrow x(v)$
Set $j \leftarrow x(v-1)$
Choose $U \in_U [0, 1]$
If $U \leq p_i/(p_i + p_j)$
  Set $x(v) \leftarrow j$
  Set $x(v-1) \leftarrow i$
Set $X_{t+1} \leftarrow x$

Figure 3.20: Adjacent transposition for MA1 distribution chain

Second, this analysis shows that the average asymptotic access time of the MA1 chain is lower than that of MTF chain, but it does not tell what that asymptotic efficiency is. Moreover, for either chain to mix, the second to lowest probability item must be selected (otherwise it may lie past the lowest probability item–a situation with low probability). But this second lowest weight can be made arbitrarily small, making the mixing time arbitrarily large. To simulate from the asymptotic distri-

bution of the MA1 chain requires extra subsidiary chains with different transitions but the same distribution as the MA1 chain.

Arbitrary transposition for MA1 distribution chain

Set $x \leftarrow X_t$
Choose $w_1 \in_U \{1, 2, \ldots, n\}$
Choose $w_2 \in_U \{1, 2, \ldots, n\} \setminus \{v_1\}$
Set $v_1 = \min\{w_1, w_2\}$
Set $v_2 = \max\{w_1, w_2\}$
Set $i \leftarrow x(v_1)$
Set $j \leftarrow x(v_2)$
Set $d \leftarrow |v_1 - v_2|$
Choose $U \in_U [0, 1]$
If $U \leq p_i^d/(p_i^d + p_j^d)$
  Set $x(v_2) \leftarrow j$
  Set $x(v_1) \leftarrow i$
Set $X_{t+1} \leftarrow x$

Figure 3.21: Arbitrary transposition for MA1 distribution chain

For example, the adjacent tranposition chain does not request a specific item, but randomly chooses a position, then swaps the position and the position immediately preceding it according to the stationary distribution. Of course, there is no reason why we must use adjacent transpositions. Arbitrary transpositions work just as well.

Note that the average value of d chosen in this fashion is $O(n)$. For weights which are different from one another by a factor of more than $(1 + c/n)$, this means that the chance that the lowest weight item is placed first is on the order of e^{-c}.

3.10 Hypercube slices

The heat bath chain for the hypercube is simple: choose a coordinate of the chain uniformly at random, and then change that coordinate (again uniformly at random) to either 0 or 1. When we are restricted to $L \leq \sum_i x(i) \leq U$, we must modify the heat bath chain to take this into account. When a switch would have the value of $\sum_i x(i)$ moving below L, we simply hold at the current value rather than making the switch. Similarly for when we hit U. This chain is of interest primarily because of

Heat bath hypercube slice chain

Set $x \leftarrow X_t$
Choose $i \in_R \{1, \ldots, n\}$
Choose $c \in_R \{0, 1\}$
If $L \leq c + \sum_{j \neq i} x(j) \leq U$
  Set $x(i) \leftarrow c$
Set $X_{t+1} \leftarrow x$

Figure 3.22: Heat bath hypercube slice chain

its relationship to algorithms for the Ising model when the magnetization is required to be constant (see [50] for details).

3.11 What remains: mixing time

Creating Markov chains is easy, but showing that the chains in this section mix in a reasonable amount of time is not. In the next chapter we introduce the idea of bounding chains for computing the mixing time, and we present bounding chains for each of the chains considered here.

Chapter 4

Bounding Chains

> The wolf Fenris ... broke the strongest fetters as if they were made of cobwebs. Finally ... the mountain spirits .. made for them the chain called Gleipnir ... when the gods asked the wolf to suffer himself to be bound with it ... he suspected their design, fearing enchantment. He therefore only consented to be bound with it upon condition that one of the gods put his hand in his mouth. Tyr alone has courage enough to do this. But when the wolf found that he could not break his fetters ... he bit off Tyr's hand, and he has ever since remained one-handed.
>
> *-Norse Myth, Bullfinch's Mythology*

Fortunately the modern practitioner of bounding chain techniques does not need to make quite as large a sacrifice as Tyr, but a constant factor loss of speed is sometimes involved. In this chapter we describe the purpose and power of the bounding chain technique.

To accurately describe why bounding chains are important, we first take a look at how Markov chains are actually simulated. Usually, random numbers are drawn which determine a function. This function is then applied to the current state to determine the next state of the chain.

Definition 4.1 *Let f be a random function from Ω into itself. We say that f is* consistent *with a Markov chain with transition matrix P if $P(f(x) = y) = P(x, y)$ for all x, y.*

Recall the Dyer-Greenhill chain for the hard core gas model of the previous chapter (Figure 3.8). Once the random node v is decided along with the uniform random variable U, we have completely chosen the function f for that time step. No matter what the state, knowing v and U allows to determine the next state of the chain.

One method, then of simulating a Markov chain is for each time t, choose a random function f_t (independent of all other $f_{t'}$) and set $X_{t+1} = f_t(X_t)$. Since each f_t is independent, this clearly has the Markov property. In general, we do not require that each f_t be identically distributed; all that is needed is that they are consistent with the Markov chain. However, in this work all of the chains which we consider can be simulated using functions f_t which are independent and identically distributed. If $X_0 = x$, then $X_t = f_{t-1}(f_{t-2}(\cdots f_0(x)\cdots))$. As shorthand, set $F_b^a = f_a \circ f_{a+1} \circ \cdots \circ f_{b-1}$ so that $X_t = F_t^0$.

Suppose that we choose two starting values for the chain X_0 and Y_0 and set $X_t = F_t^0(X_0)$ and $Y_t = F_t^0(Y_0)$. Then if $X_t = Y_t$ for some value of t, then $X_t' = Y_t'$

for all $t' \geq t$. This motivates the following definition.

Definition 4.2 *Suppose that a Markov chain is defined via a random sequence of functions* $\ldots, f_{-1}, f_0, f_1, \ldots$ *consistent with a transition matrix* P. *We refer to such a chain as a* complete coupling chain. *If* $F_t^a(X_a) = F_t^a(Y_a)$, *we say that the stochastic processes* X *and* Y *have* coupled *at time* t, *and denote the first time this occurs by* T_C. *If* F_t^a *is a constant function, then we say that the chain has* completely coupled *at time* t, *and we denote the first time this occurs by* T_{CC}.

The notion of complete coupling will be of paramount importance to us. It will not only allow for computer experiments and analytical arguments that determine upper bounds on the mixing time of a Markov chain, but also give perfect sampling algorithms. The perfect sampling aspects of complete coupling will be explored in Chapters 6 and 7. Here we concentrate on the experimental and analytical determination of mixing times.

The following theorem is an important tool in determining the mixing time of a Markov chain.

Theorem 4.1 *Suppose that we have a completely coupling chain,* X_0 *has some arbitrary distribution over* Ω *and* Y_0 *has a stationary distribution* π. *Then*

$$||F_t^0(X_0) - \pi||_{TV} \leq P(X_t \neq Y_t) = P(T_C > t).$$

That is, the total variation distance between X_t *and the stationary distribution is bounded above by the probability that the* X *and* Y *processes have coupled.*

This theorem was originally proved by Doeblin [10]. Aldous [1] is usually credited with popularizing this theorem as a tool for bounding the mixing time of a Markov

chain in the context of MCMC methods. We shall use this theorem throughout this work, and so here we present the proof, which is straightforward.

Proof: Given that Y_0 has stationary distribution π, $P(Y_t \in A) = \pi(A)$ for all $A \subset \Omega$. Also, $T_C > t$ and $T_C \leq t$ are disjoint sets, so

$$\begin{aligned}
||F_t^0(X_0) - \pi||_{TV} &= \max_{A\subset\Omega} |P(X_t \in A) - \pi(A)| \\
&= \max_{A\subset\Omega} |P(X_t \in A, T_C > t) + P(X_t \in A, T_C \leq t) - P(Y_t \in A)| \\
&= \max_{A\subset\Omega} |P(X_t \in A, T_C > t) + P(Y_t \in A, T_C \leq t) - \\
&\quad (P(Y_t \in A, T_C > t) + P(Y_t \in A, T_C \leq t)| \\
&= \max_{A\subset\Omega} |P(X_t \in A, T_C > t) - P(Y_t \in A, T_C > t)| \\
&= \max_{A\subset\Omega} [\max\{P(X_t \in A, T_C > t), P(Y_t \in A, T_C > t)\}] \\
&\leq \max_{A\subset\Omega} P(T_C > t) \\
&= P(T_C > t).
\end{aligned}$$

□

Another way of stating this result is that the mixing time $\tau_{TV}(\epsilon)$ is bounded above by $\min_t\{P(T_C > t) \leq \epsilon\}$. Note that T_C is itself bounded above by T_{CC}, that is, if every process started at time 0 has coupled then clearly a pair of processes X and Y have coupled no matter what distribution they have at time 0. Therefore we have proved the following corollary.

Corollary 4.1 *Let $\tau_{TV}(\epsilon)$ be the mixing time of the chain with respect to total variation distance. Then if at time t, $P(T_{CC} > t) < \epsilon$,*

$$\tau_{TV}(\epsilon) < t.$$

Note that this statement does not depend at all upon the stationary distribution of the chain, it is entirely dependent upon T_{CC}. Therefore, if we had a method for efficiently determining whether or not $T_{CC} > t$, then we would immediately have a procedure for estimating the mixing time of the chain.

Experimental Upper Bounds on the Mixing Time

Input: ϵ

For $i = 1$ to k
  Set $t = 0$
  Repeat
    Set $t = 2t + 1$
    Compute F_t^0
  Until F_t^0 is constant
  Set $t_i = t$
Set t_{est} to be the ϵk largest value of t_i

Figure 4.1: Experimental Upper Bounds on the Mixing Time

Johnson [28] proposed the algorithm in Figure 4 for a specific case where it is very easy to determine if $t < T_{CC}$, the case of monotonicity.

4.1 Monotonicity

A partially ordered set, or poset, consists of a base set of elements Ω together with a partial order $\preceq$ satisfying reflexivity ($x \preceq x$), antisymmetry ($x \preceq y$ and $y \preceq x$ implies $x = y$), and transitivity ($x \preceq y$ and $y \preceq z$ implies $x \preceq z$). An example of a poset is when Ω is all subsets of $\{1, \ldots, n\}$ and $x \preceq y$ if and only if $x \subseteq y$. Note that this particular subset has an element which is greater than all other elements,

as well as one which is smaller than all other elements.

Definition 4.3 *The element* $\hat{1}$ *is* maximal *for* $(\Omega, \preceq)$ *if for all* $x \in \Omega$, $x \preceq \hat{1}$. *Similarly,* $\hat{0}$ *is a* minimal *element of the poset if* $\hat{0} \preceq x$ *for all* $x \in \Omega$.

In our example of the poset of all subsets of a set, $\hat{0} = \emptyset$ is minimal and $\hat{1} = \{1, \ldots, n\}$ is maximal.

Often it is possible to place a partial order on the state space of the Markov chain such that moves on the chain respect the partial order.

Definition 4.4 *Suppose the functions* $\ldots, f_{-1}, f_0, f_1, \ldots$ *determine a completely coupling Markov chain. We will say that the Markov chain* respects *or* preserves *the partial order* $\preceq$ *if for all* x, y *in* Ω *and all times* t,

$$x \preceq y \Rightarrow f_t(x) \preceq f_t(y).$$

Now a simple induction argument shows that if the Markov chain respects a partial order, that $X_0 \preceq Y_0$ implies that $F_t^0(X_0) \preceq F_t^0(Y_0)$. Johnson's approach [28] was to use a maximal element, a minimal element, and a Markov chain which preserves the partial order to "squeeze" all the elements of Ω together.

Suppose that $\hat{0}$ and $\hat{1}$ are minimal and maximal elements of Ω. Then by definition $\hat{0} \preceq x \preceq \hat{1}$ for all $x \in \Omega$. Suppose that at time t, $F_0^t(\hat{0}) = F_0^t(\hat{1})$. Then since for all times t, $F_0^t(\hat{0}) \preceq F_0^t(x) \preceq F_0^t(\hat{1})$, we know that $F_t^0(\hat{0}) = F_t^0(x) = F_t^0(\hat{1})$ for all $x \in \Omega$. In other words, F_t^0 is constant and $T_{CC} \leq t$. Moreover, we know that the smallest time $\hat{0}$ and $\hat{1}$ meet is in fact T_{CC}, since complete coupling cannot occur while two processes have not coupled.

Example. The simplest example of a finite monotonic chain is the random walk on $\{1, \ldots, n\}$. Suppose that we are at state i, then with probability $1/2$ the next state is $\max\{i-1, 1\}$ and with probability $1/2$ the next state is $\min\{i+1, n\}$. Then with the partial order $i \preceq j$ if $i \leq j$, this Markov chain is monotonic. It has a maximal element n and a minimal element 1.

Other examples of chains which admit a monotonic partial order include the ferromagnetic Ising model and the discrete Widom-Rowlinson model on a mixture of two types. However, a wide variety of chains do not have a simple monotonic structure, and so a more general version of monotonicity was developed.

4.2 Antimonotonicity

Kendall [29] appears to be the earliest to take advantage of antimonotonicity in designing an exact sampling algorithm, although Häggström and Nelander [18] were the first to formally define the notion.

Recall that for most cases of interest, the sample space is just C^V. The partial orders for these state spaces are often constructed by putting an order $\leq$ on C, and then saying that $X \preceq Y$ if $X(v) \leq Y(v)$ for all $v \in V$. For chains on these spaces with this type of partial order, Häggström and Nelander defined the notion of antimonotonicity as follows.

Definition 4.5 *Consider a Markov chain on $\Omega = C^V$, and a partial order between configurations on any subset of V. The chain is* antimonotonic *if for all configurations $x \preceq y$ and v such that $f_t(x)(v) \neq x(v)$, we have that $f_t(x)(v) \geq f_t(y)(v)$.*

Note that in monotonic cases with this type of partial order, we have that $x \preceq y$ implies that $f_t(x)(v) \leq f_t(y)(v)$. That is why we refer to the property with the inequality in the opposite direction as antimonotonicity.

This definition is usually only applicable when the number of nodes that change color from step to step is small, such as in single node (or edge) update chains. When a single node changes, then all we need check is that for the node v that is altered, $x \preceq y$ implies that $x(v) \geq y(v)$.

We keep track of two states T_t (for top) and B_t (for bottom). These states will have the property that if $B_t \preceq X_t \preceq T_t$, then $B_{t+1} \preceq X_{t+1} \preceq T_{t+1}$. This looks similar to the monotonic case, but B and T do not evolve according to the Markov chain. At time t, let $T' = f_t(T_{t+1})$, and $B' = f_t(B_{t+1})$. For all v, set $T_{t+1} = \max\{T'(v), B'(v)\}$ and $B_{t+1} = \min\{T'(v), B'(v)\}$. Then we have guaranteed that $B_t \preceq X_t \preceq T_t \Rightarrow B_{t+1} \preceq X_{t+1} \preceq T_{t+1}$.

The algorithm proceeds just as in the monotonic case. If we can find initial states such that $B_0 \preceq x \preceq T_0$ for all $x \in \Omega$, then $B_t = T_t$ implies that F_t^0 is constant and we are done.

(Note that the above algorithm actually works for chains which are not antimonotonic, such as the Luby-Vigoda chain for the hard core gas model. Since all of these methods are specific cases of bounding chains, we will not explore them in detail here.)

The single site heat bath chain for the hard core model is an example of an antimonotonic chain. Here, a single node is chosen to be changed. If the node turns to 1, then all the neighboring nodes must have been 0. Thus to get high values

(1) at a node, all the neighbors must have small values (0). This is the essential notion of antimonotonicity, that the particular node value will be higher if the rest of the configuration is smaller, where $\leq$ is the measure of whether a value is higher or smaller. Unfortunately, many chains of interest do not obey either monotonicity nor antimonotonicity.

4.3 The bounding chain approach

The concept of bounding chains generalizes the idea of monotonicity and antimonotonicity, and is applicable to a vast number of chains. In its basic form, the idea for discrete chains was independently introduced in [21] by the author and [18] by Häggström and Nelander. In [23], [24], and [22], the author analyzed properties of the bounding chain more fully, and the idea is applied to a wide variety of different problems, including continuous Markov chains.

The original idea was developed in [21] and [18] in order to develop an exact sampling algorithm for the proper k colorings of a graph, which is neither monotonic nor antimonotonic when $k \geq 3$.

The concept is straightforward. Basically, instead of trying to show that complete coupling has occurred for every node v of the graph, we work on showing that the complete coupling has occurred for a specific node v. For some nodes v, complete coupling will have occurred, and for some it will not. Once complete coupling has occurred for every node v, we know that it has occurred for the entire Markov chain.

Given a Markov chain M_1 running on $\Omega = C^V$ with transition matrix P, we introduce a new Markov chain M_2 which runs on $\Omega_2 = \mathcal{P}(C)^V$, where $\mathcal{P}(C)$ denotes the set of nonempty subsets of C. Given that Ω is finite, this new chain will be finite as well, and a configuration on M_2 consists of giving each node a set of colors drawn from C. In other words, each node of a configuration in the chain M_2 has associated with it a color set which ranges from single colors up to the entire set C.

Definition 4.6 *Let $x \in \Omega$ and $y \in \Omega_2$. Say that $x \in y$, or x is* in *y if for all nodes v, $x(v) \in y(v)$.*

In order to obtain our experimental bounds on the mixing time, we need to show that F_t^0 is a constant. The size of Ω is often exponential in the input, therefore keeping track of $\cup_{x\in\Omega} F_t^0(x)$ is prohibitively expensive. However, keeping track of $\cup_{x\in\Omega} F_t^0(x)(v)$ is much easier. That is, we keep track of the total number of possible colors of each individual node. When $|\cup_{x\in\Omega} F_t^0(x)(v)| = 1$ for every node v, we know that $F_t^0(x)$ is constant over $x \in \Omega$. The bounding chain is the tool that makes this approach possible.

Definition 4.7 *Let M_1 on C^V and M_2 on $\mathcal{P}(C)^V$ be complete coupling chains using the random sequences $\ldots, f_{-1}, f_0, f_1, \ldots$ and $\ldots, g_{-1}, g_0, g_1$ respectively. Then we say that M_2 is a* bounding chain *for M_1 if for all t,*

$$x \in y \Rightarrow f_t(x) \in g_t(y),$$

and

$$(\forall v)(|y(v)| = 1) \Rightarrow g_t(y)(v) = f_t(y)(v).$$

The first property of bounding chains says that after one step of the chain, if $X_t \in Y_t$, then X_{t+1} will be in Y_{t+1} if the Y process is being run on a bounding chain for the X process.

The second property says that if the bounding chain has evolved to the point where the color set at each point is a single color, then the chain evolves exactly as the chain that it bounds. Therefore, the set of states where $|y(v)| = 1$ for all v is absorbing for M_2.

A simple induction argument extends the first property from single time steps to multiple steps.

Fact 4.1 *Suppose that M_2 is a bounding chain for M_1. Let X be a process run on M_1 and Y a process on M_2. Then for all $t \geq 0$.*

$$X_0 \in Y_0 \Rightarrow X_t \in Y_t.$$

Note that when $|y(v)| = 1$ for $y \in \Omega_2$, we know that if $x \in y$, then the value of x must be the singleton element in $y(v)$. This leads to the following definition.

Definition 4.8 *We say that the value of node v is* determined *by $y \in \Omega_2$ if $|y(v)| = 1$. If $|y(v)| > 1$, we say that the value of v is* unknown.

Another way of stating the second property of bounding chains is that once all of the nodes of the graph are determined, they stay determined. Once there are no unknown nodes, no node will ever be unknown again.

The following theorem is the reason bounding chains are so useful.

Theorem 4.2 *Suppose that M_2 is a bounding chain for M_1 and that $Y_0(v) = C$ for all nodes v. Then if Y_t determines v for all $v \in V$, $F_t^0(x)$ is constant.*

Proof: Suppose that $Y_0(v) = C$ and that at some time t, $|Y_t(v)| = 1$. Let $x \in \Omega$. Then $x \in Y_0$, so $F_t^0(x) \in Y_t$ by Fact 4.1. However, only one element of Ω is in Y_t since $|Y_t(v)| = 1$ for all nodes v. Therefore, $F_t^0(x)$ is that single element regardless of x, and so must be constant. $\square$

We claimed earlier that bounding chains generalized the notion of monotonicity and antimonotonicity. In monotonic chains, we trapped the states between two other states $F_0^t(\hat{1})$ and $F_0^t(\hat{2})$. Suppose that the partial order $\preceq$ is derived from a partial order $\leq$ on the color set. Set $Y_t(v) = \{c \in C : F_0^t(\hat{0})(v) \preceq c \preceq F_0^t(\hat{0})(v)\}$. Each node is given of a color set, and by the monotonic behavior of the chain, $X_t \in Y_t \Rightarrow X_{t+1} \in Y_{t+1}$. Furthermore, for $|Y_t(v)| = 1$ to occur for all v, $F_t^0(\hat{0}) = F_t^0(\hat{1})$, meaning that Y_t now evolves exactly as $X_t = F_t^0(X_0)$ does. Therefore Y_t is a bounding chain.

Antimonotonicity is similar. Again we set $Y_t(v) = \{c \in C : B_t(v) \leq c \leq T_t(v)\}$, and once more it is easy to see that because of antimonotonicity this is a valid bounding chain.

One chain which is neither monotonic nor antimonotonic is the Dyer-Greenhill chain of the previous chapter for generating samples from the hard core gas model. The method used by Dyer and Greenhill for proving that the mixing time of the chain was polynomial for restricted values of λ was path coupling [13]. Here we develop a bounding chain for this problem. This bounding chain will not only allow us to prove same theoretical mixing time result for the chain as in [13], but it will also give us a means for experimentally determining the mixing time when λ is

outside this restricted range (and it will give an exact sampling algorithm for this problem as we show in later chapters.)

4.3.1 Bounding the Dyer-Greenhill Hard Core chain

Recall that the state space for the Dyer-Greenhill chain is $\{0,1\}^V$, and so our state space for the bounding chain will be $\{\{0\},\{1\},\{0,1\}\}^V$. For notational convenience, we will use ? to denote the set $\{0,1\}$. If a node is assigned a ?, that indicates that F_t^0 at that node might not be constant.

The Dyer-Greenhill chain 3.8 chooses a vertex v uniformly at random, then decides whether to attempt to turn that node color to 1 or 0. If the attempt is to turn v to 0, the colors of the neighbors of the node do not matter; v is colored 0 regardless of what the neighbors are. If the attempt is to turn v to 1 and all neighbors are 0 then v is colored 1. If at least 2 neighbors of v are colored 1 then v is colored 0. If exactly one neighbor of v is colored 1, another roll is made which if successful means that v is colored 1, and this neighbor is switched from being colored 1 to being colored 0.

Most of the bounding chains we will develop are formed from the same process. Suppose that we wish to generate a bounding function g for a Markov chain with function f. Then $g(y)(v) = \{f(x)(v)|x \in y\}$ is such a bounding function. That is, for each possible x that could be in y, compute $f(x)$. Then for each node v, let the new value of $y(v)$ be the union over all x in y of $f(x)(v)$. Then ensures that both bounding chain properties are true. In constructing a bounding chain, first instantiate all the random variables that are needed to determine f. Then apply

f to all possible x in y. Then for each node, write down all possible outcomes for $f(x)$ at that node.

Fortunately, for most chains of interest the value of v at the next step is entirely determined by the values of the neighbors of v. We do not have to examine all $x \in y$, rather, we only have to look at all possible values for the neighbors of v that lie in y.

For the Dyer-Greenhill chain, we will first describe in words the behavior of the bounding chain. This is given in algorithmic form in Figure 4.2. We need to examine all possible types of outcomes for the function f. Suppose that the random variables comes out to be v, and $U > \lambda/(1+\lambda)$. Then $f(x)$ says to color node v with 0. Therefore $\cup_{x \in y} f(x)(v) = 0$, and $\cup_{x \in y} f(x)(w) = y(w)$ for all $w \neq v$. In this case we change $y(v)$ to $\{0\}$ and leave all the rest of the nodes unchanged.

For all of the following cases suppose that $U \leq \lambda/(1+\lambda)$. If all the neighbors w of v satisfy $y(w) = \{0\}$, the only configurations x in y also have all neighbors of v colored 0, so $x(v)$ always changes to 1. Therefore $\cup_{x \in y} f(x)(v) = \{1\}$, so $g(y)(v) = \{1\}$. All the other nodes remain unchanged.

If at least two neighbors of v have $y(v) = \{1\}$, then so do all x in y, and again v is always colored 0, so $y(v) = \{0\}$. The tricky cases arise when some of the neighbors of v are colored $? = \{0, 1\}$.

Suppose that at least two neighbors of v are colored $?$ in y and the rest are colored 0. Let x_1 be the configuration where all neighbors of v are colored 0, and let x_2 be the configuration such that $y(v) =? \Rightarrow x_2(v) = 1$. Then $x_1 \in y$ and $x_2 \in y$. Sadly, we note that $f(x_1) = 1$ while $f(x_2) = 0$, so we must color $y(v)$ with $? = \{0, 1\}$

at the next step.

These are all the functions f where we do not attempt a swap. Suppose now that exactly 1 neighbor w of v is colored $\{1\}$, and some other neighbor is colored ?. Then if we do not attempt a swap (because $U > \lambda/[4(1+\lambda)]$), we know v must be colored 0 because of the neighbor colored $\{1\}$, so $y(v) = \{0\}$ at the next step. If we do attempt a swap, then matters turn ugly. Again let x_1 be the configuration where a ? is resolved to a 0, and x_2 be the configuration where a ? is resolved to a 1. Then $f(x_1)(v) = 1$, $f(x_2)(v) = 0$, and $f(x_1)(w) = 0$ and $f(x_2)(w) = 1$. Therefore both v and w are colored ? in the next step.

However, the swap can be helpful for some configurations. Suppose that we attempt a swap and exactly one neighbor w of v is colored ?, while the rest are 0. Then again using x_1 (a ? resolves to a 0) and x_2 (a ? resolves to a 1), we see that $f(x_1)(w) = f(x_2)(w) = 0$ and $f(x_1)(v) = f(x_2)(v) = 1$, so both v and w are determined in the next time step.

Theorem 4.3 *With probability 1, the values of all the nodes will be determined in finite time.*

Proof: There is a small but positive chance that the next n moves will be to change each of the nodes to color 0 (where this chance is at least $n!/[n^n(1+\lambda)^n]$). When that happens, all of the nodes are determined and once they all are determined they all stay determined. Therefore, the time needed for complete determination is stochastically bounded by a geometric random variable, and so is finite with probability 1. □

Of course we would like to make a much stronger statement than that, and in fact we can. Let Δ refer to largest degree in the graph (where the degree of a node is the number of undirected edges adjacent to that node).

Theorem 4.4 *Let T_{BC} denote the time that all the nodes are completely determined by the bounding chain, and so $T_{CC} \leq T_{BC}$. If $\lambda \leq 2/[\Delta - 2]$, then*

$$E[T_{BC}] \leq 4 \min \left\{ \frac{2(1+\lambda)}{2-(\Delta-2)\lambda} n \log_2(2n), 2n^2(1+\lambda) \right\}.$$

Therefore the algorithm runs in $O(n \ln n)$ steps when λ is bounded away from $2/[\Delta - 2]$, and $O(n^2\lambda)$ steps in general. Moreover, T_{BC} is rarely very much larger than its expected value:

$$P[T_{BC} > keE[T_{BC}]] \leq e^{-k}$$

The proof of this theorem when $\lambda < 2/[\Delta - 2]$ will be straightforward, but when $\lambda = 2/[\Delta - 2]$ (as in the case where we are attempting to sample independent sets uniformly over graphs of bounded degree 4) will require a bit of machinery, and so we take a brief look at martingales.

4.3.2 Martingales

Let D_t denote the set of unknown nodes at a particular time step t, and A_T denote the set of determined nodes (D stands for "disagree" and A stands for "agree"). At time 0, all the nodes are colored ?, and so $|D_0| = n$. To prove the theorem, we would like to be able to show that on average the size of D_t is decreasing at each step. This is the motivation behind the introduction of martingales.

Definition 4.9 *We say that a stochastic process* $(\ldots, X_{-1}, X_0, X_1, \ldots)$ *is a* supermartingale *if (with probability 1)*

$$E[X_{t+1}|\sigma(\ldots, X_{t-1}, X_t)] \leq X_t.$$

In order to prove bounds on the behavior of supermartingales, we need the concept of a stopping time.

Definition 4.10 *Suppose that for all* t*, the event* $\tau \leq t$ *is* $\sigma(\ldots, X_{t-1}, X_t)$ *measurable. Then* τ *is a stopping time of the process.*

Roughly speaking, τ is a stopping time if at all times t we can determine if τ has occurred or not. An example a stopping time is the first time that the Markov chain enters a particular state. Given the past history of the chain, we may determine whether or not that state has already been entered.

The supermartingale property says that for single time steps the expectation of the value of the stochastic process decreases on average. This is also true for longer time intervals, and for stopping times [38].

Fact 4.2 *Let* $\{X_i\}$ *be a supermartingale and let* τ *be a stopping time greater than* i*, then*

$$E[X_\tau] \leq E[X_i].$$

The following theorem is well known, we reprove it here as an illustration of a proof technique which we will use later.

Theorem 4.5 *Suppose that we have a supermartingale on* $\{0, \ldots, n\}$*, and that 0 is an absorbing state. Furthermore, suppose that* $P(X_{t+1} \neq X_t) \geq p$*. Then the expected time until the stochastic process reaches 0 is* $O(n^2/p)$*.*

The proof technique works by bounding the number of times that the process moves across the interval from i to $i+1$.

Definition 4.11 *We say that a stochastic process X* upcrosses *(a,b) at time t, $X_t \leq a$ and $X_{t+1} \geq b$. Let $U(a,b)$ be the number of times $(X_0, X_1, \ldots)$ upcrosses (a,b). Similarly, a* downcrossing *occurs when $X_t \geq b$ and $X_{t+1} \leq a$.*

Bounding the expected number of upcrossings in a general supermartingale is normally accomplished by means of the upcrossing lemma [38]. However, for the special case for which we need it, we can prove a stronger result directly.

Lemma 4.1 *For a supermartingale on $0, 1, \ldots$ such that 0 is an absorbing state:*

$$E[U(i, i+1)] \leq i+1.$$

Proof: Suppose that $X_t \leq i$. Let τ_{i+1} be the first time that the process moves to a state greater than or equal to $i+1$. Let τ_0 be the first time that the that the process hits 0. Then $\tau = \min\{\tau_0, \tau_{i+1}\}$ is also a stopping time, and

$$\begin{aligned} E[X_t] &\geq E[X_\tau] \\ i &\geq P(\tau = \tau_0)E[X_{\tau_0}] + P(\tau = \tau_n)E[X_{\tau_n}] \\ i &\geq P(\tau = \tau_0) \cdot 0 + P(\tau = \tau_{i+1})E[X_{\tau_{i+1}}] \\ \frac{i}{i+1} &\geq P(\tau = \tau_{i+1}) \end{aligned}$$

Therefore, the probability that another upcrossing occurs is at most $i/(i+1)$, making the number of upcrossings stochastically bounded above by a geometric random

variable with parameter $1/(i+1)$. Therefore the expected number of upcrossings is bounded above by $i+1$. $\square$

Armed with these lemmas, we now proceed to prove Theorem 4.5.

Proof of Theorem 4.5: We bound the expected number of times $t > 0$ that $X_t = i$. This is a random variable which we will call N_i. Note that $X_t = i$ for one of three reasons. First, we could have had $X_{t-1} = i$ and the state stayed the same. By assumption the probability of staying at the same value is at most $(1-p)$. Second, it might have been that $X_{t-1} < i$ but $X_{t+1} = i$. This implies that there was a $(i-1, i)$ upcrossing. Finally, we could have had $X_{t-1} > i$ but $X_{t+1} = i$, which implies that there was a $(i, i+1)$ downcrossing. Note that the number of $(i, i+1)$ downcrossings is bounded above by one more than the number of $(i, i+1)$ upcrossings.

Taken together, we have that

$$\begin{aligned} E[N_i] &\leq (1-p)E[N_i] + E[U(i-1,i)] + (E[U(i,i+1)]+1) \\ E[N_i] &\leq \frac{1}{p} \cdot [E[U(i-1,i)] + E[U(i,i+1)] + 1] \\ &\leq \frac{1}{p} \cdot [i + i + 1 + 1] \\ &= \frac{2i+2}{p}. \end{aligned}$$

Since we know that X only takes on values from 0 to n, we have that the expected number of steps before hitting 0 is

$$\begin{aligned} E[\tau_0] &= \sum_{i=1}^{n} E[N_i] \\ &\leq \sum_{i=1}^{n} \frac{2i+2}{p} \end{aligned}$$

$$= \frac{(n+1)(n+2)-1}{p},$$

which is $O(n^2/p)$. □

A minor modification to the above theorem will have a noticeable impact on our ability to prove running time bounds. The only difference is that now the probability that we move is dependent upon the value of X_t.

Theorem 4.6 *Suppose that we have a supermartingale on $\{0,\ldots,n\}$ with 0 an absorbing state. Furthermore, suppose that $P(X_{t+1} \neq X_t) \geq p_{X_t}$. Then the expected time until the stochastic process reaches 0 is*

$$\sum_{i=0}^{n} \frac{2(i+1)}{p_i}.$$

Proof: As with the proof of Theorem 4.5, we proceed by bounding the expected value of N_i. Here, however, the probability of moving from N_i at any given time step is at least p_i.

$$\begin{aligned} E[N_i] &\leq (1-p_i)E[N_i] + [U(i-1,i) + U(i,i+1) + 1] \\ &\leq \frac{1}{p_i}(2i+2) \end{aligned}$$

Therefore $E[\tau_0] = \sum_{i=1}^{n} \frac{2(i+1)}{p_i}$, as desired. □

When we have a stochastic process which is better than a supermartingale, we use Wald's Inequality (see [38]).

Theorem 4.7 *Suppose that $X = (X_0, X_1, \ldots)$ is a nonnegative stochastic process such that $E[X_{t+1} - X_t | X_t] \leq -q$ when $X_t > 1$. Then if τ_0 is the first time that $X_t = 0$,*

$$E[\tau] \leq E[X_0]/q.$$

4.3.3 Running time of bounding chain for Dyer-Greenhill

We need one more well known fact from elementary probability before proving Theorem 4.4.

Fact 4.3 Markov's Inequality. *Let X be a nonnegative random variable with positive expected value $E[X]$. Then*

$$P(X \geq a) \leq \frac{E[X]}{a}.$$

Proof of Theorem 4.4: For convenience, let $\mathcal{F}_t$ be the σ-algebra generated by $\{X_{t'} | t' \leq t\}$. As before, let D_t be the set of unknown nodes at time t, and let $A_t = V \setminus D_t$ be the set of determined nodes. We will show that $|D_t|$ is a supermartingale. The second bounding chain property guarantees that when $|D_t| = 0$, it stays 0 and so 0 is an absorbing state for the $|D_t|$ process.

From the algorithm, it is clear that the size of D_t changes by at most two at each step. $|D_t|$ increases in size when nodes move from A_t to D_{t+1}, and decreases when nodes move from D_t to A_{t+1}. Let V_t be the random vertex chosen at time t by the algorithm. Then $P(v = V_t) = 1/n$ for all v, and

$$E[|D_{t+1}| \mid \mathcal{F}_t] = |D_t| + \frac{1}{n} \sum_v E[|D_{t+1}| - |D_t| \mid \mathcal{F}_t, v = V_t].$$

From Figure 4.2 we know that at each step, our choice of v places us in one of six disjoint cases depending on the color sets of the neighbors of v. So for v satisfying cases 1 through 6, we compute $E[|D_{t+1}| - |D_t| \mid v = V_t, \mathcal{F}_t]$.

Suppose first that $v \in A_t$. Let $c(v)$ denote the case that v falls into, from 1 through 6. Let C_i denote the value of $E[|D_{t+1}| - |D_t| \mid v = V_t, \mathcal{F}_t]$ given that

$c(v) = i$, and v is in A_t. Then in cases 1, 2, and 3 the node always stays in A_t, so our expectation is 0 and $C_1 = C_2 = C_3 = 0$. If v is in case 4, let w be the single unknown neighbor. With probability $\lambda/(4(1+\lambda))$, v attempts to swap with w and so w moves to A_t. With probability $3\lambda/(4(1+\lambda))$, v attempts to turn to color 1, resulting in v becoming unknown. Hence

$$\begin{aligned} C_4 &= \frac{3\lambda}{4(1+\lambda)} - \frac{\lambda}{4(1+\lambda)} \\ &= \frac{\lambda}{2(1+\lambda)}. \end{aligned}$$

In case 5, switching v leads to both v and its neighbor colored 1 to be moved to D_t, so $C_5 = 2\frac{\lambda}{4(1+\lambda)}$. Finally, in case 6, attempting to turn v to color 1 results in v moving to D_t, so $C_6 = \frac{\lambda}{1+\lambda}$.

Now suppose that v was in D_t to start. In taking a step, we do not consider at all the color of D_t. Hence we can treat such an occurrence as always moving v out of D_t, and then treating it as though it was in A_t. Hence for $v \in D_t$ and case i,

$$E[|D_{t+1}| - |D_t| | v = V_t, \mathcal{F}_t] = C_i - 1.$$

Let R_i denote the number of nodes that fall into case i. Altogether, we have that

$$\begin{aligned} E[|D_{t+1}| - |D_t| \mid \mathcal{F}_t] &= \sum_{i=1}^{6} \frac{1}{n} \left[\sum_{v \in A_t, c(v)=i} C_i + \sum_{v \in D_t, c(v)=i} (C_i - 1) \right] \\ &= -\frac{|D_t|}{n} + \sum_{i=1}^{6} \frac{1}{n} \sum_{v:c(v)=i} C_i \\ &= -\frac{|D_t|}{n} + R_4 \cdot C_4 + R_5 \cdot C_5 + R_6 \cdot C_6] \\ &\leq -\frac{|D_t|}{n} + \frac{1}{n}[R_4 + R_5 + 2R_6] \max\{C_4, C_5, \frac{1}{2}C_6\} \end{aligned}$$

Now R_4, R_5 and R_6 cannot be arbitrarily large. Every vertex counted in R_4 and R_5 is adjacent to at least one vertex in D_t. Every vertex counted by R_6 is adjacent to at least two vertices in D_t. Taken as a whole, the vertices in D_t are adjacent to at most $|D_t|\Delta$ different vertices. Hence

$$R_4 + R_5 + 2R_6 \leq \Delta|D_t|.$$

We have shown that $C_4 = C_5 = \frac{1}{2}C_6$, and so $C = \max\{C_4, C_5, \frac{1}{2}C_6\} = \frac{1}{2}\frac{\lambda}{1+\lambda}$, so that

$$\begin{aligned} E[|D_{t+1}| \mid \mathcal{F}_t] &\leq |D_t| + -\frac{|D_t|}{n} + \frac{|D_t|\Delta}{n}C \\ &= |D_t|\left(1 - \frac{\Delta\lambda/(2(1+\lambda)) - 1}{n}\right) \\ &= |D_t|\beta, \end{aligned}$$

where β is set to be the factor in parenthesis in the last expression. There are two ways to proceed in the analysis at this point. Since each gives an upper bound on the expected running time, each will give us one term in the minimum expression of the theorem. First, since $\beta \leq 1$, $|D_t|$ is a supermartingale, allowing us to use Theorem 4.6 to continue. The probability that $|D_{t+1}| \neq |D_t|$ is bounded below by the probability that a node in D_t is chosen and turned to 0. This occurs with probability $\frac{|D_t|}{n} \cdot \frac{1}{1+\lambda}$. Therefore, by Theorem 4.6,

$$\begin{aligned} E[T_{BC}] &\leq \sum_{i=1}^{n} 2(i+1)\frac{(1+\lambda)(n)}{i} \\ &\leq (2n + \ln n)n(1+\lambda) \end{aligned}$$

which is smaller than the second term in the minimum expression of the theorem.

When $\beta < 1$, we can take another approach. When we take a single step, the expectation of $|D_t|$ decreases by a constant factor. We now show by induction on t

that

$$E[|D_t|] \leq E[|D_0|]\beta^t. \tag{4.1}$$

The base case when $t = 0$ is simply an identity. As our induction hypothesis, suppose that 4.1 holds for time t. The fact that $0 \leq |D_t| \leq n$ tells us that $E[|D_t|]$ is bounded for all t, so at time $t+1$,

$$E[|D_{t+1}|] = E[E[|D_{t+1}|||D_t|]] \leq E[\beta|D_t|] = \beta\beta^t E[|D_0|] = \beta^t E[|D_0|].$$

Note that $|D_t|$ is integral, so $T_{BC} > t \Rightarrow |D_t| \geq 1$. Hence

$$\begin{aligned} P(T_{BC} > t) &= P(|D_t| \geq 1) \\ &\leq \frac{1}{E[|D_t|]} \\ &\leq \beta^t E[|D_0|] \\ &\leq n\beta^t, \end{aligned}$$

which shows that T_{BC} has an exponentially declining tail, allowing us to upper bound $E[T_{BC}]$. This upper bound is exactly the first expression in the minimum term.

The final portion of the theorem, that $P[T_{BC} > ekE[T_{BC}]] \leq \exp(-k)$ follows from the facts that $P(T_{BC} > eE[T_{BC}]) < 1/e$, and $P(T_{BC} > t+s) < P(T_{BC} > t)P(T_{BC} > s)$. □

The same basic proof outline will be used again and again as we examine different bounding chains. First, come up with some integer measure of how far away the bounding chain is from detecting complete coupling (in this case, we used $|D_t|$). Second, show that this measure is a supermartingale, or even better, than it shrinks

by a constant factor at each step. Finally, use the facts we have shown about martingales, or the fact that the measure is integral, to upper bound the time until our measure hits zero.

Of course, not every bounding chain analysis will be quite as straightforward, and the next chapter basically consists of simple tricks to allow bounding chains to be used on each of the models of Chapter 2.

Bounding chain step for Dyer-Greenhill chain

Set $Y = Y_t$
Choose a vertex v uniformly at random from V
Choose U uniformly from $[0, 1]$
If $U > \frac{\lambda}{1+\lambda}$
  Set $Y(v) = \{0\}$
Else
  Case 1: All neighbors of v are colored $\{0\}$
    Set $Y(v) = \{1\}$
  Case 2: v has exactly 1 neighbor w colored $\{1\}$, rest are $\{0\}$
    If $U \leq p_{swap}\frac{\lambda}{1+\lambda}$
      Set $Y(v) = \{1\}$, $Y(w) = \{0\}$
    Else
      Set $Y(v) = \{0\}$
  Case 3: v has more than one neighbor colored $\{1\}$
    Set $Y(v) = \{0\}$
  Case 4: One neighbor w colored $\{0, 1\}$, rest colored 0
    If $U \leq p_{swap}\frac{\lambda}{1+\lambda}$
      Set $Y(v) = \{1\}$, $Y(w) = \{0\}$
    Else
      Set $Y(v) = \{0, 1\}$
  Case 5: One neighbor w colored $\{1\}$, at least one colored $\{0, 1\}$
    if $U \leq p_{swap}\frac{\lambda}{1+\lambda}$
      Set $Y(v) = \{0, 1\}$, $Y(w) = \{0, 1\}$
  Case 6: More than one neighbor is unknown, rest are 0
    if $U \leq \frac{\lambda}{1+\lambda}$
      Set $Y(v) = \{0, 1\}$

Figure 4.2: Bounding chain step for Dyer-Greenhill chain

Chapter 5

Bounding chains for other models

> A priest asked: What is Fate, Master? And he was answered:
>
> It is that which gives a beast of burden its reason for existence.
>
> It is that which men in former times had to bear upon their backs.
>
> It is that which has caused nations to build byways from City to City upon which carts and coaches pass, and alongside which inns have come to be built to stave off Hunger, Thirst and Weariness.
>
> And that is Fate? said the priest.
>
> Fate ... I thought you said Freight, responded the Master.
>
> That's all right, said the priest. I wanted to know what Freight was too.
>
> *-Kehlog Albran, "The Profit"*

Finding the fate of a state of the Markov chain is easy once you have a bounding chain, because no matter what the starting state, the final fate of the process will be the same.

On the other hand, it turns out that the hard core gas model is one of the few examples where a Markov chain can be directly turned into a bounding chain. Often, in order for the bounding chain to detect complete coupling in a reasonable amount of time, some tweaking of the original chain into a more suitable form is necessary.

5.1 Q coloring chain

A case in point is the chain for the Q colorings of a graph (Figure 3.10). Jerrum [25] showed that the single site Metropolis Hastings chain for this problem is rapidly mixing when the number of colors $Q \geq 2\Delta$, where Δ is the maximum degree of the graph. Salas and Sokal [46] extended this result to the heat bath chain. In this section we analyze a bounding chain for this chain introduced in [21] and [18]. We will show that this bounding chain detects complete coupling in polynomial time when $Q \geq \Delta(\Delta + 1)$, although computer experiments in [18] indicate that it is polynomial even when $Q < 2\Delta$ on certain classes of graphs.

Rather than working directly with the heat bath chain, it will be easier to work with the acceptance rejection heat bath chain when describing the bounding chain. Although it is not necessary to take this approach when writing and analyzing the bounding chain, it usually does lead to some simplification.

After $\Delta + 1$ different colors are chosen we know that at least one of them had to be nonblocking. Recall that in the bounding chain at site we keep a subset of colors, $Y(v)$. In this case $Y(v)$ is at worst each of the $\Delta + 1$ colors that we tried

Single site acceptance rejection heat bath Q coloring chain

Set $x \leftarrow X_t$
Choose $v \in_U V$
Repeat
  Choose $c \in_U C$
Until c is nonblocking for v
Set $x(v) \leftarrow c$
Set $X_{t+1} \leftarrow x$

Figure 5.1: Single site acceptance rejection heat bath Q coloring chain

for node v. Therefore the size of each $Y(v)$ in the bounding chain is at most $\Delta + 1$. Hence there are at most $\Delta(\Delta + 1)$ colors in the color sets of neighbors of v, so if $Q > \Delta(\Delta + 1)$ there is some chance of choosing a color which is known not to be blocked for v.

The idea of the bounding chain is to keep selecting colors for v, adding them to the color set, until we have chosen at least $\Delta + 1$ different colors, or we have found one not in the set. Let $w \sim v$ denote that $\{v, w\}$ is an edge of the graph. For each

Bounding chain for single site Q coloring heat bath chain

Set $y \leftarrow Y_t$
Choose $v \in_U V$
Set $y(v) \leftarrow \emptyset$
Repeat
  Choose $c \in_U C$
  Set $y(v) \leftarrow y(v) \cup \{c\}$
Until $c \in \cup_{v \sim w} Y(w)$

Figure 5.2: Bounding chain for single site Q coloring heat bath chain

of v, we add to $y(v)$ those colors which could possibly be chosen for $x(v)$, and so

$X_t \in Y_t \rightarrow X_{t+1} \in Y_{t+1}$.

As long as $Q > \Delta(\Delta + 1)$, we always have a chance of $|Y(v)| = 1$ in the next round. It turns out that this condition is sufficient for the bounding chain to detect complete coupling in polynomial time.

Theorem 5.1 *Let T_{BC} be the first time that the bounding chain detects complete coupling. If $Q > \Delta(\Delta + 1)$,*

$$E[T_{BC}] \leq \frac{Qn \ln n}{Q - \Delta(\Delta + 1)}.$$

If $Q = \Delta(\Delta + 1)$,

$$E[T_{BC}] \leq 3n^2 Q.$$

Proof: As with the Dyer-Greenhill bounding chain, we will prove the result by keeping track of the size of D_t, the set of nodes such that $|Y_t(v)| > 1$. Let $d(v)$ denote the number of neighbors of v which lie in D_t. The set of nodes where $|Y_t(v)| = 1$ we denote A_t. Let $v : A_t \rightarrow D_{t+1}$ denote the event where v moves from A_t at time t to D_{t+1} in the next time step. Such a move makes D_{t+1} 1 larger than D_t.

On the other hand, when $v : D_t \rightarrow A_{t+1}$, this decreases D_{t+1} by 1 compared to D_t. Each node is selected to be altered with probability $1/n$.

$$\begin{aligned} E[|D_{t+1}| - |D_t| \mid |D_t|] &= \sum_{v \in A_t} \frac{1}{n} P(v : A_t \rightarrow D_{t+1} \mid |D_t|) \\ &\quad + \sum_{v \in D_t} \frac{1}{n}(-1) P(v : D_t \rightarrow A_{t+1} \mid |D_t|) \\ &= \sum_{v \in A_t} \frac{1}{n} P(v : A_t \rightarrow D_{t+1} \mid |D_t|) \end{aligned}$$

$$
\begin{aligned}
& + \sum_{v \in D_t} \frac{1}{n}[-1 + P(v : D_t \to D_{t+1} \mid |D_t|) \\
= \quad & \frac{-|D_t|}{n} + \frac{1}{n} \sum_v P(v \in D_{t+1} \mid |D_t|)
\end{aligned}
$$

The probability that v is in D_{t+1} depends on the number of unknown neighbors of v, which we shall denote $d(v)$. Suppose that each of these $d(v)$ neighbors has $\Delta + 1$ colors in its color set. Then we are uncertain about the status of at most $d(v)(\Delta+1)$ colors. That is, for these colors we are not sure whether or not they block v, and so choosing them for v leads to v entering D_{t+1}. This occurs with probability

$$P(v \in D_{t+1}) = \frac{d(v)(\Delta + 1)}{Q}.$$

Combining this with the fact that each node in D_t has at most Δ neighbors, and so $\sum_v d(v) \leq |D_t|\Delta$ yields

$$
\begin{aligned}
E[|D_{t+1}| - |D_t| \mid |D_t|] &= \frac{1}{n}\left[-|D_t| + \sum_v \frac{d(v)(\Delta + 1)}{Q}\right] \\
&= \frac{1}{n}\left[-|D_t| + \frac{|D_t|\Delta(\Delta + 1)}{Q}\right]
\end{aligned}
$$

which is at most 0 when $Q \geq \Delta(\Delta + 1)$, making $|D_t|$ a supermartingale. When $Q \geq \Delta(\Delta + 1)$, an easy induction gives us

$$E[|D_t|] \leq n\left(1 - \frac{Q - \Delta(\Delta + 1)}{Qn}\right)^t.$$

After $k\frac{Qn \ln n}{Q - \Delta(\Delta+1)}$ steps, this makes $E[|D_t|] \leq e^{-k}$. Given that $|D_t|$ is a nonnegative integer, Markov's inequality gives $P(|D_t| > 0) \leq e^{-k}$, which means that the expected time needed for $|D_t|$ to hit 0 is as in the first part of the theorem.

For the second half, note that $|D_t| \neq |D_{t+1}|$ with probability at least $|D_t|/(Qn)$, which is a lower bound on the chance of picking an unknown node and changing it to known. Therefore we may apply Theorem 4.6 which gives us the second half of the theorem. □

As with the bounding chain case, it is unlikely that complete coupling takes much longer than the expected time to completely couple. More formally,

Theorem 5.2 *Let T_{BC} be the time that $|Y_t(v)| = 1$ for all v, given that $Y_0(v) = C$ for all v. Let $E[T_{BC}]$ be its expected value. Then $P(T_{BC} > 2kE[T_{BC}]) \leq 2^{-k}$.*

Proof: By Markov's inequality the probability that $T_{BC} > 2E[T_{BC}]$ is at most $1/2$. The starting condition we are given, $Y_0(v) = C$, is in some sense the worse possible. If $Y_0'(v) \subset Y_0(v)$ for all v, then $|Y_t(v) = 1| \rightarrow |Y_0'(v)| = 1$.

Therefore after $2E[T_{BC}]$ steps, either we have complete coupling or we try again. Since the next steps are independent of the last, the next $2E[T_{BC}]$ steps also have a $1/2$ chance of showing complete coupling. After $2kE[T_{BC}]$ steps, the only way we could not have complete coupling is if all k sets of $2E[T_{BC}]$ steps failed, which happens with probability at most $1/2^k$.

This chain exhibits the cutoff phenomenon which was seen in the bounding chain for the Dyer Greenhill hard core chain. When the number of steps is less than $n \ln n$ there is a good chance that we have not even selected all of the nodes at least once. Therefore the bounding chain could not have converged in this time. However, when the number of colors is sufficiently high, the bounding chain always detects complete coupling after $O(n \ln n)$ steps, with an exponentially declining probability

of failure. This behavior will be seen in the Potts model bounding chain as well.

5.2 The Potts model

As with the Q coloring chain, we begin by writing the heat bath chain for the Potts model as an acceptance rejection chain. For convenience, in this section we consider the ferromagnetic model. Similar results may be shown for the antiferromagnetic model.

We continue using $b_i(v)$ to denote the number of neighbors of v which have color i. After choosing which v to change, let $w(i) = \alpha^{b_i(v)}$ be the weight associated with color i, where α is again $\exp(1/T)$. A natural upper bound on the weights is α^{Δ}. With this notation, the general acceptance rejection chain becomes:

Single site acceptance rejection heat bath Potts chain

Set $x \leftarrow X_t$
Choose $v \in_U V$
Repeat
  Choose $c \in_U C$
  Choose $W \in_U [0, 1]$
Until $W \leq \frac{\alpha^{b_c(v)}}{\alpha^{\Delta}}$
Set $x(v) \leftarrow c$
Set $X_{t+1} \leftarrow x$

Figure 5.3: Single site acceptance rejection heat bath Potts chain

Here blocking colors do not prevent v from becoming a particular color, they encourage it.

Now consider how to develop a bounding chain for this process. For each node

v and color i, let $d_i(v)$ be the number of neighbors w of v for which $|Y(w)| \geq 2$ and $i \in Y(w)$. Let $b_i(v)$ denote the number of neighbors w of v for which $|Y(w)| = 1$ and $Y(w) = \{i\}$. Then the chance of choosing color i for v may be as high as $\alpha^{b_i(v)+d_i(v)}/\alpha^{\Delta}$ or as low as $\alpha^{b_i(v)}/\alpha^{\Delta}$ for any $X \in Y$. Therefore if we fall below $\alpha^{b_i(v)}/\alpha^{\Delta}$ we always know to terminate the loop, but if we are in between this high and low value, we must add this color to the set of possible colors and repeat the loop again.

Single site acceptance rejection heat bath Potts bounding chain

Set $y \leftarrow Y_t$
Choose $v \in_U V$
Set $y(v) \leftarrow \emptyset$
Repeat
  Choose $c \in_U C$
  Choose $W \in_U [0,1]$
  If $W \leq \frac{\alpha^{b_c(v)+d_c(v)}}{\alpha^{\Delta}}$
    Set $y(v) \leftarrow y(v) \cup \{c\}$
Until $W \leq \frac{\alpha^{b_c(v)}}{\alpha^{\Delta}}$
Set $x(v) = c$
Set $X_{t+1} = x$

Figure 5.4: Single site acceptance rejection heat bath Potts bounding chain

Again this bounding chain exhibits the same kind of cutoff phenomenon as seen earlier, where we do not have a good bound until all the nodes have been hit with nonnegligible probability, and then the probability that we have not detected complete coupling declines exponentially.

Theorem 5.3 *Using this bounding chain for the ferromagnetic Potts model,*

$$P\left(T_{BC} > \frac{kn \ln n}{\Delta(1-\alpha^{-1})}\right) \leq e^{-k}.$$

Proof: Repeating the proof for the Q coloring chain, we are led to

$$E[|D_{t+1}| - |D_t| \mid |D_t| = \frac{1}{n}[-|D_t| \sum_{v \in V} P(v \in D_{t+1}).$$

A necessary condition for node v to end up in D_{t+1} is that the first color for which $W \leq \frac{\alpha^{b_c(v)+d_c(v)}}{\alpha^{\Delta}}$ also satisfies $W > \frac{\alpha^{b_c(v)}}{\alpha^{\Delta}}$. (This condition is not sufficient since we may eventually end up choosing this first unknown color as the only color picked, an event which will come into play later when we consider the Ising model.)

Using the worst possible upper bound that $d_i(v) = d(v)$ for all i (meaning that $Y(w) = C$ for all neighbors of v in D_t) we have that the probability that v ends up unknown is

$$\begin{aligned} P(v \in D_{t+1}) &\leq \frac{\alpha^{b_c(v)+d_c(v)}}{\alpha^{b_c(v)+d_c(v)}} - \frac{\alpha^{b_c(v)}}{\alpha^{b_c(v)+d_c(v)}} \\ &= 1 - \frac{1}{\alpha^{d_c(v)}} \\ &\leq 1 - \frac{1}{\alpha^{d(v)}} \end{aligned}$$

Therefore

$$E[|D_{t+1} - |D_t| \mid |D_t| \leq \frac{1}{n}[-|D_t| + \sum_{v:d(v)>0} 1 - \frac{1}{\alpha^{d(v)}}.$$

The interesting thing about the terms in the last summand is that they do not contain any information about the node v other than $d(v)$. Therefore we just consider

maximizing this sum subject to the constraint that the $d(v)$ are positive integers that sum to at most $|D_t|\Delta$.

Given this freedom, the maximum will occur when all of the $d(v) = 1$. Suppose $d(v) \geq 2$. This contributes

$$1 - \frac{1}{\alpha^{d(v)}}$$

to the sum. However, if we create a dummy node v' with $d(v') = 1$ and lower $d(v)$ by 1, then the sum of $d(w)$ over all w remains the same, but now the contribution to the sum is

$$1 - \frac{1}{\alpha^{d(v)-1}} + 1 - \frac{1}{\alpha} = 1 - \frac{\alpha - \alpha^{d(v)} + \alpha^{d(v)-1}}{\alpha^{d(v)}}.$$

To see that the numerator of this fraction is at most 1 (which would mean this contributes more to the sum than when $d(v) \geq 2$) we note that at $\alpha = 1$ the numerator equals 1. Taking its derivative with respect to α gives

$$\begin{aligned} 1 - d(v)\alpha^{d(v)-1} + (d(v)-1)\alpha^{d(v)-2} &\leq 1 - \alpha^{d(v)-1} \\ &< 0 \end{aligned}$$

Making the derivative negative for all $\alpha > 1$. Therefore by the mean value theorem the numerator is less than 1 for all positive α, making this contribution more significant.

Hence

$$\begin{aligned} E[|D_{t+1}| - |D_t| \mid |D_t|] &\leq \frac{1}{n}\left[-|D_t| + |D_t|\Delta(1 - \frac{1}{\alpha})\right] \\ &= \frac{-|D_t|}{n}\left[(\Delta - 1) - \frac{1}{\alpha}\right]. \end{aligned}$$

This will be negative when $\alpha \leq 1 + 1/(\Delta - 1)$, in which case an induction may be used to show that

$$E[|D_t| \leq n\left(1 - \frac{\Delta(1-\alpha^{-1})}{n}\right)^t,$$

and the proof is finished in the same manner as before. □

5.3 Swendsen-Wang

Swendsen-Wang is unusual in that it switches back and forth between 2 colorings on the edges of the graph and Q colorings on the nodes of the graph. Our bounding chain must be equally quixotic, coloring the edges either $\{0\}$, $\{1\}$, or $? = \{0, 1\}$ and coloring the nodes in a similar fashion.

The Swendsen-Wang bounding chain is given in Figure 3.11, where $\mathcal{C}_A$ is the set of connected components with respect to the edges in A, $w \sim v$ if w and v are connected via edges in $A \cup B$, and v_C is the component in $\mathcal{C}_A$ that contains vertex v. Swendsen-Wang has two phases and so our bounding chain does as well. It is the first phase that makes the bounding chain approach possible. Here edges are thrown out of the chain independently with probability $1-p$ (recall that $p = 1-\exp(-1/T)$ is another means of measuring the temperature of the Potts model). This means that if an edge is colored $\{1\}$ or $?$ and is thrown out, then its color will always change to $\{0\}$. When computing components, however, edges that are still colored $?$ could mean that we do not know which nodes belong to which components. Hence the colors of these nodes at the next stage of the algorithm will be uncertain.

Analyzing the change in unknown edges is easy during the removal stage. It

Swendsen-Wang bounding chain

Set $y \leftarrow Y_t$
Let $A \leftarrow \{\{v,w\} \in E : y(v) = y(w), |y(v)| = |y(w)| = 1\}$
Let $B \leftarrow \{\{v,w\} \in E : |y(v) \cap y(w)| \geq 1, |y(v)| + |y(w)| > 2\}$
For each edge e set $U(e) \in_U [0,1]$
For each node v set $k(v) \in_U \{1,\ldots,k\}$
For each edge $e \in A$
  If $U(e) < 1-p$
    Set $A \leftarrow A \setminus \{e\}$
    Set $B \leftarrow B \setminus \{e\}$
For all $C \in \mathcal{C}_A$
  Set $z(w) \leftarrow k(C_v)$ for all $w \in C$
Choose a total order uniformly at random for $C \in \mathcal{C}_A$
For all v
  Set $y(v) \leftarrow \cup_{v_C < w_C, w \sim v} z(w)$
Set $Y_{t+1} \leftarrow y$

Figure 5.5: Swendsen-Wang bounding chain

is the growth of unknown components that makes proving the following theorem difficult.

Theorem 5.4 *Set*

$$\beta = 1 - (1-p)^{\Delta} + \frac{Q-1}{2Q} \cdot \frac{p\Delta}{1-p(\Delta-1)}$$

If $\beta < 1$, the Swendsen-Wang bounding chain will have detected complete coupling by time $-\log_\beta 2n$ with probability at least 1 /2.

Proof: As before, we shall show that $|D_t|$ is a supermartingale. We use A and B as defined in the bounding chain step at time t.

The key point is that for a vertex to receive more than one color, it must be connected to another vertex using edges in B, and at least one edge in $B \setminus A$. But

for an edge to be in $B \setminus A$, it had to have been adjacent to a node in D_t. Hence nodes in D_{t+1} must be connected through edges in B to a node in D_t. This is necessary but not sufficient; a node v may be connected to a node in D_t and still end up not being placed in D_{t+1} depending on the color choices. Three possibilities that would preclude v adding w to D_{t+1} are: w is already in D_{t+1} because of another node, the color chosen for v and w is the same, and $w_C > v_C$, so w does not add the color of v to its edge set. Although we cannot analyze the first exclusion, it is easy to quantify the last two.

We shall write $w \sim v$ if nodes v and w are connected using edges in B. Every node placed in D_{t+1} is either in D_t to start with or connected to some node in D_t using edges in B. Therefore we have that

$$|D_{t+1}| \leq \sum_{v \in D_t} \left[1_{v \in D_{t+1}} + \sum_{w \sim v} 1_{k(w_C) \neq k(v_c)} 1_{w_C < v_C} \right].$$

The ordering of components is uniform over all possible orderings, which gives us $P(w_C < v_C) \leq 1/2$. The probability that the components which v and w lie in receive different colors is $(Q-1)/Q$. By linearity of expectations

$$E[|D'| \mid \mathcal{F}_{\sqcup}] \leq \sum_{v \in D_t} \left[P(v \in D_{t+1} \mid \mathcal{F}) + \sum_{w \sim v} \frac{Q-1}{2Q} \right].$$

A necessary condition for v to be in D_{t+1} is that at least one edge adjacent to v must have survived the edge removal phase. This occurs with probability $1-(1-p)^{\Delta}$. We bound the number of $w \sim v$ by using a branching process argument (see [9]).

After edge removal, the number of nodes adjacent to w is bounded above stochastically by a binomial random variable with parameters p and Δ. Each of these is a separate branching process, with number of children distributed as a binomial

random variable with parameters p and $\Delta - 1$. (The possible number of children is $\Delta - 1$ rather than Δ because one edge must be used as the parent). Let h denote the expected size of each of these child processes.

Each one of these children processes has $(\Delta - 1)p$ expected number of children, and so the expected size of each child branching process satisfies the recursion:

$$E[h] \leq 1 + (\Delta - 1)pE[h].$$

Solving, we find that $E[h] \leq 1/[1 - p(\Delta - 1)]$.

The original node v has at most (in expected value after Phase I) $p\Delta$ neighbors connected by unknown edges. Each of these neighbors is also the source of a branching process, and so altogether the expected number of $w \sim v$ is $p/[1 - p(\Delta - 1)$. Summing up, we find that

$$E[|D_{t+1}|] \leq \sum_{v \in D_t} \beta = |D_{t+1}|\beta$$

where

$$\beta = \left[1 - (1-p)^{\Delta} + \frac{Q-1}{2Q} \cdot \frac{p\Delta}{1 - p(\Delta - 1)}\right].$$

What we have shown is that $E[|D_{t+1}| \mid \mathcal{F}] \leq \beta |D_t|$. Again this yields via induction $E[|D_t||D_0] \leq \beta^t |D_0|$. D_0 is just the entire set of vertices V, and so $E[|D_t|] \leq \beta^t n$. Hence after $-\log_\beta 2n$ time steps, $E[|D_t|] \leq 1/2$. Since $|D_t|$ is integral, we have that $P(|D_t| = 0) > 1/2$ by Markov's inequality, and we are done.□

Like all of our theorems concerning the *a priori* running time of bounding chains, this one has an immediately corollary that the chain is rapidly mixing when the

condition on p is satisfied. This is similar to a result proved independently by Cooper and Frieze [8] using the technique of path coupling. The following table shows the largest value of p over which these results apply.

Table 5.1: Swendsen-Wang approach comparison

Δ	2	3	4	5	6	7
Cooper/Frieze	0.416	0.209	0.136	0.100	0.079	0.065
Bounding chain: $Q = 2$	0.410	0.260	0.188	0.148	0.121	0.103
Bounding chain: Any Q	0.318	0.202	0.147	0.116	0.095	0.081

Of course, this is at best a theoretical determination. The bounding chain approach allows for computer experimentation to determine what the actual running time is for values of p which are much larger.

5.4 Sink free orientations

In the sink free orientation chain (Figure 3.12), we choose an edge at random and choose a new orientation at random from the set of the orientations that do not create a sink.

First, note that without loss of generality we may assume that every node has degree at least 2, since a leaf of the graph must have its edge directed out of the leaf in order to avoid creating a sink.

Suppose we were to begin the bounding chain by labeling every edge $? = \{-1, 1\}$ indicating that we do not know the orientation on any edge. Then we would never

be able to gain any information about the state of the chain, since we would never know whether flipping a particular edge was a permissible move.

Therefore we let $Y = Y^1 \cup Y^2$, meaning that $Y(e) = Y^1(e) \cup Y^2(e)$ for all edges e. At the start of the bounding chain procedure, we single out a particular edge e of the chain. We set $Y^1(e) = \{1\}$, $Y^2(e) = \{-1\}$ and $Y^1(e) = Y^2(e) =?$ for all $e' \neq e$. Clearly this means that $X_0 \in Y_0 = Y_0^1 \cup Y_0^2$ for all $X_0 \in \Omega$, since for all edges e', $Y_0(e') =?$.

If we guarantee that $X_t \in Y_t^i \Rightarrow X_{t+1} \in Y_{t+1}^i$ for $i = 1, 2$ for all t, then we will have guaranteed that $X_i \in Y_t \Rightarrow X_{t+1} \in Y_{t+1}$ for all t, and we will have a bounding chain.

We know the direction of some positive number of edges in Y_t^i; therefore it is possible to learn the identity of others. As always, we say that an edge e is known if $|Y(e)| = 1$, and otherwise it is unknown. An edge is directed into i if the edge (i, j) is colored -1 or if the edge (j, i) is colored 1, and otherwise it is directed out of i. Using this terminology, the bounding chain may be written as follows.

Now, Bubley and Dyer [6] showed that the sink free orientation chain couples in expected $O(m^3)$ time. So if X^1 and X^2 are two arbitrary processes different at time 0, then $P[X_t^1 = X_t^2] \geq 1/2$ for $t \geq 2m^3$. The actual algorithm has two phases. In Phase I we completely couple using the two processes Y^1 and Y^2, hoping that each of them detects complete coupling in their half of the states. In Phase II we run the chain as Bubley and Dyer did, hoping that the two states which remain merge into a single state.

Theorem 5.5 *For this two phase approach, $E[T_{BC}]$ is $O(m^5)$.*

Single edge heat bath sink free orientation bounding chain

Set $y^k \leftarrow Y_t^k$
Choose $[e = \{i, j\}] \in_U E$, where $i < j$
Case 1: All edges besides e are known to be directed into i
  Set $y(e) \leftarrow \{1\}$
Case 2: All edges besides e are known to be directed into j
  Set $y(e) \leftarrow \{-1\}$
In the remaining cases, choose $U \in_U [0, 1]$
Case 3: $U \leq 1/2$ and no edges (besides e) known to leave i
  Set $y(e) \leftarrow \{1, -1\}$
Case 4: $U > 1/2$ and no edges (besides e) known to leave j
  Set $y(e) \leftarrow \{1, -1\}$
Case 5: $U \leq 1/2$ and an edge known to leave i
  Set $y(e) \leftarrow \{-1\}$
Case 6: $U > 1/2$ and an edge known to leave j
  Set $y(e) \leftarrow \{1\}$

Figure 5.6: Single edge heat bath sink free orientation bounding chain

Proof: The probability that a single Phase I/Phase II pass detects complete coupling is the probability that three events occur. Let t_1 be the time at the end of Phase I, and t_2 be the time at the end of Phase II. Two of the three events that must occur are $|Y_{t_1}^k(v)| = 1$ for all v and $k = 1, 2$. Let X^1 and X^2 be the states defined by Y^1 and Y^2 should this occur. Then the third event that must happen is that $X_{t_2}^1 = X_{t_2}^2$, that is, the two remaining states have coupled by the end of Phase II.

The probability that the two states couple in Phase II was already shown [6] to be at least $1/2$ when the time the chain is run is $O(m^3)$. Therefore here we bound the time needed for Phase I to run to completion.

As always, we will show that $|D_t|$ is a supermartingale.

$$E[|D_{t+1}| - |D_t| \mid |D_t|) \leq \frac{1}{m}\left[\sum_{e \in A_t} P(e : A_t \to D_{t+1} \mid |D_t|) - \sum_{e \in D_t} P(e : D_t \to A_{t+1} \mid |D_t|)\right]$$

Consider how an edge might move from A_t to D_{t+1}. Suppose that $e = \{i, j\}$, with $i < j$. Then we know the value of the edge already, so if the roll of U is such that e does not change direction, we will still know it. Therefore the probability of an edge becoming unknown is at most $1/2$.

An edge cannot become unknown unless it is adjacent to an unknown edge. Moreover, it cannot become unknown if it is pointed towards that unknown edge. For example, suppose e is currently colored -1, and so is oriented (j, i). Then if another edge adjacent to i is unknown then that unknown edge cannot make (j, i) unknown, since either we continue with e colored -1, or we reverse the direction to (i, j), which does not create a conflict at i no matter what the direction of the unknown edge is.

If however, we have an unknown edge adjacent to j, then switching edge e to (i, j) might create a sink, or it might not, so edge e would become unknown.

Suppose that we have an unknown edge adjacent to i. How many other edges adjacent to i can it possibly help to become unknown? We have seen that the only edges it can make unknown are those which are directed (i, j) for some j. Suppose that there are at least 2 such edges. We only selected one edge at a time to change, so changing 1 such edge to (j', i) would leave at least one edge still directed (i, j), so the value of the unknown edge is utterly irrelevant.

Therefore the unknown edge can at most make one edge adjacent to i unknown. Suppose that the unknown edge is $\{i,k\}$ and the edge which it might make unknown is (i,j). Then if $\{i,k\}$ is chosen and the direction chosen is (k,i) we know that this move would not create a sink because we have an edge leaving i, namely, (i,j). Each edge has an equal chance of being chosen, therefore

$$P((i,j) : A_t \to D_{t+1} \mid |D_t|) = \frac{1}{2m} = P(\{i,k\} : D_t \to A_{t+1} \text{ as } (k,i) \mid |D_t|).$$

Similarly, if this unknown edge could create an unknown edge from edge (k,ℓ), then

$$P((k,\ell) : A_t \to D_{t+1}) = \frac{1}{2m} = P(\{i,k\} : D_t \to A_{t+1} \text{ as } (i,k)).$$

Summing over all known and unknown edges gives us

$$\sum_{e \in A_t} P(e : A_t \to D_{t+1} \mid |D_t|) \leq \sum_{e \in D_t} P(e : D_t \to A_{t+1} \mid |D_t|),$$

thereby showing that $|D_t|$ is a supermartingale.

The difference between this process and the others that we have considered so far is that not only is $|D_t| = 0$ an absorbing state, so is $|D_t| = n$. Fortunately, due to our trick of using Y^1 and Y^2, we start with $|D_t| = n - 1$ for both bounding processes at the beginning of Phase I.

At least one edge in D_t must be adjacent to an edge which is pointing away from their common node, otherwise D_t would be known. If we select this unknown edge and direct it towards this common point, then it will become known. The point is that the probability that $|D_{t+1}|$ does not equal $|D_t|$ is at least $1/(2m)$.

Theorem 4.5 deals with the case when 0 is the only absorbing state. If n is also an absorbing state of the process, then the expected time until the state is

absorbed at either 0 or n is bounded above by the expected time until the state hits 0, which is $O(m^2/[1/(2m)]) = O(m^3)$. Therefore after $3m^3$ steps, the probability that the process reaches absorption is 1/2. The probability that $|D_t|$ reached 0 is $1/m$, and so the probability that the number of unknowns went to 0 for both Y^1 and Y^2 is $1/m^2$. Therefore after m^2 expected runs of length $3m^3$, Phase I will have condensed the bounding chain to the point where it only contains two processes X^1 and X^2. Phase II then couples these in time m^3, making the total running time $O(m^5 + m^3) = O(m^5)$.

5.5 Hypercube slices

To bound the behavior of the hypercube slices chain, we again use two phases. Unlike the sink free orientation bounding chain, however, this approach will allow us to show the mixing time of the chain within a constant factor.

Consider this alternative version of the heat bath chain for hypercube slices shown in Figure 3.22. In the earlier chain, we have a 1/2 chance of just holding and not switching the random coordinate. This chance is taken care of by the test $U_1 < 1/2$. If we do decide to switch, the chance that we switch a node colored 1 to 0 is $|C_1|/n$, and the chance that we switch a node colored 0 to 1 is $|C_0|/n$. The test for U_2 determines which event actually occurs.

The reason for writing the Markov chain in this form is that now the value of $|C_1|$ is itself a Markov chain. It either goes up 1 with probability $(n - |C_1|)/n$, or down 1 with probability $|C_1|/n$.

Alternative heat bath hypercube slice chain

Set $x \leftarrow X_t$
Choose $U_1 \in_U [0,1]$
Choose $U_2 \in_U [0,1]$
Set $C_1 \leftarrow \{i : x(i) = 1\}$
Set $C_0 \leftarrow \{i : x(i) = 0\}$
If $U_1 < 1/2$
  If $U_2 \leq |C_1|/n$
    Choose $i \in_U C_1$
    Set $x(i) = 0$
  Else **Choose** $i \in_U C_0$
    Set $x(i) = 1$
Set $X_{t+1} \leftarrow x$

Figure 5.7: Alternative heat bath hypercube slice chain

In Phase I all we keep track of is the value of $|C_1|$. We know that at the beginning $L \leq |C_1| \leq U$. We run the chain in such a fashion so that if $|C_1(x)| \leq |C_1(y)|$, then $|C_1(f(x))| \leq |C_1(f(y))|$, that is, the stochastic process $|C_1(X_1)|$ is monotonic.

If we have monotonicity, start a hypothetical process with $|C_1(x)| = L$ and another with $|C_1(y)| = U$, and at some future time t, $|C_1(F_0^t(x))| = |C_1(F_0^t(y))|$, then we know that $|C_1(F_0^t(z))| = |C_1(F_0^t(x))|$ for all $z \in \Omega$. The statistic $|C_1|$ will be the same for all processes. In Phase I, let C_L be $|C_1(F_0^t(x))|$ where $|C_1(x)| = L$, and C_U be $|C_1(F_0^t(y))|$ where $|C_1(y)| = U$. Phase I ends when the $|C_1|$ values for the X and Y processes merge.

How do we guarantee monotonicity? If $U_1 < 1/2$ then we have all the processes with $|C_1|$ odd move, and all the processes with $|C_1|$ even hold. If $U_1 \geq 1/2$, we have all the processes with $|C_1|$ hold, and all the even ones move. Since $|C_1|$ changes by at most one at each step, if $|C_1(x)| < |C_1(y)|$ then either $|C_1(y)| - |C_1(x)|$ is

even, in which case their difference is at least two. Each term changes by at most one, so their difference changes by at most two and $|C_1(f(x))| \leq |C_1(f(y))|$. If their difference is odd then at each step at most one of the values changes, so their difference changes by at most 1 and again we have that $|C_1(f(x))| \leq |C_1(f(y))|$.

If $|C_1(x)| = |C_1(y)|$ then $|C_1(f(x))| = |C_1(f(y))|$, so altogether we have that $|C_1|$ is monotonic. Once Phase I is over we need only deal with states that contain

Heat bath hypercube slice bounding chain Phase I

Input: C_L, C_U

Set $y \leftarrow Y_t$
Choose $U_1 \in_U [0,1]$
Choose $U_2 \in_U [0,1]$
Choose $U_3 \in_U [0,1]$
If ($U \leq 1/2$ and C_L is odd) or ($U \geq 1/2$ and C_L is even)
  If $U_2 \leq |C_L|/n$
    Set $C_L \leftarrow \max\{C_L - 1, L\}$
  Else **Set** $C_L \leftarrow \min\{C_L + 1, U\}$
If ($U \leq 1/2$ and C_U is odd) or ($U \geq 1/2$ and C_U is even)
  If $U_2 \leq |C_U|/n$
    Set $C_U \leftarrow \max\{C_L - 1, L\}$
  Else **Set** $C_U \leftarrow \min\{C_L + 1, U\}$

Figure 5.8: Heat bath hypercube slice bounding chain Phase I

the same number of coordinates colored 1.

In the alternative heat bath chain, suppose that we decided to change a node colored 1 to color 0. Some of the nodes are known to be colored 1 in the bounding chain, that is, $Y(v) = \{1\}$, so we roll another die to see if we are choosing from these nodes, or not. If we are not, then we choose a node to switch via a two step

process. First, pick a node i from the unknown nodes. For a particular process X lying in the bounding chain, if $X(v) = 1$, we switch $X(v)$ to 0. If $X(v) = 0$, then we pick another node from the unknown nodes where $X(v) = 1$, and switch that. So this is a form of acceptance rejection sampling, where after the first rejection we give up and just choose from the points meeting our criteria.

So we pick a node, and if it doesn't meet our criteria of being colored 1, we reject it and pick a node that is colored 1 for sure. Here's the rub: the first node that we choose will always be colored 0 at the end of the step. If it was colored 1 at the beginning of the step then we switch it to 0. If it was colored 0 then we pick a different node altogether and switch the value of that other node. In other words, at the end of the step in the bounding chain, we know that $Y(v) = 0$.

As in all of the other bounding chains we have considered, let D_t denote the set of nodes such that $|Y(v)| > 1$. Let K_1 be those nodes that are known to be colored 1 ($Y(v) = \{1\}$), and K_0 be those nodes that are known to be colored 0.

This bounding chain will always detect complete coupling very quickly.

Theorem 5.6 *After* $2n \ln[2n/\epsilon]$ *time steps, the probability that this bounding chain will have detected complete coupling is at least* $1 - \epsilon$.

Proof: Phase I ends when $|C_1|$ is the same for $F_0^t(z)$ for all z in the state space. Phase II ends when $F_0^t(z)$ is a constant. We will show that the probability that Phase I has not ended by time $n \ln[2(U - L)/\epsilon]$ is at most $\epsilon/2$ and the probability that Phase II has not ended by time $n \ln[2n/\epsilon]$ is also at most $\epsilon/2$. The union bound for failure will then complete the proof.

Heat bath hypercube slice bounding chain Phase II

Input: Y_t, $|C_1|$

Set $y \leftarrow Y_t$
Choose $U_1 \in_U [0,1]$
Choose $U_2 \in_U [0,1]$
Choose $U_3 \in_U [0,1]$
If $U \leq 1/2$
  If $U \leq |C_1|/n$ and $|C_1| > L$
    If $U_3 < |K_1|/|C_1|$
      Choose $i \in_U K_1$
      Set $y(v) \leftarrow \{0\}$
    Else
      Choose $i \in_U D_t$
      Set $y(v) \leftarrow \{0\}$
  Else if $U > |C_1|/n$ and $|C_1| < U$
    If $U_3 < |K_0|/|C_1|$
      Choose $i \in_U K_0$
      Set $y(v) \leftarrow \{1\}$
    Else
      Choose $i \in_U D_t$
      Set $y(v) \leftarrow \{1\}$
Set $Y_{t+1} \leftarrow y$

Figure 5.9: Heat bath hypercube slice bounding chain

We begin with Phase I. Let C'_L and C'_U denote the values of C_L and C_U after one time step, and let a denote the change in the difference between the upper and lower bounds, so that

$$a = (C'_U - C'_L) - (C_U - C_L).$$

Since $|C'_U - C_U| \leq 1$ and $|C'_L - C_L| \leq 1$, we know that a is either $0, \pm 1$, or ± 2.

Consider two cases, in the first case, C'_L and C'_U have the same parity. Then with probability $1/2$ they don't move at all, and with probability $1/2$ they move

with according to the value of U_2. If $U_2 \leq |C_L|/n$, either $a = 0$ or $a = -1$ (where this latter event occurs when $|C_L| = L$. If $|C_L|/n < U_2 \leq |C_U|/n$, then $a = -2$, and if $|C_U|/n < U_2$, then $a = 0$ or $a = -1$ (with the second event occurring when $|C_U| = U$). Therefore,

$$\begin{aligned} E[a|\text{same parity}] &\leq \frac{1}{2}0 + \frac{1}{2}\left[-2\frac{C_U - C_L}{n}\right] \\ &= -\frac{C_U - C_L}{n} \end{aligned}$$

Now suppose that C_U and C_L have different parity. Then with probability 1/2 C_L moves. With probability $(n - C_L)/n$, a will be -1. With probability at most C_L/n a will be 1 (this is an upper bound on the probability since C_L could be L). The other possibility is that C_U moves, again with probability 1/2. In this case a is -1 with probability C_U/n and 1 with probability at most $(n - C_U)/n$.

Adding everything up, we get that

$$\begin{aligned} E[a|\text{different parity}] &\leq \frac{1}{2}\left[-\frac{n - C_L}{n} + \frac{C_L}{n}\right] + \frac{1}{2}\left[-\frac{C_U}{n} + \frac{n - C_U}{n}\right] \\ &= -\frac{C_U - C_L}{n}. \end{aligned}$$

And so it does not matter whether or not we started with C_L and C_U having the same parity or not, the bound on $E[a]$ is the same.

Another way of writing this bound is

$$E[C'_U - C'_L | C_U - C_L] \leq \left(1 - \frac{1}{n}\right)(C_U - C_L).$$

Following our usual program, we note that this implies that the expected value of the difference after nk time steps is at most $(U - L)e^{-k}$. By Markov's inequality

and the fact that $C_U - C_L$ is integer, this is also an upper bound on the probability that $C_U \neq C_L$. Therefore, after $n \ln[(U-L)/\epsilon]$ steps, the probability that Phase I has not ended is at most ϵ.

Now we tackle Phase II. Let D_t be the number of unknown steps at time t. We first note that D_t is more than a supermartingale–it never goes up! Whenever a site that is unknown is hit, that site permanently changes to known. The probability of hitting a site in D_t is just $|D_t|/n$, and this changes the number of unknowns by 1, so

$$E[|D_{t+1}| \mid \mathcal{F}_t] \leq |D_t| \left(1 - \frac{1}{n}\right)$$

so once more we have that $E[|D_{nk}|] \leq ne^{-k}$ and running for $n \ln n/\epsilon$ steps, Phase II will have ended with probability at least $1 - \epsilon$.

5.6 Widom-Rowlinson

As pointed out in Chapter 3, there are several different chains for this model. The birth death swapping chain will have the strongest theoretical characteristics. However, one of the other chains might be the faster depending on the implementation, and so we consider bounding chains for all of the chains given in Chapter 3.

5.6.1 Nonlocal conditioning chain

First we consider the nonlocal chain. For this chain, we chose a color at random, then rolled a uniform random variable for each node where that color was not blocked.

An acceptance rejection approach to this problem would be to roll a uniform for every node. Then if a node rolls to accept the color and is blocked, leave the node uncolored.

This idea leads naturally to a bounding chain approach where a node which might be blocked by an unknown node itself becomes unknown. Let $N(v)$ denote the node v together with all the neighbors of v.

Nonlocal bounding chain for Widom-Rowlinson

Set $y \leftarrow Y_t$
Choose $i \in_U \{1, \ldots, Q\}$
Let $D = \{v : |Y(v)| > 1\}$
For all nodes v
  Set $y(v) \leftarrow y(v) \setminus \{i\}$
For all nodes v
  Choose $U_v \in_U [0, 1]$
  If $U_v \leq \lambda_i/(1 + \lambda_i)$
    Case 1: All nodes in $N(v)$ are colored $\{0\}$
      Set $y(v) \leftarrow \{i\}$
    Case 2: there exists $w \in N(v) \cap D_t$ such that $w \neq \{0, i\}$
      Set $y(v) \leftarrow y(v) \cup \{i\}$

Figure 5.10: Nonlocal conditioning chain for Widom-Rowlinson

Theorem 5.7 *Let* $\bar{\lambda}_i = \lambda_i/(1 + \lambda_i)$, *and*

$$\beta = (\Delta + 1)\left[\sum_i \bar{\lambda}\right] - \min_i \bar{\lambda}.$$

If $\beta < 1$, *then the bounding chain will have detected complete coupling in the nonlocal Widom-Rowlinson chain after* $-2Q\ln(2nQ)/(1 - \beta)$ *steps with probability at least 1 / 2.*

If $Q = 2$, then we have a tighter bound for β,

$$\beta = (\Delta + 1)\left[\frac{1}{2} + \frac{1}{2}\sqrt{\bar{\lambda}_1 \bar{\lambda}_2}\right].$$

Proof: Let D_t^i denote the set of vertices v in D_t such that $i \in y(v)$. Then we proceed by looking at $E[|D_{t+1}^i| \mid \mathcal{F}]$. The first thing to note is that if color i is chosen, all of D_t^i is thrown out when i is removed from $y(v)$ for all v.

Therefore all of D_t^i is constructed in the second step. For a node to be added to D_t^i, it must be adjacent to or on top of a node in D_t^j for some $j \neq i$. Let $d^j(v)$ denote the number of neighbors of v which are in D_t^j.

$$\begin{aligned}
E[|D_{t+1}^i \mid \mathcal{F}] &= \frac{Q-1}{Q}|D_t^i| + \frac{1}{Q}\sum_{v \in V} P(v \in D_{t+1}^i \mid \mathcal{F}) \\
&\leq \frac{Q-1}{Q}|D_t^i| + \frac{1}{Q}\sum_{v \in V}\sum_{j \neq i} 1_{d^j(v)>0}\frac{\lambda_i}{\lambda_i + 1} \\
&= \frac{Q-1}{Q}|D_t^i| + \frac{1}{Q}\sum_{j \neq i}\sum_{v \in V} 1_{d^j(v)>0}\frac{\lambda_i}{\lambda_i + 1} \\
&= \frac{Q-1}{Q}|D_t^i| + \frac{1}{Q}\sum_{j \neq i}[\Delta + 1]|D_t^j|\bar{\lambda}_i,
\end{aligned}$$

where $\bar{\lambda}_i = \lambda_i/(1 + \lambda_i)$. Let z be the Q dimensional vector $[|z_1||z_2|\ldots|z_Q|]^T$ at time step t. What we have shown is that

$$E[z_{t+1}] = E[E[z_{t+1}|\mathcal{F}]] \leq AE[z_t],$$

with A being a Q by Q matrix:

$$\frac{[\Delta+1]}{Q}\begin{bmatrix} 0 & \bar{\lambda}_2 & \bar{\lambda}_3 & \dots & \bar{\lambda}_Q \\ \bar{\lambda}_1 & 0 & \bar{\lambda}_3 & \dots & \bar{\lambda}_Q \\ \vdots & & & & \\ \bar{\lambda}_1 & \bar{\lambda}_2 & \bar{\lambda}_3 & \dots & 0 \end{bmatrix} + \left(1 - \frac{1}{Q}\right) I$$

An inductive argument shows that $E[z_t] = A^t E[z_0]$. It is well known from linear algebra [20] that $||A^t x|| \leq \alpha^t ||x||$, where α is an upper bound on the magnitude of the eigenvalues of A. In our case, $||E[z_0]|| \leq \sum_i \bar{\lambda}_i n$, and so $||E[z_t]|| \leq \alpha^t n \sum_i \bar{\lambda}_i n$.

After $\ln(2n^2 \sum_i \bar{\lambda}_i)$ steps, $||E[z_t]|| \leq 1/(2n)$, so that $E[|z_i|] \leq 1/(2n)$ for all i. Therefore by Markov's inequality the probability that $|z_i| > 0$ is at most $1/(2n)$. Using the union bound, the chance that all of the $|z_i|$ are identically 0 is at most $1/2$.

It remains to bound α. Of course given actual values for the λ_i, it is a simple matter to numerically compute the value of α. When $Q = 2$, this may also be done analytically, yielding $\alpha = 1/2 + 1/2\sqrt{\bar{\lambda}_1 \bar{\lambda}_2}$. For $Q > 2$, the method of Gershgorin disks may be used to show that

$$\alpha \leq 1 - \frac{1}{Q} + \frac{1}{Q} \max_i \left\{ \sum_{j \neq i} \bar{\lambda}_i \right\} = 1 - \frac{1}{Q} \left[1 - \left[\sum_i \bar{\lambda} i \right] - \min_i \bar{\lambda}_i \right],$$

which completes the proof.□

Häggström and Nelander [18] gave a bounding chain for the local heat bath chain, and showed that for the specific case of the Widom-Rowlinson model where all the λ_i are equal to λ, the bounding chain efficiently detects complete coupling when $Q\lambda\Delta < 1$.

Note that for the nonlocal chain, our result gives a polynomial time guarantee when $(Q-1)\lambda[\Delta+1] < 1$. When $Q = 2$, this is almost a factor of 2 improvement for large Δ. To get a bound with both the $Q-1$, and only a factor of Δ instead of $\Delta+1$, we consider a stronger version of the bounding chain than was found in [18]. For this chain we will be able to show that it converges in polynomial time when $(Q-1)\lambda\Delta < 1$.

5.6.2 The single site heat bath chain

The base Markov chain we use is the single site heat bath chain. The acceptance rejection single site heat bath chain for Widom-Rowlinson works as follows. Choose a node v uniformly at random. Choose a color c for that node where $P(c=0) = 1/(1+\sum_i \lambda_i)$ and $P(c=i) = \lambda_i/(1+\sum_i \lambda_i)$. If color c is blocked at node v, then pick a new color, repeating until a nonblocking color is chosen for v.

Acceptance rejection single site heat bath Widom-Rowlinson bounding chain

Set $y \leftarrow Y_t$
Choose $v \in_U V$
Set $y(v) = \emptyset$
Repeat
  Choose $c \in_R$ so that $P(c=0) = 1/(1+\sum_i \lambda_i)$
    and $P(c=i) = \lambda_i/(1+\sum_i \lambda_i)$ for all $1 \le i \le Q$
  If for all w neighboring v, $y(w) \neq \{i\}$
    Set $y(w) \leftarrow y(w) \cup \{i\}$
Until $c = 0$ or c not blocked by a neighbor of v

Figure 5.11: Acceptance rejection single site heat bath Widom-Rowlinson bounding chain

In [18] it was noted that in bounding chains like the one above, the possibility exists of always changing the node to a particular color regardless of the values of the neighbors. In our case we always have a fixed probability of changing the color to 0. When this probability is greater than $\Delta/(\Delta+1)$, the bounding chain will detect complete coupling in polynomial time. Therefore this chain was previously known to converge when $1/(1+\sum_i \lambda_i) \geq \Delta/[\Delta+1]$, or equivalently, when $\sum_i \lambda_i \leq 1/\Delta$.

Experimentally, it was noted in [18] that this bound was quite loose, especially when $Q=2$. In fact, this chain has exactly the same behavior as the nonlocal chain.

Theorem 5.8 *Let* $\bar{\lambda}_i = \lambda_i/(1+\lambda_i)$, *and*

$$\beta = \Delta\left[\sum_i \bar{\lambda}\right] - \min_i \bar{\lambda}.$$

If $\beta < 1$, *then the bounding chain will have detected complete coupling in the nonlocal Widom-Rowlinson chain after* $2n\ln(2nQ)/(1-\beta)$ *steps with a probability that is at least 1 / 2.*

If $Q=2$, *then we have a tighter bound for* β,

$$\beta = \Delta\left[\frac{1}{2} + \frac{1}{2}\sqrt{\bar{\lambda}_1\bar{\lambda}_2}\right].$$

One immediate difference is that the number of steps is larger by a factor of n, owing to the fact that the nonlocal chain alters all n nodes simultaneously, whereas the local chain just alters one at a time. Here just measuring running time in terms of Markov chain steps can be misleading.

Proof: The proof is essentially the same as for the nonlocal chain. A node gets moved into z_i if it is chosen to be the changing node, i is chosen sometime during

the acceptance rejection process, and it is adjacent to a node in z_j for $j \neq i$. Let z_i' denote the unknown set for color i after the step is taken. Then if color 0 is chosen in the acceptance rejection process the repeat loop stops. Therefore, the probability of choosing i is at most the probability that i gets chosen before 0. But this is just the probability that i is chosen conditioned on either i or 0 being chosen, and is $\lambda_i/(1+\lambda_i)$. For each node v, let $d_i(v)$ denote the number of neighbors of v that lie in z_j for some $j \neq i$.

$$\begin{aligned}
E[|z_i'| - |z_i| \mid \mathcal{F}] &= \frac{1}{n}\left(\sum_{v\in z_i}(-1)P(v \notin z_i' \mid \mathcal{F}) + \sum_{v\notin z_i} P(v \in z_i' \mid \mathcal{F})\right) \\
&\leq \frac{1}{n}\left(\left[\sum_v P(v \in z_i' \mid \mathcal{F})\right] - |z_i|\right) \\
&\leq \frac{1}{n}\left[\sum_v \bar{\lambda}_i 1_{d_i(v)>0} - |z_i|\right] \\
&\leq \frac{1}{n}\left[\sum_v d_i(v)\bar{\lambda}_i\right] \\
&\leq \frac{1}{n}\left[\bar{\lambda}_i \sum_{j\neq i} |z_j|\Delta\right]
\end{aligned}$$

Note that this is exactly the same equation we derived for the nonlocal chain, except now $E[z_{t+1}] \leq AE[z_t]$, where

$$A = \frac{\Delta}{n}\begin{bmatrix} 0 & \bar{\lambda}_2 & \bar{\lambda}_3 & \dots & \bar{\lambda}_Q \\ \bar{\lambda}_1 & 0 & \bar{\lambda}_3 & \dots & \bar{\lambda}_Q \\ \vdots & & & & \\ \bar{\lambda}_1 & \bar{\lambda}_2 & \bar{\lambda}_3 & \dots & 0 \end{bmatrix} + \left(1 - \frac{1}{n}\right) I$$

This in the same as the matrix for the nonlocal chain, except instead of a $(\Delta+1)/Q$ factor in front of the first term, we have a Δ/n term, which is why our new running

time is $O^*(n)$ rather than $O^*(Q)$. Working through the eigenvalue bounds as before gives the result in the theorem. □

By considering a stronger bounding chain, we have increased the range of λ_i where have a polynomial guarantee by a factor of $Q/(Q-1)$. To further increase the range, we introduce a bounding chain for the birth death swapping chain.

5.6.3 The birth death swapping chain

For the bounding chain for the birth death swapping chain, we may prove the following.

Theorem 5.9 *Let* $\bar{\lambda}_i = \lambda_i/(1+\lambda_i)$, *and*

$$\beta = \frac{1}{2}(\Delta+1)\left[\sum_i \bar{\lambda}\right] - \min_i \lambda_i.$$

If $\beta < 1$, *then the bounding chain will have detected complete coupling in the nonlocal Widom-Rowlinson chain after* $2n\ln(2nQ)/(1-\beta)$ *steps with probability at least 1 / 2.*

If $Q = 2$, *then we have a tighter bound for* β,

$$\beta = (\Delta+1)min\left\{\frac{2}{3}\sqrt{\bar{\lambda}_1\bar{\lambda}_2}, \frac{1}{2}\max_i \lambda_i\right\}$$

The discrete bounding chain is derived from the birth death swapping bounding chain for the continuous Widom-Rowlinson model. The proof of this theorem and description of the bounding chain is completely analogous to the continous case, and will be postponed until Chapter 8.

5.7 The antivoter model

As mentioned in chapter 2, the antivoter model and the voter model are closely related with one important difference–the antivoter model has a stationary distribution (given a nonbipartite graph) whereas the voter model has absorbing states where all the nodes are colored the same way.

On the other hand, they are linked in that the mixing time for the antivoter chain is bounded above by the absorption time for the voter model. In fact, as we shall show, the time needed for the bounding chain to detect complete coalescence of the antivoter chain is the same as the time until absorption for the voter chain.

To facilitate complete coalescence, we add a symmetric move that at each step randomly permutes the color set. With probability 1/2, nodes colored 1 all flip to color 0 and nodes colored 0 all flip to color 1. All this accomplishes is to allow us at the first step to say exactly what the color of at least 1 node is, either 0 or 1. As with the sink free orientations chain, we need to know the color of at least one site at all times in order to make any progress. But once at least some of the nodes are known, then selecting an unknown node and a known neighbor cause the unknown node to become known.

Instead of labeling each node $\{0,1\}$ or unknown, assign each node a variable x_i, where $x_i \in \{0,1\}$. The bounding chain proceeds by flipping the value of the variable instead of the known value. Then the value of $y(i)$ is a monomial, either x_j or $1-x_j$ for some j in V. Once all the monomials $y(i)$ contain but a single variable x_j, we say that we have completely coupled. All nodes colored x_j will be a single

Antivoter bounding chain

Set $y \leftarrow Y_t$
Choose $v \in_U V$
Choose $U \in_U [0,1]$
If $U \leq 1/2$
  For all $v \in V$
    Set $y(v) = 1 - y(v)$
Choose w uniformly from the neighbors of v
Set $y(w) \leftarrow 1 - y(v)$
Set $Y_{t+1} \leftarrow y$

Figure 5.12: Antivoter bounding chain

color, and all nodes colored $1 - x_j$ will be a different color.

However, note that we may run the voter model on two colors in a similar fashion, except at each step $y(w) \leftarrow y(v)$ instead of $1 - y(v)$. Absorption occurs when all the nodes are colored x_j, which occurs at exactly the same time that all the nodes for the antivoter model are either x_j or $1 - x_j$. A more detailed analysis can show that the original antivoter chain without the flipping of color classes is rapidly mixing; here we just show that our modified chain rapidly mixes.

Theorem 5.10 *After $n^3\Delta/c_{min}$ time steps (where c_{min} is the size of the unweighted minimum cut in the graph), the probability that this chain detects complete coupling (alternatively, that the two color voter model reaches an absorption state) is at least $1 - \epsilon$.*

Proof: At each step, let j be the value such that the most nodes are colored either x_j or $1 - x_j$. Let A_t denote this set of nodes, and let $D_t = V \setminus A_t$. Define

$\phi_t = \sum_{v \in D_t} deg(v)$. Since $deg(v)$ is always positive, when $\phi_t = 0$ we know that $|D_t| = 0$, our standard goal.

A node w moves from A_t to D_{t+1} if a neighbor v is selected, and then w is the random neighbor of v which is selected to be changed, an event which occurs with probability $\frac{1}{n} \cdot \frac{d(v)}{deg(v)}$, where $d(v)$ is the number of neighbors of v which lie in D_t. This event changes the value of ϕ by $deg(v)$. Similarly, a node w moves from D_t to A_t if a neighbor v is selected followed by w being selected as the neighbor. This changes the value of ϕ by $-deg(v)$. The only way that $\phi_{t+1} \neq \phi_t$ is for one of these two events to occur.

$$\begin{aligned} E[\phi_{t+1} \mid \mathcal{F}_t] &= \phi_t + \frac{1}{n}\left[\sum_{v \in A_t} \deg(v)\frac{d(v)}{deg(v)} + \sum_{v \in D_t}(-deg(v))\frac{deg(v) - d(v)}{deg(v)}\right] \\ &= \phi_t - \frac{1}{n}\phi_t + \frac{1}{n}\sum_v d(v) \\ &= \phi_t - \frac{1}{n}\phi_t + \frac{1}{n}\phi_t \\ &= \phi_t \end{aligned}$$

Hence ϕ_t is not only a supermartingale, it is in fact a martingale as well. The sets A_t and D_t must be connected by a number of edges c_{min}, where c_{min} is the size of the minumum unweighted cut in the graph. Since $deg(v) \leq \Delta$ for all v, we know that each edge $\{v, w\}$ in the graph is selected with probability at least $1/(n\Delta)$ So the probability that ϕ changes value is bounded below by $c_{min}/(n\Delta)$, and using Theorem 4.5 completes the proof.

5.8 The list update problem

The bounding chain for the list update chains are straightforward, but unfortunately seem to take longer to detect complete coupling. Consider the direct MA1 chain, which selects an item at random according to a distribution p, then swaps that item with the chain directly in front of it. To create a bounding chain, we need only to keep track of where each possible item could be. When $|y(v)| = 1$ for all

MA1 list update chain

Set $y \leftarrow Y_t$
Request $i \in_R \{1, \ldots, n\}$ with the probability of choosing i is p_i
If for some j, $y(j) = \{i\}$
  Set $y(j) \leftarrow y(j-1)$
  Set $y(j-1) \leftarrow \{i\}$
Else
  For all j from n down to 2
    If $i \in y(j)$
      Set $y(j) \leftarrow y(j) \setminus \{j\}$
      Set $y(j) \leftarrow y(j) \cup y(j-1)$
      Set $y(j-1) \leftarrow y(j-1) \cup \{j\}$
Set $Y_{t+1} \leftarrow y$

Figure 5.13: MA1 list update chain

$v \in \{1, \ldots, n\}$, the bounding chain has detected complete coalescence.

The arbitrary transposition chain for the MA1 list update process has a similar bounding chain. Our approach is brute force. Given the two positions picked, one has a color set and the other has a color set. For each pair, we use our random uniform U to decide the ordering, and update the color sets accordingly. Note that taking a single step of the bounding chain may take up to n^2 time (within the for

Arbitrary transposition for MA1 bounding chain

Set $y \leftarrow Y_t$
Choose $w_1 \in_U \{1, 2, \ldots, n\}$
Choose $w_2 \in_U \{1, 2, \ldots, n\} \setminus \{v_1\}$
Set $v_1 \leftarrow \min\{w_1, w_2\}$
Set $v_2 \leftarrow \max\{w_1, w_2\}$
Set $y_1 \leftarrow y(v_1)$
Set $y_2 \leftarrow y(v_2)$
Set $y(v_1) \leftarrow \emptyset$
Set $y(v_2) \leftarrow \emptyset$
Choose $U \in_U [0, 1]$
For each (i, j) with $i \in y_1$ and $j \in y_2$
  If $U \leq p_i^d/(p_i^d + p_j^d)$
    Set $y(v_2) \leftarrow y(v_2) \cup \{j\}$
    Set $y(v_1) \leftarrow y(v_1) \cup \{i\}$
  Else **Set** $y(v_2) \leftarrow y(v_2) \cup \{i\}$
    Set $y(v_1) \leftarrow y(v_1) \cup \{j\}$
Set $Y_{t+1} \leftarrow y$

Figure 5.14: Arbitrary transposition for MA1 bounding chain

loop). This is in sharp contrast to many of our other chains where the time for a single step was roughly the same as for the original Markov chain.

When the weights p_i are geometric, so that $p_i = \theta^i$, the higher weight items tend to be placed at the front with very high probability. Under these conditions, a modified version of this bounding chain where a step takes only as long as the Markov chain step may be used. Moreover, this chain converges quickly under certain conditions. This chain takes advantage of the fact that the high weight items tend to collect on the left side. We let m be the smallest value such that items 1 through m are all known. Let $LEFT$ be the set $\{1, \ldots, m\}$ and $RIGHT$ be the rest of the nodes. Then instead of just wildly choosing positions, we first

decide whether positions come from $LEFT$ or $RIGHT$ and proceed from there.

Note that for any permutation, choosing a random position in that permutation is equivalent to choosing a random item. This equivalence will be very useful in constructing a bounding chain.

Theorem 5.11 *Suppose that* $p_{i+1}/p_i \leq \theta \leq 1/5$ *for all* i. *Then*

$$E[\textit{time until complete coupling}] \leq 10n^3.$$

Proof: Given that the weights decrease geometrically, we know that it is likely that the high weight items will be at the front. Therefore, in this proof we take a departure from keeping track of $|D_t|$, and instead keep track of m_t, where m_t is the largest value such that $\{1, \ldots, m_t\}$ are in A_t. Some value $i > m_t + 1$ might also be in A_t, but we will not use that information in our analysis.

We shall show that on average the value of m_t grows larger at each time step, so that $n - m_t$ is a supermartingale. Once $m_t = n$, complete coupling has occurred and we are done.

In case 1, both records are chosen from nodes in A_t, therefore $m_{t+1} = m_t$ since we are only moving around known nodes.

The value of m_t can change in the other two cases. For instance, if both values come from $RIGHT$, then it is conceivable that the value of m_t can go up if we know the item which gets placed in position $m_t + 1$.

The probability that both choices come from $RIGHT$, that one choice has position $m + 1$, and the other choice is the largest record in $RIGHT$ is at least $2/n^2$ (it could be $1/n$ if these two positions are the same). The ratio p_j/p'_j is at least θ,

so the probability that this largest record gets sorted with the record in position $m_t + 1$ is at least $1/(1+\theta)$. Hence $P(m_{t+1} = m_t + 1) \geq 2/[n^2(1+\theta)]$.

Unfortunately, m_t may go down if we choose one position from $LEFT$ and the other position from $RIGHT$. This happens when $a \leq m$. Potentially, $m_{t+1} = a - 1$. To upper bound the probability that this occurs, we first note that the probability that position a and a particular unknown record j are both chosen is just $2/n^2$. Now the closest that an unknown record can be to positon a is $m + 1 - a$, and so the probability that sorting occurs is at least $p_\ell^{m+1-a}/(p_\ell^{m+1-a} + p_s^{m+1-a})$. Therefore the probability of becoming unknown is at most

$$\frac{p_s^{m+1-a}}{p_\ell^{m+1-a} + p_s^{m+1-a}} = \frac{1}{(p_\ell/p_s)^{m+1-a} + 1}.$$

This ratio is largest when p_ℓ/p_s is smallest. We know that $p_i/p_{i+d} \geq 1/\theta^d$ and $p_{i-d}/p_i \geq 1/\theta^d$.

Let i be the record at position a. For each d there are at most two records whose label is d away from i. Moreover, the distance between these two records and a is at least $m+1-a$. Hence the probability that position a becomes unknown is bounded above by

$$2 \sum_{d=1}^{(n-m)/2} \frac{\theta^{d(m+1-a)}}{1 + \theta^{d(m+1-a)}} \leq 2\theta^{m+1-a}/(1 - \theta^{m+1-a}).$$

Combining these terms, we have that

$$\begin{aligned}
E[m_{t+1}|m_t] &\geq (m_t + 1)\frac{2}{n^2} \cdot \frac{1}{1+\theta} + 2\sum_{a=1}^{m_t}(a-1)\frac{2}{n^2} \cdot \frac{\theta^{m_t+1-a}}{1+\theta^{m_t+1-a}} \\
E[m_{t+1} - m_t|m_t] &\geq \frac{2}{n^2}\left[\frac{1}{1+\theta} + 2\sum_{a=1}^{m_t}(a-1-m_t)\frac{\theta^{m_t+1-a}}{1+\theta^{m_t+1-a}}\right] \\
&\geq \frac{2}{n^2}\left[\frac{1}{1+\theta} + 2\sum_{a'=1}^{m_t}(-a')\frac{\theta^{a'}}{1-\theta^{a'}}\right]
\end{aligned}$$

$$\begin{aligned} &\geq \frac{2}{n^2}\left[\frac{1}{1+\theta} - 2\frac{\theta}{(1-\theta)^3}\right] \\ &\geq \frac{1}{10n^2} \end{aligned}$$

Therefore, the expected number of steps needed until $m_t = n$ is $n/(1/10n^2)$, or just $10n^3$ by Wald's identity.

5.8.1 Application to nonparametric testing

This problem of sampling from the list update distribution with geometric weights has another application quite unrelated to the list update problem. Suppose that we have sampled 100 people about their favorite ice cream. Ranking the results gives us a permutation. We wish to construct a test for a specific statistic of some sort, such as how many ice cream makers in the top five positions are located in North Dakota.

In order to determine if our test is statistically significant, we first need some model of how the possible rankings are distributed. One common method for accomplishing this is to assume that given the true distribution of rankings μ_1, a sampled permutation μ_2 is randomly chosen with weight proportional to $\theta^{d(\mu_1,\mu_2)}$ where $\theta < 1$ so that it is unlikely that our surveyed rankings will be a great distance from the true set of rankings.

This definition begs the question, what distance do we use to measure $d(\mu_1, \mu_2)$? One possibility is the sum of squares distance, that is

$$d(\mu_1, \mu_2) = \sum_i (\mu_1(i) - \mu_2(i))^2,$$

an idea known as Molloy's Rule.

There are some obvious variations on this, such as using $|\mu_1(i) - \mu_2(i)|$ instead of the squares. However, we shall look further at the sum of squares distance and see where it leads. Expanding, we have that

$$\begin{aligned} d(\mu_1, \mu_2) &= \sum_i \mu_1(i)^2 - 2\mu_1(i)\mu_2(i) + \mu_2(i)^2 \\ &= \left(\sum_i \mu_1(i)^2\right)\left(\sum_i \mu_2(i)^2\right)\left(\sum_i -2\mu_1(i)\mu_2(j)\right). \end{aligned}$$

Two facts help us out. First, assume that μ_1 is the identity permutation, this changes nothing if we assume that μ_1 has been applied to μ_2. Second, the first two terms in this product are constants, and so θ raised to these terms are constants as well. Therefore the distribution $\pi(\mu) = \theta^{d(\mu_1,\mu_2)}/Z$ satisfies:

$$\begin{aligned} \pi &\sim \theta^{\left(\sum_i -2i\mu_2(j)\right)} \\ &\sim \prod_i \left(\theta^{-2i}\right)^{-\mu_2(i)} \\ &\sim \prod_i \left(\theta^{-2i}\right)^{n-\mu_2(i)} \end{aligned}$$

where again we may insert the n in the final equation because it is simply a multiplicative constant. This is exactly the form of the geometrically weighted list update problem with ratio θ^2, and so if $\theta^2 < 1/5$ our previous theorem tells us that we may exactly sample for this problem as well.

5.9 Other applications of bounding chains

We have already seen that having a bounding chain available gives us a means for experimental determination of the mixing time of the chain. However, the potential of bounding chains is much greater. In the next chapter, we discuss coupling from the past (CFTP), a means for generating perfect samples from a distribution. Bounding chains give a way to use CFTP for a particular chain. Moreover, upper bounds on the time needed for bounding chains to detect complete coupling will provide an upper bound on the running time of the algorithm.

Coupling from the past has but one weakness, that the user must commit to running the algorithm until termination in order not to introduce bias into the sample eventually obtained. Therefore in the chapter following our discussion of CFTP we examine another method for generating perfect samples based on the concept of strong stationary times. Again the existence of bounding chains will be an essential component in creating these perfect samplers.

Arbitrary transposition for MA1 bounding chain II
Set $y \leftarrow Y_t$
Set $A_t \leftarrow \{v : |y(v)| = 1\}$
Set $m \leftarrow \min_i\{y(1), \ldots, y(i)\} \subset A_t$
Set $LEFT \leftarrow \{y(1), \ldots, m\}$, $RIGHT \leftarrow \{y(m+1), \ldots, m\}$
Choose U_1 uniformly at random from $[0, 1]$
Case 1 $U_1 \leq \left(\frac{m}{n}\right)^2$
  Choose $i \in_U LEFT$, **Choose** j uniformly from the items in $LEFT$
  Let a be the position of record i
Case 2 $\left(\frac{m}{n}\right)^2 < U_1 \leq \left(\frac{m}{n}\right)^2 + 2\left(\frac{m}{n}\right)\left(\frac{n-m}{n}\right)$
  Choose i uniformly from the items in $LEFT$
  Choose j uniformly from the items in $RIGHT$
  Let a be the positions of record i
Case 3 $\left(\frac{m}{n}\right)^2 + \left(\frac{m}{n}\right)\left(\frac{n-m}{n}\right) < U_1$
  Choose U_2 uniformly at random from $[0, 1]$
  If $U_2 < 2/(m-n)^2$
  Set $a = m + 1$, **Set** j to be the highest probability item in $RIGHT$
Choose U_3 uniformly at random from $[0, 1]$
If $a = m + 1$ and j is the highest probability unknown record
  Let j' be second highest probability among unknown records
  If $U_3 \leq \frac{1}{1+(p'_j/p_j)}$
    Let $y(a) = \{j\}$
Else if $a \leq m$
  Set $p_\ell \leftarrow \max\{p_i, p_j\}$, **Set** $p_s \leftarrow \min\{p_i, p_j\}$
  If record j is in $LEFT$
    If $U_3 \leq \frac{1}{1+(p_s/p_\ell)^{|b-a|}}$
      Sort items i and j
    Else
      Antisort items i and j
  If record j is in $RIGHT$
    If $U_3 \leq \frac{1}{1+(p_s/p_\ell)^{(m+1-a)}}$
      Let $y(a) = \{\ell\}$
    Else
      Let $y(a) =?$
Set $Y_{t+1} \leftarrow y$

Figure 5.15: Arbitrary transposition for MA1 bounding chain

Chapter 6

Perfect sampling using coupling from the past

> It is the mark of an educated mind to rest satisfied with the degree of precision which the nature of the subject admits and not to seek exactness where only an approximation is possible.
>
> *-Aristotle*

The traditional Monte Carlo Markov chain method yields an answer that is only approximately distributed according to the desired distribution, and this was the state of the art for decades. Fortunately, certain researchers never read Aristotle, and in the last five years methods have been discovered which allow exact sampling from the stationary distribution of a chain. Such a method is often referred to as a *perfect sampling* algorithm.

Propp and Wilson [41] introduced coupling from the past (CFTP) as a simple

algorithm for generating samples drawn exactly from the stationary distribution of a chain. In only a few years, Monte Carlo Markov chain practitioners have expanded the scope of CFTP, applying the procedure to dozens of different chains.

The general idea is quite simple. We have often stated that the stochastic processes which we consider have index set over all integers, including negative ones. We now describe how to simulate such a chain.

6.1 Reversing the chain

Suppose that X_0 has distribution p_0. Then the distribution of X_1 will be $p_1 = p_0 P$. We wish to find a distribution p_{-1} such that $p_0 = p_{-1} P$. If P has a unique stationary distribution π, then $\pi P = \pi$, and setting X_i to have distribution π for all i (including negative i) gives the result.

Hence we suppose that each X_i is distributed according to the unique stationary distribution of the chain. Given X_0, we wish to simulate $X_{-1}, X_{-2}, \ldots$. Furthermore, we wish to insure that $P(X_i = x_i | X_{i-1} = x_{i-1}) = P(x_{i-1}, x_i)$. Using the fact that X_i, X_{i-1} are stationary gives

$$\begin{aligned} P(X_i = x_i | X_{i-1} = x_{i-1}) &= \frac{P(X_i = x_i, X_{i-1} = x_{i-1})}{P(X_{i-1} = x_{i-1})} \\ &= \frac{P(X_{i-1} = x_{i-1} | X_i = x_i)}{P}(X_i = x_i) P(X_{i-1} = x_{i-1}) \\ &= P(x_{i-1}, x_i) \frac{\pi(x_i)}{\pi(x_{i-1})} \end{aligned}$$

This relationship inspires the following definition.

Definition 6.1 *For a Markov chain with transition matrix P and unique stationary*

distribution π*, let*

$$\tilde{P}(y,x) = \frac{\pi(x)}{\pi(y)} P(x,y)$$

be the reversibilization *of* P*. If a chain is reversible, then* $\tilde{P} = P$*.*

This is why the detailed balance condition is also known as reversibility. Given a reversibile chain, and X_0 started in the stationary distribution, we can run the chain backwards according to P in order to obtain $X_0, X_{-1}, X_{-2}, \ldots$ which are also distributed according to π, and $\ldots, X_{-2}, X_{-1}, X_0$ will be distributed as a stochastic process on a Markov chain.

6.2 Coupling from the past

Coupling from the past utilizes this characterization to obtain samples that are drawn exactly from the stationary distribution. Suppose that X_0 is stationary. Then using our construction, X_{-t} will also be stationary for all t. Therefore, CFTP starts at X_{-t} and runs forward up to time 0. Suppose that F_0^{-t} is constant. Then we have that $X_0 = F_0^{-t}(X_{-t})$ which is a known value. Therefore we have obtained a perfect sample from the stationary distribution, X_0.

The only difficulty arises when F_0^{-t} is not constant. Then we simply increase t until F_0^{-t} is constant. As long as this eventually occurs with probability 1, this algorithm will almost surely terminate.

Coupling from the past

Set $t = 0$
Repeat
  Set $t \leftarrow 2t - 1$
  Run chain from t to $(t+1)/2$
Until $F_{(t+1)/2}^{-t}$ is constant
Output $F_t^0(\Omega)$ as our answer

Figure 6.1: Coupling from the past (CFTP)

6.2.1 CFTP and bounding chains

Propp and Wilson [41] noted that CFTP may be used anywhere F_0^{-t} may be shown to be constant, although most of their examples dealt with monotonic Markov chains. For monotonic chains, recall that we need only keep track of $F_0^{-t}(\hat{1})$ and $F_0^{-t}(\hat{0})$, and wait until they meet. It was shown in [41] that the expected time until they meet is of the same order as the mixing time of the Markov chain.

With bounding chains, we only know that the complete coupling time gives an upper bound for the mixing time of the chain, but the reverse may not be true. Still, bounding chains do allow us to determine when F_0^{-t} is constant, and so immediately the results of the last two chapters indicate that we have algorithms for perfect sampling from the hard core gas model using the Dyer-Greenhill chain, the Potts model using single site update or Swendsen-Wang, the k colorings of a graph, the sink free orientations of a graph, the antivoter model, the restricted hypercube, and the list access problem.

The running time of $CFTP$ is of the same order as the time needed for complete

coupling to be detected by the bounding chain. Therefore our results showing that the bounding chain detects complete coupling immediately extend to give a priori bounds on the running time of CFTP.

The amount of work needed for CTFP comes from the memory required to store each $f_{-t'}$ for $t \leq t' \leq 0$. In practice, random seeds are used for a pseudorandom number generator that creates the same sequence of "random" numbers each time (given the same seed). Therefore, the memory requirements are reduced to retaining the seed for $t = -1, -2, -4, \ldots$, which is usually on the order of $\ln n$.

6.2.2 Coupling from the future

As Stephen Hawking pointed out, "Disorder increases with time because we measure time in the direction in which disorder increases." In coupling from the past, we found X_0 by starting at a point $-t$ in time and then running forward. The counterintuitive part of this process from a physical point of view is that disorder, as measured by the number of possible states admitted by the bounding chain, is decreasing as time moves forward. However, we could modify our algorithm to start at a point $t > 0$ in time and run backwards using the reversed transition matrix $\tilde{P}$. We shall refer to this time reversed version of the algorithm as coupling from the future (CFTF).

Practically, CFTF is no different from CFTP. Most of the chains we consider here are reversible, and so running the chain forwards or backwards makes no difference whatsoever to running time or memory requirements. This notion of CFTF does have some nice theoretical implications, however.

The first concerns the time arrow associated with entropy (disorder). A general notion of entropy is the logarithm of the number of states that the system could possibly be in. With bounding chains, the entropy of a state Y_t is $\sum_v \ln |Y_t(v)|$. When the entropy is 0, $|Y_t(v)| = 1$ for all v and we have completely coupled. In coupling from the past, the entropy decreases as time moves forward, contrary to the usual physical use of the term.

With CFTF, the bounding chain is working on the reversed chain, and so as the entropy decreases we are moving backwards in time, exactly as intuition would suggest.

Second, with coupling from the future we need only define our stochastic process on index set $0, 1, \ldots$. We only need the reversibilization for moving backwards on a finite set of indices, not for the entire set of negative integers.

Third, the method of attack in Chapter 7 will also concentrate on a process $X_0, X_1, \ldots$ where X_0 is assumed to be stationary. This forward way of looking at things makes clearer the connection between the two.

CFTF, like CFTP, is an example of an uninterruptible perfect sampling algorithm. Once a run is started to compute X_0, the user must commit to finishing the run in order to obtain unbiased samples. In the next chapter we show how this limitation can be removed with some extra work.

Chapter 7

Perfect sampling using strong stationary times

> Technological progress has merely provided us with more efficient means for going backwards.
>
> *-Aldous Huxley*

The reversibilization of a Markov chain, the ability to go backwards in time, provides the cornerstone for another algorithm for generating exact samples. Unlike CFTP, this algorithm will be interruptible, in that the user can give up, shut off the computer, and go home at any time in the algorithm without introducing bias into the sample.

Suppose that we have a Markov process $(X_0, X_1, \ldots)$ where X_0 is begun in the unique stationary distribution π of the chain. Then X_t will also be stationary for all times t. Unfortunately, the simulator must start in a specified state, so instead

of having the situation $P(X_t = x) = \pi(x)$, we must deal with $P(X_t = x|X_0 = x_0)$ which in general will not be $\pi(x)$.

We "counteract" the effect of conditioning on the value of X_0 using a strong stationary stopping time. Recall that a stopping time is any random variable τ such that the event $\tau \leq t$ is $\sigma(X_0, \ldots, X_t)$ measurable. A strong stationary stopping time is one that obviates the effects of starting in a particular state.

Definition 7.1 *Say that τ is a* strong stationary stopping *time if*

$$P(X_t = x|\tau \leq t, X_0 = x_0) = \pi(x).$$

Here we concentrate our efforts on a perfect sampling technique introduced by Fill [14] for a class of chains which are stochastic monotonic (a relaxation of the monotonicity property discussed earlier). In this paper Fill commented that the method could be generalized, and Murdoch and Rosenthal [37] specified an algorithm which was applicable to a broader range of chains. Here we shall refer to this more general algorithm as FMR.

Murdoch and Rosenthal developed FMR as an algorithm. Here we show that their idea also leads to a strong stationary stopping time. Using this idea, we develop bounds on the running time of FMR in terms of the complete coupling time and stationary mixing time of the chain. Rather than follow Murdoch and Rosenthal's development, we begin by examining how general strong stationary times may be developed using complete coupling.

Consider again $X_0, X_1, \ldots$ where each X_t has distribution π. Suppose that τ is any stopping time which occurs by time t with positive probability and both $\pi(x_0)$

and $\pi(x)$ are positive. Then

$$\begin{aligned} P(X_t = x | \tau \leq t, X_0 = x_0) &= \frac{P(X_t = x, X_0 = x_0 | \tau \leq t)}{P(X_0 = x_0 | \tau \leq t)} \\ &= \frac{P(X_0 = x_0 | \tau \leq t, X_t = x)}{P(X_0 = x_0 | \tau \leq t)} P(X_t = x) \\ &= \frac{P(X_0 = x_0 | \tau \leq t, X_t = x)}{P(X_0 = x_0 | \tau \leq t)} \pi(x) \end{aligned}$$

Our goal is to construct a τ such that the fraction multiplying $\pi(x)$ is equal to 1. Recall that for a process such as X, the reversibilization $\tilde{P}(y, x) = \pi(x)P(x, y)/\pi(y)$ allows us to simulate the chain in the reverse time direction. Therefore, one method of generating the random vector $X_0, \ldots, X_t$ is to generate X_t according to π, and then run the chain backwards using $\tilde{P}$.

Just as we use functions f_t (where $X_{t+1} = f_t(X_t)$) to take moves on the Markov chain in the forward direction, let $\tilde{f}_t$ (where $X_{t-1} = \tilde{f}_t(X_t)$) take moves in the reverse direction. For $a < b$, let $\tilde{F}_b^a = f_b \circ f_{b-1} \circ f_{a+1}$ so that $X(a) = \tilde{F}_b^a(X(b))$.

Now suppose that F_b^a is a constant. Then this constant is a random variable with a distribution that is independent of $\sigma(X_t)$. Therefore, we let τ be the first time t such that F_t^0 is constant. This means that

$$P(X_0 = x_0 | \tau \leq t, X_t = t) = P(X_0 = x_0 | \tau \leq t)$$

and

$$P(X_t = x | \tau \leq t, X_t = t) = \pi(x).$$

Recall that in CFTF (and CFTP) the goal was to determine the value of the fixed random variable X_0. Now we are more flexible. We are willing to accept any

random variable X_t as stationary, as long as complete coupling has occurred moving backwards from time t to time 0.

Algorithmically, this may be used to take perfect samples as follows. Start X_0 in an arbitrary state x_0. Run the chain forward to time t. This generates a path $X_0, X_1, X_2, \ldots, X_t$. Now run the chain backwards from time t to 0 conditioned on the moves made on the path. If these backwards moves completely couple the chain, then X_t will be stationary.

The functions $\tilde{f}_t$ move the process backwards in time. By conditioning on the backwards path, we mean that we choose $\tilde{f}_t$ conditioned on the event that $\tilde{f}_t(X_t) = X_{t-1}$.

FMR Perfect Sampling

Input: T, $X_0, X_1, \ldots, X_T$

Run the chain backwards, conditioned on the path $X_T, \ldots, X_0$,
as a complete coupling chain
If the backward chain completely couples by time 0,
Output $\tau \leq t$ is true
Else
Output $\tau \leq t$ is false

Figure 7.1: Complete coupling strong stationary times

Use of a strong stationary stopping time has one great advantage over CFTP (and CFTF). It is interruptible. The user runs the chain forward until $\tau \leq t$. If this takes too long, the user can abort the process without introducing any bias into samples which might be taken later. As Fill notes [14], this is really more of a theoretical advantage than a practical one, since when CFTP has small expected

running time, the probability that the actual running time is larger by a factor of k declines exponentially in k.

The amount of work can be much greater for this procedure than for CFTP. We must run the backward chain conditioned on the forward path, which may be quite difficult to do. However, for local update algorithms this is usually quite easy, and later we describe how this procedure may be applied to the Dyer-Greenhill chain for the hard core gas model.

On the other hand, the amount of work can also be much smaller. We are conditioning on the forward path started at x_0. The choice of x_0 can make it more likely that the chain will have completely coupled. Consider the case of single site update for the hard core gas model. If we start x_0 at the state of all nodes colored 0 (the empty independent set), then in the backwards moves it is more likely to color a node 0. We have seen in the bounding chain that when a node is colored 0 it immediately moves from unknown to known. Therefore it is possible that starting will all nodes colored 0 makes it more likely that the backwards bounding chain will show complete coupling.

7.0.3 Upper bounds on the strong stationary stopping time

Recall from Chapter 1 that the separation distance to π is defined as

$$||p - \pi||_S = \sup_{A|\pi(A)>0} (1 - p(A)/\pi(A)) .$$

Just as the coupling theorem allows us to bound total variation mixing time in terms of a coupling stopping time, Diaconis and Aldous [3] showed how separation

mixing time is bounding by strong stationary stopping times.

Theorem 7.1 *Suppose that a Markov chain has a strong stationary stopping time* τ*, then*

$$||p_x^t - \pi||_S \leq P(\tau > t).$$

Therefore, we do not expect that our strong stationary stopping time will run faster than the separation mixing time. Recall that $\tau_S(1/2)$ is the first time that the separation distance starting at an arbitrary state falls below $1/2$, and T_{BC} is the time that the bounding chain detects complete coupling.

Theorem 7.2 *Let* $T = 2E[T_{BC}] + \tau_S(1/2)$*. Then for any positive* t*,*

$$P(\tau > t) \leq (3/4)^{\lceil t/T \rceil}.$$

where $E[\tau] \leq 2E[T_{BC}] + \tau_S(1/2)$*. For reversible chains, we may also set* $T = 6E[T_{BC}]$*.*

Proof: Intuitively, running the chain for $2E[T_{BC}]$ steps gives the bounding chain time to detect complete coupling in the reverse direction. Running the chain for another T_S steps after that down to 0 allows the chain to almost reach the stationary distribution, so that conditioning on the value of X_T does not change the probability of complete coupling too much. Let BC_0^T denote the event that the bounding chain detects complete coupling from time T to time 0. Then given that we start at x_0, the probability that $\tau \leq T$ is just $P(BC_0^T | X_0 = x_0)$. Let $T = 2E[T_{BC}] + \tau_S(1/2)$ By Markov's inequality, running the chain backwards from from time T to time $T - 2E[T_{BC}] = \tau_S(1/2)$ gives us at least a $1/2$ chance of the bounding chain detecting

complete coupling. The remaining time from T' down to 0 gives us at least a $1/2$ chance that X_0 will be stationary no matter what the value of state $X_{\tau_S(1/2)}$. Hence, even if we know that the bounding chain detected complete coupling from time T down to time $\tau_S(1/2)$, we know that X_0 will still be close to stationary.

$$\begin{aligned} P(BC_0^T | X_0 = x_0) &= \frac{P(BC_0^T, X_0 = x_0)}{P(X_0 = x_0)} \\ &\geq \frac{P(BC_{T-2E[T_{BC}]}^T, X_0 = x_0)}{P(X_0 = x_0)} \\ &= \frac{P(BC_{T-2E[T_{BC}]}^T) P(X_0 = x_0 | BC_{T-2E[T_{BC}]}^T)}{P(X_0 = x_0)} \\ &\geq \frac{1}{2} \cdot \frac{1/2\pi(x_0)}{\pi(x_0)} \\ &= 1/4. \end{aligned}$$

Therefore $P(\tau \geq T) \leq 1/4$ and since the intervals $[0, T], [T+1, 2T], \ldots$ are independent, $P(\tau \geq kT) \leq (3/4)^k$, which gives the first result.

For reversible chains, it is well known that $\tau_S(1/2) \leq 4\tau_{TV}(1/2) \leq 2E[T_{BC}]$ [2] which yields the final result. $\Box$.

This shows that when dealing with reversible chains, the running time of FMR will be similar to the running time of CFTP in number of steps, with the added bonus of only requiring a set amount of time and memory before the algorithm begins. (Note that for the specific case of monotonic reversible chains, Fill proved a tighter running time bound of just T_S which might be faster than T_{CC} [14].)

7.1 Application to local update chains

In this section, we apply this strong stationary stopping time procedure to the hard core gas chain of Dyer and Greenhill 3.8 and the single site Widom-Rowlinson heat bath chain, both of which are local update Markov chains.

7.1.1 The Hard Core Gas Model

Recall from Chapter 3 the Dyer-Greenhill chain for the hard core gas model. The use of FMR requires two things: first, an efficient means for determining when complete coupling has occurred; and second, a way of running the chain forward conditioned on a single path outcome.

One technique for determining complete coupling is the bounding chain given in Chapter 4 as Figure 4.2. For the remainder of this section, then, we discuss the requirement specific to FMR, that of running the chain conditioned on the outcome of a single particle.

Consider the path $X_0, X_1, \ldots, X_T$. If we needed to keep track of the entire state of X_t for every t in $0, \ldots, T$, then FMR would require vastly more memory than CFTP. Fortunately, for local update chains this is not necessary. Just as with CFTP, it is enough to record the move made from X_t to X_{t+1} for all t in $0, \ldots, T-1$.

For the Dyer-Greenhill chain, these moves consist of three types. HOLD, where $X_{t+1} = X_t$, FLIP(v), where the move consists entirely of flipping the color of node v from 0 to 1 (or vice versa), and SWAP(v, w), where node v was colored 0, node w was colored 1, and the chain switched their values. We deal with each of these

moves in turn.

For HOLD, we must choose at random f_t such that $f_t(X_t) = X_t$. This may be accomplished through acceptance rejection. Pick a random f_t as usual. If $f_t(X_t) = X_t$, then keep this move, otherwise reject it and begin again. The expected time needed to make a move is just the inverse of the probability that a random f_t fixes X_t. If v is already 0, then this is the probability that v is chosen to be 0. If v is 1, then this is the probability that v is chosen to be 1. A lower bound on the probability of holding is the minimum of these two chances. Hence

$$P(f_t(X_t) = X_t) \geq \min\left\{\frac{1}{1+\lambda}, \frac{\lambda}{1+\lambda}\right\},$$

and the expected number of choices of f_t we must make until acceptance is just $(1+\lambda)\max\{1, 1/\lambda\}$.

For the FLIP(v) move, clearly the node chosen for f_t is v. If node v moved from 0 to 1, then U is uniform over $[1/(1+\lambda), 1]$, and if v moved from 1 to 0, then U is uniform over $[0, 1/(1+\lambda)]$.

Finally for the SWAP(v, w) move, v was again the chosen node, while U is uniform over $[0, \lambda/(4(1+\lambda))]$. Hence (treating λ as a constant), the memory needed is similar to CFTP where the randomness is coming from true random numbers.

7.1.2 Single site Widom-Rowlinson

The discrete Widom-Rowlinson case is similar. There are three types of moves, HOLD, where $X_{t+1} = X_t$. In MOVE1(v,c,) node v is changed to color c, where neighbors of v include at least one colored c, and MOVE2(v,c) where v is changed

to color c, but all of the neighbors of v has color 0. Again the $HOLD$ move may be accomplished using acceptance/rejection sampling. The expected time needed will be at most $(1+\lambda)\max 1, Q/\lambda$. The MOVE1($v$,$c$) case has the choice of node immediately being v, and knowing that the color moved indicates that U is uniform over $[1/(1+\lambda), 1]$. Finally, MOVE2(v,c) has choice of node v, and U uniform over the range of values which give color c.

As in the hard core gas model case, the information about moves which needs to be saved is in fact quite small, and the path in its entirety does not need to be saved.

7.2 Application to nonlocal chains

Of course, just because a chain is nonlocal does not mean that FMR cannot be applied. Recall the nonlocal update chain for discrete Widom-Rowlinson (Figure 3.14) drew a set of points of a particular color which are Poisson distributed. Now suppose that we are given two states X_{t+1} and X_t, and we are trying to compute $\tilde{f}_{t+1}$ conditioned on the fact that $f_{t+1}(X_{t+1}) = X_t$.

In going from X_t to X_{t+1}, a color c was chosen, all points of that color were removed, and the color c was added independently for each nonblocked node with probability λ_c. In going from X_{t+1} to X_t, then, we clearly must choose the same color c. Now the location of the points of color c give us partial information about what our choices for each node was. Basically, if a node is unblocked in X_t and not colored c, then we do not color that node c. If it is colored c in X_t, then that node

will be colored c if it is not blocked. Finally, if a node in X_t is blocked, we must randomly choose whether or not to color the node c, since the value at X_t imparts no information.

In other words, at each step of the chain we distribute color c as a Poisson point process with rate λ_c on the nodes. The distribution of points of color c in X_t tells us the value of the point process on all nonblocked nodes, which can then be easily extended to a point process on all of the nodes. Total memory requirement is again the memory needed to record a single step of the chain, making the memory needs roughly the same as with CFTP.

The theoretical advantages of FMR do not appear likely to outweigh the algorithm complexity over CFTP. However, it is interesting to note when an interruptible perfect sampling algorithm exists that is competitive with CFTP.

Chapter 8

Continuous Models

> I am so in favor of the actual infinite that instead of admitting that Nature abhors it, as is commonly said, I hold that Nature makes frequent use of it everywhere, in order to show more effectively the perfections of its Author.
>
> *-Georg Cantor*

Until now we have only dealt with discrete state spaces Ω. However, Monte Carlo Markov chain methods can also be used to obtain samples when the state space is continuous. We begin by introducing some new techniques and notation to deal with continous state spaces.

8.1 Continous state space Markov chains

The construction and properties of a Markov chain over an arbitrary sample space Ω are quite similar to the case when Ω is finite. First, we note that we must have a set of measurable sets on Ω, say $\mathcal{F}$. Consider the stochastic process $\ldots, X_3, X_4, X_5, \ldots$ such that each X_i is a random variable drawn from a probability space on $(\Omega, \mathcal{F})$. Let $\sigma(\ldots, X_{-1}, X_0, X_1, \ldots, X_i)$ be the σ-algebra generated by $\ldots, X_{i-1}, X_i$.

Definition 8.1 *Let $C \in \mathcal{F}$. The stochastic process $X = (\ldots, X_{-1}, X_0, X_1, \ldots)$ on $(\Omega, \mathcal{F})$ is a* Markov chain *if*

$$P(X_{i+1} \in C | \sigma(X_0, X_1, \ldots, X_i)) = P(X_{i+1} \in C | X_i).$$

Instead of a transition matrix, we now record probabilities of moving via a transition kernal K that behaves in a very similar way to its discrete counterpart.

Definition 8.2 *A kernal $K : \Omega \times \mathcal{F} \to [0, 1]$ is a* transition kernal *for a Markov chain if*

$$K(x, C) = P(X_{t+1} \in C | X_t = x)$$

for all x in Ω and $C \in \mathcal{F}$.

Definition 8.3 *Let p be a probability distribution on $(\Omega, \mathcal{F})$ and K a kernal. Then for $C \in \mathcal{F}$,*

$$pK(C) = \int_{x \in \Omega} K(x, C) p(dx).$$

Fact 8.1 *Suppose that the random variable X_t has distribution p. Then X_{t+1} has distribution pK.*

Definition 8.4 *Define $K^0 = I$, the kernal mapping probability distributions to themselves. Recursively define (for all $C \in \mathcal{F}$)*

$$K^{t+1}(x, C) = \int_{y \in \Omega} K(y, C) K^t(x, dy).$$

Fact 8.2 *Note that $K^1 = K$. Moreover, for all $C \in \mathcal{F}$*

$$P(X_{t+s} \in C | X_t = x) = K^s(x, C).$$

The notions of irreducibility and aperiodicity also extend in a natural way to the continuous world.

Definition 8.5 *A Markov chain is* irreducible *if there exists a measure ϕ on $\mathcal{F}$ such that for all C with $\phi(C) > 0$, and for all $x \in \Omega$, there exists a t such that*

$$K^t(x, C) > 0.$$

Definition 8.6 *Suppose that we have an irredicible Markov chain, and that Ω is partitioned into k sets $\mathcal{E} = \{E_0, \ldots, E_{k-1}\}$. If for all $i = 0, \ldots, m-1$ and all $x \in E_i$, $K(x, \Omega \setminus E_j) = 0$ for $j = i + 1 \mod k$, then $\mathcal{E}$ forms a* k-cycle *in the Markov chain. The* period *of a Markov chain, is the largest value of k for which a k-cycle exists. If $k = 1$, the Markov chain is said to be* aperiodic.

Definition 8.7 *A Markov chain is ergodic if it is both irreducible and aperiodic. It is geometrically ergodic if there exists a probability distribution π, constant $\omega > 1$ and a function $\eta(x)$ such that*

$$||K^t(x, \cdot) - \pi(\cdot)||_{TV} < \eta(x)\omega^{-t}.$$

The chain is uniformly ergodic if $\eta(v)$ is constant over $v \in \Omega$.

Our goal is to bound ω away from 1, and show that $\eta(x)$ is not exponential in the input size for our starting value x, thereby showing that convergence to the desired stationary distribution occurs in polynomial time.

As in the discrete case, the concept of reversibility will allow us to take exact samples from the stationary distribution of a chain.

Definition 8.8 *A kernal K satisfies the* detailed balance condition *or is* reversible *with respect to π if for all $A, B \in \mathcal{F}$*

$$\pi(A)K(A,B) = \pi(B)K(B,A).$$

Any distribution which is reversible with respect to K is stationary for K. If $\pi(dx)$ is a density function $s(x)$ and $K(x,dy)$ is a density $k(x,y)$, then the reversibility condition becomes $s(x)k(x,y) = s(y)k(y,x)$.

Finally, we note that the coupling theorem carries through to the continuous case.

Theorem 8.1 *Suppose that $X_0 = x$, Y_0 is distributed according to some stationary distribution π and the two stochastic processes are coupled. Then if T_C is the coupling time,*

$$||K^t(x,\cdot) - \pi(\cdot)||_{TV} \leq P(T_C > t).$$

8.2 Continuous time Markov chains

Another extension of Markov chains is to the index set of the stochastic process. Previously, our stochastic process $\{\ldots, X_{-6}, X_{-5}, X_{-4}, \ldots\}$ was indexed by the in-

tegers. For continous time Markov chains, the stochastic process will be indexed by an increasing sequence of real values $\ldots, k_{-1}, k_0 = 0, k_1, \ldots$.

Roughly speaking, continuous time Markov chains introduce "clocks" to changes. Suppose that we are at state x. Instead of a random change in the state occurring at every time step, a clock is attached to every random change. This clock is an exponential random variable with rate given by a rate kernel, $W(x, A)$. We wish to ensure that the rate at which clocks expire is not too high, so that with probability 1 only a finite number of events occur in a finite amount of time.

Definition 8.9 *We say that $W : \Omega \times \mathcal{F} \to [0, \infty)$ is a rate kernal if*

$$\int_{y \in A} W(x, dy) = W(x, A)$$

and

$$\int_{x \in \Omega} W(x, dx) < W(x, \Omega) < \infty.$$

Finally, we require that if $W(x, dy) > 0$, then $W(x, A) < \alpha W(y, A)$ for some constant α for all $A \in \mathcal{F}$. (This insures that no state is left instantaneously.)

These conditions guarantee that a finite amount of time elapses between changes, that is, $P(x_{t+s} = x_t) \geq e^{-W(x,\Omega)}$.

Now, because we are dealing with exponential random variables, we first gather some facts about their distribution. Suppose that $E(a)$ denotes an exponential random variable with rate a.

Fact 8.3 *Exponential random variables have the forgetfulness property, so that for $t, s > 0$,*

$$P(E(a) > t + s | E(a) > s) = P(E(a) > t) = e^{-at}.$$

Fact 8.4 *The rate of the minimum of two exponential random variables is the sum of their rates.*

$$\min\{E(a), E(b)\} \sim E(a+b).$$

Fact 8.5 *For two exponential random variables, $E(a)$ and $E(b)$,*

$$P(E(a) \leq E(b)) < \frac{a}{a+b}.$$

This first fact tells us something very important about continuous time Markov chains. If we are at state x, and s time passes without a clock expiration, then the distribution of each clock variable is exactly the same as it was before. Nothing happening over a time period does not give any information about what will be the next event.

The second fact allows us to give a full description of how the stochastic process $\ldots, X_{k_i}, X_{k_{i+1}}, \ldots$ may be formed. Suppose that $X_{k_i} = x$. Then $P(k_{i+1} - k_i > t) = \exp\{-W(x,\Omega)t\}$, and

$$P(X_{k_{i+1}} \in A) = \frac{W(x,A)}{W(x,\Omega)}.$$

For notational convenience, we set $X_t = X_{k_i}$, where $k_i \leq t < k_{i+1}$.

Definition 8.10 *The transition kernel K is defined as*

$$K^t(x,A) = P(X_t \in A | X_0 = x).$$

The third fact will be useful when trying to determine the probability that a particular event occurred given that some event occurred.

Intuitively, a distribution is stationary if the rate at which probability leaves a state is equal to the rate at which it is entering. More precisely,

Theorem 8.2 *Suppose that π is stationary for the Markov chain with rate kernel W. Then $\pi W = 0$.*

This important property allows us to "add" two continuous time Mark chains with a common stationary distribution.

Theorem 8.3 *Let W_1 and W_2 be two rate kernels for which π is stationary. Then $W_1 + W_2$ is the rate kernel for a new chain for which π is also stationary.*

In other words, if we add new moves to the chain satisfying $\pi W_{new} = 0$, then the stationary distribution of the chain will remain unchanged.

8.3 The Continuous Hard Core Gas Model

In the discrete hard core gas model, a configuration consisted of a set of vertices on a graph colored 1, and the rest colored 0. In the continuous case, we again have a set of points colored 1, but now the points come from a continuous state space, such as a subset of $\mathbf{R}^d$.

Let $x \subset S$, where $S \subset \mathbf{R}^d$ is a bounded Borel set. Then x is a *configuration* if the number of points in x is finite. We will write $x = \{x_1, \ldots, x_n\}$, and let $n(x)$ be the number of points in x. In the hard core gas model, each point has a "hard core" of radius $R/2$ around it. Let $\rho(a, b)$ be the distance between two points a and b. The fact that the core is hard means that two cores cannot intersect, or equivalently in $\mathbf{R}^d$, no two points of a configuration are allowed to be within a distance R of each other. We shall refer to a configuration satisfying this property as *valid*. The

probability distribution for the hard core gas model is

$$s(x) = \frac{\lambda^{n(x)}}{Z_\lambda},$$

if $\rho(x_i, x_j) > R$ for all $i \neq j$, and 0 otherwise. As in the discrete case, Z_λ is the normalization constant that makes s a probability distribution.

To obtain samples from this distribution, we will use continuous time Markov chains. The first chain we consider was proposed by Lotwick and Silverman [32] who showed that the chain does converge geometrically to the correct stationary distribution. In other words, for each starting state there exists constants C_1 and C_2 such that the total variation distance between the state of the chain after t steps and the stationary distribution is at most $C_1 \exp(-t/C_2)$. Of course, this result is not helpful in practice if the constant C_2 is exponentially large. While we do not present a full analysis of the convergence rate, for some ranges of λ we will show that this chain does mix in polynomial time.

The chain of Lotwick and Silverman is a spatial birth death chain. These chains have been extensively studied (see [43], [44], [36], [49]) and have been widely used for generating point processes on subsets of $\mathbf{R}^d$. The idea is simple. There are two types of events, births and deaths. Births add points to the configuration and deaths remove points. Let x be a configuration and z a point in space S. If the birth rate at point z given that we are currently in configuration x is $b(z, x)$, then the probability that a point in dz is added to x in time interval dt is $b(z, x)dxdz$. Similarly if z is a point in x, then $d(z, x)$ is the rate at which the point z dies. In time interval dt the probability that point z is removed from configuration x is

$d(z,x)dt$. Preston [40] showed the following version of reversibility for these birth death processes.

Theorem 8.4 *Let x be a configuration and z be a point in the state space. If*

$$b(z,x)f(x) = d(z,x)f(x \cup \{z\}),$$

then $f(x)$ is a stationary density for the birth death process.

To show that the Lotwick-Silverman chain has the correct stationary distribution is an easy application of birth death reversibility. Let $U(x)$ denote the volume that is within distance R of x, and suppose that we have scaled the problem so that the volume of S is 1. Then new points are added to the set x (born) in the area not within distance R of x. Points in x are removed from the set (die) at rate 1.

Note that if we remove the restriction that the cores must not intersect, the density just becomes $f(x) = \lambda^n(x)$ and we have a Poisson point process on S.

When actually simulating the chain, it is quite expensive to compute $1 - U(x)$, so instead an acceptance/rejection method is used. A point is chosen uniformly at random from M at rate λ. If it lies within distance R of x, it is not added, but if no cores intersect the core of the new point, it is added. This is a thinned Poisson process and so will have rate equal to the old rate times the probability of acceptance, or exactly $\lambda(1 - U(x))$ as desired.

Since this chain is a birth death chain, we need only verify that birth death reversibility is satisfied. The death rate for any point is constant independent of x, so $d(z,x) = 1$ for all z and x. The birth rate for z such that z is not within distance

Lotwick-Silverman Continuous Hard Core Chain

Set $X = X_t$
Choose Δt exponential with rate $\lambda + n(x)$
Choose U uniformly from $[0, 1]$
Case 1: $U < \lambda/(\lambda + n(x))$
  Choose v uniformly at random from S
  If $\rho(v, x) > R$
    Set $X = X \cup \{v\}$
Case 2: $U > \lambda/(\lambda + n(x))$
  Choose i uniformly from $\{1, \ldots, n\}$
  Set $X = X \setminus \{x_i\}$
  Set $t = t + \Delta t$
  Set $X_t = X$

R of any point in x is also a constant, λ. Hence

$$b(z, x)f(x) = \lambda \frac{\lambda^{n(x)}}{Z_\lambda} = d(z, x)f(x \cup \{z\})$$

and $\pi(dx) = f(x)dx$ is a stationary distribution. The set of points in a configuration may be though of as a queue. That π is the unique stationary distribution is a consequence of the coupling lemma and the fact that for any state, there is a small but positive probability that after 1 unit of time, the queue will be empty. Even if the queue does not empty in this time, it will not have grown too much larger with high probability.

8.4 Continuous bounding chains

In using CFTP, the goal is the same as in the discrete case: given an unknown state at time $-t$, determine whether or not the state becomes known at time 0. However,

two difficulties make use and analysis of CFTP more difficult. First, it is quite difficult to show that no matter which state we started in at time $-t$, we ended up in the same state at time 0. The number of possible states to consider is too high.

Therefore the brute force approach that worked quite well in the discrete case will not avail us here. Instead, we use the approach of Kendall [29] where we learn something about the unknown state at time $-t$ by looking farther back in time.

Definition 8.11 *Say that a point $z \in S$ has birth death interval $[b_z, d_z]$ if it was born at time b_z and dies at time d_z.*

Note that just because a point was born at time b_z does not guarantee that it was added to the set. However, whether or not it was added to the set, it is removed from the set at all times greater than d_z. Therefore, the only points which are even possibly part of the configuration at time $-t$ are those whose birth death interval $[b_z, d_z]$ contain the time $-t$. This approach allows us to initialize a bounding chain.

The idea for continuous state space bounding chains is straightforward. A configuration is a set of points. The bounding chain keeps track of two sets of points, points that are known to be in the set, x_A, and sets that are possibly in the set, x_D. At each step, we say that (x_A, x_D) bounds x if $x_A \subset x \subset x_A \cup x_D$. We shall refer to points in X_A as known, and points in X_D as unknown.

Our initialization procedure says that at time $-t$, the points x_D consists of all points whose birth death interval contains $-t$. We wish to take steps in the Markov chain such that if (x_A, x_D) bounds x at time t, it will also bound it at time $t' > t$. The procedure works as follows. Suppose a new point is born. Points which are

blocked by points in x_A are definitely not added to the set. Also, points which are not blocked by either x_A or x_D are definitely added to the set, and so are added to x_A. The only uncertainty comes when attempting to add points which are blocked by points in x_D. It is unknown whether these points are added or not, and so they are placed in x_D.

Suppose that we are given a list of birth and death times over a time interval $[a, b]$, and a list of z such that $b_z \leq a < d_z$. We first initialize the bounding chain setting X_D to be this set of z whose lifetime falls across a. We then proceed in sorted time order, examining and dealing with births and deaths as they arise. Again, this may be done in linear time, and so utilizing the bounding chain does not increase the order of complexity of running the Markov simulation. Note that generation of the initial X_D is not difficult. If we have no prior knowledge about the time before a, then we simply use a Poisson point process with parameter λ, generating lifetime lengths for these points and then (using the forgetfulness property of exponentials) generate death times for each of these points. If we already know something about birth and death times before a, we simply condition on that information when creating the Poisson point process.

We may use the bounding chain exactly as we did in the discrete case for either determination of mixing time, or as a black box for a CFTP perfect sampling algorithm. As with the discrete algorithms, if we wish to experimentally determine mixing time, we simply check whether X_D is empty at time b. If it is, the state $x = \emptyset$ will have coupled with the stationary path. Nothing in our presentation of the bounding chain prevents us from modifying b on the fly, and continuing until

Bounding Chain for Lotwick-Silverman

Input: List of events, interval $[a, b]$, $\{z : b_z \leq a < d_z\}$
Output: $X_b = (X_A, X_D)$

Set $X_D = \{z : b_z \leq a < d_z\}$
For each event e in sorted time order
  If $e = d_z$ for some z
    If $z \in X_D$
      Set $X_D \leftarrow X_D \setminus \{z\}$
    If $z \in X_A$
      Set $X_A \leftarrow X_A \setminus \{z\}$
  If $e = b_z$ for some z
  If $\rho(z, X_A \cup X_D) > R$
    Set $X_A = X_A \cup \{z\}$
  If $\rho(z, X_A) > R$ and $\rho(z, X_D) \leq R$
    Set $X_D = X_D \cup \{z\}$

Figure 8.1: Bounding Chain for Lotwick-Silverman

X_D is empty.

For coupling from the past, we need to be a little more careful how we generate the list of birth death times and lifetime crossing times. We generate birth death times in reverse, that is, starting at time 0, we move backwards in time and have deaths appearing at rate λ. For each death, we then have a corresponding birth occur at a time earlier, such that the difference is exponential with rate 1. Because we generate our data death first, then birth, it follows that for any time $-t$, we immediately know which birth death intervals cross $-t$ and have death times in $[-t, 0]$. It remains to consider death times which cross $-t$ and have death times in $(0, \infty)$. To limit the possibilities that we must consider, note that any lifetime which starts before $-t$ and ends after 0 must cross time 0 as well. The set of points

whose lifetime crosses 0 is just a Poisson point process with parameter λ, and so we generate this list of points. Anytime we need to see how many cross $[-t, 0]$, we just examine this list and see how many that died after 0 were also born before $-t$.

In the discrete case, we found that when $\lambda \leq 2/(\Delta - 2)$, we could show that the bounding chain converged in polynomial time. For the continuous case, we now show the following. Our result is stated in terms of the number of events needed to occur, since it is this value that represents the amount of work actually needed to be performed in running the algorithm.

Theorem 8.5 *Consider the hard core gas model with parameters λ and R, and suppose that we run the bounding chain for the Lotwick-Silverman chain forward from time 0 for k events. Then if $\lambda \leq 1/V_R$, after $O(\lambda^3 \ln(1/\epsilon))$ steps the probability that we have converged is*

$$P(X_D \neq \emptyset) \leq \epsilon.$$

As in Chapter 4 where we first introduced bounding chains, we will require the use of supermartingales in our analysis.

8.4.1 More on supermartingales

Recall that a supermaringale is a stochastic process such that with probability one,

$$E[X_{t+1} | \sigma\{\ldots, X_t\}] \leq X_t.$$

In expectation, a supermartingale decreases as time goes on. Now suppose that the supermartingale never grows too fast, so that

$$X_{t+1} - X_t \leq c$$

for some constant c. Then Azuma's inequality [4] limits the probability that the supermartingale grows too large.

Theorem 8.6 *Let $X_0, X_1, \ldots$ be a supermartingale satisfying $X_{t+1} - X_t \leq c$. Then*

$$P[X_t - X_0 \geq \alpha] \leq e^{\frac{\alpha^2}{2tc^2}}.$$

That is, the probability that X_t rises too far above X_0 is exponentially small in the time t.

Our method of proving Theorem 8.5 will be to show that the number of unknown nodes stays above 0 with an exponentially declining probability. The sequence $n(X_D)$ has particular properties, such as 0 being an absorbing state. The following lemma shows how these properties force the process to hit 0 quickly.

Lemma 8.1 *Suppose that X is a nonnegative stochastic process satisfying: 0 is an absorbing state, $X_{t+1} - X_t \leq c$, and $E[X_{t+1} - X_t | X_t] \leq q$ where q is a constant between 1 and 0. Finally, if X_0 is distributed as a Poisson process with parameter λ, $\epsilon > 0$, and*

$$t \geq \frac{2(c+q)^2[\ln(1/\epsilon) + (e^{2q} - 1)\lambda]}{q^2},$$

then

$$P(X_t > 0) \leq \epsilon.$$

Proof: Let τ_0 be the first time that the process X_t hits 0. Note that the stochastic process

$$N_t = \begin{cases} X_t + tq & X_t > 0 \\ \tau_0 q & X_t = 0 \end{cases}$$

is a supermartingale. The fact that $E[X_{t+1} - X_t | X_t] \leq q$ insures that $E[N_{t+1} - N_t | N_t] \leq 0$ when $X_t > 0$. Furthermore, N_t is constant for all $t \geq \tau_0$, so it is certainly a supermartingale. We also have that $N_{t+1} - N_t \leq c + q$ so we may apply Azuma's inequality. Note $N_0 = X_0$ so $P(N_0 = i) = e^{-\lambda}\lambda^i/i!$. Altogether, we have that

$$\begin{aligned}
P(X_t > 0) &= P(N_t > tq) \\
&= \sum_{i=0}^{\infty} P(N_t > tq | N_0 = i) P(N_0 = i) \\
&= \sum_{i=0}^{\infty} P(N_t - N_0 > tq - i | N_0 = i) P(N_0 = i) \\
&\leq \sum_{i=0}^{\infty} e^{-(tq-i)^2/(2t(c+q)^2)} e^{-\lambda}\lambda^i/i! \\
&= \sum_{i=0}^{\infty} e^{(-tq^2+2qi-i^2/t)/(2(c+q)^2)} e^{-\lambda}\lambda^i/i! \\
&\leq \sum_{i=0}^{\infty} e^{-tq^2/(2(c+q)^2)+(e^{2q}-1)\lambda} e^{-\lambda e^{2q}} (e^{2q}\lambda)^i/i! \\
&= e^{-tq^2/(2(c+q)^2)+(e^{2q}-1)\lambda}.
\end{aligned}$$

For this last value to fall below ϵ, we must have that

$$\begin{aligned}
\frac{-tq^2}{2(c+q)^2} + (e^{2q} - 1)\lambda &\leq \ln \epsilon \\
\frac{2(c+q)^2[\ln(1/\epsilon) + (e^{2q} - 1)\lambda]}{q^2} &\leq t
\end{aligned}$$

which completes the proof. □

Proof of Theorem 8.5: We utilize our (by now) standard procedure of looking at how the size of X_D changes with certain events, in this case the set of births and the

deaths of points in X_D. Let X_D^i be the set X_D after i such events have occurred, and let $\mathcal{F}_\rangle$ denote the σ-algebra generated by all the configurations up to the time of the ith event. We wish to compute a bound on $E[n(X_D^{i+1}) \mid \mathcal{F}_\rangle]$. The rate of births is λ, and the rate of deaths is $n(X_D)$. From Fact 8.5, we know that the probability that one of our events is a birth is $\lambda/(\lambda + n(X_D)$, and the probability that an unknown point dies is $n(X_D)/(\lambda + n(X_D))$. When a point in X_D dies, $n(X_D)$ goes down by 1. When a point is born, it only joins X_D if it tries to be born in an area blocked by a point in X_D. This occurs with probability at most $n(X_D)V_R$. (This is an upper bound since some of the unknown points' blocked areas might overlap.) Therefore

$$\begin{aligned} E[n(X_D^{i+1} \mid \mathcal{F}_\rangle] &\leq (-1)\frac{n(X_D^i)}{\lambda + n(X_D^i)} + \frac{\lambda}{\lambda + n(X_D^i)}[V_D n(X_D^i)] \\ &= \frac{n(X_D^i)}{\lambda + n(X_D^i)}[V_D\lambda - 1]. \end{aligned}$$

Therefore when $n(X_D^i) \geq 1$,

$$E[n(X_D^{i+1}) - n(X_D^i)|\mathcal{F}_\rangle] \leq \frac{V_D\lambda - 1}{\lambda + 1}.$$

We are given that $1 \leq V_D\lambda$, this means that $n(X_D^i)$ satisfies the conditions of Lemma 8.1 with $q = \frac{1-V_D\lambda}{\lambda+1}$. Note that $q \leq 1$, which means that

$$P(n(X_D^i) > 0) \leq e^{7\lambda} e^{-\frac{t}{18}\left(\frac{1-V_D\lambda}{\lambda+1}\right)^2}.$$

Finally, we note that the number of events where a node in X_A died is bounded above by the number of births. Therefore after k events, at least $k/2$ events had to have been either births or deaths of nodes in X_D, which completes the proof.□

8.4.2 The Swapping Continuous Hard Core chain

Greenhill and Dyer [12] increased the ability to analyze the rate of convergence of the discrete hard core chain by introducing a Broder type swapping move. We now show that introducing the same move into the continuous hard core chain yields a similar increase in our analytic ability. The move is as follows. If a point attempts to be born and is blocked by exactly one point, then the blocking point might be removed and the new point added. This swap is executed with probability 1/4. Algorithmically, this new chain may be written as in Figure 8.2.

Swapping continuous hard core chain

Set $x \leftarrow X_t$
Choose δt exponential with rate $\lambda + n(x)$
Choose U uniformly from $[0, 1]$
Case 1: $U < \lambda/(\lambda + n(x))$
  Choose v uniformly at random from S
  If $\rho(v, x) > R$
    Set $X \leftarrow X \cup \{v\}$
  If $\exists z \in x$ such that $\rho(v, z) \leq R$ and $\rho(v, x \setminus \{z\}) > R$
    With probability 1/4 set $x \leftarrow x \setminus \{z\} \cup \{v\}$
Case 2: $U > \lambda/(\lambda + n(x))$
  Choose i uniformly from $\{1, \ldots, n\}$
  Set $x \leftarrow x \setminus \{x_i\}$
  Set $t \leftarrow t + \delta t$
  Set $X_t \leftarrow x$

Figure 8.2: Swapping continuous hard core chain

This swap move only moves between states with the same stationary probability, and $W_{swap}(x, dy) = W_{swap}(y, dx)$ no matter what the probability that we execute the swap (when the value of 1/4 was chosen to make the analysis as tight as possible).

Hence π is also stationary for this rate function, so by Theorem 8.3 this new chain has the same stationary distribution as the old one. The new moves do not increase the number of points in the configuration, therefore the same argument that showed geometric ergodicity for the old chain applies here as well.

The bounding chain for this swap chain is analogous to that introduced for the Dyer Greenhill chain. Deaths occur exactly as before. If a point in x_D dies, it is removed from x_D. If a point in x_A dies, it is removed from x_A.

Suppose that a new point is born. If it is not blocked by any point, either in x_A, or x_D, then it is added to x_A. If the point is blocked by one point in x_D, and we choose not to swap, then the point is not added. If the point is blocked by two or more points in x_A, the point is not added. If the point is blocked by two or more points in x_D, and no points in x_A, we are unsure whether or not to add the point to the configuration, and so the point is added to x_D. All of these moves are the same as for the bounding chain for the Lotwick-Silverman chain.

The interesting cases come when the point added is a candidate for a swap, and the algorithm chooses to execute the swap. Suppose the point is blocked by exactly one point in $x_A \cup x_D$. Then the blocking point is removed, and the new point is added to x_A. If the point is blocked by exactly one point in x_A and at least one point in x_D, then we are unsure about two points, the new point might be added and the blocking point in x_A might be removed. Therefore both of these points must be placed in x_D.

Theorem 8.7 *Consider the hard core gas model with parameters λ and R, and suppose that we run the bounding chain for the Lotwick-Silverman chain forward*

from time 0 for k events. Then if $\lambda < 2 - \lambda V_R$, then after $O(\lambda^3 \ln(1/\epsilon))$ steps the probability that we have converged is

$$P(X_D \neq \emptyset) \leq \epsilon.$$

Swapping continuous hard core bounding chain

Input: List of events, interval $[a, b]$, $\{z : b_z \leq a < d_z\}$,
the swap execute rolls for each birth
Output: $X_b \leftarrow (X_A, X_D)$

Set $X_D \leftarrow \{z : b_z \leq a < d_z\}$
For each event e in sorted time order
If $e = d_z$ for some z
If $z \in X_D$
Set $X_D \leftarrow X_D \setminus \{z\}$
If $z \in X_A$
Set $X_A \leftarrow X_A \setminus \{z\}$
If $e = b_z$ for some z
Let D be the number of points y in X_D such that $\rho(z, y) \leq R$
Let A be the number of points y in X_A such that $\rho(z, y) \leq R$
If $D = A = 0$
Set $X_A \leftarrow X_A \cup \{z\}$
If $D \geq 1, A = 0$ and we do not execute a swap at this birth
Set $X_D \leftarrow X_D \cup \{z\}$
If $D = 1, A = 0$ and we execute the swap
Set $X_D \leftarrow X_D \setminus \{y \in X_D : \rho(y, z) \leq R\}$
Set $X_A \leftarrow X_A \cup \{z\}$
If $D \geq 2, A \leq 1$ and we execute the swap
Set $X_D \leftarrow X_D \cup \{z\} \cup \{y \in X_A : \rho(y, z) \leq R\}$
Set $X_A \leftarrow X_A \setminus \{y \in X_A : \rho(y, z) \leq R\}$

Figure 8.3: Swapping continuous hard core bounding chain

The computer time needed to generate a simulation is the number of events which occur, which is why we phrase this theorem using events rather than time.

Proof: The proof is very similar to the case of the nonswapping chain, and so we will utilize the same notation as in that proof. We again will only consider those events capable of changing $n(X_D^i)$, namely, all births, and deaths of nodes in X_D^i. These events occur at rate $r = \lambda + n(X_D^i)$.

The death rate contribution to $E[n(X_D^{i+1}) \mid \mathcal{F}_i]$ is the same as before; what changes is the contribution of the birth rate. Before, births could only increase $n(X_D^{i+1})$, but with the introduction of the swap move, it can also decrease the number of unknowns.. Suppose that a point is blocked by a single point in X_D^i. Then a birth and swap at that point removes a point from X_D^i, and so $n(X_D^i)$ decreases by 1. If there is a birth and no swap, then the new point joins X_D^i and $n(X_D^i)$ increases by 1.

If, however, a new point is blocked by a point in X_D^i and X_A^i, then a swap results in two points being added to X_D. If the point chooses not to swap, then it cannot be born since it is blocked by X_A^i.

Let A_1 denote the area blocked by a single point in X_D^i and no points of X_A^i. Let A_2 denote the area blocked by more than one point in X_D^i and no points of X_A^i, and let A_3 denote the area blocked by at least one point in X_D^i and exactly one point in X_A^i.

$$\begin{aligned} E[n(X_D^{i+1}) - n(X_D^i) \mid \mathcal{F}_\rangle] \;\leq\; & (-1)\frac{n(X_D^i)}{r} + (-1)\left(\frac{1}{4}\right)\left(\frac{\lambda A_1}{r}\right) \\ & + \left(\frac{3}{4}\right)\left(\frac{\lambda A_1}{r}\right) + \frac{\lambda A_2}{r} \\ & + (2)\left(\frac{1}{4}\right)\left(\frac{\lambda A_3}{r}\right) \end{aligned}$$

$$= \frac{-n(X_D^i) + \frac{1}{2}\lambda[A_1 + 2A_2 + A_3]}{r}$$

Note that each point in A_1 and A_3 is blocked by at least point within distance R of an unknown point, and each point in A_2 is blocked by at least two unknown points. Hence $A_1 + 2A_2 + A_3 \leq n(X_D^i)V_R$. Therefore

$$E[n(X_D^{i+1}) - n(X_D^i) \mid \mathcal{F}_\backslash] \leq \frac{n(X_D^i)[-1 + n(X_D^i)V_R/2]}{\lambda + n(X_D^i)}$$

and we may apply Lemma 8.1. As in the nonswapping chain, at least half of the events must be births or deaths of points in X_D, completing the proof.□

We have proven that this chain converges rapidly for values of λ which are twice as high as for the nonswapping chain. This swap move may be added to other chains as well to increase their performance.

8.5 Widom-Rowlinson

The Widom Rowlinson model was originally conceived as a continuous model [49], where the density of a configuration of points $x = (x_1, x_2, \ldots, x_Q)$ is

$$\frac{\lambda_1^{n(x_1)}\lambda_2^{n(x_2)}\ldots\lambda_Q^{n(x_Q)}}{Z}.$$

This is exactly analogous to the discrete case, and each of the chains we examined there has a corresponding continuous chain.

The birth death chain given for the discrete model actually comes from a discretization of the birth death chain for the continuous model. We say that a point

v and color i is blocked if $\rho(v, x_j) < R$ for some j not equal to i.

Birth death continuous Widom-Rowlinson chain
Set $x \leftarrow X_t$
Choose δt exponential with rate $\sum_i \lambda_i + n(x)$
Choose $U \in_U [0, 1]$
Set $p_0 \leftarrow 1/(1 + \sum_i \lambda_i)$
For all $0 < i < n$
  Set $p_i \leftarrow \lambda_i/(1 + \sum_i \lambda_i)$
Choose $i \in_R \{0, \ldots, Q\}$ according to p
If $i = 0$
  Choose $v \in_U x_1 \cup \cdots \cup x_Q$
  Let j be the color such that $v \in x_j$
  Set $x_j \leftarrow x_j \setminus \{j\}$
Else
  If color i is not blocked for node v
    Set $x(v) = i$
Set $X_{t+1} \leftarrow x$

Figure 8.4: Birth death continuous Widom-Rowlinson chain

As pointed out in [16], this chain exhibits a monotonic structure when $Q = 2$. For $Q > 2$, we must use a bounding chain. As with the hard core case, a state y of the bounding chain will be $(x_1, x_2, \ldots, x_Q, z_1, z_2, \ldots, z_Q)$, where each x_i contains points which are known to be colored i, while z_i refers to those points that are might or might not be in the configuration. If they are in the configuration, though, they are definitely colored i. Let $n(x) = |x_1 \cup \cdots \cup x_Q|$ and $n(z) = |z_1 \cup \cdots \cup x_Q|$. Consider a point v and color i. If $\rho(v, x_j) < R$ for some $j \neq i$, then we say that v is blocked for i by a known node. If $\rho(v, z_j) < R$ for some $j \neq i$, then we say that v is blocked for i by an unknown node.

Theorem 8.8 *Consider the Widom-Rowlinson model with parameters λ_i, i running*

Birth death continuous Widom-Rowlinson bounding chain

Set $y \leftarrow Y_t$
Choose Δt exponential with rate $\sum_i \lambda_i + n(x) + n(z)$
Choose U uniformly from $[0, 1]$
Set $p_0 \leftarrow 1/(1 + \sum_i \lambda_i)$
For all $0 < i < n$
  Set $p_i \leftarrow \lambda_i/(1 + \sum_i \lambda_i)$
Choose $i \in_R \{0, \ldots, Q\}$ according to p
Case 1: $i = 0$
  Choose $v \in_U x_1 \cup \cdots \cup x_Q$
  Let j be the color such that $v \in x_j$
  Set $x_j \leftarrow x_j \setminus \{j\}$
  Set $z_j \leftarrow z_j \setminus \{j\}$
Case 2: $i > 0$, v not blocked for i
  Set $x_i \leftarrow x_i \cup \{i\}$
Case 3: $i > 0$, v blocked for i only by unknown nodes
  Set $z_i \leftarrow z_i \cup \{i\}$
Set $t \leftarrow t + \delta t$
Set $Y_t \leftarrow y = (x_1, \ldots, x_Q, z_1, \ldots, z_Q)$

Figure 8.5: Birth death continuous Widom-Rowlinson chain

from 1 *to* Q *and* R*, and suppose that we run the bounding chain for this simple birth death chain forward from time 0 for* k *events. For* $Q = 2$ *let*

$$\alpha = \frac{1}{2} + \frac{1}{2}\sqrt{\lambda_1 \lambda_2},$$

and let

$$\alpha = \sum_i \lambda_i - \min_i \lambda_i$$

for $Q > 2$*. If* $\alpha < \lambda V_R$*, then after* $O(\frac{1}{\lambda V_R - \alpha} \ln(1/\epsilon))$ *events the probability that we have converged is*

$$P(n(z) = 0) \leq \epsilon.$$

Proof: We consider only those events capable of changing the size of $n(z)$, namely, births of new points and deaths of points in z. Let z_i denote the unknown set of points of color i at the beginning of a bounding chain step, and z_i' the set afterwards. Let $\mathcal{F}$ denote the sigma algebra generated by events up to the time of z.

A bad event occurs when a point of color i is blocked by a point in z_j, where $j \neq i$. Given that we have a birth, a point in z_j blocks a volume V_R, and so the total amount of volume blocked could be as high as $V_R[n(z) - |z_i|]$.

$$\begin{aligned} E[z_i' - z_i|\mathcal{F}] &= P(\text{birth of color } i)P(\text{birth blocked}) - P(\text{death of color } i) \\ &= \frac{\lambda_i}{n(z) + \sum_i \lambda_i} V_R[n(z) - |z_i|] - \frac{|z_i|}{n(z) + \sum_i \lambda_i} \\ &= \frac{\lambda_i}{n(z) + \sum_i \lambda_i}. \end{aligned}$$

And so just as for the discrete chain we have that $E[z'] = E[E[z'|z]] \leq E[AE[z]] = AE[z]$, since the elements of A are all nonnegative. Let z^t denote the unknown set of points at time t. An induction yields $E[z^t] \leq A^t E[z^0]$, so that $||E[z^t]|| \leq \alpha^t ||E[z^0]||$, where α is the highest magnitude eigenvalue of A.

As in the discrete case, the only remaining problem is how to bound α. When $Q = 2$, we preform the computation directly, and $\alpha = V_R[1/2 + 1/2\sqrt{\lambda_1 \lambda_2}]$. For $Q > 2$, the method of Gershgorin disks may be used to show that

$$\alpha \leq 1 - \frac{1}{Q} + \frac{1}{Q} V_R \max_i \left\{ \sum_{j \neq i} \lambda_i \right\} = 1 - \frac{1}{Q} \left[1 - V_R \left[\sum_i \lambda i \right] - \min_i \lambda_i \right].$$

To complete the proof, we recall that $(1 - \delta/Q)^{Q/\delta} \leq 1/e$. $\square$

8.5.1 Continuous swapping chain

The swap move is not difficult to describe in the continuous case, and it will double the range of λ_i over which we can prove that the bounding chain detects complete coupling in polynomial time. When a point of color i is blocked by any number of points of color j different from i in the birth death chain, that point is not born. In a swap move, if a point of color i is blocked by exactly one point of color $j \neq i$, then with probability p_{swap} a swap move is executed and the point is born, and the blocking point is removed from the set.

Swapping birth death continuous Widom-Rowlinson chain

Set $x \leftarrow X_t$
Choose δt exponential with rate $\sum_i \lambda_i + n(x)$
Choose $U \in_U [0,1]$
Set $p_0 \leftarrow 1/(1 + \sum_i \lambda_i)$
For all $0 < i < n$
  Set $p_i \leftarrow \lambda_i/(1 + \sum_i \lambda_i)$
Choose $i \in_R \{0, \ldots, Q\}$ according to p
Case 1: $i = 0$
  Choose $v \in_U x_1 \cup \cdots x_Q$
  Let j be the color such that $v \in x_j$
  Set $x_j \leftarrow x_j \setminus \{j\}$
Case 2: $i > 0$ is not blocked for node v
  Set $x_i \leftarrow x_i \cup \{v\}$
Case 3: $i > 0$ is blocked for v by exactly one point $w \in x_j$
  If $U \leq p_{swap}$
    Set $x_j \leftarrow x_j \setminus \{w\}$
    Set $x_i \leftarrow x_i \cup \{v\}$
Set $X_{t+1} \leftarrow x$

Figure 8.6: Swapping birth death continuous Widom-Rowlinson chain

The bounding chain must now take into account the possibility of a swap type

move. We have the same cases that arise in the discrete case.

As in all cases dealing with the swapping move, changing the value of p_{swap} affects how quickly it actually converges. When $p_{swap} = 1/4$, good things happen. The only difference between the theorem below and the theorem for the nonswapping chain is that here α may be larger by a factor of 2.

Theorem 8.9 *Consider the Widom-Rowlinson model with parameters λ_i, i running from 1 to Q and R, and suppose that we run the bounding chain for this simple birth death chain forward from time 0 for k events. For $Q > 2$, let*

$$\alpha = \frac{1}{2}\sum_i \lambda_i - \min_i \lambda_i.$$

If $p_{swap} = 1/4$ and $\alpha < \lambda V_R$, then after $O(\frac{1}{\lambda V_R - \alpha}\ln(1/\epsilon))$ events the probability that we have converged is

$$P(n(z) = 0) \leq \epsilon.$$

For $Q = 2$, either let $p_{swap} = 1/4$ and

$$\alpha = \frac{1}{2}\max_i \lambda_i,$$

as above, or let $p_{swap} = 1/3$ and

$$\alpha = \frac{2}{3}\left(\frac{1}{2} + \frac{1}{2}\sqrt{\lambda_1\lambda_2}\right).$$

Either way, if $\alpha < \lambda V_R$, the same probability of convergence holds.

Proof: Let $|D_t|$ be the set of unknown points at time t. Then these points cover area in three ways. (These are similar to the three classifications of nodes seen in

the discrete case.) To change the value of $|D_t|$, either an unknown point died, or a point being born resulted in one or two nodes being switched to unknown. The rate at which points are born or die from the unknown set is $r = |D_t| + \sum_i \lambda_i$.

Let D^1 denote the area covered by exactly one unknown point of some color. When a point is born in D^1, we are in case 4 in the swapping bounding chain. Let D_i^2 denote the area covered by exactly one known point of color i and at least one unknown point of color i, this corresponds to case 5. Finally, let D_i^3 denote the area covered by exactly at least two unknown points of color i and no known points, so that when a point is born here we are in case 3.

Suppose that $Q > 2$, so that $\alpha = \sum_i \lambda_i - \min_i \lambda_i$. Case 4 can have two outcomes. If we choose to swap, then D_{t+1} is smaller than D_t by one node, but if we do not choose to swap it grows by one. For a point $v \in D^1$, let b_v be the color of the point i such that v is being blocked by a point in z_i. The point is that births of color b_v are not blocked and do not affect $|D_t|$ if the birthing point has color i, but might change if the new point has color $j \neq i$. This event occurs with probability λ'_{b_v}/r. Given that we have a birth event but that we have not yet selected a position, the expected change due to points in D^1 will be

$$\begin{aligned}
\int_{v \in D^1} (+1)(1 - p_{swap})\frac{\lambda'_{b_v}}{r} - p_{swap}\frac{\lambda'_{b_v}}{r} dv &= \int_{v \in D^1} (1 - 2p_{swap})\frac{\lambda'_{b_v}}{r} dv \\
&\leq \int_{v \in D^1} (1 - 2p_{swap})\frac{\alpha}{r} dv \\
&= D^1(1 - 2p_{swap})\frac{\alpha}{r}
\end{aligned}$$

where the inquality is valid as long as $p_{swap} \leq 1/2$. We will later set $p_{swap} = 1/4$, so this is true.

For case 5, choosing to swap makes $|D_{t+1}| - |D_t| = 2$, but is unchanged if we do not swap. The contribution from this case is

$$\int_{v \in D^2} 2p_{swap} \frac{\lambda'_{b_v}}{r} \leq D^2(2p_{swap})\frac{\alpha}{r}.$$

In case 3, the new node is always part of D_{t+1} so $|D_{t+1}| - |D_t| = 1$. Therefore the contribution is independent of p_{swap},

$$\int_{v \in D^3} \frac{\lambda'_{b_v}}{r} \leq D^3\frac{\alpha}{r}.$$

The final event that may occur to affect $|D_t|$ is the death of an unknown node, which occurs with probability $|D_t|/r$ and results in the removal of exactly one unknown node. We now set $p_{swap} = 1/4$, so that

$$\begin{aligned} E[|D_{t+1}| - |D_t| \mid \mathcal{F}_t] &\leq D^1(1 - 2p_{swap})\frac{\alpha}{r} + D^2(2p_{swap})\frac{\alpha}{r} + D^3\frac{\alpha}{r} - \frac{|D_t|}{r} \\ &\leq \frac{\alpha}{2r}\left[D^1 + D^2 + 2D^3\right] - \frac{|D_t|}{r}. \end{aligned}$$

Points in D^1 and D^2 are covered by at least one node of D_t, and points in D_2 are covered by at least two nodes of D_t, so $D^1 + D^2 + 2D^3 \leq V_R|D_t|$. (The existence of this bound is why we choose p_{swap} as we did.

Following our usual derivation, $E[|D_t|] \leq \beta^t E[|D_0|]$ where

$$\beta = 1 - \frac{\alpha V_R/2 - 1}{r}.$$

We have that $\beta < 1$ when $2\alpha V_R < 1$ and since $|D_t|$ is integral and $|D_0|$ is poisson with rate $\sum_i \lambda_i$, Markov's inequality finishes the proof.

For $Q = 2$, $\lambda = \max_i \lambda_i$, so we may use that for α. However, we may also go into more detail, splitting apart D^1 into D^1_1 and D^1_2. Let D^1_1 be the area blocked

for 2 by exactly one unknown point of color 1 and D_2^1 be the area blocked for 1 by exactly one unknown point of color 2 (these areas may overlap). Split D^2 and D^3 in a similar fashion.

As in the regular birth death chain, let z_i denote the number of unknowns of color $i \in \{1, 2\}$, and z_i' denote the number of unknown points after the step is taken. Then from our discussion above, we know that

$$\begin{aligned} E[z_1' - z_1 \mid \mathcal{F}] &= \frac{\lambda_1}{r}\left[D_2^1(1 - p_{swap}) + 2D_2^2(p_{swap})\right] - \frac{\lambda_2}{r}D_1^1 p_{swap} - \frac{|D_t|}{r} \\ &\leq \frac{\lambda_1}{r}\frac{2}{3}\left[D_2^1 + D_2^2\right] \\ &\leq \frac{\lambda_1}{r}\frac{2}{3}z_1 V_R \end{aligned}$$

using the fact that $D_2^1 + D_2^2 \leq z_1 V_R$. An analogous equation may be derived for the expected change in z_2.

This is the same recurrence that we have seen for the $Q = 2$ case in the regular birth death chain (for both the continuous and discrete case.) The eigenvalues of the system are smaller than in the previous case by a factor of $2/3$, giving the result. □

The moral of this chapter is: everything that works in the discrete world carries over to the continuous realm with no problems.

Swapping birth death continuous Widom-Rowlinson bounding chain

Set $y \leftarrow Y_t$
Choose Δt exponential with rate $\sum_i \lambda_i + n(x) + n(z)$
Choose U uniformly from $[0, 1]$
Set $p_0 \leftarrow 1/(1 + \sum_i \lambda_i)$
For all $0 < i < n$
  Set $p_i \leftarrow \lambda_i/(1 + \sum_i \lambda_i)$
Choose $i \in_R \{0, \ldots, Q\}$ according to p
Case 1: $i = 0$
  Choose $v \in_U x_1 \cup \cdots x_Q$
  Let j be the color such that $v \in x_j$
  Set $x_j \leftarrow x_j \setminus \{j\}$
  Set $z_j \leftarrow z_j \setminus \{j\}$
Case 2: $i > 0$, v not blocked for i
  Set $x_i \leftarrow x_i \cup \{i\}$
Case 3: $i > 0$, v blocked for i by at least two unknown nodes
  Set $z_i \leftarrow z_i \cup \{i\}$
Case 4: $i > 0$, v blocked for i by exactly one unknown node $w \in z_j$
  If $U < p_{swap}$
    Set $z_j \leftarrow z_j \setminus \{w\}$
    Set $x_i \leftarrow x_i \cup \{v\}$
  Else **Set** $z_i \leftarrow z_i \cup \{v\}$
Case 5: $i > 0$, v blocked for i by exactly one known node $w \in x_j$,
    and at least one unknown nodes
  If $U < p_{swap}$
    Set $x_j \leftarrow x_j \setminus \{w\}$
    Set $z_j \leftarrow z_j \cup \{w\}$
    Set $z_i \leftarrow z_i \cup \{v\}$
Set $t \leftarrow t + \delta t$
Set $Y_t \leftarrow y = (x_1, \ldots, x_Q, z_1, \ldots, z_Q)$

Figure 8.7: Swapping birth death continuous Widom-Rowlinson chain

Chapter 9

Final thoughts

> "I don't see much sense in that." said Rabbit. "No," said Pooh humbly, "there isn't. But there was going to be when I began it. It's just that something happened to it along the way."
>
> *-A.A. Milne*

The theoretical bounds on the mixing time derived using bounding chains are always weaker than what may be found experimentally. For example, in chapter 5 we saw a bounding chain for the list update problem that is guaranteed to run quickly when the probabilities of selecting items are geometric, i.e., $p_i/p_{i+1} \leq \theta$ for some $\theta \leq 1/5$. When $\theta > 1/5$, experiments must be run to see how quickly the bounding chain converges.

As shown in the graph below, this bounding chain appears to be polynomial when $\theta = 0.4$ or lower, but it also clearly exponential when $\theta = 0.5$. These running times are an average of 1000 runs of the bounding chain.

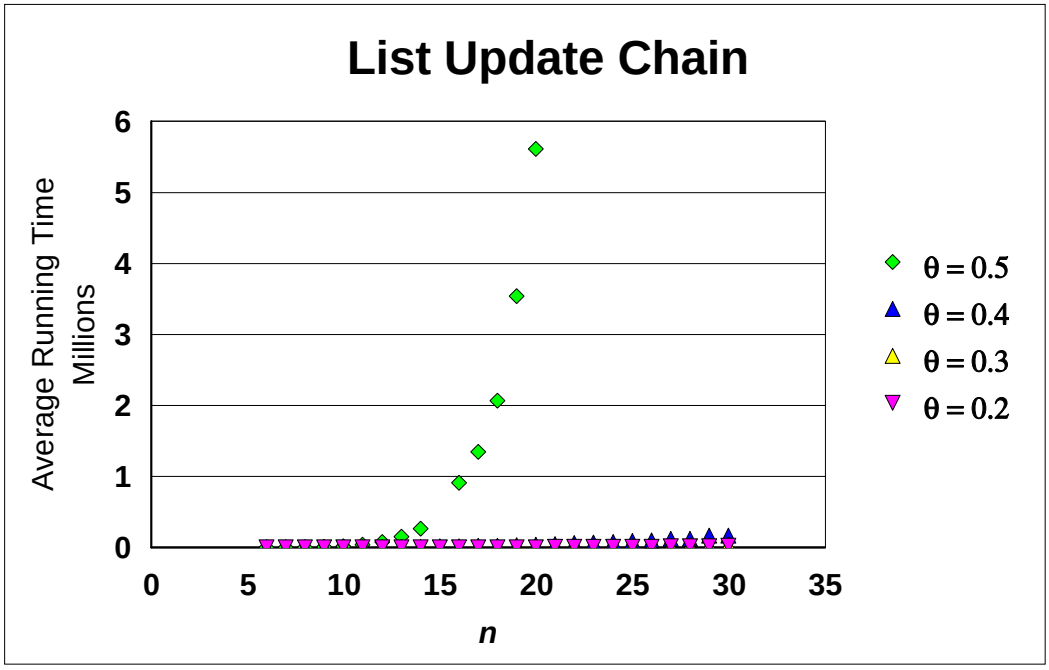

Figure 9.1: T_{BC} for list update chain

Given that practice always wins over theory, there are of course several reasons why we are still interested in theoretical bounds on the time bounding chains need to detect complete coupling. First, the theoretical range of parameters over which the chain is rapidly mixing is a solid indication of the actual usefulness of the chain. As we have seen, the simple local analysis of bounding chains tends to give results that are within a constant factor of the true answer. This is a trivial statement for bounding chains such as the list update chain for geometric weights, where the parameter θ seems independent of n. However, for chains such as the antiferromagnetic Potts model where the range of λ where the chain is rapidly mixing depends quite strongly on Δ, this is a somewhat surprising fact.

Second, as an immediate corollary of the bounding chain mixing time we get an immediate bound on the mixing time of the chain, and a priori bounds on the running time of CFTP and FMR for perfect sampling. It is always nice to know that certain problems, such as randomly sampling from the sink free orientations of a graph, have polynomial (expected) running time algorithms.

Coupling from the past is a simple idea with extraordinary consequences for the practice of Monte Carlo Markov chain methods. However, using CFTP is not always easy. Bounding chains are an important tool for using perfect sampling algorithms such as CFTP and FMR, or even just for determining the mixing time of a chain. The fact that it leads to relatively straightforward local analyses of the original chains is an added perk that makes this method surprisingly powerful.

Bibliography

[1] D. Aldous. Random walks on finite groups and rapidly mixing Markov chains. In A. Dold and B. Eckmann, editors, *Séminaitre de Probabilités XVII 1981/1982*, volume 986 of *Springer-Verlag Lecture Notes in Mathematics*, pages 243–297. Springer-Verlag, New York, 1983.

[2] D. Aldous and J.A. Fill. *Reversible Markov Chains and Random Walks on Graphs*. Monograph in preparation, 1994.

[3] D.J. Aldous and P. Diaconis. Strong uniform times and finite random walks. *Adv. in Appl. Math.*, 8:69–97, 1987.

[4] K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19:357–367, 1967.

[5] Petr Beckman. *A History of* π. St. Martin's Press, 1971.

[6] Russ Bubley and Martin Dyer. Graph orientations with no sink and an approximation for a hard case of ♯SAT. *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 248–257, 1997.

[7] P. J. Burville and J. F. C. Kingman. On a model for storage and search. *Journal of Applied Probability*, 10:697–701, 1973.

[8] Colin Cooper and Alan Frieze. Mixing properties of the Swendsen-Wang process on classes of graphs. *Preprint*, 1998.

[9] Luc Devroye. *Branching Processes and Their Applications in the Analysis of Tree Structures and Tree Algorithms*, pages 249–306. Springer, 1998.

[10] W. Doeblin. Exposé de la théorie des chains simples constantes de markov à un nombre fini d'états. *Rev. Math. de l'Union Interbalkanique*, 2:77–105, 1933.

[11] Martin Dyer, Alan Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. Technical Report ECS-LFCS-98-391, University of Edinburgh, 1998.

[12] Martin Dyer and Catherine Greenhill. On Markov chains for independent sets. *Preprint*, 1997.

[13] Martin Dyer and Catherine Greenhill. A genuinely polynomial-time algorithm for sampling two-rowed contingency tables. In *25th International Colloquium on Automata, Languages and Programming*, pages 339 – 350. EATCS, 1998.

[14] James A. Fill. An interruptible algorithm for perfect sampling via Markov chains. *Annals of Applied Probability*, 8:131–162, 1998.

[15] V. Gore and M. Jerrum. The Swensen-Wang process does not always mix rapidly. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 674–681, 1997.

[16] O. Häggström, M.N.M. van Leishout, and J. Moller. Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. *Bernoulli*, 96. to appear.

[17] Olle Häggström and Karin Nelander. Exact sampling from anti-monotone systems, 1997. Preprint.

[18] Olle Häggström and Karin Nelander. On exact simulation from Markov random fields using coupling from the past. *Scand. J. Statist.*, 1998. To Appear.

[19] W. J. Henricks. The stationary distribution of an interesting markov chain. *Journal of Applied Probability*, 9:886–890, 1973.

[20] R. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[21] Mark L. Huber. Exact sampling and approximate counting techniques. In *Proceedings of the 30th Annual Symposium on the Theory of Computing*, pages 31–40, 1998.

[22] Mark L. Huber. Exact sampling for the Widom-Rowlinson mixture model. 1998. Preprint.

[23] Mark L. Huber. Exact sampling using Swendsen-Wang. In *Proceeding of the Symposium on Discrete Algorithms*, 1998. To Appear.

[24] Mark L. Huber. Perfect sampling from independent sets. 1998. Preprint.

[25] M. Jerrum. A very simple algorithm for estimating the number of k-colourings of a low-degree graph. *SIAM Journal on Computing*, 22:1087–1116, 1995.

[26] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[27] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. In *Proceedings of the 17th ICALP*, pages 462–475. EATCS, 1990.

[28] V. E. Johnson. Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *Journal of the American Statistical Association*, 91:154–166, 1996.

[29] Wilfrid S. Kendall. Perfect simulation for the area-interaction point process. In *Proceedings of the Symposium on Probabiity Towards the Year 2000*, 1995.

[30] D. E. Knuth. *The Art of Computer Programming, Volume I, Sorting and Searching*. Addison-Wesley, 1973.

[31] J.L. Lebowitz and G. Gallavotta. Phase transitions in binary lattice gases. *Journal of Mathematical Physics*, 12:1129–1133, 1971.

[32] H.W. Lotwick and B.W. Silverman. Convergence of spatial birth-and-death processes. *Math. Proc. Cambridge Philos. Soc.*, 90:155–165, 81.

[33] Michael Luby and Eric Vigoda. Approximately counting up to four (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 682–687, 1997.

[34] J. McCabe. On serial files with relocatable records. *Operations Research*, 12:609–618, 1965.

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

[36] J. Møller. On the rate of convergence of spatial birth-death processes. *Ann. Inst. Statist. Math.*, 41:565–581, 1989.

[37] Duncan J. Murdoch and Jeffrey S. Rosenthal. An extension of Fill's exact sampling algorithm to non-monotone chains. 1998. Preprint.

[38] Sidney C. Port. *Theoretical Probability for Applications.* Wiley-Interscience, 1994.

[39] R. B. Potts. Some generalised order-disorder transformations. *Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.

[40] C.J. Preston. Spatial birth-and-death processes. *Bull. Inst. Internat. Statist.*, 46(2):371–391, 1977.

[41] James G. Propp and David B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1 and 2):223–252, 1996.

[42] Dana Randall and David Wilson. Sampling spin configurations of an Ising system. In *Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[43] B. D. Ripley. Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society Series B*, 39:172–212, 1977.

[44] B. D. Ripley. Simulating spatial patterns: dependent samples from a multivariate density. *Applied Statistics*, 28:109–112, 1979.

[45] Ronald Rivest. On self-organizing sequential search heuristics. *Communications of the ACM*, 19:63–67, 1976.

[46] J. Salas and A. D. Sokal. Absence of phase transition for antiferromagnetic Potts models via the Dobrushin uniqueness theorem. *Journal of Statistical Physics*, 86(3–4):551–579, 1997.

[47] Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

[48] R. Swendsen and J-S. Wang. Non-universal critical dynamics in Monte Carlo simulation. *Physical Review Letters*, 58:86–88, 1987.

[49] B. Widom and J.S. Rowlinson. New model for the study of liquid-vapor phase transition. *Journal of Chemical Physics*, 52:1670–1684, 1970.

[50] Thomas Yan. A rigorous analysis of the Creutz demon algorithm for the microcanonical, one-dimensional Ising model. *Ph.D. Thesis*, 1999.