Annealing and Tempering for Sampling and Counting

A Thesis
Presented to
The Academic Faculty

by

Nayantara Bhatnagar

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Algorithms, Combinatorics and Optimization
Georgia Institute of Technology
August 2007

Annealing and Tempering for Sampling and Counting

Approved by:

Professor Dana Randall, Advisor
College of Computing,
Georgia Institute of Technology.

Professor Eric Vigoda, Advisor
College of Computing,
Georgia Institute of Technology.

Professor Prasad Tetali,
School of Mathematics,
Georgia Institute of Technology.

Professor Robin Thomas,
School of Mathematics,
Georgia Institute of Technology.

Professor Santosh Vempala,
College of Computing,
Georgia Institute of Technology.

Date Approved: 22^{nd} June, 2007

For my parents.

ACKNOWLEDGEMENTS

I would like to thank my advisors Dana Randall and Eric Vigoda for their guidance and support, which they gave generously. I was fortunate to have the opportunity to work with two of the leading researchers in my field. Dana never ceased to surprise me with her quick insights and ability to get to crux of a problem. I thoroughly enjoyed working with Eric and am always inspired by his persistent approach to solving hard problems.

During my stay at Georgia Tech, I had the opportunity to interact with other faculty in theory group and the ACO program and I would like to thank them as well. I would like to thank Prasad Tetali especially, who is the nicest of people, and from whose methodical approach to research I learned a lot. I'd like to thank Vijay Vazirani for his timely advice as well as for some enjoyable discussions on problems. I'd like to thank Santosh Vempala for some fun discussions as well.

Thanks to Ivona Bezáková, Juan Vera and Sam Greenberg for enjoyable discussions in the course of our collaborations. I'd like to thank all the friends I made at Georgia Tech; Aranyak, Vangelis, Amin, Nikhil, Apurva, Ashok, Tejas, Deeparnab, Gagan and Rishi: your presence made the working environment at Tech very cordial and friendly.

I would like to thank my family; my parents Rakesh and Nirupama, and my sister Melanie for their constant support and encouragement.

Lastly, I'd like to thank Parikshit for having faith in me when my own belief often faltered. Our collaboration was the one I enjoyed the most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The Markov Chain Monte Carlo (MCMC) method has been widely used in practice since the 1950's in areas such as biology, statistics, and physics. However, it is only in the last few decades that powerful techniques for obtaining rigorous performance guarantees with respect to the running time have been developed. Today, with only a few notable exceptions, most known algorithms for approximately uniform sampling and approximate counting rely on the MCMC method. This thesis focuses on algorithms that use MCMC combined with an algorithm from optimization called simulated annealing, for sampling and counting problems.

Annealing is a heuristic for finding the global optimum of a function over a large search space. It has recently emerged as a powerful technique used in conjunction with the MCMC method for sampling problems, for example in the estimation of the permanent and in algorithms for computing the volume of a convex body. We examine other applications of annealing to sampling problems as well as scenarios when it fails to converge in polynomial time.

We consider the problem of randomly generating 0-1 contingency tables. This is a well-studied problem in statistics, as well as the theory of random graphs, since it is also equivalent to generating a random bipartite graph with a prescribed degree sequence. Previously, the only algorithm known for all degree sequences was by reduction to approximating the permanent of a 0-1 matrix. We give a direct and more efficient combinatorial algorithm which relies on simulated annealing. An interesting aspect of the annealing algorithm we define is that the high temperature distribution for the annealing is defined algorithmically.

Simulated tempering is a variant of annealing used for sampling in which a temperature parameter is randomly raised or lowered during the simulation. The idea is that by extending the state space of the Markov chain to a polynomial number of progressively smoother distributions, parameterized by temperature, the chain could cross bottlenecks in

the original space which cause slow mixing. The conventional wisdom is that tempering could speed up the convergence time exponentially, or at worst, it could be slower by at most a polynomial in the number of distributions. We first show that simulated tempering mixes torpidly for the 3-state ferromagnetic Potts model on the complete graph. The torpid mixing is caused by a first order phase transition, a fundamental difference in the behavior of this model from the Ising model, for which simulated tempering is known to converge at all temperatures. Moreover, we disprove the conventional belief and show that simulated tempering can converge at a rate that is slower than the algorithm at a fixed temperature by at least an exponential factor.

CHAPTER I

INTRODUCTION

Counting problems arise naturally in mathematics as well as computer science. Counting the number of primes less than a number n and counting the number of partitions of n into positive integers are two well-studied problems in number theory [41]. In enumerative combinatorics, counting problems from different areas of discrete mathematics are studied and the goal is to obtain closed form expressions or asymptotics for the number of objects of a given size that satisfy a certain property [81]. Algorithms for counting problems can be useful when there is no closed form expression known.

One of the aims of theoretical computer science is to classify the computational complexity of algorithmic tasks. In this setting, a counting problem is a particular type of computational problem where the objective is to count the number of objects satisfying a given property. We are interested in *efficient* algorithms for counting that run in time that is polynomial in the size of the objects, even though the number of objects may be exponential in the size. There is a large body of ongoing work dedicated to understanding the complexity of counting problems. A well-studied problem in combinatorics and computer science is that of estimating the permanent of a 0-1 matrix, which is equivalent to the problem of counting perfect matchings in a bipartite graph, #Bip-Perfect-Matching. This problem has played a pivotal role in our understanding of the complexity of counting problems [87].

Computational counting problems arise naturally from many different areas. Given a graph G, what is an algorithm for the problem #Perfect-Matching, counting the number of perfect matchings G contains? In a continuous setting, natural problems that arise in geometry include computing the volume of a convex body (#Volume) and integrating a multidimensional function. In statistical mechanics, computing an average over energies of configurations of particles (a function of "microscopic" interactions between particles) as a

function of temperature gives information about "macroscopic" thermodynamic properties of the system. Examples of problems studied in this context include computing the partition function of the Ising model [46], computing the number of "dimer coverings" or perfect matchings of a lattice [32], or counting the number of self-avoiding walks in a lattice [63]. For a large class of natural problems, Jerrum, Valiant and Vazirani [48] demonstrated there is a close connection between the complexity of counting and sampling algorithms.

1.1 The Computational Complexity of Counting

Formally, a counting problem aims to compute a function $f : \Sigma^* \to \mathbb{N}$ from strings over an alphabet Σ to the natural numbers. We can ask whether polynomial time counting algorithms exist for counting problems whose decision version is solvable in polynomial time. Over 150 years ago, Kirchoff [55] showed that the number of spanning trees of a graph is given by the determinant of its Laplacian, a matrix related to the adjacency matrix of the graph (see [90] for a proof). This formulation gives a polynomial time algorithm for counting the number of spanning trees, since the determinant can be computed in polynomial time by Gaussian elimination. In 1961, Fisher, Kasteleyn and Temperley [31, 53, 84] independently gave a polynomial time algorithm for computing the number of perfect matchings of a lattice. Their technique generalizes to counting the number of perfect matchings of any planar graph. Interestingly, both these problems can be reduced to the problem of computing a determinant.

Unfortunately, these are among the few problems for which exact counting algorithms are known. An explanation for this is given by Valiant's theory of *#P-completeness* [87]. Valiant defined the counting class #P to be the class of counting problems f where f is the number of accepting computations of a *non-deterministic polynomial time Turing Machine*. The class #P includes #Sat, the problem of computing the number of satisfying assignments to a SAT formula. A problem is said to be #P-complete if it is in #P and if every problem in #P is *polynomial time reducible* to it (see [71] for more details). Valiant showed that #Perfect-Matching, whose decision version is in P, is *#P-complete*. Thus an algorithm for #Perfect-Matching would imply an algorithm for #Sat. Since the

latter is at least as hard as SAT, one does not expect an efficient algorithm for #PERFECT-MATCHING, or indeed for any #P-complete problem.

Many natural problems with decision versions in P, such as computing the number of matchings of all sizes (#MATCHINGS), counting the number of independent sets (#IS), or #VOLUME, are #P-complete. Often the problems remain #P-complete even when restricted to natural classes of graphs, such as #MATCHINGS for lattices or bipartite graphs [42, 44]. Therefore, much of the algorithmic work on counting problems has focused on obtaining efficient approximations.

1.1.1 Counting by Sampling

One very successful approach to obtaining efficient approximation algorithms has been through the connection between approximate counting and approximately uniform sampling established by Jerrum, Valiant and Vazirani [48]. They showed that for "self-reducible" problems (explained below), approximating the size of the set Ω can be reduced to sampling elements of Ω approximately uniformly at random, and vice-versa. Informally, a self-reducible function is one which can be expressed in terms of the same function for a smaller input. For example, #SAT is self-reducible since the number of satisfying assignments to a SAT formula is the sum of the number of assignments to the smaller SAT instances obtained by setting the first variable to 1 and then to 0. A formal treatment of self-reducibility as well as the equivalence between approximate counting and sampling can be found in [80].

In general, the equivalence can be phrased in the language of *partition functions*. In statistical mechanics, one objective is to study the behavior of large collections of interacting particles. The particles can be in certain allowed *configurations*, and each configuration has an associated weight. Let Ω be a set of allowable configurations, and let the weight of $x \in \Omega$ be $w(x)$. The weight can be a function of a *temperature* parameter. The *partition function* is defined to be the sum

$$Z = \sum_{x \in \Omega} w(x). \tag{1}$$

If all the weights are 1, then the partition function is just $|\Omega|$. The partition function

is significant for physical systems because thermodynamic properties of the system such as specific heat and heat capacity are functions of the derivatives of $\log(Z)$ [35]. We are interested in efficient algorithms for approximating the partition function of a system.

Definition 1.0.1. *A* fully polynomial time randomized approximation scheme *or FPRAS, for computing $f : \Sigma^* \to \mathbb{N}$ is a randomized algorithm $A(\cdot)$ which on input $x \in \Sigma^*$ outputs a number $\hat{f}$ such that*

$$(1-\varepsilon)f(x) \leq \hat{f} \leq (1+\varepsilon)f(x)$$

with probability at least $1-\delta$ and runs in time that is polynomial in n, ε^{-1} and $\log(\delta^{-1})$.

Definition 1.0.2. *A* fully polynomial approximately uniform sampler *or FPAUS for sampling from Ω with distribution π, is a randomized algorithm $A(\cdot)$ which takes as input x and outputs an element of Ω according to some distribution whose variation distance[1] from π is at most ε and takes time that is polynomial in n and ε^{-1}.*

Theorem 1.1 ([48]). *For self-reducible functions, almost uniform generation and randomized approximate counting can be reduced to one another.*

As an example, suppose that Ω is the set of all independent sets of a graph $G = (V, E)$. Recall that an independent set $I \subseteq V$ in graph is a subset of the vertices such that no two vertices in I are adjacent. For a parameter $\lambda > 0$, the weight of an independent set I is defined to be $w(I) = \lambda^{|I|}$. The partition function is

$$Z_G(\lambda) = \sum_{I \in \Omega} \lambda^{|I|}.$$

When $\lambda = 1$, $Z_G(\lambda)$ counts the number of independent sets in the graph.

The equivalence between approximate counting and approximate sampling implies an FPRAS for $Z_G(\lambda)$ if we have an FPAUS for sampling from Ω according to the distribution π where $\pi(I)$ is proportional to $w(I)$. For the partition function $Z_G(\lambda)$, self-reducibility means that $Z_G(\lambda)$ can be expressed as a sum of partition functions for smaller graphs as follows

$$Z_G(\lambda) = Z_{G \setminus v}(\lambda) + \lambda Z_{G \setminus \{v \cup N(v)\}}(\lambda).$$

[1] *This measure of distance between distributions is defined in Section 1.3.*

The first term on the right hand side corresponds to the independent sets not containing v, which are exactly the independent sets of the graph $G \setminus v$, the graph obtained by deleting v and the edges containing it, from G. The second term corresponds to the independent sets containing v, which are the independent sets of the graph where v and its neighbors $N(v)$ are deleted.

The equivalence between approximate counting and approximate sampling is more general, and is known to hold for a variety of problems which are not self-reducible, such as #VOLUME, or the problem of counting the number of k-colorings of a graph.

Thus, the problem of obtaining efficient approximate counting algorithms for many problems of interest can be reduced to designing an approximate sampling algorithm. One of the most powerful techniques we have for approximate sampling from a set is to use a randomized algorithm based on simulating a Markov chain on the set of objects.

1.2 Markov Chains

Markov chains were first studied in 1906 by the Russian mathematician Andrey Markov, who was interested in the extension of the law of large numbers to dependent events. A Markov chain is a sequence of random variables $X_0, X_1, \cdots$ taking values in a finite set Ω satisfying the "Markov property", meaning that conditioned on the current state at time t, the state at $t+1$ is independent of the state at time $t-1$ and all previous times. Markov chains can be used to model a variety of stochastic phenomena such as Brownian motion, birth-death processes, gambling problems, shuffling decks of cards and queuing processes. They are applied in several areas of computer science and in other disciplines such as biology and statistical physics. For instance, the web-search algorithm employed by Google can be viewed as a Markov chain on an appropriately defined graph of web-pages [13]. In biology, genetic mutations, genome rearrangement and population processes are typically modeled as Markov chains [25].

The classical theory of Markov chains did not include a consideration of the rate of convergence, and it turns out that this plays an important role in the design of efficient sampling algorithms. Much of the theoretical analysis of Markov chains in computer science

has been on the Markov chain Monte Carlo (MCMC) method for randomly generating combinatorial objects. The idea is to construct a graph called the *Markov kernel* whose vertices are the states in Ω and whose edges are determined by defining a neighborhood structure for each state. Often, a natural choice for the kernel is to connect two combinatorial objects if one is a small perturbation of the other. The Markov chain performs a random walk on the Markov kernel, by choosing a random neighbor to move to from the current state according to fixed transition probabilities.

We are interested in designing *efficient* MCMC algorithms for sampling from the space Ω according to a distribution π. At a high level, this means that the number of steps required for the Markov chain to output a sample from a distribution that is "close" to π is polylogarithmic in the size of Ω.

1.2.1 Markov Chain Basics

Let $\mathfrak{M} = (X_t)_{t=0}^{\infty}$ be a stochastic process on the finite space Ω. Let P be a non-negative stochastic *transition matrix* of size $|\Omega| \times |\Omega|$ where the rows and columns are indexed by the states of Ω. That is, it satisfies the constraint that

$$\sum_{x_j \in \Omega} P(x_i, x_j) = 1 \qquad \text{for every } x_i \in \Omega.$$

The stochastic process $\mathfrak{M}$ is a *Markov chain* if for every time t, and states $x_0, \cdots, x_t$

$$\mathbb{P}[X_t = x_t | X_{t-1} = x_{t-1}, \cdots, X_0 = x_0] = \mathbb{P}[X_t = x_t | X_{t-1} = x_{t-1}] = P(x_{t-1}, x_t).$$

We will consider only Markov chains which are *time homogeneous*, so that for any time t_0 and pair of states x, x',

$$\mathbb{P}[X_{t_0+1} = x | X_{t_0} = x'] = P(x, x').$$

Together with the Markov property, time-homogeneity implies that the t-step transition probabilities are given by

$$\mathbb{P}[X_{t_0+t} = x | X_{t_0} = x'] = P^t(x, x').$$

A Markov chain is *ergodic* if it satisfies the following two technical conditions:

i) *Irreducibility:* For every $x, x' \in \Omega$, there exists a time t such that $P^t(x, x') > 0$.

ii) *Aperiodicity:* For every $x \in \Omega$, $gcd(t : P^t(x, x) > 0) = 1$.

A distribution π on Ω is *stationary* if $\pi P = \pi$.

Theorem 1.2 (Fundamental Theorem of Markov Chains, [30]). *An ergodic Markov chain on a finite space Ω has a unique limiting stationary distribution π, that is,*

$$\lim_{t\to\infty} P^t(x, x') = \pi(x') \qquad \text{for every } x, x' \in \Omega.$$

A distribution μ is *reversible* with respect to the transition matrix P of a Markov chain if for every $x, x' \in \Omega$,

$$\mu(x)P(x, x') = \mu(x')P(x', x). \tag{2}$$

Then the following can easily be verified.

Theorem 1.3. *If the distribution μ is reversible with respect to P, then it is a stationary distribution.*

We can use the above fact to define a Markov chain with the desired stationary distribution. This is the principle of the Metropolis-Hastings Markov chain [70]. Let P and μ be the transition matrix and stationary distribution of an irreducible Markov chain on Ω. We can construct a transition matrix Q with stationary distribution π on Ω as follows. Define

$$Q(x, y) = \begin{cases} P(x, y) \min\left(\frac{\pi(y)P(y,x)}{\pi(x)P(x,y)}, 1\right) & \text{if } y \neq x \\ 1 - \sum_{z\neq x} P(x, z) \min\left(\frac{\pi(z)P(z,x)}{\pi(x)P(x,z)}, 1\right) & \text{if } y = x \end{cases}$$

Then, it can be checked that π is reversible with respect to Q and hence is a stationary distribution.

Suppose we wish to sample weighted independent sets in G with parameter λ according to the distribution $\pi(I) \propto w(I)$. The *heat bath Glauber dynamics* Markov chain M_{IS} is given as follows. Let I_t denote the independent set at time t.

1. Choose $v \in V$ uniformly at random.

2. With probability $\frac{\lambda}{1+\lambda}$ attempt to add v to I_t. If $I_t \cup v$ is an independent set, set $I_{t+1} = I_t \cup v$, otherwise, set $I_{t+1} = I_t$.

3. With probability $\frac{1}{1+\lambda}$, set $I_{t+1} = I_t$.

Theorems 1.2 and 1.3 imply that M_{IS} will converge in the limit to the distribution π. Heat bath Glauber dynamics is a general Markov chain used for sampling in a class of models called *spin systems* which we introduce next.

1.2.2 Spin Systems and Glauber dynamics

In statistical mechanics, spin systems are used to model the behavior of finite collections of interacting particles. We are interested in sampling from configurations of the spin system to understand the properties of "typical" configurations. A spin system consists of an underlying graph $G = (V, E)$ and a set of q *spins*. The set of configurations is $\Omega \subseteq [q]^V$ and each $x \in \Omega$ satisfies some local constraints at each vertex. Each configuration has a weight and the objective is to sample from the configurations with probabilities proportional to these weights. We illustrate below with some examples. The third example, the q-state Potts model, will be the focus of Chapter 4.

Independent Sets: The set of spins is $\{0, 1\}$, and $q = 2$. A vertex is assigned 1 if it is in the independent set and 0 if not. The local constraint is that for each edge $(i, j) \in E$, $x(i) + x(j) \leq 1$. The weight of a configuration x is given by $w(x) = \sum_{i \in V} x(i)$. We wish to sample from the distribution π on Ω given by

$$\pi(x) = \frac{\lambda^{w(x)}}{Z_G(\lambda)}.$$

The parameter λ is also referred to as the *activity* or *fugacity*.

Ising Model: The Ising model was first defined in the 1920's to study ferromagnetism in solids. It is now studied in a much broader context [17]. The set of spins is $\{-1, +1\}$ corresponding to the magnetic moment of an atom of the solid. The set of states Ω is an assignment of spins to each vertex of G. In the case without any external magnetic field, the *Hamiltonian* of a configuration x is given by

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

Define the inverse temperature to be $\beta = \frac{1}{kT}$, where k is Boltzmann's constant. The *Gibbs distribution* at inverse temperature β is given by

$$\pi_\beta(x) = \frac{e^{\beta H(x)}}{Z(\beta)},$$

where $Z(\beta) = \sum_{y \in \{\pm 1\}^n} e^{\beta H(y)}$, the normalizing factor, is the partition function.

Note that when $\beta > 0$, configurations with a large number of edges with the same spin on both endpoints are favored in the stationary distribution. For large values of β, a typical configuration will have large components of vertices of the same spin with small clusters of the opposite spin. At small values of β the spins in a typical configuration will look fairly independent.

q-state Potts Model: The q-state Potts model was defined by R.B. Potts in 1952 [72], and generalizes the Ising model to more than two spins. It models particles of a crystalline solids and was defined in order to understand the behavior of ferromagnetism and other solid-state phenomenon. The set of spins is $\{1, \cdots, q\}$. The space Ω of the q-state ferromagnetic Potts model is the set of all q^n q-colorings of G. The Hamiltonian of a configuration x is given by

$$H(x) = \sum_{(i,j) \in E(G)} J \cdot \delta(x(i), x(j)),$$

where δ is the Kronecker-δ function that takes the value 1 if its arguments are equal and zero otherwise. When $J > 0$ the model corresponds to the *ferromagnetic* case where neighbors prefer the same color, while $J < 0$ corresponds to the *anti-ferromagnetic* case where neighbors prefer to be differently colored. The Gibbs distribution at inverse temperature β is given by

$$\pi_\beta(x) = \frac{e^{\beta H(x)}}{Z(\beta)},$$

where $Z(\beta) = \sum_{y \in [q]^n} e^{\beta h(y)}$ is the partition function. Note that when $J < 0$, in the limit as $\beta \to \infty$, the distribution tends to the uniform distribution over proper q-colorings of G.

In each of these cases, there is a natural way to define a neighborhood structure on states. Two independent sets are adjacent if they differ by exactly one vertex. Configurations of the Ising and Potts model are adjacent if the spin at exactly one vertex differs. Glauber dynamics is a random walk on the graph of configurations defined by these adjacencies.

The *heat bath Glauber dynamics Markov chain* can be defined for sampling from the distribution π for a spin system as follows. Let X_t denote the configuration at time t.

1. Choose $i \in V$ uniformly at random.
2. Choose X_{t+1} with the conditional distribution $\pi(\cdot | X_t(j)\ j \neq i)$ conditioned on $X_{t+1}(j) = X_t(j)$ for every $j \neq i$.

The fundamental theorem of Markov chains (Theorem 1.2) guarantees that Glauber dynamics will converge to the Gibbs distribution in the limit. Although the above exposition focused on sampling from the Gibbs distribution for spin systems, in fact, by the general principle used in the Metropolis-Hastings algorithm, we can construct a Markov chain which converges to the desired stationary distribution in the limit. However, from an algorithmic perspective this is not sufficient. We would like to bound the number of steps the Markov chain must be run until we obtain samples from a distribution which is a good approximation to the stationary distribution and this can vary significantly for different Markov chains. The algorithmic issues are made precise below.

1.3 Markov Chain Monte Carlo - Algorithmic Considerations

In order to get efficient algorithms for sampling and counting, we need to guarantee that the Markov chain reaches, or gets "close" to the stationary distribution in a reasonable amount of time. By reasonable, we mean that the time should grow at most polylogarithmically in $|\Omega|$. Typically, $|\Omega|$ is exponentially large in the size of the object we wish to generate. For instance, the number of independent sets can be an exponential in the number of vertices of the graph but we would like the mixing time to be bounded by a polynomial in the number of vertices. The rate of convergence of the Markov chain to the stationary distribution is quantified by the mixing time, as defined below.

The *total variation distance* (or variation distance) between two distributions μ, ν on Ω is given by

$$d_{tv}(\mu, \nu) = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|.$$

The *mixing time* from the starting state x, $\tau_x(\delta)$ is given by

$$\tau_x(\delta) = \min\{t \geq 0 \mid d_{tv}(P^t(x, \cdot), \pi) \leq \delta\}$$

If the mixing time $\tau_x(\delta)$ from any starting point is bounded by a polynomial in n and $\ln \delta^{-1}$, then the chain is *rapidly mixing*. If the mixing time from any state is bounded from below by an exponential in n^{ε} for any $\varepsilon > 0$, the chain is *torpidly mixing*. The requirement that the mixing time be polynomial has resulted in an extensive study of the mixing rate of Markov chains, producing a wide array of techniques for proving both rapid and torpid mixing (See the survey by Randall [73] and the monograph by Jerrum [43] for a comprehensive introduction). There is now a large body of work on methods for bounding the mixing time of a Markov chain such as coupling, spectral gap characterization, conductance and isoperimetry, multicommodity flows, comparison, and decomposition. Some of these which are used in the work in this thesis are explained in more detail in Chapter 2.

These techniques have been applied with great success to the analysis of the mixing time for natural Markov chains for many central problems including computing the partition function for the ferromagnetic Ising model [46], computing the volume of a convex body [28], sampling k-colorings of a graph when the maximum degree is large [88] and estimating the permanent of a 0-1 matrix [47]. On the negative side, it has been shown that there are instances where Markov chains such as Glauber dynamics, which make "local" updates, will mix torpidly [12, 27, 67, 74, 85].

1.3.1 Torpid Mixing of Local Markov Chains

Torpid mixing for Glauber dynamics is a feature of systems which exhibit *phase transitions* (see Section 4.1.1) where there is an abrupt change in what typical configurations look like. For example, in the Ising model, at low temperatures, typical configurations are "ordered", with most of the spins of the same kind. As the temperature is increased, at a critical point most of the weight of the Gibbs distribution is on configurations that are "disordered," where the spins appear independent. In the ordered phase, there may be multiple classes of configurations that dominate in the Gibbs measure. In the Ising model at low temperature, one type of configuration that dominates the measure is when most of the vertices have spin

$+1$ (the other type being when most spins are -1). In this scenario, Glauber dynamics mixes torpidly. At a high level, this is because for the Markov chain to go from configurations that are predominantly $+1$ to those that are predominantly -1, it must pass through balanced configurations that are highly unlikely in the distribution [67, 85].

Other examples of this phenomenon include the torpid mixing of Glauber dynamics for independent sets of the lattice $\mathbb{Z}^2$ for large enough λ [74], independent sets and the q-state Potts model on the lattice $\mathbb{Z}^d$ in a region around certain critical values of λ and β respectively [12], Glauber dynamics for sampling independent sets of the d-dimensional hypercube for large enough λ [34], and Glauber dynamics for sampling independent sets in graphs of maximum degree at least 6 [27].

From the perspective of rapid mixing, what these examples have in common is that there is a "bottleneck" or "cut" in the distribution over the space. Roughly, there are two or more regions in the space containing most of the probability mass of the stationary distribution, separated (in the sense that deleting these states from the Markov kernel would disconnect it) by a region with exponentially small measure. Local Markov chains fail to cross this cut in the state space in polynomial time. This intuition can be formalized by using the fact that the minimum cut in the state space in fact characterizes the mixing time. The *conductance* Φ (see Chapter 2 for precise definitions) is the minimum over all subsets $S \subseteq \Omega$ of the probability of the Markov chain leaving the set conditioned on being in S. Jerrum and Sinclair [49] showed that the mixing time of a reversible Markov chain is polynomial if and only if the conductance is at least inversely polynomial. To prove torpid mixing, it is sufficient to show that the conductance is smaller than any inverse polynomial.

It can be shown that if Ω can be partitioned into three disjoint sets S_1, S_2, S_3 such that states of S_1 and S_3 are connected by the Markov chain only through states of S_2, then the conductance Φ is bounded by

$$\max\left(\frac{\pi(S_2)}{\pi(S_1)}, \frac{\pi(S_2)}{\pi(S_3}\right).$$

Consider the case of independent sets of a bipartite graph. In [27] it is shown that there are bipartite graphs of degree 6 for which any chain that adds and deletes at most some constant fraction of the vertices will mix torpidly. We present a simplified argument

here, showing that there are bipartite graphs of some constant degree for which M_{IS} mixes torpidly.

The idea is that in a dense bipartite graph, large independent sets will have most vertices in one bipartition or the other. The independent sets which are roughly balanced will have to be small in size by the assumption on density. The following theorem shows that there exist constant degree bipartite graphs that are sufficiently dense.

Theorem 1.4 ([75]). *For every $0 < \delta < 1$ and sufficiently large n, there is a bipartite graph $([n], [n], E)$ of degree $O(\delta^{-1} poly(\log(\delta^{-1})))$ such that for every pair of subsets $A \subseteq [n], B \subseteq [n]$ of the two partitions, if $|A| \geq \delta n$ or $|B| \geq \delta n$, there is at least one edge in the graph induced by $A \cup B$.*

The above theorem follows by a probabilistic argument.

Theorem 1.5. *There exists $0 < \delta < 1$ such that for sampling independent sets of the graph above with the parameter δ, the Glauber dynamics Markov chain M_{IS} has exponentially small conductance.*

Proof. We define a cut with exponentially small conductance as follows: Let S_1 be the independent sets with greater than δn vertices in the left bipartition. The set S_2 consists of independent sets where the left bipartition has exactly δn vertices. We set $S_3 = \Omega \setminus S_1 \setminus S_2$. Note that S_2 contains no independent sets with more that δn vertices in the right bipartition. To reach such an independent set from S_1, the Markov chain must pass through S_2.

We can bound the sizes of these sets as follows. Then,

$$|S_2| \leq \binom{n}{\delta n}^2 \leq \left(\frac{e}{\delta}\right)^{2\delta n}$$

and $S_1, S_3 \geq 2^n$. Hence, for δ sufficiently small,

$$\Phi \leq \frac{|S_2|}{\min(|S_2|, |S_3|)} \leq e^{-\Omega(n)}.$$

□

Thus, in order to make effective use of the counting-sampling paradigm for such problems, one needs to either design Markov chains which modify the states in a non-local

fashion, or to modify the existing local sampling scheme so that it can overcome the bottlenecks which cause it to mix torpidly. Simulated annealing, a heuristic for optimization, as well as simulated tempering, a Markov chain algorithm are methods that attempt to take this latter approach.

1.4 Annealing and Simulated Tempering

Simulated annealing is a heuristic for optimization over a large search space that attempts to improve on local search which can get trapped in local optima [15, 56]. Annealing uses a temperature parameter so that at high temperatures, with some non-zero probability, the algorithm makes unfavorable moves that allow it to move out of local optima. The annealing starts at high temperature and gradually the temperature is lowered so that unfavorable moves become less and less likely.

We first define the annealing algorithm in the context of optimization and then in the context of MCMC algorithms for sampling.

1.4.1 Annealing in Optimization

Let H be a function defined over the finite search space Ω. In an optimization problem, we would like to find $x \in \Omega$ such that $H(x) = \max_{y \in \Omega} H(y)$. Let P be the transition matrix of a Markov chain defined on Ω. Let T_t denote the temperature at time t. The sequence $(T_t)_{t=1}^{\infty}$ is a *cooling schedule* if $\lim_{t \to \infty} T_t = 0$. Let $\beta = 1/T$ be an inverse temperature parameter. The transition probabilities at temperature β are given by

$$P_\beta(x,y) = \begin{cases} P(x,y) \min\left(1, \frac{e^{\beta H(y)}}{e^{\beta H(x)}}\right) & \text{if } y \neq x \\ 1 - \sum_{z \neq x} P_\beta(x,z) & \text{if } y = x \end{cases}$$

The simulated annealing algorithm is defined as follows. Start at an arbitrary point x_0 in Ω. For each time $t = 1, \cdots T$,

i) Let $\beta = 1/T_t$ the inverse temperature as defined by the cooling schedule.

ii) From the current point x_{t-1}, choose the point X_t according to the distribution on states given by $P_\beta(x_{t-1}, \cdot)$.

The algorithm thus defines a random sequence of states $(X_t)_{t=1}^{\infty}$. When T is large, the moves are made with less regard for improvement in the function, but as T decreases, unfavorable moves become less likely. At the high temperature, the dynamics converges to the uniform distribution over Ω. As the temperature becomes lower, the limiting distribution of the dynamics becomes more biased towards the optimal states. Let

$$\hat{\Omega} = \{x : H(x) = \max_y H(y)\}$$

be the set of global maxima. The simulated annealing algorithm is convergent if

$$\lim_{t\to\infty} \mathbb{P}[X_t \in \hat{\Omega}] = 1.$$

A heuristic justification given for annealing is that performing unfavorable moves with some probability will allow the local algorithms to cross the barriers that cause it to get trapped at local optima at low temperatures. There are only a few settings where the convergence of the algorithm has been analyzed [40, 54, 51], although for the graph bisection problem studied in the last work, it was shown by Carson and Impagliazzo [14] that local search also suffices and annealing is not required.

1.4.2 Annealing in the MCMC Framework

A relatively recent development has been the analysis of algorithms using annealing in conjunction with MCMC for sampling and counting problems. Two examples are the simulated annealing algorithm of Jerrum, Sinclair and Vigoda for estimating the permanent [47] and the algorithm of Lovász and Vempala for computing the volume of a convex body [59], which is currently the fastest algorithm for that problem.

Suppose that we wish to sample from Ω at an inverse temperature β^* so that $\pi(x) \propto e^{\beta^* H(x)}$. If the Markov chain on Ω is symmetric, i.e., $P(x, y) = P(y, x)$, then the Markov chain with transition matrix P_{β^*} as defined above will converge to π. The annealing algorithm is defined as before, that is, at each time, we take a step from the current state using the transition probabilities at the temperature specified by the cooling schedule. In this case we say the simulated annealing algorithm converges if starting with an initial distribution

$\mathbf{x_0}$ over states

$$\lim_{t\to\infty} \mathbf{x_0} P_{\beta_1} \times \cdots \times P_{\beta_t} = \pi_{\beta^*}.$$

The following toy example illustrates how annealing can help overcome bottlenecks for local Markov chains. Let $\Omega = [n]$ and for each $1 \leq i \leq n-1$, let i be adjacent to $i+1$ in the Markov kernel. Suppose for some β^* we are interested in sampling from the distribution

$$\pi_{\beta^*}(i) = \begin{cases} \frac{e^{\beta^* n}}{(n-1)e^{\beta^* n}+1} & \text{if } i \neq n/2 \\ \frac{1}{(n-1)e^{\beta^* n}+1} & \text{if } i = n/2. \end{cases}$$

We can abstract out the important ideas by thinking of just 3 points on a line, where the stationary probabilities of the end points are $\frac{e^{\beta n}}{2e^{\beta n}+1}$ while the stationary probability of the middle point is $\frac{1}{2e^{\beta n}+1}$.

Suppose that at temperature $\beta^* > 0$, we start with the probability mass entirely on one endpoint so that $x_0 = (1, 0, 0)$. If we apply the transition matrix P_{β^*} repeatedly to x_0, we find that the time taken for the distribution to come within $1/4$ in variation distance of the stationary distribution is exponentially large. This corresponds to the fact that starting at the endpoint with exponentially large mass, we would have to wait for exponential time before a move to the middle point is accepted.

We define an annealing algorithm for the sampling problem. The cooling schedule consists just of two temperatures, 0 and β^*. For some constants C_1, C_2, the schedule is given by $\beta_t = 0$ for $t \leq C_1 n^2$ and $\beta_t = \beta^*$ for $C_1 n^2 < t \leq C_2 n^2$.

Theorem 1.6. *The simulated annealing algorithm converges to within ε of the distribution π_{β^*} in time $O(n^2 \ln \varepsilon^{-1})$.*

A sketch of the proof is as follows. The transition matrix P_β of the chain at temperature β is given by

$$\begin{pmatrix} 1-\alpha & \alpha & 0 \\ 1/3 & 1/3 & 1/3 \\ 0 & \alpha & 1-\alpha \end{pmatrix},$$

where $\alpha = \frac{1}{3e^{\beta n}}$.

After first $N = C_1 n^2$ steps, the distribution is uniform over each of the points, since the transition matrix at $\beta = 0$ is just the reflecting random walk on the line. Now, if we iterate the matrix P_{β^*}, it can be calculated that the variation distance from the stationary distribution decreases at the rate of roughly $1/3$ in each step. Thus in $O(\ln \varepsilon^{-1})$ steps we would be within ε of the stationary distribution. Going back to the line on n points, taking into account that it takes $O(n^2)$ time to mix on either half of the line, we get that after $O(n^2 \ln \varepsilon^{-1})$ steps of running the chain at β^* the distribution is at most ε in distance away from π_{β^*}. In this case, the speedup in the convergence time was because annealing at $\beta = 0$ gave us a good starting distribution for the Markov chain at β^*.

1.4.3 Simulated Tempering

The *simulated tempering* Markov chain [65] is a variant of annealing where the temperature is chosen randomly in each time step. Suppose that we wish to sample from the distribution π_M at a temperature β_M. To use simulated tempering, we define a sequence of distributions $\pi_{M-1}, \cdots, \pi_0$, parameterized by inverse temperatures $\beta_{M-1}, \cdots, \beta_0 = 0$. The state space of the tempering chain is $\Omega \times [M]$, a tuple consisting of a state and a temperature. Suppose that we are in the state (x, i). At each step of the chain, either the temperature is kept fixed at β_i and the first co-ordinate, i.e., the state is randomly updated or we attempt to randomly change the temperature to β_{i+1} or β_{i-1} keeping the state fixed. We describe the transitions of the chain precisely in Section 4.2.1 of Chapter 4.

The heuristic justification for this Markov chain is that by extending to a polynomial number of smoother distributions at lower inverse temperatures, the Markov chain may be able to cross bottlenecks at low temperatures without paying a large penalty in the running time.

Annealing and tempering provide a generic framework that can be applied in principle to any sampling problem. However, issues such as how to choose the distributions for tempering or how to choose the cooling schedule are not addressed and depend on the specifics of the problem at hand. Nevertheless, such techniques have been successfully applied to the estimation of the permanent [47] and the computation of volume [59]. These methods

are also popular in practice and this strongly suggests the need for a better understanding of the power and limitations of these techniques. In particular, we would like to answer questions of the following kind:

1. For which problems can annealing and tempering speed up the mixing rate of a Markov chain?

2. Can one apply these techniques to *any* torpidly mixing chain and hope that we will only do better with regard to mixing time, or not worse by more than a polynomial in the number of temperatures?

1.5 Contributions of This Thesis

In this thesis we address the questions above and the contributions are two-fold. On the positive side, we demonstrate the power and flexibility of the annealing method by applying it to the problem of sampling and counting labeled bipartite graphs with given degrees. Unlike previous approaches, the annealing algorithm required a careful choice of the starting distribution for annealing. The main novelty was that the starting distribution was found using a combinatorial algorithm. Our algorithm bypasses the reduction to computing the permanent thus improving on the previously best known running time. Finally, our algorithm can be extended to the case of sampling subgraphs with given degrees of an input graph. This work appears in Chapter 3 and is based on joint work with Ivona Bezáková and Eric Vigoda and appeared in *Random Structures and Algorithms, 2007* [7]. A preliminary version appeared in *Proceedings of the Symposium on Discrete Algorithms, 2006.*

On the negative side, we disprove the belief that these heuristics can always be tried in practice since they can only improve the mixing time of fixed temperature algorithms, or at worst slow them down by a polynomial factor. We show that the mixing time for the simulated tempering Markov chain for sampling from configurations of the 3-state ferromagnetic Potts model on the complete graph is exponentially large. Our analysis reveals that the torpid mixing is due to a first order phase transition in the system, at a critical inverse temperature. Moreover, simulated tempering will mix exponentially slowly regardless of the intermediate temperatures chosen to define the tempering algorithm. Finally,

the mixing rate is actually slower by an exponential factor compared to the mixing time of the Metropolis algorithm at a fixed temperature. This work appears in Chapter 4 and is based on joint work with Dana Randall and appeared in the *Proceedings of the Symposium on Discrete Algorithms, 2004* [9].

CHAPTER II

MARKOV CHAIN BACKGROUND

In this chapter we state some of the now classical techniques for bounding the mixing time of a Markov chain. These will be the main tools we use in the analysis of the Markov chains we study subsequently.

2.1 Eigenvalue Gap

The inverse of the spectral gap of the transition matrix of a Markov chain characterizes the mixing time. Let $\lambda_0, \lambda_1, \ldots, \lambda_{|\Omega|-1}$ be the eigenvalues of an ergodic reversible Markov chain with transition matrix P, so that $1 = \lambda_0 > |\lambda_1| \geq |\lambda_i|$ for all $i \geq 2$. Let the spectral gap be $Gap(P) = \lambda_0 - |\lambda_1|$.

Theorem 2.1 ([1, 24]). *For $\delta > 0$,*

i) $\tau_x(\delta) \leq \frac{1}{Gap(P)} \ln\left(\frac{1}{\pi(x)\delta}\right)$.

ii) $\max_x \tau_x(\delta) \geq \frac{|\lambda_1|}{2Gap(P)} \ln\left(\frac{1}{2\delta}\right)$.

However, in general, the eigenvalue gap is not easy to compute since the size of the transition matrix is large.

2.2 Conductance

The *conductance*, introduced by Jerrum and Sinclair, provides a measure of the mixing rate of a chain [49]. The conductance is in fact related to the spectral or eigenvalue gap (see for example [80]), a connection which was studied independently by Lawler and Sokal [57]. A similar relationship between the second largest eigenvalue and the "expansion" of a graph was established by Alon [2] and Alon and Milman [3].

Bounding the conductance often gives an easier means for bounding the gap, in order

to bound the mixing time. For $S \subset \Omega$, let

$$\Phi_S = \frac{F_S}{C_S} = \frac{\sum_{x \in S, y \notin S} \pi(x) P(x,y)}{\pi(S)}.$$

Then, the conductance is given by

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

Jerrum and Sinclair [49] showed that the conductance upper and lower bounds the mixing time.

Theorem 2.2. *For any reversible Markov chain with conductance* Φ

$$\frac{1-2\Phi}{2\Phi} \ln \varepsilon^{-1} \leq \max_x \tau_x(\delta) \leq \frac{2}{\Phi^2} \left(\ln \delta^{-1} + \ln \frac{1}{\pi_{\min}} \right)$$

where $\pi_{\min} = \min_{x \in \Omega} \pi(x)$. Thus, to lower bound the mixing time it is sufficient to show that the conductance is small.

2.3 Multicommodity Flow

The multicommodity flow method for bounding the mixing rate of Markov chain $\mathfrak{M}$ was introduced by Sinclair in [79]. Multicommodity flow is roughly a dual notion to the conductance and it characterizes rapid mixing as well. Here we state the upper bound on mixing time in terms of the congestion of the multicommodity flow. Let $\mathcal{K}$ denote the Markov kernel underlying $\mathfrak{M}$ so that $T = (M, M')$ is an edge of $\mathcal{K}$ if $P(M, M') > 0$. Let $\mathcal{P}_{IF}$ be the set of all directed paths from I to F in $\mathcal{K}$. A *flow* in the Markov kernel is a function $g : \bigcup_{I,F \in \Omega, I \neq F} \mathcal{P}_{IF} \to \mathbb{R}_0^+$ such that $\sum_{p \in \mathcal{P}_{IF}} g(p) = \pi(I)\pi(F)$. The *congestion* of the flow g is defined as:

$$\rho(g) = \ell(g) \max_{T=(M,M')} \left\{ \frac{1}{\pi(M) P(M,M')} \sum_{p \ni T} g(p) \right\}$$

where $\ell(g)$ is the length of the longest path p such that $g(p) > 0$. Note, the summation is over all $p \in \cup_{I,F} \mathcal{P}_{IF}$, and T is restricted to be an edge of the Markov kernel so that $P(M, M') > 0$.

This implies the following bound on the mixing time, from [79],

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

Multicommodity flow is a generalization of the *canonical paths* method [24, 79] where all the flow between two states is sent along a single path.

Two auxiliary techniques we make use of are Comparison and Decomposition, which are methods for bounding the spectral gap of one Markov chain in terms of the spectral gap of related Markov chains.

2.4 Comparison

The comparison theorem of Diaconis and Saloff-Coste [23] is useful in bounding the mixing time of a Markov chain when the mixing time of a related chain on the same state space is known.

Let $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be two Markov chains on Ω. Let P_1 and π_1 be the transition matrix and stationary distributions of $\mathfrak{M}_1$ and let P_2 and π_2 be those of $\mathfrak{M}_2$. Let $E(P_1) = \{(x, y) : P_1(x, y) > 0\}$ and $E(P_2) = \{(x, y) : P_2(x, y) > 0\}$ be sets of directed edges. For $x, y \in \Omega$ such that $P_2(x, y) > 0$, define a *path* γ_{xy}, a sequence of states $x = x_0, \cdots, x_k = y$ such that $P_1(x_i, x_{i+1}) > 0$. Let $\Gamma(z, w) = \{(x, y) \in E(P_2) : (z, w) \in \gamma_{xy}\}$ denote the set of endpoints of paths that use the edge (z, w).

Theorem 2.3. (Diaconis and Saloff-Coste [23])

$$Gap(P_1) \geq \frac{1}{A} \cdot Gap(P_2),$$

where

$$A = \max_{(z,w)\in E(P_1)} \left\{ \frac{1}{\pi_1(z)P_1(z, w)} \sum_{\Gamma(z,w)} |\gamma_{xy}|\pi_2(x)P_2(x, y) \right\}.$$

2.5 Decomposition

Decomposition theorems are useful for breaking a complicated Markov chain into simpler chains that are easier to analyze [62, 66, 50]. Let $\mathfrak{M}$ be a Markov chain with transition matrix P. Let $\Omega_1, \cdots, \Omega_m$ be a disjoint partition of Ω. For each $i \in [m]$, define the Markov chain $\mathfrak{M}_i$ on Ω_i whose transition matrix P_i, the *restriction* of P to Ω_i is defined as

- $P_i(x, y) = P(x, y)$, if $x \neq y$ and $x, y \in \Omega_i$;

- $P_i(x,x) = 1 - \sum_{y \in \Omega_i, y \neq x} P_i(x,y), \quad \forall x \in \Omega_i.$

The stationary distribution of $\mathfrak{M}_i$ is $\pi_i(A) = \frac{\pi(A \cap \Omega_i)}{\pi(\Omega_i)}$. Define the *projection* $\overline{P}$ to be the transition matrix on the state space $[m]$:

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

The decomposition theorem says that the spectral gap of the chain $\mathfrak{M}$ is at least one half the product of the spectral gap of the projection chain and the spectral gap of the slowest restriction chain.

Theorem 2.4. (Martin and Randall [66])

$$Gap(P) \geq \frac{1}{2} Gap(\overline{P}) \left(\min_{i \in [m]} Gap(P_i) \right).$$

CHAPTER III

A SIMULATED ANNEALING ALGORITHM FOR RANDOMLY GENERATING BINARY CONTINGENCY TABLES

3.1 Introduction and Motivation

Given a pair of non-negative integer sequences $r = r_1, \ldots, r_n$ and $c = c_1, \ldots, c_m$, a binary contingency table satisfying the sequences r, c is a 0-1 matrix where the i-th row sums to r_i and the j-th column sums to c_j, for all $1 \leq i \leq n, 1 \leq j \leq m$. We can also think of the binary contingency table as the adjacency matrix of a bipartite graph $G = U \cup V$ with vertex set $U = \{u_1, \ldots, u_n\}$ indexing the rows and $V = \{v_1, \ldots, v_m\}$ indexing the columns. Then the vertices u_i have degree r_i and the vertices v_j have degree c_j. We will use the matrix and graph views interchangeably throughout.

Gale and Ryser gave a necessary and sufficient condition on the row and column sums for such a matrix to exist. For $0 \leq k \leq \max(n, m) = M$, let c_k^* be the number of column sums that are at least as large as k.

Theorem 3.1 ([33, 76]). *There is a binary contingency table with row sums $r_1 \geq \cdots \geq r_n$ and column sums $c_1 \geq \cdots \geq c_n$ if and only if for each $1 \leq k \leq M$,*

$$\sum_{i=1}^{k} r_i \leq \sum_{i=1}^{k} c_i^* \tag{3}$$

where r_i or c_i are 0 if $i \geq n$ or m respectively.

Clearly this condition can be checked in time that is polynomial in n and m. It is well known that a simple greedy algorithm can be used to construct a matrix with the given row and column sums (if one exists) in polynomial time. In his paper, Ryser remarks that a more difficult problem is to determine the number of such matrices $N(r, c)$. Interestingly, it is not known to be #P-hard to compute $N(r, c)$. However, at this time, all algorithms

for computing $N(r, c)$ obtain only approximations to it, and are randomized. Thus it may very well be that there is a deterministic algorithm for computing $N(r, c)$ exactly which has so far eluded us. In addition to computing $N(r, c)$, we are also interested in the problem of generating a uniformly random bipartite graph with a given degree sequence.

Counting and sampling the number of binary contingency tables satisfying given marginals is an important problem in statistics [21, 22]. The number of binary contingency tables satisfying given marginals can be used to analyze the dependence between variables in the data that the table represents [77, 6]. In the theory of random graphs, an algorithm for generating a random graph with given degrees is useful if one wants to make a statement about typical properties of such graphs [54].

3.2 Previous Algorithmic Work

There have been two broad approaches to randomly generating bipartite graphs with a given degree sequence. In the Markov chain Monte Carlo approach, the idea is to define a Markov chain on the space of desired graphs so that after running the chain a sufficiently long time, we generate graphs with the required degrees almost uniformly at random.

The second approach has focused on defining efficient algorithms for generating a graph with the required degrees with *exactly* uniform probabilities. For example, a naive method, which is far from efficient, is to randomly, with equal probability choose each entry in the matrix to be 0 or 1 and reject the matrices which violate the required marginals. Clearly, every matrix is generated with the same probability.

Both the above methods have been applied to the problem of randomly generating *graphs* with a given degree sequence, i.e., without the bipartiteness restriction. In most cases there is a natural extension of the algorithm to the case when we restrict to bipartite graphs, so for simplicity, we only state the results for graphs.

Jerrum and Sinclair use a Markov chain approach in [45] to generate graphs with given degrees approximately uniformly at random. The Markov chain $\mathfrak{M}_1$ they defined (a weighted version of which will be the basis of our algorithm, see Section 3.6) uses auxiliary states which are graphs with degrees close to the required degrees. They show that if the degree

sequence is *stable*, that is, under small perturbations of r, c, the number of graphs does not change by more than a polynomial in the number of vertices, then $\mathfrak{M}_1$ mixes in polynomial time. The same arguments can be extended to the bipartite graph sampling problem. There, examples of stable degree sequences include regular degrees, when for all i, j, $r_i = c_j$; and "bounded" sequences, when for all i, j, $r_i, c_j \leq \sqrt{n}$. In fact the regularity condition for bipartite graphs can be relaxed to the degrees being regular in one bipartition, but not necessarily the other.

A second Markov chain $\mathfrak{M}_2$ can be defined on the space of binary contingency tables, which chooses two columns and two rows and attempts to add the matrix

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

to the 2×2 matrix defined by the chosen rows and columns. This is known as the Diaconis or "switch" Markov chain. It can be shown that these moves connect the space of all the tables with the given marginals. Kannan, Tetali and Vempala [52] showed that $\mathfrak{M}_2$ mixes in polynomial time for regular sequences; for sequences where $r_i = c_j = d$ and $m = n$, the mixing time of the chain is bounded by $O^*(n^{13}d^{13})$, neglecting logarithmic factors.

Observe that in the bipartite graph corresponding to a 0-1 table, moves of $\mathfrak{M}_2$ pick two vertices in each of the bipartitions, say i_1, i_2 and j_1, j_2 and attempt to add the edges $(i_1, j_1), (i_2, j_2)$ and delete the edges $(i_1, j_2), (i_2, j_1)$. The move succeeds if and only if there are exactly two edges in the induced subgraph, and it "switches" how the vertices are connected. Cooper, Dyer and Greenhill [19] show that the analogous Markov chain on graphs with the required degree sequence that performs switches of pairs of edges mixes in polynomial time. They obtain a mixing time of $O^*(n^9d^{16})$ for the Markov chain.

The first polynomial time algorithm for approximating $N(r, c)$ was given by Jerrum, Sinclair and Vigoda in [47], where they give an FPRAS for estimating the permanent of a 0-1 matrix. It is known by a reduction due to Tutte [86] that computing $N(r, c)$ can be reduced to computing the permanent of a 0-1 matrix. The basis of their algorithm is a Markov chain which can also be used to approximately uniformly generate random bipartite graphs with the given degree sequences. The reduction from a degree sequence of size $n+m$

results in a permanent computation for a matrix of size $O(n^2+m^2)$. Combined with recent improvements for the running time of the permanent algorithm in [8], this results is a mixing time of $O^*(n^{14})$.

The mixing times of the above Markov chains are large polynomials and would have to be tightened much more before there is any hope of practical MCMC algorithms. Next, we discuss two alternative methods of random generation, using the configuration model and importance sampling, both of which usually have very reasonable running times.

3.2.1 The Configuration Model

Consider the following simple algorithm of Bender and Canfield and Bollobás [5, 11] for generating a random regular graph with degree d. Clone each vertex into d copies and take a random perfect matching on the resulting nd vertices. Next, shrink all the clones into one vertex, resulting in a multigraph. It is not difficult to see that all simple graphs can be constructed in an equal number of ways. If we reject the sample if the graph is not simple, this would give the correct distribution over graphs. Unfortunately, the probability of the graph being simple is bounded by $e^{-d^2/4}$, so the sampling algorithm would be polynomial only if $d = O(\ln^{1/2}(n))$.

To go beyond the low degree barrier, Steger and Wormald [82] defined a modification to the configuration model which builds the random matching one edge at a time, avoiding loops and multiple edges. For this algorithm, the difficulty is in showing that the distribution over simple graphs obtained is uniform. Steger and Wormald showed that the distribution was uniform for degrees $d = o(n^{1/28})$. Kim and Vu [54] improved this and demonstrated that for $d = o(n^{1/3})$ the distribution is asymptotically uniform. They conjecture that the distribution is uniform even beyond $o(n^{1/3})$. The running time of this algorithm is only $O(nd^2)$. These arguments can usually be extended to the case of non-regular degrees, though the analysis becomes complicated and there may be restrictions on the deviations of the degrees from the average. This approach for random generation also has the attraction that it can be used to prove properties of random regular graphs, which is not the case for MCMC algorithms.

3.2.2 Importance Sampling

Importance sampling is a general purpose technique that is used to reduce the variance of an estimator for a quantity [58]. Suppose that our objective is to estimate the expectation of a function f on Ω with respect to the distribution π. The expectation is given by

$$\overline{f} = \mathbb{E}_\pi[f] = \sum_{x \in \Omega} f(x)\pi(x)$$

and can be estimated by taking samples $x_1, \cdots, x_n$ drawn according to the distribution π and computing

$$\sum_{i=1}^{n} f(x_i)\pi(x_i).$$

In the limit, as the number of samples $n \to \infty$, the above quantity will converge to the expectation $\overline{f}$. However, the number of samples that must be taken in order to approximate $\overline{f}$ depends on the variance of the estimator.

Instead, suppose we draw the x_i according to another distribution μ on Ω. Then, $\frac{f \cdot \pi}{\mu}$ is an estimator for $\overline{f}$ since

$$\mathbb{E}_\mu \left[\frac{f \cdot \pi}{\mu} \right] = \sum_{x \in \Omega} \frac{f(x) \cdot \pi(x)}{\mu(x)} \mu(x) = \overline{f}$$

A guideline for choosing μ is to put more weight where π is concentrated and f is also large. For example, if μ had the same shape as $f \cdot \pi$, then the variance of the estimator would be zero. In practice, importance sampling with a distribution μ may also be useful if it is not known how to generate samples according to the distribution π, but we still wish to estimate $\overline{f}$.

If the function f over the space of binary contingency tables is uniform and takes value $1/N(r, c)$, then its expectation is just the number of tables $N(r, c)$. In order to estimate $N(r, c)$, Chen et al., [16] proposed a sequential importance sampling algorithm where the columns are filled in sequentially according to a distribution given by asymptotics for the number of 0-1 contingency tables [38] (the resulting distribution over tables is the importance sampling distribution). Subsequently Blanchet [10] analyzed the proposed algorithm and showed that it can be used to generate uniformly random bipartite graphs with

$d_{max} = o(D^{1/4})$ in time $O(D^2)$, where D is the number of edges in the graph. Bayati, Kim and Saberi [4] give a very efficient importance sampling algorithm when the maximum degree is $d_{max} = O(D^{1/4-\varepsilon})$. The expected running time of their algorithm is $O(Dd_{max})$.

Importance sampling is used widely in practice, often without rigorous guarantees and Bezáková et al., in [8], show that examples can be constructed which violate the regularity conditions above, where importance sampling will require exponentially many trials to produce a good estimate of $N(r, c)$.

Our emphasis is on algorithms that are provably efficient for arbitrary degree sequences. More precisely, we are seeking an FPRAS for $N(r, c)$. Until now, the only method known for approximating $N(r, c)$ was by reducing the problem to approximating the permanent [47, 45]. We present a new algorithm for binary contingency tables with arbitrary degree sequences, by directly exploiting the combinatorial structure of the problem. The resulting algorithm is faster than permanent-based algorithms, although it is still far from practical.

3.3 High Level Description of the Algorithm

Our algorithm is very much inspired by the permanent algorithm of Jerrum, Sinclair, and Vigoda [47], but requires an interesting algorithmic twist. The new algorithmic idea relies on a combinatorial property of bipartite graphs satisfying a given degree sequence.

The basis of our algorithm is a Markov chain which walks on bipartite graphs with the desired degree sequence and graphs with exactly two deficiencies. We say a graph has a deficiency at vertices u_i and v_j if they have degree $r(i) - 1$ and $c(j) - 1$, respectively, and all other vertices have the desired degree. The number of graphs with the desired degree sequence might be exponentially fewer than the number of graphs with two deficiencies (see [52] for an explicit example). Thus, we need to weight the random walk defined by the Markov chain so that graphs with the desired degree sequence are "likely" in the stationary distribution.

Let $w(i, j)$ denote the ratio of the number of graphs with the desired degree sequence versus the number of graphs with deficiencies at u_i and v_j. It turns out that given rough approximations to $w(i, j)$, for all i, j, the Markov chain weighted by these ratios quickly

reaches its stationary distribution, and samples from the stationary distribution can then be used to get arbitrarily close estimates of $w(i,j)$. This type of bootstrapping procedure for recalibrating the ratios $w(i,j)$ was central to the algorithm for the permanent.

For the permanent there is an analogous Markov chain on perfect matchings and matchings with at most two unmatched vertices (or holes) where the corresponding ratios, denoted as $\widehat{w}(i,j)$, are the number of perfect matchings divided by the matchings with holes at u_i, v_j. In the case of the permanent, a bootstrapping algorithm for computing the ratios $\widehat{w}$ yields a natural simulated annealing algorithm. Consider an unweighted bipartite graph G that we wish to compute the number of perfect matchings of. In the complete bipartite graph, denoted as G_0, it is trivial to exactly compute the ratios $\widehat{w}(i,j)$ for every i,j. From G_0, we then slightly decrease the weight of edges not appearing in G, constructing a new weighted graph G_1. Using $\widehat{w}$ for G_0 we use the bootstrapping to closely estimate $\widehat{w}$ for G_1. Then we, alternately, decrease (slightly) the weight of non-edges of G creating a new graph G_i, and then use the estimates of $\widehat{w}$ for G_{i-1} to bootstrap $\widehat{w}$ for G_i. A crucial element of this algorithmic approach is that the quantities $\widehat{w}(i,j)$ are trivial to compute in the initial graph, which in this case is the complete bipartite graph.

For contingency tables, what is a starting instance where we can easily estimate the corresponding ratios $w(i,j)$'s? Recall that our final goal is to sample subgraphs of the complete bipartite graph with a given degree sequence. It is not clear that there is some trivial graph which we can use to start the simulated annealing algorithm. This is the key problem we overcome.

We prove that if we construct a graph G^* with the desired degree sequence using a particular Greedy algorithm, then we can estimate the ratios $w(i,j)$ in the weighted complete bipartite graph where edges of G^* have weight 1 and non-edges have sufficiently small (non-zero) weight (call this graph G_0). Our aim is to estimate the ratios when all the edge weights are 1. Once we have estimated the ratios for G_0, they can be used to bootstrap the annealing algorithm in order to compute the ratios for the graphs $G_1, G_2 \cdots$ with larger and larger weights on the non-edges of G^*, until the edge weights are all 1. A high-level outline of the bootstrapping algorithm to compute the ratios is shown in Figure 1 and a

Bootstrapping Algorithm.
Input: Degree sequences r, c and parameters $0 < \varepsilon < 1$.
Output: A $1 \pm \varepsilon$ approximation to the ratios $w(i,j)$.

1. Initialize $k = 0$, $\lambda_0 = \varepsilon(nm)^{-D}$.

2. Let G_k be the weighted graph with edges weights of 1 on the edges of G^* and λ_k on the non-edges.

3. For each pair i, j, compute a $1 \pm \varepsilon$ approximation to $w(i,j)$ for G_0, denoted by $w_0(i,j)$.

4. While $\lambda_k \leq 1$, for each i, j such that there are graphs with the required degree sequence with deficiencies at u_i and v_j,
   - Let $\lambda_{k+1} = \lambda_k \left(1 + \frac{\ln(2^{1/4})}{D}\right)$.
   - Start with a constant factor approximation to w_{k+1} by setting $w_{k+1}(i,j) := w_k(i,j)$.
   - Boost the constant factor approximation to $w_{k+1}(i,j)$ to a $(1 \pm \varepsilon)$ factor approximation by sampling.
   - Set $k := k + 1$.

Figure 1: Bootstrapping algorithm

more precise description can be found in Sections 3.4.3 and 3.7.1 (it may be helpful to skim these before proceeding).

The algorithm to estimate the ratios for G_0 follows from the following property of G^*. For every pair of vertices u_i, v_j, there is a short alternating path between u_i and v_j, or there is no graph with the degree sequence with deficiencies at u_i, v_j. (An alternating path is a path which alternates between edges and non-edges of G^*.) Moreover, the alternating path is of length at most 5, which implies an easy algorithm to count the number of minimum length alternating paths. This in turn gives a polynomial time algorithm for estimating the ratios $w(i,j)$ to within a small relative error if the weights on the non-edges of G^* are sufficiently small. The above combinatorial fact is the main result of this work. Interestingly this combinatorial property fails to hold for many other natural variants of Greedy and max-flow algorithms for constructing a graph with a specified degree sequence.

The algorithmic consequence of our work is an $O((nm)^2 D^3 d_{\max} \log^5(n+m))$ time algorithm to approximately count the number of bipartite graphs with the desired degree sequence, where $D = \sum r(i) = \sum c(j)$ is the total degree (or total number of edges) and

$d_{\max} = \max\{\max_i r(i), \max_j c(j)\}$ is the maximum degree. In the worst case this translates to an $O(n^{11} \log^5 n)$ algorithm for an $n \times n$ matrix since $D = O(n^2)$ and $d_{\max} = O(n)$. Moreover, we can count subgraphs with the given degrees of any input graph, rather than only the complete bipartite graph (see Section 3.4.5 for a discussion of this extension). The following is a precise statement of our main result.

Theorem 3.2. *For any labeled bipartite graph $G = (U \cup V, E)$ where $U = \{u_1, \ldots, u_n\}$ and $V = \{v_1, \ldots, v_n\}$, any degree sequence $r(1), \ldots, r(n); c(1), \ldots, c(m)$, and any $0 < \varepsilon, \eta < 1$, we can approximate the number of labeled subgraphs of G with the desired degree sequence (i.e., u_i has degree $r(i)$ and v_j has degree $c(j)$, for all i, j) in time $O((nm)^2 D^3 d_{\max} \log^5(nm/\varepsilon)$ $\varepsilon^{-2} \log(1/\eta))$ where $D = \sum_i r(i) = \sum_j c(j)$ is the total degree and $d_{\max} = \max\{\max_i r(i),$ $\max_j c(j)\}$ is the maximum degree. The approximation is guaranteed to be within a multiplicative factor $(1 \pm \varepsilon)$ of the correct answer with probability $\geq 1 - \eta$.*

The permanent is a special case of the problem statement in Theorem 3.2 when $m = n$ and for $1 \leq i \leq n$, $r_i = c_i = 1$. In fact, the problem statement is not a generalization of the permanent, but is equivalent to it, since it can be reduced to computing the permanent by a reduction similar to the Tutte reduction. However, as mentioned, the reduction causes a quadratic increase in the size of the instance. The running time of our algorithm when the degrees are all constant is $O(n^7 \log^5 n)$ which matches the running time of the fastest algorithm for the permanent [8].

In Section 3.4 we give the basic definitions and present a high level description of our simulated annealing algorithm. This section aims to motivate our work on the particular variant of the Greedy algorithm we study. We prove our main result about short alternating paths in the graph constructed by a particular variant of Greedy in Section 3.5. In Section 3.6 we analyze the mixing time of the Markov chain which is used in the simulated annealing algorithm. We conclude with the details of the simulated annealing algorithm in Section 3.7. For completeness we sketch the standard reduction from counting to sampling in Section 3.8. Finally we give a breakup of the running time stated in Theorem 3.2 in Section 3.9.

3.4 Preliminaries

3.4.1 Definitions

We use U and V to denote the partitions of vertices of the bipartite graph on $n+m$ vertices. The desired degree sequences are denoted by r and c where $r : U \to \mathbf{N}_0, c : V \to \mathbf{N}_0$, and $\mathbf{N}_0$ is the set of non-negative integers.

For every vertex $v \in V(G)$, let $N(v)$ denote its neighborhood and let $\overline{N}(v) = V \setminus N(v)$ if $v \in U$, and $\overline{N}(v) = U \setminus N(v)$ if $v \in V$. We will use a and u to denote vertices in U and b and v to denote vertices in V.

Definition 3.2.1. *We say that a bipartite graph with partitions U, V* corresponds to the degree sequences $r : U \to \mathbf{N}_0, c : V \to \mathbf{N}_0$ *if* $\deg(a) = r(a)$ *for every $a \in U$ and* $\deg(b) = c(b)$ *for every $b \in V$. A pair of degree sequences r, c is* feasible *if there exists a corresponding bipartite graph.*

Let $\mathcal{P} = \mathcal{P}(r, c)$ be the set of all graphs corresponding to r, c. Recall, our overall aim is to approximate $|\mathcal{P}|$. By a standard reduction [48] this can be done by sampling almost uniformly at random from $\mathcal{P}$.

It is easy to construct a graph with the desired degree sequence, or determine that no such graph exists, using a Greedy algorithm (of which there are many valid variants) or a max-flow algorithm. We study one such variant of Greedy in Section 3.5. Hence, we can assume that r, c defines a feasible degree sequence.

In our simulated annealing algorithm, graphs with the desired degree sequence, except at two vertices, called holes (or deficiencies), will play a central role. This is akin to the role of near-perfect matchings in algorithms for the permanent.

Definition 3.2.2. *Let $u \in U, v \in V$ and let r, c be a pair of degree sequences on U, V. We define degree sequences with holes at u, v as follows:*

$$r^{(u)}(a) := \begin{cases} r(a) & \text{if } a \neq u \\ r(a) - 1 & \text{if } a = u \end{cases} \qquad c^{(v)}(b) := \begin{cases} c(b) & \text{if } b \neq v \\ c(b) - 1 & \text{if } b = v \end{cases}$$

We say that u, v is a pair of feasible holes for the degree sequences *r, c if the pair of sequences $r^{(u)}$, $c^{(v)}$ is feasible.*

Let $\mathcal{N}(u,v)$ be the set of all graphs corresponding to $r^{(u)}, c^{(v)}$ where $u \in U$, $v \in V$, and let $\mathcal{N} = \cup_{u,v}\mathcal{N}(u,v)$. Let $\Omega = \Omega(r,c) = \mathcal{P} \cup \mathcal{N}$.

3.4.2 High-level Description of the Annealing

We give a rough description of the simulated annealing algorithm for binary contingency tables. This is not the novel aspect of the algorithm as it is very much inspired by algorithms for the permanent. Our emphasis in this section is to motivate our main result about the graph constructed by the Greedy algorithm.

The simulated annealing algorithm will consider a sequence of activities on edges of the complete bipartite graph, i.e., for all pairs (x,y) where $x \in U, y \in V$. There will be a subgraph corresponding to the Greedy algorithm which always has activity 1 on each edge, and the other edges will initially have activities $\lambda \approx 0$, and these edges will slowly increase their activities to $\lambda = 1$. More precisely, let G^* denote the graph with the desired degree sequence constructed by Greedy algorithm which is formally defined in Section 3.5. (The details of this graph are not relevant at this stage.) For a positive parameter λ, we define the activity of edge $e = (x,y)$, $x \in U$, $y \in V$, as:

$$\lambda(e) = \begin{cases} 1 & \text{if } e \in E(G^*) \\ \lambda & \text{if } e \notin E(G^*) \end{cases}$$

The activity of a graph $G \in \Omega$ is then defined as:

$$\lambda(G) = \prod_{e \in E(G)} \lambda(e) = \lambda^{|E(G) \setminus E(G^*)|}$$

Finally, the activity of a set of graphs is $\lambda(S) = \sum_{G \in S} \lambda(G)$.

3.4.3 Bootstrapping

A key quantity is the following collection of *ideal weights*:

$$w^*_\lambda(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

These weights are "ideal" in the sense that given close approximations to them, there is a Markov chain which can be used to efficiently generate samples from $\mathcal{P}$ weighted by λ. Thus, using these ideal weights for $\lambda = 1$ we can efficiently sample graphs with the

desired degree sequence. Given rough approximations to the ideal weights w^* (say within a constant factor), samples from the Markov chain can be used to boost these weights into an arbitrarily close approximation of the ideal weights. This is the bootstrapping procedure and the same approach was used for the approximation of the permanent.

Using the bootstrapping procedure (further details of which can be found in Section 3.7.1) to refine rough estimates of the ideal weights we can obtain a simulated annealing algorithm for sampling binary contingency tables. We start with λ_0 close to 0 (specifically with $\lambda_0 = \varepsilon(nm)^{-D}$), where D is the total number of edges. For a particular choice of G^*, it turns out to be possible to compute a $(1 \pm \varepsilon)$ approximation of the ideal weights $w^*_{\lambda_0}$ in a straightforward manner. We will then raise λ slightly to a new value λ_1. For example, suppose we set $\lambda_1 = \left(1 + \ln(2^{1/4})/D\right)\lambda_0$. Then for any graph G, $\lambda_1(G)$ is within a factor of $2^{1/4}$ of $\lambda_0(G)$. This implies that $\lambda_1(\mathcal{P})$ and $\lambda_1(\mathcal{N}(u,v))$ will be within a factor of $2^{1/4}$ respectively of $\lambda_0(\mathcal{P})$ and $\lambda_0(\mathcal{N}(u,v))$. Then, the ideal weights $w^*_{\lambda_0}$ for λ_0 will be a $\sqrt{2}$-approximation to the ideal weights for λ_1. We use the bootstrapping procedure to boost these to get arbitrarily close estimates of $w^*_{\lambda_1}$. We can then continue to alternately raise λ by a factor $\left(1 + \ln(2^{1/4})/D\right)$, and then bootstrap new estimates of the ideal weights. In $O(D^2 \log(mn))$ steps, λ becomes 1 and we will have a suitable approximation of the ideal weights for $\lambda = 1$. It turns out that we can use a more efficient algorithm for updating λ, so that the ideal weights are still constant factor approximations for the successive ideal weights, see Section 3.7.2.

Algorithms for the permanent use a similar simulated annealing approach, but instead start at the complete bipartite graph and slowly remove edges not appearing in the input graph. We instead start at a graph which depends on the desired degree sequence. We then slowly add in non-edges until we reach the complete bipartite graph. In some sense we are doing a reverse annealing.

3.4.4 Estimating Initial Weights

Now we can address how we estimate the ideal weights for λ sufficiently small. Note, $\lambda(\mathcal{P})$ and $\lambda(\mathcal{N}(u,v))$ are polynomials in λ. In particular,

$$\lambda(\mathcal{P}) = \sum_{k=0}^{D} p_k \lambda^{D-k},$$

where p_k denotes the number of graphs corresponding to r, c which contain exactly k edges of G^*. Similarly,

$$\lambda(\mathcal{N}(u,v)) = \sum_{k=0}^{D-1} p_k^{u,v} \lambda^{D-1-k},$$

where $p_k^{u,v}$ is the number of graphs corresponding to $r^{(u)}, c^{(v)}$ which contain exactly k edges of G^*.

For λ sufficiently small, to approximate $\lambda(\mathcal{N}(u,v))$, it suffices to determine the leading non-zero coefficient, i.e., $p_j^{u,v}$ such that $p_k^{u,v} = 0$ for $k > j$. Note that the sum of all the coefficients in the polynomial is at most $(nm)^D$. Then, for $\lambda \leq \varepsilon/(nm)^D$, for some $\varepsilon > 0$, we claim that $x_{u,v} = p_j^{u,v} \lambda^{D-1-j}$ is a $(1+\varepsilon)$ approximation to $\lambda(\mathcal{N}(u,v))$. Formally,

$$\begin{aligned} x_{u,v} \leq \lambda(\mathcal{N}(u,v)) &= x_{u,v} + \sum_{k=0}^{j-1} p_k^{u,v} \lambda^{D-1-k} \\ &\leq x_{u,v} + \lambda^{D-j} \sum_{k=0}^{j-1} p_k^{u,v} \\ &\leq x_{u,v} + \varepsilon \lambda^{D-j-1} \\ &\leq (1+\varepsilon) x_{u,v} \end{aligned}$$

The second to last inequality follows because $\lambda(nm)^D \leq \varepsilon$. The last inequality follows since $x_{u,v} \geq \lambda^{D-1-j}$.

The graph G^* constructed by Greedy has degree sequence r, c, and hence it has exactly one subgraph (G^* itself) that has this degree sequence. Thus, the constant term of $\lambda(\mathcal{P})$ is 1 and we can approximate $\lambda(\mathcal{P})$ by 1. For $u \in U, v \in V$, if $(u,v) \in E(G^*)$, then the subgraph with edges $E(G^*) \setminus (u,v)$ has holes at u, v, and this is the only subgraph with degree sequence $r^{(u)}, c^{(v)}$. In this case we can also approximate $\lambda(\mathcal{N}(u,v))$ by 1.

If $(u,v) \notin E(G^*)$, then there is no subgraph of G^* with holes at u, v, i.e., degree sequence $r^{(u)}, c^{(v)}$ so that $p_{D-1}^{u,v} = 0$. Note, we cannot approximate $\lambda(\mathcal{N}(u,v))$ by 0, since we need an approximation that is close within a multiplicative factor. We instead need to determine a non-zero coefficient of lowest degree in the polynomial. Since $p_k^{u,v}$ is the number of graphs corresponding to $r^{(u)}, c^{(v)}$ with exactly k edges of G^*, the degree of the leading non-zero term in $\lambda(\mathcal{N}(u,v))$ is ℓ where $2\ell+1$ is the length of the shortest alternating path between u and v in G^*. We prove that for our particular choice of G^*, for every u, v there is an alternating path from u to v of length at most 5, or u, v are infeasible holes (in which case we do not need to consider their polynomial). Since these alternating paths are so short, in polynomial time we can simply enumerate all possible such paths, and exactly determine the leading non-zero coefficient, thereby obtaining a good approximation to $\lambda(\mathcal{N}(u,v))$.

This will result in the following theorem, whose proof we present in section 3.5.

Theorem 3.3. *Let r, c be a feasible degree sequence and let $\varepsilon > 0$ and $\lambda \leq \frac{\varepsilon}{(nm)^D}$. There exists a graph G^* (independent of ε and λ) such that for any pair of feasible holes u, v we can compute a weight $w(u,v)$ satisfying*

$$(1-\varepsilon)w(u,v) \leq w_\lambda^*(u,v) \leq (1+\varepsilon)w(u,v).$$

in time $O(nmd_{\max}^2)$. Overall, the construction of G^* together with the computation of $w(u,v)$ for all feasible holes u, v takes time $O((nmd_{\max})^2)$.

3.4.5 Subgraphs of Arbitrary Input Graph

The above high-level algorithm description applies to the contingency tables problem, where we are generating a random subgraph of the complete bipartite graph $K_{n,m}$ with the desired degree sequence. Our approach extends to subgraphs of any bipartite graph $G = (V, E)$.

The general algorithm proceeds as in Section 3.4.2. Thus, regardless of G, we construct G^* using the Greedy algorithm and approximate the initial weights. For non-edges of G^*, their activity is slowly raised from $\lambda \approx 0$ to $\lambda = 1$. At this stage all edges have activity $\lambda = 1$, and thus we can generate random subgraphs of $K_{n,m}$ with the desired degree sequence. Then for non-edges of G, i.e., $(u,v) \notin E$, we slowly lower their activity from $\lambda(u,v) = 1$ to $\lambda(u,v) \approx 0$.

Lowering the activities is analogous to raising the activities, and simply requires that the weights w^* at the previous activities can be used to bootstrap the weights w^* at the new activities. Finally, the algorithm ends with close approximations to the weights w^* for the graph with activities of $\lambda(u,v) = 1$ for all $(u,v) \in E$ and $\lambda(u,v) \approx 0$ for all $(u,v) \notin E$. Therefore, we can generate random subgraphs of G with the desired degree sequence.

3.4.6 Analysis Details

The analysis of the Markov chain underlying the simulated annealing algorithm requires considerable technical work. It combines many of the ideas in the recent works of Cooper, Dyer and Greenhill [19], Kannan, Tetali and Vempala [52], Jerrum, Sinclair and Vigoda [47], and Bezáková et al. [8]. This analysis is contained in Section 3.6. In Section 3.7 we give the details of the simulated annealing algorithm and analyze its running time. In the next section we prove Theorem 3.3.

3.5 Greedy graph

In this section we prove that in the graph constructed by a variant of the greedy algorithm, that we call Greedy, for all u, v, either there is a short alternating path from u to v or there is no graph with holes at u, v. This immediately implies Theorem 3.3. The variant of the greedy algorithm we analyze uses a specific rule to break ties which will be described shortly,

Definition 3.3.1. *Let $G = (U, V, E)$ be a bipartite graph with partitions U, V and edge set E, and let $u \in U$, $v \in V$. We say that there exists an* alternating path from u to v *of length $2k+1$, if there exists a sequence of vertices $u = w_0, w_1, \ldots, w_{2k}, w_{2k+1} = v$ such that $w_{2i} \in U, w_{2i+1} \in V$ and $(w_{2i}, w_{2i+1}) \in E$ for every $i \in \{0, \ldots, k\}$, and $(w_{2i-1}, w_{2i}) \notin E$ for every $i \in \{1, \ldots, k\}$.*

The Greedy algorithm depends on an ordering of the vertices. We need an ordering which is consistent with the degree sequence in the following sense.

Definition 3.3.2. *Fix $c : V \to \mathbf{N}_0$ and let π be a total ordering on V. We say that π is* consistent with *c, if for every b_1, b_2 with $c(b_1) > c(b_2)$, vertex b_1 precedes b_2 in π (i.e.*

$b_1 \prec_\pi b_2$*).*

We now define the Greedy algorithm which is the focus of our analysis. It can be viewed as a recursive procedure which matches the highest degree vertex in U, say x, to $r(x)$ highest degree vertices in V. Then the procedure recurses on the residual degree sequence obtained from the original sequence by setting the degree of x to zero and decrementing the degrees of all its neighbors, until all residual degrees equal zero. However, we need to specify how to break ties when two vertices have the same residual degree. This turns out to be the crucial aspect of our algorithm. For this purpose we introduce an additional parameter of the algorithm, a preference relation π which is initially consistent with c. For the recursive call we use a relation $\hat{\pi}$ induced by π on the residual sequence $\hat{c}$. Here is the formal description of the algorithm:

Procedure GREEDY(r, c, π),

Input: $r : U \to \mathbf{N}_0$, $c : V \to \mathbf{N}_0$ are degree sequences and π is a total ordering on V consistent with c

- Let $G = (U, V, \emptyset)$ be a bipartite graph with partitions U, V and no edges.
- If $\sum_{a \in U} r(a) \neq \sum_{b \in V} c(b)$, return "Sequences not feasible".
- If $\sum_{a \in U} r(a) = 0$, return G.
- Let $x \in U$ be a vertex for which $r(x)$ is maximum (if there is more than one, choose arbitrarily).
- Let $Y \subseteq V$ be the first $r(x)$ vertices in the ordering π.
- If Y contains a vertex of degree 0, return "Sequences not feasible".
- For every $y \in Y$, add the edge (x, y) to G.
- Let $\hat{G} :=$ GREEDY$(\hat{r}, \hat{c}, \hat{\pi})$, where

$$\hat{r}(a) = \begin{cases} r(a) & a \in U \setminus x \\ 0 & a = x \end{cases} \qquad \hat{c}(b) = \begin{cases} c(b) - 1 & b \in Y \\ c(b) & b \in V \setminus Y \end{cases}$$

  and $\hat{\pi}$ is a total ordering on V defined by: $b_1 \prec_{\hat{\pi}} b_2$ if and only if $\hat{c}(b_1) > \hat{c}(b_2)$ or $\hat{c}(b_1) = \hat{c}(b_2)$ and $b_1 \prec_\pi b_2$.
- Add the edges of $\hat{G}$ to G and return G.

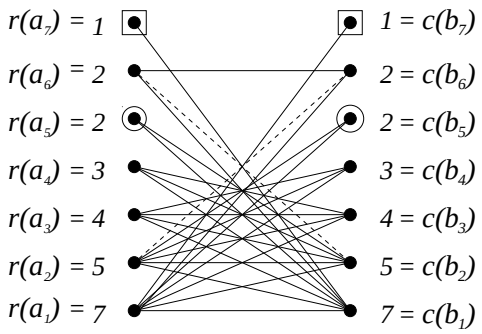

Figure 2: The Greedy graph on the sequence $(1, 2, 2, 3, 4, 5, 7)$, $(1, 2, 2, 3, 4, 5, 7)$.

Now we are ready to present the main combinatorial result. We claim that in the graph constructed by Greedy there is a short (constant-length) alternating path between any two *feasible* holes. One such graph is depicted in Figure 2. It shows a pair of feasible vertices a_5, b_5 and an alternating path $a_5, b_2, a_6, b_6, a_2, b_5$ between them of length 5. Notice that the holes a_7, b_7 are infeasible, thus there is no alternating path between them. In contrast with our main result, in Proposition 3.9 at the end of this section, we construct a family of graphs which require alternating paths of linear length for certain pairs of holes. Each graph in this family is an output of a greedy algorithm which breaks ties arbitrarily.

Theorem 3.4. *Let r, c be a pair of feasible degree sequences and let π be a total ordering on V consistent with c. Let $G = (U, V, E)$ be the graph constructed by* Greedy*(r, c, π). Then for any pair of feasible holes $u \in U$, $v \in V$ in G there exists an alternating path from u to v of length ≤ 5.*

Proof. We prove the theorem by induction on the number of non-zero entries in r. In the base case, there is a single non-zero entry in r. For any pair of feasible holes u, v, the non-zero entry is $r(u)$ and G contains the edge (u, v). Thus u, v forms an alternating path of length 1.

For the inductive hypothesis, assume that the theorem is true for every triple (r', c', π'), where r', c' are feasible degree sequences, r' contains fewer non-zero entries than r, and π' is a total ordering consistent with c'. Let $u \in U$, $v \in V$ be a pair of feasible holes for r, c. Suppose that $U = \{a_1, \ldots, a_n\}$ and the edges adjacent to $a_i \in U$ are added in the i-th

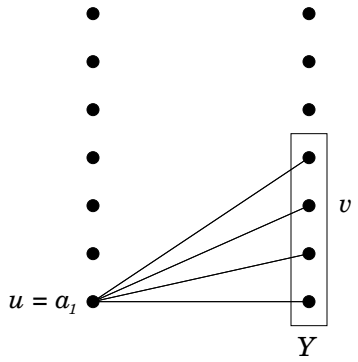

Figure 3: $u = a_1$ and $v \in Y$

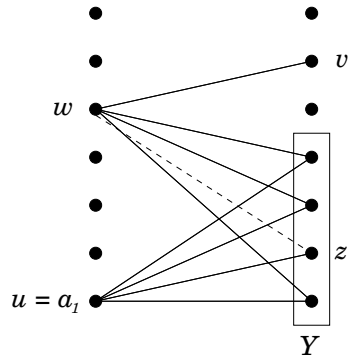

Figure 4: $u = a_1$ and $v \in V \setminus Y$

iteration (or recursive call) of GREEDY(r, c, π). We say that this is the recursive call when a_i is *processed*. In the first recursive call $x = a_1$. Recall that Y denotes the set of a_1's neighbors.

- If $u = a_1$, we construct a short alternating path from u to v as follows (Figures 3,4).
  - If $v \in Y$, then (u, v) is an edge in G and thus u, v forms an alternating path of length 1.
  - If $v \notin Y$, let w be any neighbor of v. Such a neighbor exists since $\deg(v) > 0$, since u, v are feasible holes. Since u is the vertex of the highest degree, $\deg(w) \leq \deg(u)$. Hence there exists a vertex $z \in Y$ which is not a neighbor of w. (If not, then $\deg(w) \geq 1 + |Y| > \deg(u)$, a contradiction.) Then u, z, w, v forms an alternating path of length 3.
- Suppose $u \neq a_1$. Recall that $\hat{r}$, $\hat{c}$ are the reduced degree sequences corresponding to

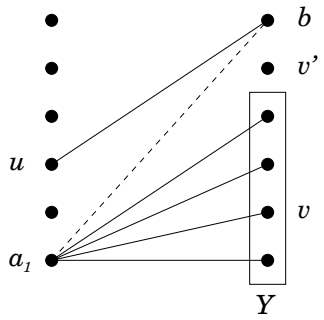

Figure 5: u has a neighbor $b \in V \setminus Y$

the graph $\hat{G}$ obtained from G by removing all edges adjacent to a_1.

- If u, v are also feasible holes for $\hat{r}, \hat{c}$, then we may use the inductive hypothesis to conclude that there exists an alternating path from u to v in $\hat{G}$ of length ≤ 5. Note that the correctness of GREEDY and the assumption that r, c are feasible imply that $\hat{r}, \hat{c}$ are feasible sequences, and $\hat{\pi}$ is consistent with $\hat{c}$ by definition. Hence we can indeed apply induction. Since G and $\hat{G}$ differ only in edges adjacent to a_1 and the path in $\hat{G}$ does not use a_1 (because $\hat{c}(a_1) = 0$), the path is also an alternating path of length ≤ 5 in G.
- Suppose that u, v are not feasible holes for $\hat{r}, \hat{c}$. We use the following claim:

  Claim 3.5. *If u, v are not feasible holes for $\hat{r}, \hat{c}$, then $v \in Y$ is of degree $c(v) = 1$ and there exists $v' \in V \setminus Y$ also of degree $c(v') = 1$.*

  Before we prove the claim, we check what it implies about the existence of a short alternating path between u and v.

  By the claim, $v \in Y$ and there exists another $v' \in V \setminus Y$ with $c(v) = c(v') = 1$. If u has an edge to a vertex $b \in V \setminus Y$, then u, b, a_1, v forms an alternating path of length 3 (see Figure 5). Therefore, we may assume that all of u's neighbors lie in Y. Let r_j, c_j be the residual degree sequences just before the greedy algorithm for r, c starts adding edges adjacent to $u = a_j$ (i.e., r_j, c_j are GREEDY's inputs to the recursive call in which u is processed). In other words, r_j, c_j are the parameters of the j-th recursive call originated from GREEDY(r, c, π). Let b be a vertex of the highest remaining degree in $V \setminus Y$ at the start of the j-th recursive call (notice

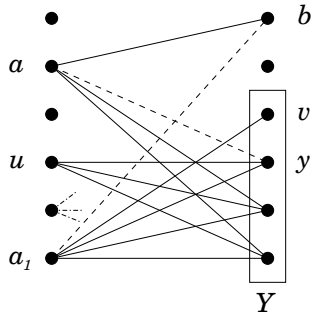

Figure 6: Vertex $b \in V \setminus Y$ of residual degree ≥ 1

that $V \setminus Y$ is nonempty since $v' \notin Y$). The existence of a short alternating path follows from this claim:

Claim 3.6. *If u, v are feasible, then $c_j(b) = 1$.*

The proof of the claim is included in Section 3.5.1. By the claim, for feasible u, v there is a vertex $a \in U$ adjacent to b which is processed after u (see Figure 6). This follows from the fact that all of u's neighbors are in Y. Hence, $\deg(a) \leq \deg(u)$, and therefore, there exists $y \in Y$ which is a neighbor of u but it is not a neighbor of a. (If not, then $\deg(a) \geq 1 + \deg(u)$, a contradiction.) Then u, y, a, b, a_1, v is an alternating path of length 5.

□

3.5.1 Proofs of Claims 3.5 and 3.6 and Theorem 3.3

To finish the proof of the Theorem 3.4, it remains to prove the two claims. We re-state both claims, together with their assumptions.

Theorem 3.7 (Claim 3.5). *Recall that $\hat{r}, \hat{c}$ denote the residual sequences after the greedy algorithm matches the first vertex $a_1 \in U$. Assume that u, v are feasible holes for r, c, where $u \neq a_1$. If u, v are not feasible for $\hat{r}, \hat{c}$, then $v \in Y = N(a_1)$ and there exists a vertex $v' \in V \setminus Y$ such that $c(v) = c(v') = 1$.*

Proof of Claim 3.5. Since u, v are feasible for r, c, there exists a graph with degree sequence $r^{(u)}, c^{(v)}$ (the sequence with holes at u, v). Let $G^{(u,v)}$ be the graph returned by

GREEDY$(r^{(u)}, c^{(v)}, \pi^{(u,v)})$ where $\pi^{(u,v)}$ is the total order obtained from π by repositioning v right after all the vertices of degree $c(v)$ (thus v comes before all vertices of degree $c(v)-1$). We will compare $G^{(u,v)}$ with G to establish conditions under which u, v are feasible for $\hat{r}, \hat{c}$.

Notice that if $v \notin Y$ or if $v \in Y$ but $c(v) > c(b)$ for every $b \in V \setminus Y$, then the neighborhoods of a_1 in G and $G^{(u,v)}$ are identical (because the first $r(a_1) = |Y|$ elements of π and $\pi^{(u,v)}$ are the same). Thus, in this case, $\hat{c}^{(v)}$ (the sequence $\hat{c}$ with hole at v) is identical to the sequence $\widehat{c^{(v)}}$, the residual sequence used in the recursive call of GREEDY$(r^{(u)}, c^{(v)}, \pi^{(u,v)})$. Moreover, since $u \neq a_1$, we have $\hat{r}^{(u)} = \widehat{r^{(u)}}$. By the correctness of the greedy algorithm, $\widehat{r^{(u)}}, \widehat{c^{(v)}}$ are feasible. Therefore, if a_1 has the same neighbors in G and $G^{(u,v)}$, we can conclude that u, v are feasible holes for $\hat{r}, \hat{c}$.

We are left with the case when $v \in Y$ and there exists $v' \in V \setminus Y$ of the same degree $c(v') = c(v)$. We will show that if $c(v) > 1$, the holes u, v are feasible for $\hat{r}, \hat{c}$. This implies the claim.

Suppose $c(v) = c(v') > 1$ for $v \in Y$ and $v' \in V \setminus Y$. Since u, v are feasible for r, c, by symmetry u, v' are also feasible for r, c. However, $v' \notin Y$ and thus, as before, we can conclude that $\hat{r}^{(u)}, \hat{c}^{(v')}$ are feasible and there exists a corresponding graph H. Notice that $\hat{c}^{(v')} = c(v') - 1$ (because $v' \notin Y$ is a hole) and that $\hat{c}^{(v')} = c(v) - 1$ (because $v \in Y$). Since $c(v') = c(v) > 1$, vertices v and v' have the same non-zero degree in H. We will modify H to obtain H', a graph corresponding to $\hat{r}^{(u)}, \hat{c}^{(v)}$. This will prove the feasibility of u, v for $\hat{r}, \hat{c}$. To get H', we need to decrease the degree of v and increase the degree of v' by one while keeping the other degrees intact.

If there is a vertex $a \in U$ which is adjacent to v but not v' in H, we may simply set $H' = H \cup (a, v') \setminus (a, v)$. If there is no such a, then the neighborhood sets of v and v' in H are identical (see Figure 7). If there is a vertex $y \in Y$ for which there exists a neighbor a of v' (and v) in H which is not adjacent to y, then we construct H' as follows. Since $y \in Y$, and $v' \notin Y$ is of the same degree in G as $v \in Y$, by the definition of Y we have $c(y) \geq c(v) = c(v')$. Therefore the degree of y in H is not smaller than the degree of v' in H, i.e. $\hat{c}^{(v')}(y) \geq \hat{c}^{(v')}(v')$. Thus there must exist y's neighbor a' in H which does not neighbor v'. It suffices to set $H' = H \cup \{(a, y), (a', v')\} \setminus \{(a, v), (a', y)\}$ (see Figure 7).

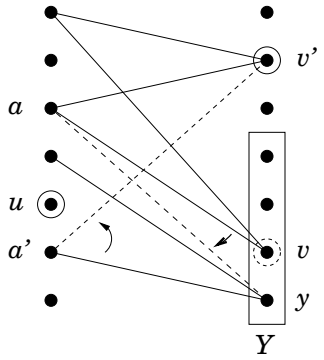

Figure 7: Neighborhoods of v, v' are identical

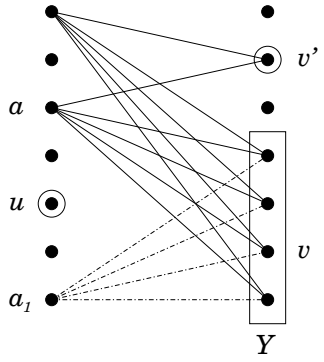

Figure 8: Every $y \in Y$ is adjacent to a

The last case happens when v and v' share the same set of neighbors in H and every $y \in Y$ is adjacent to every neighbor of v' (see Figure 8). By contradiction we will show that this case never happens. Notice that since $r(a) \leq r(a_1)$ for every $a \in U$ and the degree of a in H remains $r(a)$ except for $u \neq a_1$ which decreases by one, the degree of every $a \in U$ in H is upper bounded by $r(a_1)$, i.e. for all $a \in U$, $\hat{r}^{(u)}(a) \leq |Y|$. Let a be any neighbor of v' (by the assumption $c(v') > 1$, the neighborhood set of v' is non-empty). Then a is adjacent to every vertex in $Y \cup \{v'\}$ and therefore $\hat{r}^{(u)}(a) > |Y|$, a contradiction. □

Theorem 3.8 (Claim 3.6). *Let $u \neq a_1$ and $v \in Y$ be such that $c(v) = c(v') = 1$ for some $v' \in V \setminus Y$. Suppose* Greedy*(r, c, π) processes vertices from u in order $a_1, \ldots, a_n$. Let r_i, c_i, π_i be the parameters to the i-th recursive call of* Greedy*, i.e. the call when a_i is processed. Let $u = a_j$. If $c_j(b) = 0$ for all $b \in V \setminus Y$, then u, v are not feasible for r, c.*

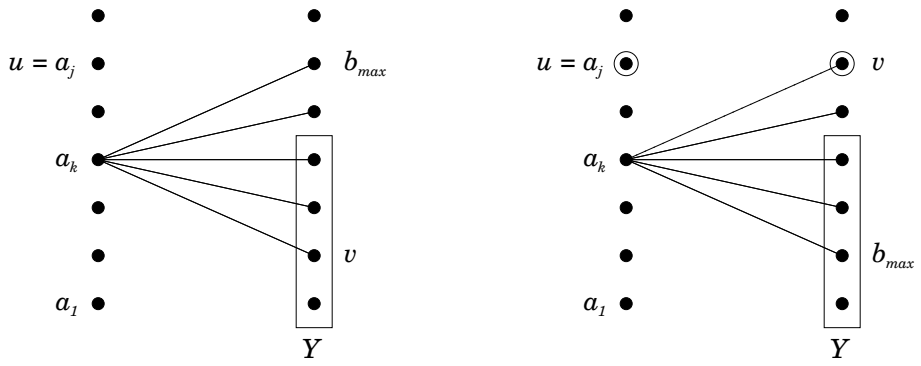

Figure 9: Constructing G and $G^{(u,v)}$ in k-th iteration

Proof of Claim 3.6. Since $c(v) = 1$, for every $b \in V \setminus Y$ we have $c(b) \leq 1$. Let $b_{\max} \in V \setminus Y$ be the vertex of degree 1 ordered last in π (there is at least one such vertex, since $c(v') = 1$). We create π' by swapping the positions of v and $b_{\max}$ in π. Notice that this ordering is consistent with $c^{(v)}$, the sequence obtained from c by decreasing the degree of v by one. We will compare the execution of $\textsc{Greedy}(r, c, \pi)$ and $\textsc{Greedy}(r^{(u)}, c^{(v)}, \pi')$ (see Figure 9). The idea is that both executions will behave similarly, with the roles of v and $b_{\max}$ reversed. Once $\textsc{Greedy}(r, c, \pi)$ gets to matching $b_{\max}$ to a vertex a_k from U, $\textsc{Greedy}(r^{(u)}, c^{(v)}, \pi')$ will attempt to match a_k to a vertex in V of residual degree zero and it will fail. Thus, by the correctness of Greedy, u, v cannot be feasible holes for r, c. We describe the idea in detail below.

Notice that $\textsc{Greedy}(r, c, \pi)$ and $\textsc{Greedy}(r^{(u)}, c^{(v)}, \pi')$ process all vertices a_i for $i < j$ in the same order (assume that if the second execution has multiple choices for x, it chooses the same x as the first execution, if possible). This follows from the fact that the only vertex whose degree is changed is vertex a_j and its degree only decreased. Moreover, the order of processing vertices of U is independent of c (or $c^{(v)}$), assuming the executions do not fail.

Let r_i, c_i, π_i and $r_i^{(u)}, c_i^{(v)}, \pi_i'$ be the parameters of the i-th recursive call originated from $\textsc{Greedy}(r, c, \pi)$ and $\textsc{Greedy}(r^{(u)}, c^{(v)}, \pi')$, respectively. Let $a_k \in U$ be the vertex matched to $b_{\max}$ by $\textsc{Greedy}(r, c, \pi)$. By the assumption of the claim, $c_j(b_{\max}) = 0$, and hence $k < j$, i.e. a_k is processed before $u = a_j$. By induction on i one can verify that for $i < k$ the following hold:

1. $r_i^{(u)}(a) = r_i(a)$ for $a \in U \setminus \{u\}$ and $r_i^{(u)}(u) = r_i(u) - 1$.

2. $c_i^{(v)}(b) = c_i(b)$ for $b \in V \setminus \{v, b_{\max}\}$, $c_i^{(v)}(b_{\max}) = c_i(v)$, $c_i(b_{\max}) = 1$ and $c_i^{(v)}(v) = 0$.

3. Let $supp(f) = \{x \mid f(x) \neq 0\}$.

   - If $v \in supp(c_i)$, then $supp(c_i) = supp(c_i^{(v)}) \cup \{v\}$. The total order π_i restricted to $supp(c_i)$ and the total order π_i' restricted to $supp(c_i^{(v)}) \cup \{v\}$ are identical except that the positions of $v, b_{\max}$ are reversed.
   - If $v \notin supp(c_i)$, then $supp(c_i) \setminus \{b_{\max}\} = supp(c_i^{(v)})$, and the orderings π_i restricted to $supp(c_i)$ and π_i' restricted to $supp(c_i^{(v)}) \cup \{v\}$ are identical except that b_{max} appears in π_i where v appears in π_i'.

4. For every $b \succ_{\pi_i} b_{\max}$, $c_i(b) = 0$. In words, $b_{\max}$ is the last vertex of degree 1 in π_i, if it is indeed of degree 1.

For the induction, the base case $i = 1$ is clear in each case. The case of $i = 2$ also follows in each case, and it can be checked that it holds for the second claim in 3. Now assume the claims are true up to some $2 \leq i \leq k - 2$. We show that they hold for $(i + 1)$.

1. Since only the degree of $a_i \neq u$ decreases in both sequences by $r_i(a_i) = r_i^{(u)}(a_i)$.

2. Since $b_{\max}$ is matched only in the k-th recursive call, $c_{i+1}(b_{\max}) = c_i(b_{\max}) = 1$. Also, $c_{i+1}^{(v)}(v) = c_i^{(v)}(v) = 0$. For $i = 1$, when v is matched by the outermost recursive call of $\textsc{Greedy}(r, c, \pi)$, $b_{\max}$ is matched by $\textsc{Greedy}(r^{(u)}, c^{(v)}, \pi')$, hence $c_{i+1}^{(v)}(b_{\max}) = c_{i+1}(v)$. It follows that $c_{i+1}^{(v)}(b) = c_{i+1}(b)$ for $b \in V \setminus \{v, b_{\max}\}$ since if the statement is true for i and the orderings are identical on vertices of non-zero residual degree other than $v, b_{\max}$ by 3., then exactly the same set of vertices are used by both in the i-th recursive call.

3. This part is true by the definitions $\pi_{i+1} = \hat{\pi}_i$ and $\pi_{i+1}' = \hat{\pi}_i'$ and the fact that 1, 2, and 3 hold for i.

4. This is clear by the definition of the orderings π_{i+1} and π_{i+1}' and the fact that $c_i(b_{\max}) = 1$.

Therefore a_k is joined to the first $r(a_k)$ elements in π_k by $\textsc{Greedy}(r, c, \pi)$, and by 4. the last of them is $b_{\max}$. However, by 3., the execution of $\textsc{Greedy}(r^{(u)}, c^{(v)}, \pi')$ will attempt to connect a_k to v (or some vertex with remaining degree 0), which is impossible since $c_k^{(v)}(v) = 0$. Thus, $\textsc{Greedy}(r^{(u)}, c^{(v)}, \pi')$ fails to construct a corresponding graph, and hence u, v are not feasible holes for r, c. □

This finishes the proof of the Theorem 3.4. As mentioned earlier, Theorem 3.3 is a corollary of the theorem.

Proof of Theorem 3.3. We will prove that for the greedy graph G^* for any $\varepsilon > 0$ and any $\lambda \leq \frac{\varepsilon}{(nm)^D}$ we can efficiently estimate $w^*(u, v) = \lambda(\mathcal{P})/\lambda(\mathcal{N}(u, v))$ (within a $1 \pm \varepsilon$ factor) for every feasible u, v. We have already observed that $\lambda(\mathcal{P})$ and $\lambda(\mathcal{N}(u, v))$ are polynomials in λ. We will show how to approximate $\lambda(\mathcal{P})$ and $\lambda(\mathcal{N}(u, v))$.

First we observe, that each of $\lambda(\mathcal{P})$ and $\lambda(\mathcal{N}(u, v))$ has a positive small-degree coefficient. In particular, the absolute coefficient of $\lambda(\mathcal{P})$ is 1, since G_0 is the only graph corresponding to r, c sharing exactly D edges with G^*. Moreover, by Lemma 3.4, there exists a graph $G' \in \mathcal{N}(u, v)$ which can be obtained from G^* by swapping the edges of an alternating path of length ≤ 5. Therefore G' shares at least $D - 3$ edges with G^* and thus the coefficient of x^d for some $d \leq 2$ in $\lambda(\mathcal{N}(u, v))$ is positive. Moreover,

$$|\mathcal{P}| \leq \binom{nm}{D} \leq (nm)^D,$$

where the first inequality follows from the fact that $\binom{nm}{D}$ counts the number of bipartite graphs (with partitions of sizes n, m) with exactly D edges. Thus,

$$1 \leq \lambda(\mathcal{P}) = 1 + \sum_{k=0}^{D-1} p_k \lambda^{D-k} \leq 1 + \lambda \sum_{k=0}^{D-1} p_k \leq 1 + \lambda(nm)^D \leq 1 + \varepsilon$$

To approximate $\lambda(\mathcal{N}(u, v))$, we will enumerate all graphs corresponding to $r^{(u)}, c^{(v)}$ which share at least $D - 3$ edges with G^*. This can be done by going through all possible alternating paths from u to v of length ≤ 5 and through all alternating cycles of length 4 (corresponding to the case when the symmetric difference of G^* and the graph with the degree sequence $r^{(u)}, c^{(v)}$ consists of the edge (u, v) and a 4-cycle). This way, for fixed u, v,

in time $O(nmd_{\max}^2)$ we can compute

$$x_{u,v} := p_{D-1}^{u,v} + p_{D-2}^{u,v}\lambda + p_{D-3}^{u,v}\lambda^2.$$

Then, $x_{u,v}$ is a $(1+\varepsilon)$-approximation of $\lambda(\mathcal{N}(u,v))$:

$$x_{u,v} \leq \lambda(\mathcal{N}(u,v)) = x_{u,v} + \sum_{k=0}^{D-4} p_k^{u,v}\lambda^{D-1-k} \leq x_{u,v} + \lambda^3 \sum_{k=0}^{D-4} p_k^{u,v} \leq x_{u,v} + \varepsilon\lambda^2 \leq (1+\varepsilon)x_{u,v},$$

where the last inequality follows from $x_{u,v} \geq \lambda^2$ since there exists $j \in [3]$ for which $p_{D-j}^{u,v} \geq 1$.

Therefore in time $O((nmd_{\max})^2)$ we can compute $x_{u,v}$ for every u, v and $1/x_{u,v}$ is a $(1+\varepsilon)$-approximation of $w_\lambda^*(u,v)$. □

Finally, we present a family of graphs, each resulting from a greedy algorithm breaking ties arbitrarily, which for some feasible holes u, v require an alternating path from u to v of linear length. This is in contrast to the result above showing that by choosing the rule for breaking ties in the algorithm carefully, each feasible pair of holes is joined by a constant length alternating path in the resulting graph.

Proposition 3.9. *For every $n \geq 0$, there exist degree sequences r_n, c_n and corresponding graphs G_n such for some feasible pair of holes u, v, there is no alternating path from u to v of length $\leq 2n$ in G_n.*

Proof. Denote the vertices in the two bipartitions by $U = \{u_1, \cdots, u_{n+1}\}$ and $V = \{v_1, \cdots, v_{n+1}\}$. For $n = 0$, let $r_0 = c_0 = (1)$. For $n \geq 1$ let $r_n = c_n = (1, 1, 2, 3, \ldots, n)$. Construct G_n inductively as follows.

---

1. If $n = 0$, set $E(G_n) = \{(u_1, v_1)\}$.

2. If $n = 1$, set $E(G_n) = \{(u_1, v_2), (u_2, v_1)\}$.

3. For $n \geq 2$,

   i) Set $E(G_n) := \bigcup_{v \in U \setminus \{v_1\}} (u_{n+1}, v) \cup \bigcup_{u \in V \setminus \{u_1\}} (u, v_{n+1})$.

ii) The degree requirements of $u_2, u_{n+1}, v_2, v_{n+1}$ are now satisfied. The residual degree sequence is of the form r_{n-2}, c_{n-2} on the vertices $u_1, u_3, \cdots, u_n$ and $v_1, v_3, \cdots, v_n$ if $n \geq 3$, and on u_1, v_1 if $n = 2$.

- If $n \geq 3$, construct the graph G'_{n-2} on $U' = \{u_3, u_1, \cdots, u_n\} = \{u'_1, \cdots, u'_{n-1}\}$ and $V' = \{v_3, v_1, \cdots, v_n\} = \{v'_1, \cdots, v'_{n-1}\}$. (Note that the order of u_1, u_3 and v_1, v_3 are reversed, so that u_n, v_n will be joined to all the vertices of V', U' except v_3, u_3 respectively.)
- If $n = 2$, construct the graph G'_{n-2} on $U' = \{u_1\}$ and $V' = \{v_1\}$.

iii) Set $E(G_n) := E(G_n) \cup E(G'_{n-2})$.

For every $n \geq 1$, u_2, v_2 is a pair of feasible holes. In the base cases, we can check that , the shortest alternating path in G_0 from u_2 to v_2 is of length 1, u_2, v_2, and the shortest alternating path in G_1 from u_2 to v_2 is of length 3, u_2, v_1, u_1, v_2. In G_2, the shortest alternating path from u_2 to v_2 is of length 5, $u_2, v_3, u_1, v_1, u_3, v_2$. Assume the statement is true for all $k < n$ for $n \geq 3$. We claim that the shortest alternating path from u_2 to v_2 in G_n is of length $\geq 2n + 1$. Any alternating path from u_2 to v_2 must begin with the sequence of vertices u_2, v_{n+1}, u_1, and end with v_1, u_{n+1}, v_2, and consist of an alternating path from u_1 to v_1, not using the vertices $u_2, u_{n+1}, v_2, v_{n+1}$. I.e., an alternating path in G'_{n-2} from u'_2 to v'_2. By induction, the path in G'_{n-2} has length $\geq 2n - 3$, and hence any alternating path in G_n from u_2 to v_2 has length $2n + 1$. □

3.6 The Markov Chain

Our Markov chain is analogous to the chain used in algorithms for the permanent [44, 47] and is also an appropriately weighted version of the Markov chain defined in [45]. Recall that $\mathcal{P}$ denotes the set of graphs with the required degree sequence, and $\mathcal{N}(u, v)$ denotes the set of graphs with deficiencies at u, v and $\mathcal{N} = \cup_{u,v}\mathcal{N}(u, v)$. The state space of the chain is $\Omega = \mathcal{P} \cup \mathcal{N}$.

The Markov chain is characterized by an activity $\lambda > 0$ and a *weight function* w :

$U \times V \to \mathbf{R}^+$. The weight of a graph $G \in \Omega$ is defined as

$$w(G) = \begin{cases} \lambda(G) & \text{if } G \in \mathcal{P} \\ \lambda(G)w(u,v) & \text{if } G \in \mathcal{N}(u,v) \end{cases}$$

The transitions $G_t = (U, V, E_t) \to G_{t+1} = (U, V, E_{t+1})$ of the Markov chain MC are:

1. If $G_t \in \mathcal{P}$, choose an edge e uniformly at random from E_t. Set $G' = G_t \setminus e$.

2. If $G_t \in \mathcal{N}(u,v)$, choose an edge $e = (x,y)$ uniformly at random from the multi-set[1] $E_t \cup \{(u,v)\}$ and choose W uniformly from U, V.

   (a) If $e = (u,v)$ and $(u,v) \notin E_t$, let $G' = G_t \cup (u,v)$.

   (b) If $W = U$ and $(u,y) \notin E_t$, let $G' = G_t \setminus (x,y) \cup (u,y)$.

   (c) If $W = V$ and $(x,v) \notin E_t$, let $G' = G_t \setminus (x,y) \cup (x,v)$.

   (d) Otherwise, let $G' = G_t$.

3. With probability $\min\{1, w(G')/w(G_t)\}$, set $G_{t+1} = G'$; otherwise, set $G_{t+1} = G_t$.

It is straightforward to verify that the stationary distribution π of the chain is proportional to the weights w, i.e., for $G \in \Omega$, $\pi(G) = w(G)/Z$ where $Z = \sum_G w(G)$. The main result of this section is to show that if the weights $w(u,v)$ are within a constant factor of their ideal values $w^*(u,v)$, MC mixes in polynomial time.

We continue with some standard definitions before formally stating the main result on the convergence time of the Markov chain. The *total variation distance* between two distributions μ, ν on Ω is given by

$$d_{tv}(\mu,\nu) = \frac{1}{2}\sum_{x\in\Omega} |\mu(x) - \nu(x)|$$

Let P denote the transition matrix of the chain MC, and thus $P^t(x,\cdot)$ denotes the distribution after t steps of the chain, with starting state x. The *mixing time* $\tau_x(\delta)$ of MC starting at state $x \in \Omega$ is defined as

$$\tau_x(\delta) = \min\{t \geq 0 \mid d_{tv}(P^t(x,\cdot), \pi) \leq \delta\}$$

We can now state our main result on the mixing time of MC.

[1] It may be that (u,v) is already in the set of edges E_t.

Theorem 3.10. *Assuming the weight function w satisfies inequality*

$$w^*(u,v)/2 \leq w(u,v) \leq 2w^*(u,v) \tag{4}$$

for every feasible hole pattern $u \in U, v \in V$, then the mixing time of the Markov chain MC started at G is $\tau_G(\delta) = O(nmD^2 d_{max}(\ln(1/\pi(G)) + \log \delta^{-1}))$, where $d_{max} = \max\{\max_i r(i), \max_j c(j)\}$.

3.6.1 Analyzing the mixing time

We will bound the mixing time of MC using the multicommodity flow method, see Chapter 2. To define the flow g, for each $I, F \in \Omega \times \Omega$ we must specify how to route the flow along directed paths going from I to F. As in [47] it is convenient to first define a flow f between all pairs $I \in \Omega$ and $F \in \mathcal{P}$. The flow f can be extended to a flow g between all pairs by routing the flow between a pair of near-perfect tables I, F through a random perfect table. Extending the flow from f to g causes only a modest increase in the congestion. If $\widehat{\rho}(f)$ denotes the congestion restricted to pairs $(I, F) \in \Omega \times \mathcal{P}$, then

$$\rho(g) \leq 2\frac{\pi(\mathcal{N})}{\pi(\mathcal{P})}\widehat{\rho}(f) \leq 8nm\widehat{\rho}(f) \tag{5}$$

The first inequality follows by a result of Schweinsberg, Corollary 3 in [78]. The second inequality follows assuming by (4) that the weights $w(u,v)$ approximate the ideal weights $w^*(u,v)$ up to a factor of 2. Hence it will suffice to define the flow f, which we do in the next section, and bound $\widehat{\rho}(f)$ to bound the mixing time.

3.6.2 Defining the Canonical Flow

We use the flows defined by Cooper, Dyer and Greenhill [19]. Therefore, we follow their notation. Let I, F be perfect or near-perfect contingency tables. We wish to define a set of canonical paths between them by decomposing $H = I \oplus F$ into a sequence of edge-disjoint alternating circuits. Different circuit decompositions will correspond to distinct paths. A circuit of H is a sequence of vertices $w_0, \cdots, w_\ell$, such that $(w_i, w_{i+1}), (w_\ell, w_0) \in E_H$, and each of these edges is distinct, though the vertices may be repeated. The set

$E_H = (E_F \setminus E_I) \cup (E_I \setminus E_F)$. Let the edges in $E_F \setminus E_I$ be red and the edges in $E_I \setminus E_F$ be blue. At each vertex, we will choose a *pairing* of the red and blue edges. A pairing of $I \oplus F$ consists of a pairing at every vertex. Let $\Psi(I, F)$ be the set of all such pairings of $I \oplus F$. For each pairing in $\Psi(I, F)$, we will construct a canonical path from I to F, carrying a total flow of $\pi(I)\pi(F)/|\Psi(I, F)|$. We define the pairings and the corresponding circuit decompositions below.

$\underline{I, F \in \mathcal{P}}$: In this case, at each vertex of H, the red degree is equal to the blue degree. A pairing is constructed by pairing up the red edges at a vertex with the blue edges at each vertex. Hence, if the red (and blue) degree in H of a vertex v is γ_v, $|\Psi(I, F)| = \Pi_v \gamma_v!$. Fix a pairing $\psi \in \Psi(I, F)$. We define an edge disjoint circuit decomposition of H, $\mathcal{C}^\psi = (C_1^\psi, \cdots, C_s^\psi)$, and then define how to "unwind" each circuit to go from I to F. To simplify notation, we omit the superscript henceforth. Let the lexicographically smallest edge in E_H be (w_0, w_1). Choose the (w_i, w_{i+1}) to be the next edge of the circuit if (w_i, w_{i+1}) is paired with (w_{i-1}, w_i) at w_i by ψ (so that we choose (w_1, w_2) if it is paired with (w_0, w_1) by ψ at w_1). This procedure terminates with the circuit $C_1 = w_0, \cdots, w_{k-1}, w_k$ when the edge (w_k, w_0) is paired with (w_0, w_1) at w_0. If $E_H = C_1$, set $\mathcal{C} = (C_1)$. Otherwise, generate C_2 by starting with the lexicographically smallest edge not in C_1. Continue until $E_H = C_1 \cup \cdots \cup C_s$. Then set $\mathcal{C} = (C_1, \cdots, C_s)$. Note that the circuits $C_1, \cdots, C_s$ are edge disjoint by construction and the edges of the circuits are alternately blue and red.

The canonical path p^ψ corresponding to the pairing ψ is defined by the concatenation of the sequence of moves which unwind $C_1, \cdots, C_s$. Let $C_r = a_0, b_0, \ldots, a_\ell, b_\ell$ be a circuit whose lexicographically smallest blue edge is (a_0, b_0). First remove the edge (a_0, b_ℓ). Then for $i = 0, \ldots, \ell - 1$, slide the edge (a_{i+1}, b_i) into (a_i, b_i). Finally, add (a_ℓ, b_ℓ). Since the set of circuits corresponding to different pairings are distinct, the corresponding canonical paths are distinct as well. Set $f(p^\psi) = \pi(I)\pi(F)/|\Psi(I, F)|$ for each path p^ψ.

$\underline{I \in \mathcal{N} \text{ and } F \in \mathcal{P}}$: Suppose $I \in \mathcal{N}(u, v)$. Then, in the graph $H = I \oplus F$, every vertex except u, v is incident with an equal number of red and blue edges. The vertices u, v are each adjacent to one more red edge than the number of blue edges. Let the number of red

edges adjacent to the vertex v in H be γ_v. Define the pairing ψ as follows. At each vertex other than u, v choose a pairing of red and blue edges. At each of u, v choose one red edge which remains unpaired, and pair up the remaining red and blue edges. If $\Psi(I, F)$ is the set of such pairings then, $|\Psi(I, F)| = \prod_{v \in V} \gamma_v!$. For each pairing $\psi \in \Psi(I, F)$ we decompose H into a set of circuits $\mathcal{C}$ and a walk W as follows. Let (w_0, w_1) be the red edge adjacent to $u = w_0$ which is unmatched by ψ. Choose the edge (w_i, w_{i+1}) to be the next edge of the walk if (w_i, w_{i+1}) is paired with (w_{i-1}, w_i) at w_i by ψ. The procedure terminates with the walk W, given by $u = w_0, \cdots, w_\ell = v$ when the red edge $(w_{\ell-1}, w_\ell)$ which is unpaired by ψ at v is paired with $(w_{\ell-2}, w_{\ell-1})$ at $w_{\ell-1}$. If $E_H = W$, we are done, otherwise, start with the lexicographically smallest unused edge of E_H and define the circuits $C_1, \cdots, C_s$. To define the canonical path corresponding to ψ, we unwind the walk W and then the circuits $\mathcal{C}$ in their canonical order. To augment the walk $a_0, b_0, \ldots, a_\ell, b_\ell$, we slide the edges (a_{i+1}, b_i) to (a_i, b_i) for $i = 0, \ldots, \ell - 1$. Then we add the edge (a_ℓ, b_ℓ). Set $f(p^\psi) = \pi(I)\pi(F)/|\Psi(I, F)|$ for each path p^ψ.

This completes the definition of the flow f between pairs I, F in $\Omega \times \mathcal{P}$.

3.6.3 Analyzing the Flow

To prove Theorem 3.10, we analyze the mixing time of the Markov chain which uses the ideal weights $w^*(u, v)$. We will see that the theorem then follows immediately from the condition (4). By the construction of the flow, $\ell(f) \leq D$ and $\ell(g) \leq 2D$. Hence

$$\widehat{\rho}(f) \leq 2D \max_{T=(M,M')} \left\{ \frac{1}{\pi(M)P(M, M')} \sum_{p \ni T} f(p) \right\}$$

where the sum is over paths $p \in \Psi(I, F)$ for I, F in $\Omega \times \mathcal{P}$. Moreover, by the definition of the Markov chain MC, for any transition $T = (M, M')$ which has non-zero probability, $\pi(M)P(M, M') \geq \frac{1}{2D} \min\{\pi(M), \pi(M')\}$. Hence,

$$\widehat{\rho}(f) \leq 4D^2 \max_{T=(M,M')} \left\{ \frac{1}{\min\{\pi(M), \pi(M')\}} \sum_{p \ni T} f(p) \right\} \tag{6}$$

Thus to bound $\widehat{\rho}(g)$, it is enough if we bound

$$\max_{T=(M,M')} \left\{ \frac{1}{\pi(M)} \sum_{p \ni T} f(p) \right\}$$

for every transition (M, M'), since then the bound holds for the reverse transition (M', M) as well.

Let $T = (M, M')$ be any transition of the Markov chain so that $P(M, M') > 0$. Let

$$f_T = \{(I, F) \in \Omega \times \mathcal{P} : \exists\ \psi \in \Psi(I, F) \text{ s.t. } p^{\psi} \ni T\}.$$

We will show that for every transition $T = (M, M')$ of the Markov kernel,

$$\sum_{(I,F) \in f_T} \frac{\pi(I)\pi(F)}{\pi(M)} \frac{|\Psi_T(I, F)|}{|\Psi(I, F)|} = O(d_{max}) \tag{7}$$

By equations (5),(6), and (7), this implies $\rho(g) = O(mnD^2 d_{max})$. This then implies the bound on the mixing time. We divide the proof of (7) into two cases according to the type of transition, in the following two subsections.

We will use the following notation. For $y, u \in U$ and $x, v \in V$ distinct, let

$$\widehat{\mathcal{N}}(y, x, (y, v), (x, u)) = \{M \in \mathcal{N}(y, x) : (y, v), (x, u) \notin M\}$$

Also, let

$$\widehat{\mathcal{P}}(u, v) = \{P \in \mathcal{P} : (u, v) \notin \mathcal{P}\}$$

We also require notation for tables with up to 4 deficiencies. For $y, u \in U$ and $v, x \in V$ (not necessarily distinct), let $\mathcal{N}(y, x, u, v)$ denote the set of tables with deficiencies at u, v, x, y. If any of the vertices y, u, v, x are the same, this means the degree at that vertex is *two* less than its required degree.

Recall, for a transition T, f_T denotes the set of $(I, F) \in \Omega \times \mathcal{P}$ which use T for some of its flow. Let

$$f_T^{u,v} = \{(I, F) \in f_T : I \in \mathcal{N}(u, v)\}$$

3.6.3.1 Transitions of Type 2b or 2c.

Lemma 3.11. *For a transition T of type 2b or 2c,*

$$\sum_{(I,F)\in f_T} \frac{\pi(I)\pi(F)}{\pi(M)} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|} = O(d_{max})$$

To prove the lemma, we use results analogous to the combinatorial lemmas proved in [8], tailored to the canonical flows in this case. We first state and prove the combinatorial results and then show how the lemma follows.

Lemma 3.12. *Let T be a transition between near-perfect tables, so that $M \in \mathcal{N}(u,v)$, $M' \in \mathcal{N}(u',v)$ where $u,u' \in U$, $v \in V$ and $M' = M \setminus (u',x) \cup (u,x)$ for some $x \in V$.*

i)

$$\sum_{\substack{(I,F)\in f_T \\ I\in\mathcal{P}}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\lambda(I)\lambda(F) \leq \sum_{\substack{y\in U \\ (y,v)\neq(u,x)}} \lambda(u,x)\lambda(y,v)\lambda(\widehat{\mathcal{N}}(y,x,(y,v),(x,u)))\lambda(M)$$

ii) For all $s \in U$

$$\sum_{(I,F)\in f_T^{s,v}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\lambda(I)\lambda(F) \leq \lambda(u,x)\lambda(\widehat{\mathcal{N}}(s,x,(x,u)))\lambda(M)$$

iii) For all $s \in U$ and $z \in V$,

$$\sum_{(I,F)\in f_T^{s,z}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\lambda(I)\lambda(F) \leq \sum_{\substack{y\in U \\ (y,v)\neq(u,x)}} \lambda(u,x)\lambda(y,v)\lambda(\widehat{\mathcal{N}}(s,z,y,x,(y,v),(x,u)))\lambda(M)$$

Proof. i) Let $I \in \mathcal{P}$ (blue), and $F \in \mathcal{P}$ (red).

Fix a pairing $\psi \in \Psi_T(I,F)$ (at x, the edge (u,x) is always paired with (u',x)), which gives a decomposition of $I \oplus F$ into red-blue alternating circuits. Since the pairing corresponds to a path from I to F through the transition T, the vertices u,v,x lie on some circuit C. Let y be the vertex adjacent to v in C so that the edge (v,y) is blue (for the first sliding transition in the unwinding of C, y is the vertex u so that the order of vertices on the circuit is v,y,x,u'). Clearly, since (u,x) is an edge of the transition, and (y,v) is the first removed edge, $(y,v) \neq (u,x)$. In M, the circuits ordered before C agree with F, while the circuits after C agree with I.

Define the graph $E_\psi(I,F) = I \oplus F \oplus (M \cup M') \setminus \{(v,y)\}$. Then $E_\psi(I,F) \in \widehat{\mathcal{N}}(y,x,(y,v),(x,u))$. Given M and (u,x), we can recover $M \cup M' = M \cup (u,x)$, and from this, given $E_\psi(I,F)$ and (y,v), we can recover $I \oplus F = (E_\psi(I,F) \cup (y,v)) \oplus (M \cup M')$. We have that $I \cup F = M \cup E_\psi(I,F) \cup \{(u,x),(y,v)\}$, and hence

$$\frac{1}{|\Psi(I,F)|}\lambda(I)\lambda(F) = \frac{1}{|\Psi(I,F)|}\lambda(M)\lambda(E_\psi(I,F))\lambda(u,x)\lambda(y,v) \tag{8}$$

Let $\Psi'(E_\psi(I,F))$ be defined as the set of triples (I',F',ψ') with $\psi' \in \Psi(I',F')$ such that $E'_\psi(I',F') = E_\psi(I,F)$. We claim that $|\Psi'(E_\psi(I,F))| \leq |\Psi(I,F)|$. Assuming this, we claim the lemma follows. If we add up (8) for each $(I,F) \in f_T$ such that $I \in \mathcal{P}$, and each $\psi \in \Psi_T(I,F)$, then on the left hand side, each term $\lambda(I)\lambda(F)$ is counted $|\Psi_T(I,F)|$ times. On the right hand side of (8), for every graph $E \in \widehat{\mathcal{N}}(y,x,(y,v),(x,u))$ such that $E = E_\psi(I,F)$ for some I,F,ψ, the term $\lambda(E)\lambda(M)\lambda(u,x)\lambda(v,y)$ is counted $|\Psi'(E)|$ times. Formally,

$$\begin{aligned}
\sum_{\substack{(I,F)\in f_T \\ I\in\mathcal{P}}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\lambda(I)\lambda(F) &= \sum_{\substack{E\in\widehat{\mathcal{N}}(y,x,(y,v),(x,u)) \\ E=E_{\psi'}(I',F')}} \frac{|\Psi'(E_{\psi'(I',F')})|}{|\Psi(I',F')|}\lambda(u,x)\lambda(y,v)\lambda(E)\lambda(M) \\
&\leq \sum_{\substack{E\in\widehat{\mathcal{N}}(y,x,(y,v),(x,u)) \\ E=E_{\psi'}(I',F')}} \lambda(u,x)\lambda(y,v)\lambda(E)\lambda(M) \\
&\leq \sum_{\substack{y\in U \\ (y,v)\neq(u,x)}} \lambda(u,x)\lambda(y,v)\lambda(\widehat{\mathcal{N}}(y,x,(y,v),(x,u)))\lambda(M)
\end{aligned}$$

Suppose that from E_ψ and T we recover $H = I \oplus F$. Then H has even degree at every vertex. Color an edge of H *green* if it is in M and *yellow* if it is in E_ψ. To bound the number of triples $|\Psi'(E_\psi)|$, we use the fact that the pairing ψ of red and blue edges is a pairing of yellow and green edges at most vertices. A pairing of the yellow and green edges defines a decomposition of $I \oplus F = I' \oplus F'$ into alternating circuits, and further, using the transition T we can recover I' and F'. Thus the number of triples $|\Psi'(E_\psi)|$ is bounded by the number of yellow-green pairings.

In H, every vertex except possibly u,v,x,y has equal yellow and green degree. Two edges of H remain uncolored, (u,x) and (y,v).

a) Suppose $u \neq y, v \neq x$. The vertices y, x have one extra green degree and u, v have one extra yellow degree. To define the pairings at each vertex of H, define the pairings as usual for all vertices except x, y, u, v. At u, think of (u, x) as a green edge, while at x, think of it as a yellow edge. At y, think of (y, v) as a yellow edge, while at v think of it as a green edge. This ensures that there is a yellow-green pairing corresponding to the original red-blue pairing, because we know that in the red-blue pairing at v, the edge (v, y) was paired with a red edge (from F), which is now colored yellow (from E_ψ). Similar arguments can be made at the vertices y, u, x. In addition, at x, we know from T that the edge (u, x) should be paired with (u', x). The number of yellow green pairings is at most $|\Psi(I, F)|$, since if we take into account the "bicolored" edges (v, y) and (u, x), the number of yellow green pairings at each vertex is at most the number of red-blue pairings originally.

b) Suppose that $u \neq y, v = x$. Then in H at every vertex except u, y, the green degree is equal to its yellow degree. Meanwhile, y has an extra green degree and u an extra yellow degree. The pairings at each vertex are constructed as in the previous case, with the same rules for the edges (u, x) and (v, y). Again, it can be seen that the number of yellow-green pairings is the same as the number of red-blue pairings.

c) Suppose $u = y$. Note that this implies $v \neq x$. Every vertex in H except v, x has equal yellow and green degree, but again, coloring the edges (v, y) and (u, x) as before to define the pairings at v, y, x can be used to show that the number of yellow green pairings that we can construct are at most $|\Psi(I, F)|$.

Note that once the bicolored edges are taken into account, in each of the above cases, every vertex of H has green degree equal to yellow degree.

ii) Let $I \in \mathcal{N}(s, v)$ (blue) and $F \in \mathcal{P}$ (red).

Fix a pairing $\psi \in \Psi_T(I, F)$ and define $E_\psi(I, F) = I \oplus F \oplus (M \cup M')$. Then $E_\psi \in \widehat{\mathcal{N}}(s, x, (x, u))$. We claim $|\Psi'(E_\psi(I, F))| \leq |\Psi(I, F)|$. Suppose that from $E_\psi(I, F)$ and T we recover $H = I \oplus F$. Then H has even degree at every vertex except s, v. Color an edge of H

green if it is in M and yellow if it is in $E_\psi(I,F)$. The edge (u,x) of H remains uncolored.

We can show the bound on $|\Psi'(E_\psi(I,F))|$ by exactly the same steps as i), by substituting y in that case, with s here. The only difference is that we no longer take into consideration the edge (s,v) for constructing the pairings, as it is not in H. Notice that here, the fact that s and v have extra red degree will be compensated for by considering (u,x) to be bicolored for the purposes of constructing the yellow-green pairing. This results in s having a green edge and v having a yellow edge remaining effectively unpaired in the yellow-green pairing of H. Note that the red edges which were adjacent to s,v in the circuit being unwound will appear yellow adjacent to v and green adjacent to s in the yellow-green coloring of H. Thus the pairing red-blue ψ does indeed correspond to a yellow-green pairing in H.

iii) Let $I \in \mathcal{N}(s,z)$ (blue) and $F \in \mathcal{P}$ (red).

Fix a pairing $\psi \in \Psi_T(I,F)$. Then, ψ decomposes $I \oplus F$ into a sequence of red-blue alternating circuits and an alternating walk from s to z whose initial and final edges are red. Since T is a transition along the path corresponding to ψ from I to F, u,v,x lie on some circuit C. Let y be the vertex adjacent to v in C so that the edge (v,y) is blue.

Define the graph $E_\psi(I,F) = I \oplus F \oplus (M \cup M') \backslash \{(v,y)\}$. Then $E_\psi(I,F) \in \widehat{\mathcal{N}}(s,z,y,x,(y,v),(x,u))$. Suppose that from $E_\psi(I,F)$ and T we recover $H = I \oplus F$. Then H has even degree at every vertex except s,z. Color an edge of H green if it is in M and yellow if it is in $E_\psi(I,F)$.

First assume that the vertices u,v,s,z,x,y are distinct. Then, every vertex except u,v,s,z,x,y has equal yellow and green degree in H. Both s and z have one green degree more than their yellow degree. This is because they are the endpoints of a walk which has already been unwound, and hence the red edges adjacent to s,z which are left unpaired by ψ both appear in M and are green. Two edges of H remain uncolored, (u,x) and (y,v). The vertices y,x have one extra green degree and u,v have on extra yellow degree. To define the pairings at each vertex of H, define the pairings as usual for all vertices except x,y,u,v. At u, think of (u,x) as a green edge, while at x, think of it as a yellow edge. At

y, think of (y, v) as a yellow edge, while at v think of it as a green edge. The number of yellow green pairings that we can construct are at most $|\Psi(I, F)|$.

Now, in case the vertices are not distinct, there are in all 21 possibilities, taking into account the bipartition the vertices are in and the fact that $(u, x) \neq (v, y)$. However, in each case, suppose that at u, we think of (u, x) as a green edge, while at x, we think of it as a yellow edge and at y, we think of (y, v) as a yellow edge, while at v think of it as a green edge. Then, except at s, z, the yellow degree at every vertex is equal to the green degree in H. At s, z, the green degree exceeds the yellow degree by 1. Hence, the number of yellow green pairings we can construct are at most $|\Psi(I, F)|$. □

Lemma 3.13. *i) Let $u, y \in U$ and $v, x \in V$ such that $(y, v) \neq (u, x)$.*

$$\lambda(u, x)\lambda(\mathcal{N}(u, v)) \sum_{y} \lambda(v, y)\lambda(\widehat{\mathcal{N}}(y, x, (y, v), (x, u))) \leq 6d_{max}\lambda(\mathcal{P})^2$$

ii) Let $s, u \in U$ and $v, x \in V$.

$$\lambda(u, x)\lambda(\mathcal{N}(u, v))\lambda(\widehat{\mathcal{N}}(s, x, (s, u))) \leq 4\lambda(\mathcal{P})\lambda(\mathcal{N}(s, v))$$

iii) Fix $s \in U, z \in V$. Let $u, y \in U$ and $v, x \in V$ such that $(y, v) \neq (u, x)$.

$$\lambda(u, x)\lambda(\mathcal{N}(u, v)) \sum_{y} \lambda(v, y)\lambda(\widehat{\mathcal{N}}(s, z, y, x, (y, v), (x, u))) \leq 2d_{max}\lambda(\mathcal{P})\lambda(\mathcal{N}(s, z))$$

Proof. i) Let $N_1 \in \mathcal{N}(u, v)$ and $N_2 \in \bigcup_y \widehat{\mathcal{N}}(y, x, (y, v), (x, u))$. We will consider the symmetric difference $N_1 \oplus N_2$ and define a modified (multi)graph $H'(N_1, N_2)$ and a set of pairings of H', $\Psi(N_1, N_2)$. From N_1, N_2 and a pairing in $\Psi(N_1, N_2)$ we construct graphs $N_3 \in \mathcal{P}$ and $N_4 \in \mathcal{P}$, and a pairing of $H'(N_3, N_4)$. The graphs will satisfy $N_1 \cup N_2 \cup \{(u, x), (v, y)\} = N_3 \cup N_4$ where the union takes into account multiplicities. Given N_3, N_4, and the pairing of $H'(N_3, N_4)$ we will be able to reconstruct N_1, N_2 and the original pairing given an additional $3 \times [d_{max}] \times \{0, 1\}$ amount of information. We then show that the number of pairings of $H'(N_3, N_4)$ is at most the number of pairings of $H'(N_1, N_2)$, and this implies the claimed inequality.

First assume the vertices u, y and v, x are distinct. Consider the symmetric difference $H = N_1 \oplus N_2$ so that the edges from N_1 are blue, and those from N_2 are red. Then, x, y

each have blue degree 1 more than their red degree while u, v have red degree one more than their blue degree. We will fix a pairing of the red and blue edges as follows. The graph H may or may not contain the edges $(u, x), (v, y)$ depending on whether or not they are present in N_1. If either is present, it is colored blue. To define pairings at each vertex, first add the *uncolored* edges $(u, x), (v, y)$ to H, i.e. let $H' = H \cup \{(u, x), (v, y)\}$ and retain the color of all the edges from H. Note, H' may have double edges. To define the pairings at u, v, think of both the uncolored edges $(u, x), (v, y)$ as blue and define an exact pairing of the red and blue edges. At y, x, we think of the uncolored edges as red and define an exact pairing of the red and blue edges at these vertices such that the red edge (x, u) (resp. (y, v)) is always paired with the blue (x, u) (resp. (y, v)), if it is present, for example, see Figure 10 *a*). For all other vertices, the red degree is equal to the blue degree, and we pair them up. Call this pairing in H' ψ, and let the set of pairings be $\Psi(N_1, N_2)$.

Then ψ defines a decomposition of H' into alternating circuits of even length. These are shown for the example in Figure 10 *b*). The idea of the map is to traverse the circuits and put edges alternately in $N_3 \in \mathcal{P}$ and $N_4 \in \mathcal{P}$. For each circuit not containing the uncolored edges, put edges alternately in N_3 and N_4 making the convention that the blue edges are put into N_4 and the red edges into N_3. There is only one way for a circuit to contain the uncolored edges; such a circuit must contain both. (There cannot be two distinct circuits each containing one uncolored edge, since the circuit is even, and the edges alternate red-blue, ignoring the uncolored edge). For the circuit containing the uncolored edges, put edges alternately in N_3 and N_4 starting with the uncolored (y, v) in N_3. Edges which are in both N_1, N_2 are added to both N_3, N_4. Note that this set never includes the edges $(u, x), (v, y)$, so we never attempt to add them to N_3 or N_4 twice. By the definition of ψ, if H' has any double edges, then both copies do not go into the same graph since they appear consecutively in a circuit or walk. Then, $N_3 \in \mathcal{P}$ and $N_4 \in \mathcal{P}$. The bit b of the map is set to 1 if the blue edge (v, y) was present in $N(u, v)$ and was traversed *after* the uncolored (v, y). The set $[d_{max}]$ is used to encode the vertex y.

To invert the map, consider two tables, $N_3 \in \mathcal{P}$ and $N_4 \in \mathcal{P}$, and their symmetric difference $N_3 \oplus N_4$. If the pairing ψ of H' was known, we claim we can recover N_1, N_2

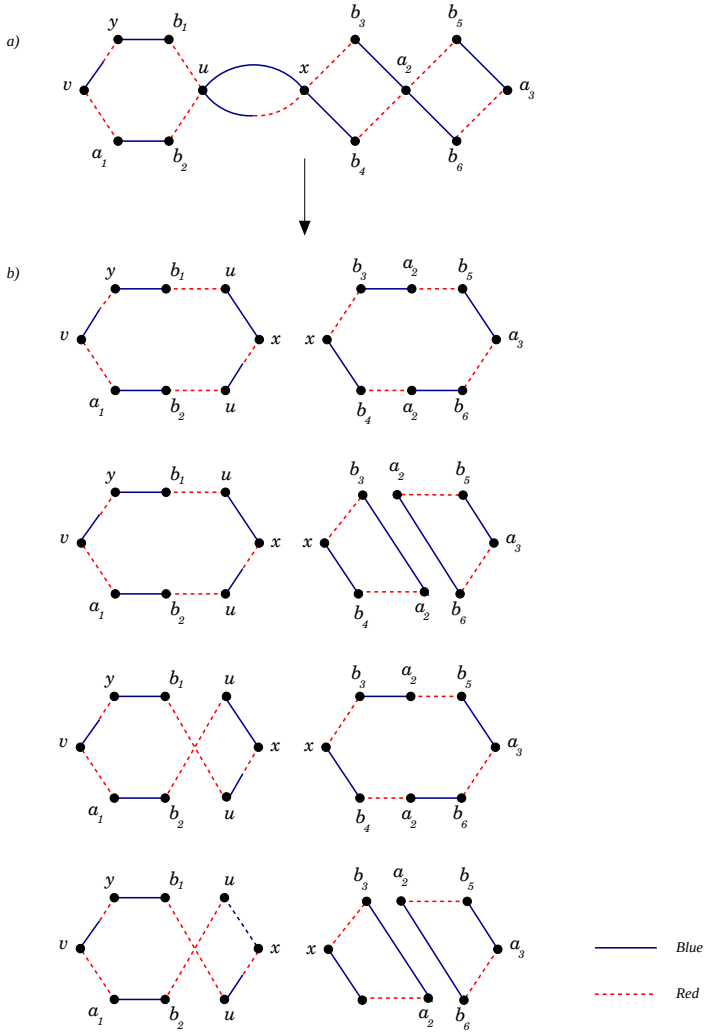

Figure 10: *a*) The graph H', *b*) Decompositions of H' into alternating circuits

uniquely. We can reconstruct H' as follows. If (u, x) (resp. (y, v)) appears in $N_3 \oplus N_4$, then it was not present in N_1, and hence appears once in H'. On the other hand, if (u, x) (resp. (y, v)) does not appear in $N_3 \oplus N_4$, then it was present in N_1, and hence appears as a double edge in H'. Thus, we can reconstruct H' from $N_3 \oplus N_4$ by adding in two copies of the edge if necessary. If ψ was known, we could partition the edges of H' into N_1, N_2 as follows. The pairing ψ determines the decomposition of H' into alternating circuits. There will be exactly one circuit which contains the edges (u, x) and (v, y). For the other circuits and the walk, we put the edges coming from N_3 into N_2, and the edges from N_4 into N_1. If there is a circuit containing $(u, x), (v, y)$, proceed as follows. If (y, v) does not appear as a double edge, start with the edge in the circuit after (y, v), and put edges alternately in N_1 and N_2, and also skipping one copy of the edge (u, x). If (y, v) appears twice in the circuit, we can determine which copy was the uncolored one by looking at the bit b. If b is 1, it is the first one, and if b is 0, it is the second one. Proceed as before, start with the edge after the uncolored (v, y), and assign edges alternately to N_1 and N_2, and also skip one copy of (u, x). Finally, put all other common edges of N_3, N_4 into both N_1 and N_2.

Color the edges of H' green if they come from N_3, and yellow if they come from N_4. Since we do not have the pairing ψ of H', instead, we use the fact that a pairing of the original red and blue edges is a pairing of the yellow and green edges of H' at all the vertices. We know that at x, y if there is a double edge, they are colored yellow, and green, and must be matched. Also, at u, v the double edges are not paired. Hence the number of valid yellow-green pairings in H' is at most as the number of original red-blue pairings $|\Psi(N_1, N_2)|$, and so there cannot be too many initial pairs of tables mapping to N_3, N_4. This is illustrated with an example in Figure 11.

In the case that the vertices are not distinct, there 2 other possibilities :

a) $u = y, v \neq x$

b) $u \neq y, v = x$

The two cases are symmetric, except that in the second case, we have to keep track of y so we only give the argument for a). Let $N_1 \in \mathcal{N}(u, v)$ (blue) and $N_2 \in \widehat{\mathcal{N}}(u, x, (u, v), (x, u))$

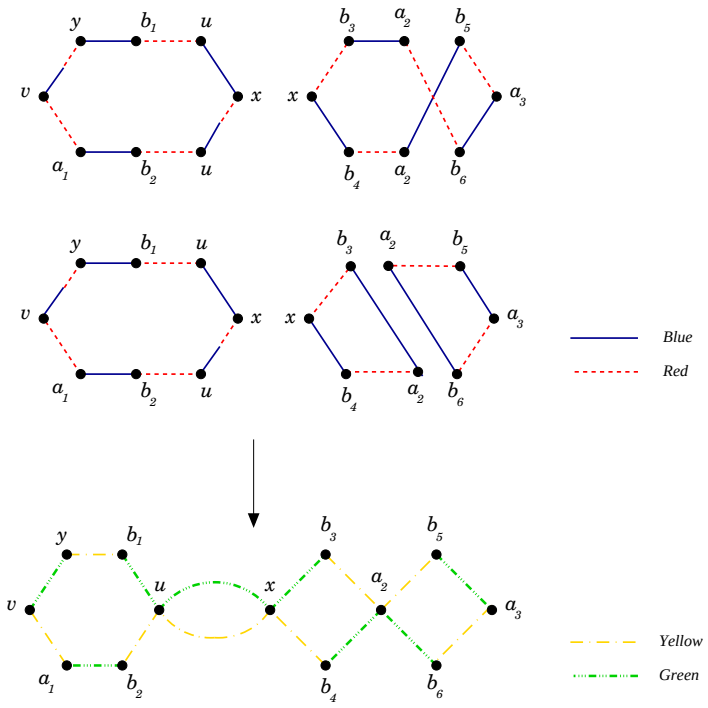

Figure 11: Graphs N_1, N_2 which map to a pair $N_3, N_4 \in \mathcal{P}$

(red). Then, in $H = N_1 \oplus N_2$, the vertex u has equal red and blue degree, while v has 1 extra red degree and x has 1 extra blue degree. Also, if the edges (u, v) or (u, x) are present, they are blue. Construct H' as before, and define the pairings as before. Thus at x we think of the uncolored (u, x) as red (and pair it with the blue (u, x) if it is present), while at u we think of it as blue. At v, think of the uncolored (u, v) as blue, while at u, we think of it as red, and always pair it with the blue (u, v) if it is present. The remainder of the argument is the same as when the vertices are distinct.

Now, given which case we are in (there are 3 cases in all), and $N_3, N_4 \in \mathcal{P}$, and the vertex y, the inequality follows since the number of yellow-green pairings is at most $|\Psi(N_1, N_2)|$.

ii) Let $N_1 \in \mathcal{N}(u, v)$ and $N_2 \in \bigcup_y \widehat{\mathcal{N}}(s, x, (x, u))$. As before, we will define a modified (multi)graph $H'(N_1, N_2)$ and a set of pairings of H', $\Psi(N_1, N_2)$. From N_1, N_2 and a pairing in $\Psi(N_1, N_2)$ we will construct graphs $N_3 \in \mathcal{N}(s, v)$ and $N_4 \in \mathcal{P}$, and a pairing of $H'(N_3, N_4)$. The graphs will satisfy $N_1 \cup N_2 \cup (u, x) = N_3 \cup N_4$, taking into account the multiplicity of the edges. Given N_3, N_4, and the pairing of $H'(N_3, N_4)$ we will be able to reconstruct N_1, N_2 and the original pairing given a constant amount of additional information. We then show that the number of pairings of $H'(N_3, N_4)$ is at most the number of pairings of $H'(N_1, N_2)$, and this implies the claimed inequality.

First assume the vertices s, x, u, v are distinct. Consider the symmetric difference $H = N_1 \oplus N_2$ so that the edges from N_1 are blue, and those from N_2 are red. In H, s, x each have blue degree 1 more than their red degree while u, v have red degree one more than their blue degree. Let $H' = H \cup \{(u, x)\}$ and retain the color of all the edges from H leaving the new edge (u, x) uncolored. Define a pairing ψ of H' as follows. At s, choose one blue edge which remains unpaired, and pair up the remaining red and blue edges. At v, choose one red edge to remain unpaired and pair up the others. To define the pairing at u, think of the uncolored edge (u, x) as blue and define an exact pairing of the red and blue edges. At x, we think of the uncolored edge as red and define an exact pairing of the red and blue edges at these vertices such that the red edge (x, u) is always paired with the blue (x, u), if it is present. For all other vertices, the red degree is equal to the blue degree, and we pair

them up as usual. Let the set of such pairings be $\Psi(N_1, N_2)$.

Then ψ defines a decomposition of H' into circuits of even length and a walk of odd length from s to v whose initial edge is blue, final edge is red, and contains the uncolored edge (u, x), since the length of the walk is odd. The idea of the map is the same as in the previous case, to put edges from the circuits and walks alternately in N_3 and N_4 with the same color conventions as before. When we traverse the walk, starting with N_4 we put each edge alternately into N_3 and N_4. Then, $N_3 \in \mathcal{N}(s, v)$ and $N_4 \in \mathcal{P}$.

To invert the map, consider the symmetric difference $N_3 \oplus N_4$. If the pairing ψ of H' was known, we can recover N_1, N_2 uniquely. We can reconstruct H' as follows. If (u, x) appears in $N_3 \oplus N_4$, then it was not present in N_1, and hence appears once in H'. On the other hand, if (u, x) does not appear in $N_3 \oplus N_4$, then it was present in N_1, and hence appears as a double edge in H'. Thus, we can reconstruct H' from $N_3 \oplus N_4$ by adding in two copies of the edge if necessary. If ψ was known, we could partition the edges of H' into N_1, N_2. The pairing ψ determines the decomposition of H' into circuits and a walk of odd length. The walk contains (all the copies of) the edge (u, x) since the circuits are all even length. For each circuit as well as the walk, we put the edges coming from N_3 into N_2, and the edges from N_4 into N_1. Put all other common edges of N_3, N_4 into both N_1 and N_2.

Color the edges of H' green if they come from N_3, and yellow if they come from N_4. Since we do not have the pairing ψ of H', instead, we use the fact that a pairing of the original red and blue edges is a pairing of the yellow and green edges of H' at all the vertices. We know that at x if there is a double edge, they are colored yellow, and green, and must be matched. Also, at u, the double edges are not paired.

If the vertices are not distinct, there are 3 cases:

a) $u = s, v \neq x$. In this case, add the uncolored (u, x) to H. At u, think of the uncolored edge as blue, and fix a pairing by leaving out one blue edge. At x, fix the pairing by always pairing the uncolored/red (u, x) with the blue copy of (u, x) if it is present. Now in H', v has one extra red degree, while u has an extra blue degree taking into account the uncolored (u, x). Hence the pairing determines an alternating walk from s to v with initial edge blue, and final edge red, containing the uncolored edge, and alternating

circuits. Put the edges along the walk alternately in N_3 and N_4. Thus we ensure N_3, N_4 each contain at most one copy of (u, x). Inverting the map is easy if the pairing of H' is known, and we can bound the number of yellow-green pairings as before.

b) $u \neq s, v = x$. In this case the argument is similar to the above, except that in H', to fix a red-blue pairing, think of the the uncolored edge (u, x) as blue at u and red at x.

c) $u = s, v = x$. This case becomes trivial. Let $N_1 \in \mathcal{N}(u, v)$ and $N_2 \in \widehat{\mathcal{N}}(u, v, (v, u))$. Set $N_3 = N_1$ and $N_4 = N_2 \cup (u, x)$. Clearly, $N_4 \in \mathcal{P}$, $N_3 \in \mathcal{N}(s, v)$, and the map is easily invertible.

Hence the number of yellow-green pairings in H' is at most the number of original red-blue pairings $|\Psi(N_1, N_2)|$. Given which case we are in (which is a factor of 4), the inequality follows.

iii) Let $N_1 \in \mathcal{N}(u, v)$ and $N_2 \in \bigcup_y \widehat{\mathcal{N}}(s, z, y, x, (y, v), (x, u))$. As before, we define a modified (multi)graph $H'(N_1, N_2)$ and a set of pairings of H', $\Psi(N_1, N_2)$. From N_1, N_2 and a pairing in $\Psi(N_1, N_2)$ we construct graphs $N_3 \in PP$ and $N_4 \in \mathcal{P}$, and pairing of $H'(N_3, N_4)$. The graphs will satisfy $N_1 \cup N_2 \cup \{(u, x), (v, y)\} = N_3 \cup N_4$, taking into account multiplicity of edges. Given N_3, N_4, and the pairing of $H'(N_3, N_4)$ we will be able to reconstruct N_1, N_2 and the original pairing given an additional $[d_{max}] \times \{0, 1\}$ amount of information. We then show that the number of pairings of $H'(N_3, N_4)$ is at most the number of pairings of $H'(N_1, N_2)$, and this implies the claimed inequality.

First assume the six vertices are distinct. Consider the symmetric difference $H = N_1 \oplus N_2$ so that the edges from N_1 are blue, and those from N_2 are red. Then, s, z, x, y each have blue degree 1 more than their red degree while u, v have red degree one more than their blue degree. Define H' as in i). We will fix a pairing ψ of the red and blue edges in H' as follows. At s, z, choose one blue edge which remains unpaired, and pair up the remaining red and blue edges. The pairing at all other vertices is defined as in i). Let the set of such pairings be $\Psi(N_1, N_2)$.

Then ψ defines a decomposition of H' into circuits of even length and a walk of odd

Table 1: Enumeration of 21 cases

	$u \neq y, v \neq x$	$u = y, v \neq x$	$u \neq y, v = x$
$s = u$	• $z = v$ • $z = x$ • $z \neq v, x$	• $z = v$ • $z = x$ • $z \neq v, x$	• $z = v = x$ • $z \neq v, x$
$s = y$	• $z = v$ • $z = x$ • $z \neq v, x$	Counted in the case $s = u$	• $z = v$ • $z \neq v, x$
$s \neq u, y$	• $z = v$ • $z = x$ • $z \neq v, x$	• $z = v$ • $z = x$ • $z \neq v, x$	• $z = v$ • $z \neq v, x$

length from s to z whose initial and final edges are blue. Traverse the walk, and starting with N_4 put each edge alternately into N_3 and N_4. The rest of the edges of H' are partitioned as in i). Then, $N_3 \in \mathcal{N}(s, z)$ and $N_4 \in \mathcal{P}$. The bit b of the map is set to 1 if the blue edge (v, y) was present in $N(u, v)$ and was traversed *after* the uncolored (v, y). The set $[d_{max}]$ is used to encode the vertex y.

We can reconstruct H' using the symmetric difference $N_3 \oplus N_4$ and the edges $(u, x), (v, y)$ exactly as in i). Since we do not have the pairing ψ of H', to recover N_1, N_2 we use the fact that a pairing of the original red and blue edges is a pairing of the yellow and green edges of H' at all the vertices. Color the edges of H' green if they come from N_3, and yellow if they come from N_4. We know that at x, y if there is a double edge, they are colored yellow, and green, and must be matched. Also, double edges at u or v are never paired. Hence the number of yellow-green pairings in H' is at most the number of original red-blue pairings $|\Psi(N_1, N_2)|$.

Lastly, we handle the various cases in which the vertices are not distinct. There are 21 possible distinct cases depending on which of the vertices u, y, v, x, s, z are the same. These are enumerated in Table 1 for completeness.

In each of these cases, when we add the uncolored edges $(u, x), (v, y)$, so that we think of them as red at y and x and blue at u and v, in order to define the pairing of H', we find that each of s, z have 1 extra blue degree, and at every other vertex, the blue degree equals the red degree. Then, we can restrict to the same kinds of pairings as in the case when the vertices are distinct, and the lemma follows, once we factor in which of the 22 cases we are

in, and the vertex y is known in each case.

Note that in some of the cases, the map can be defined by adding the edges $(u,x),(v,y)$ to the tables, but the map can be defined in this way through the pairings as well. Since the map is defined in the same way in each case, we do not even need to retain information about which of the 22 cases we are in, and the bound now follows. □

With the above inequalities in hand, the proof of Lemma 3.11 is a matter of plugging them in to the expressions which bound the congestion through a transition.

Proof of Lemma 3.11. When T is a transition of type 2b or 2c the flow through T can come from 3 sources. First, due to being on an alternating circuit between pairs of perfect tables. Second, the congestion due to being on the augmenting walk between a near-perfect table and a perfect table. Lastly, due to being on an alternating circuit between a near-perfect table and a perfect table. The proof of the bound is similar in each of these cases, and the bottleneck is the third case. In each case, let $T = (M, M')$, where $M \in \mathcal{N}(u,v)$ and $M' \in \mathcal{N}(u',v)$, with x as the pivot vertex, so that $M' = M \cup (u,x) \setminus (u',x)$.

We can bound the congestion due to $(I,F) \in \mathcal{P} \times \mathcal{P}$ through T as follows.

$$\sum_{\substack{(I,F)\in f_T \\ I\in\mathcal{P}}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\pi(I)\pi(F)}{\pi(M)}$$

$$= \frac{1}{w(\Omega)} \sum_{\substack{(I,F)\in f_T \\ I\in\mathcal{P}}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\lambda(I)\lambda(F)}{\lambda(M)}\frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{P})}$$

$$\text{(By Lemma 3.12, i)} \quad \leq \quad \frac{1}{w(\Omega)} \sum_{\substack{y \\ (y,v)\neq(u,x)}} \lambda(u,x)\lambda(y,v)\frac{\lambda(\widehat{\mathcal{N}}(y,x,(y,v),(x,u)))\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{P})}$$

$$\text{(By Lemma 3.13, i)} \quad \leq \quad \frac{6d_{max}\lambda(\mathcal{P})}{w(\Omega)}$$

$$\leq \quad \frac{6d_{max}}{nm}$$

Next, we bound the congestion due to $(I,F) \in \mathcal{N} \times \mathcal{P}$ through T when T is on the alternating walk. Note that in this case at least one of the holes of I, v is the same as a

hole of M.

$$\begin{aligned}
&\sum_{s\in U}\sum_{(I,F)\in f_T^{s,v}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\pi(I)\pi(F)}{\pi(M)} \\
= \quad & \frac{1}{w(\Omega)}\sum_s \frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(s,v))}\sum_{(I,F)\in f_T^{s,v}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\lambda(I)\lambda(F)}{\lambda(M)}
\end{aligned}$$

$$\begin{aligned}
\text{(By Lemma 3.12, ii)} \quad &\leq \quad \frac{1}{w(\Omega)}\sum_s \lambda(u,x)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(s,v))}\lambda(\widehat{\mathcal{N}}(s,x,(s,u))) \\
\text{(By Lemma 3.13, ii)} \quad &\leq \quad \frac{4n\lambda(\mathcal{P})}{w(\Omega)} \\
&\leq \quad \frac{4}{m}
\end{aligned}$$

Lastly, we bound the congestion due to $(I,F)\in\mathcal{N}\times\mathcal{P}$ when T is on an alternating circuit.

$$\begin{aligned}
&\sum_{s\in U,z\in V}\sum_{(I,F)\in f_T^{s,z}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\pi(I)\pi(F)}{\pi(M)} \\
= \quad & \frac{1}{w(\Omega)}\sum_{s,z} \frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(s,z))}\sum_{(I,F)\in f_T^{s,z}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\lambda(I)\lambda(F)}{\lambda(M)} \\
\leq \quad & \frac{1}{w(\Omega)}\sum_{s,z} \lambda(u,x)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(s,z))}\sum_{\substack{y \\ (y,v)\neq(u,x)}} \lambda(y,v)\lambda(\widehat{\mathcal{N}}(s,z,y,x,(y,v),(x,u)))
\end{aligned}$$

(By Lemma 3.12, iii)

$$\leq \quad \frac{2d_{max}nm\lambda(\mathcal{P})}{w(\Omega)}$$

(By Lemma 3.13, iii)

$$\leq \quad 2d_{max}$$

Adding the congestion from each of these sources, the congestion through a sliding transition T is bounded by $O(d_{max})$. □

3.6.3.2 Transitions of Type 2a or 1.

Lemma 3.14. *For a transition T of type 2a or 1,*

$$\sum_{(I,F)\in f_T} \frac{\pi(I)\pi(F)}{\pi(M)}\frac{|\Psi_T(I,F)|}{|\Psi(I,F)|} = O(1) \tag{9}$$

To prove the lemma, we again tailor the corresponding combinatorial inequalities of [8] for the case of canonical flows. We first state and prove the combinatorial results and then show how the lemma follows.

Lemma 3.15. *Let $T = (M, M')$ be a transition between a near-perfect table in $\mathcal{N}(u, v)$ and a perfect table, so that the edge (u, v) is either deleted or added. Let N be the near-perfect table of M and M'. Then,*

i)

$$\sum_{\substack{(I,F)\in f_T \\ I\in\mathcal{P}}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\lambda(I)\lambda(F) \leq \lambda(u,v)\lambda(\widehat{\mathcal{P}}(u,v))\lambda(N)$$

ii) For all $s \in U$ and $z \in V$,

$$\sum_{(I,F)\in f_T^{s,z}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\lambda(I)\lambda(F) \leq \lambda(u,v)\lambda(\widehat{\mathcal{N}}(s,z,(u,v)))\lambda(N)$$

Proof. i) Let $I \in \mathcal{P}$ (blue) and $F \in \mathcal{P}$ (red).

Fix a pairing $\psi \in \Psi_T(I, F)$. Define the graph $E_\psi(I, F) = I \oplus F \oplus (M \cup M')$. Then, $E_\psi(I, F) \in \widehat{\mathcal{P}}(u, v)$. Given $E_\psi(I, F), T$ and ψ, we can recover I and F. Since $I \cup F = N \cup E(I, F) \cup (u, v)$,

$$\frac{1}{|\Psi(I,F)|}\lambda(I)\lambda(F) = \frac{1}{|\Psi(I,F)|}\lambda(u,v)\lambda(E_\psi(I,F))\lambda(N)$$

As before, color the edges of $I \oplus F$ yellow and green depending on whether they come from E_ψ or M. The number of yellow-green pairings of $I \oplus F$ is bounded by $\Psi(I, F)$, and the inequality follows.

ii) Let $I \in \mathcal{N}(s, z)$ (blue) and $F \in \mathcal{P}$ (red).

Fix a pairing $\psi \in \Psi_T(I, F)$. Define the graph $E_\psi(I, F) = I \oplus F \oplus (M \cup M')$. Then, $E_\psi(I, F) \in \widehat{\mathcal{N}}(s, z, (u, v))$. Given $E_\psi(I, F), T$ and ψ, we can recover I and F. Since $I \cup F = N \cup E(I, F) \cup (u, v)$,

$$\frac{1}{|\Psi(I,F)|}\lambda(I)\lambda(F) = \frac{1}{|\Psi(I,F)|}\lambda(u,v)\lambda(E_\psi(I,F))\lambda(N)$$

As before, color the edges of $I \oplus F$ yellow and green depending on whether they come from E_ψ or M. The number of yellow-green pairings of $I \oplus F$ is bounded by $\Psi(I, F)$, and the inequality follows. □

Lemma 3.16. *i) Let $u \in U, v \in V$. Then,*

$$\lambda(u,v)\lambda(\widehat{\mathcal{P}}(u,v))\lambda(\mathcal{N}(u,v)) \leq \lambda(\mathcal{P})^2$$

ii) Fix $s \in U, z \in V$. Let $u \in U, v \in V$. Then,

$$\lambda(u,v)\lambda(\widehat{\mathcal{N}}(s,z,(u,v)))\lambda(\mathcal{N}(u,v)) \leq 4\lambda(\mathcal{N}(s,z))\lambda(\mathcal{P})$$

Proof. i) Let $N_1 \in \mathcal{N}(u,v)$ (blue) and $N_2 \in \widehat{\mathcal{P}}(u,v)$ (red).

Consider the symmetric difference $H = N_1 \oplus N_2$. Both u, v have red degree one more than their blue degree. H may or may not contain the edge (u,v). If it is present, it is colored blue. We define a red-blue pairing of H to partition the edges into two perfect tables N_3, N_4. To define the pairing, we first define the multigraph $H' = H \cup (u,v)$, so that the new edge (u,v) is colored blue. Now, let Ψ_{good} be the set of possible pairings of H' so that for $\psi \in \Psi_{good}$, the corresponding decomposition of H' into alternating circuits, there is not circuit containing both copies of (u,v). In case H' contained only one copy of (u,v) all pairings are 'good'. If H' did indeed contain two copies of (u,v), we claim that $|\Psi_{good}|$ is at least $1/2$ fraction of all possible pairings. To see this, take any pairing whose circuit decomposition contains a circuit with both copies of (u,v). From this pairing, we can obtain a 'good' pairing by switching the red edges that the blue copies of (u,v) are paired with at u. Note that two such distinct pairings will always give distinct 'good' pairings.

Now, fix $\psi \in \Psi_{good}$. Let C_1, C_2 be the circuits containing the edge (u,v). For every other circuit, send all the blue edges to N_3 and the red edges to N_4. Do the same for the circuit of C_1, C_2 in which for the edge (u,v), v is adjacent to a lower numbered vertex through a red edge. For the remaining circuit, put the red edges in N_3, and the blue edges in N_4. Lastly, put all edges in $N_1 \cap N_2$ into both N_3, N_4. Then, $N_3, N_4 \in \mathcal{P}$.

As before, we can recover the uncolored H' from N_3, N_4. If the pairing of H' was known, the map can be inverted, and N_1, N_2 recovered. Since the pairing is not known, proceed

as follows. Color the edges of H' green if they are from N_3 and yellow if from N_4. Now the total number of yellow-green pairings is equal to the total number of possible red-blue pairings. However, we can eliminate the ones in which, say at u the copies of (u, v) are paired, since this would give a cycle decomposition which was impossible for a pairing from Ψ_{good}. If the yellow degree of u in H is $d \geq 2$ (which is the case if there were 2 (blue) copies of (u, v) in H'), this eliminates at least $(d-1)!/(d!) \geq 1/2$ of all yellow-green pairings. Hence not too many N_1, N_2 pairs can map to N_3, N_4.

ii) In the case that $s \neq u$ and $z \neq v$, the proof is analogous to the previous case. The other cases are:

a) $s = u, z \neq v$. Let $N_1 \in \mathcal{N}(u, v)$ be blue and $N_2 \in \widehat{\mathcal{N}}(s, z, (u, v))$ be red. Then, in the symmetric difference, u has equal red and blue degree, v has 1 extra red degree, and z has one extra blue degree. If we add an extra blue edge (u, v), then s has an extra blue degree while v get equal red and blue degree. Hence in a pairing of H', there is an alternating walk from s to z whose initial and final edges are blue. As before, to take care that the two copies of (u, v) don't end up in the same table, we can exchange the pairing at one end, say u (on either the walk or any circuit), to get a pairing where the two edges are not part of the same circuit or walk.

b) $s \neq u, z = v$. The argument in this case is similar to *a*).

c) $s = u, z = v$. This case is trivial. If $N_1 \in \mathcal{N}(u, v)$ and $N_2 \in \mathcal{N}(s, z)$ such that the edge (u, v) is not present in N_2, set $N_3 = N_1 \in \mathcal{N}(s, z)$, and set $N_4 = N_2 \cup (u, v) \in \mathcal{P}$. The map is clearly invertible.

Since in all, there are 4 cases, accounting for a factor of 4, given which case we are in, we obtain the claimed bound. □

We now plug the above bounds into the expressions for congestion through a transition of the chain which either adds or deletes an edge.

Proof of Lemma 3.14. Let T be a transition which either adds or deletes an edge (a move in the Markov chain of type 1 or 2a). In each case, let $T = (M, M')$, where $M \in \mathcal{N}(u, v)$ and $M' \in \mathcal{P}$ (the proof in the case that the transition deletes an edge is along the same lines, with the appropriate modification to Lemmas 3.15 and 3.16). We bound the left hand side of (9) by bounding the contribution firstly, due to a pair of perfect tables, and secondly due to a near perfect and a perfect table.

We bound the congestion through T due to $(I, F) \in \mathcal{P} \times \mathcal{P}$ as follows.

$$\begin{aligned}
& \sum_{\substack{(I,F)\in f_T \\ I\in\mathcal{P}}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\pi(I)\pi(F)}{\pi(M)} \\
= \quad & \frac{1}{w(\Omega)} \sum_{\substack{(I,F)\in f_T \\ I\in\mathcal{P}}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\lambda(I)\lambda(F)}{\lambda(M)}\frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{P})} \\
\text{(By Lemma 3.15, i)} \quad \leq \quad & \frac{1}{w(\Omega)}\lambda(u,v)\lambda(\widehat{\mathcal{P}}(u,v))\frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{P})} \\
\text{(By Lemma 3.16, i)} \quad \leq \quad & \frac{\lambda(\mathcal{P})}{w(\Omega)} \\
\leq \quad & \frac{1}{nm}
\end{aligned}$$

Next, we bound the congestion through T due to $(I, F) \in \mathcal{N} \times \mathcal{P}$.

$$\sum_{s\in U, z\in V} \sum_{(I,F)\in f_T^{s,z}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\pi(I)\pi(F)}{\pi(M)}$$

$$= \quad \frac{1}{w(\Omega)} \sum_{s,z} \frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(s,z))} \sum_{(I,F)\in f_T^{s,z}} \frac{|\Psi_T(I,F)|}{|\Psi(I,F)|}\frac{\lambda(I)\lambda(F)}{\lambda(M)}$$

$$\begin{aligned}
\text{(By Lemma 3.15, ii)} \quad & \leq \quad \frac{1}{w(\Omega)} \sum_{s,z} \lambda(u,v)\lambda(\widehat{\mathcal{N}}(s,z,(u,v)))\frac{\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(s,z))} \\
\text{(By Lemma 3.16, ii)} \quad & \leq \quad \frac{4nm\lambda(\mathcal{P})}{w(\Omega)} \\
& \leq \quad 4
\end{aligned}$$

Adding the congestion from each of these sources, the congestion through a transition that adds or deletes an edge is bounded by $O(1)$. □

Lemmas 3.11 and 3.14 imply the Inequality (7). It can be seen from the proofs of the lemmas in this section, that if the weights $w(u, v)$ satisfy (4), the bound holds for the weights w up to a small constant factor. Hence, we have that $\rho(f) = O(nmD^2 d_{max})$. This implies that the mixing time of the chain started at G is bounded by $\tau_G(\delta) = O(nmD^2 d_{max}(\ln(1/\pi(G)) + \log \delta^{-1}))$. This completes the proof of Theorem 3.10.

3.7 Approximating Ideal Weights by Simulated Annealing

Recall that our goal is to find the ideal weights $w^*_\lambda(u, v)$ (or, rather, a constant factor approximation of the ideal weights) for $\lambda = 1$.

As mentioned earlier, we will do this by progressively increasing the value of λ. We start with λ close to 0, for which it is possible to compute a $(1 + \varepsilon)$ approximation of the ideal weights in a straightforward manner, see Theorem 3.3. However, later in the algorithm we will only have a constant factor, say 2, approximation of the ideal weights. We will use samples of the corresponding Markov chain to obtain a better approximation of the ideal weights; this in turn allows us to increase the value of λ slightly so that the improved approximation of the ideal weights of the old λ sufficiently approximates the ideal weights of the new λ. Eventually, λ becomes 1 and we will have a suitable approximation of the ideal weights for $\lambda = 1$. In this section we discuss these steps in more detail.

3.7.1 Bootstrapping

For each pair u, v suppose we have weights $w_\lambda(u, v)$ which are a 2-approximation to the weights $w^*_\lambda(u, v)$. That is, suppose that $w^*_\lambda(u, v)/2 \leq w_\lambda(u, v) \leq 2w^*_\lambda(u, v)$. We want to use the Markov chain to tighten this approximation to a factor $c \in (1, 2)$. The following computation closely mimics the computation of [47, Section 3].

Recall, that π_λ denotes the stationary distribution of the Markov chain. To simplify notation, we will omit the subscript λ. Recall, that for a given activity λ the ideal weights are defined as $w^*_\lambda(u, v) = \lambda(\mathcal{P})/\lambda(\mathcal{N}(u, v))$. Note that if $w(u, v) = w^*(u, v)$ for every $u \in U, v \in V$, then

$$w(\mathcal{N}(u, v)) = w^*(u, v)\lambda(\mathcal{N}(u, v)) = \lambda(\mathcal{P}) = w(\mathcal{P}).$$

Thus for the Markov chain run with weights $w = w^*$, the stationary distribution of the chain satisfies $\pi(\mathcal{N}(u,v)) = \pi(\mathcal{P})$. For arbitrary weights w, note that

$$\pi(\mathcal{N}(u,v)) = \frac{w(u,v)\lambda(\mathcal{N}(u,v))}{w(\Omega)} = \frac{w(u,v)\lambda(\mathcal{P})}{w(\Omega)w^*(u,v)} = \pi(\mathcal{P})\frac{w(u,v)}{w^*(u,v)}$$

Rearranging terms, we have

$$w^*(u,v) = w(u,v)\frac{\pi(\mathcal{P})}{\pi(\mathcal{N}(u,v))} \tag{10}$$

This implies a bootstrapping procedure to boost rough approximations to w^* into arbitrarily close approximations. By sampling from the stationary distribution of the chain with weights w, we can estimate $\pi(\mathcal{P})/\pi(\mathcal{N}(u,v))$, and thus using (10) we can estimate $w^*(u,v)$.

Here are the details. The idea is to obtain a $c^{1/2}$-approximation of both $\pi(\mathcal{P})$ and $\pi(\mathcal{N}(u,v))$. Then we will have a c-approximation, say z, of $\pi(\mathcal{P})/\pi(\mathcal{N}(u,v)) = w^*(u,v)/w(u,v)$. In other words,

$$z/c \leq \frac{w^*(u,v)}{w(u,v)} \leq cz$$

and thus it suffices to set the weight approximations $w_{\text{new}}(u,v) := w(u,v)z$ to get c-approximations of $w^*(u,v)$.

We can use the indicator random variables X and $X_{u,v}$ for the events "a sample from π is in $\mathcal{P}$" and "a sample from π is in $\mathcal{N}(u,v)$" as estimators of $\pi(\mathcal{P})$ and $\pi(\mathcal{N}(u,v))$. However, by running the Markov chain we cannot obtain a sample from π, rather a sample from $\hat{\pi}$ which is δ-close to π in total variation distance. Thus, $E[X] = \hat{\pi}(\mathcal{P})$ and $E[X_{u,v}] = \hat{\pi}(\mathcal{N}(u,v))$. It is sufficient to set δ so that $\hat{\pi}(\mathcal{P})$ and $\hat{\pi}(\mathcal{N}(u,v))$ approximate $\pi(\mathcal{P})$ and $\pi(\mathcal{N}(u,v))$, respectively, by a factor of $c^{1/4}$. Then we can use several samples of X and $X_{u,v}$ to approximate $\hat{\pi}(\mathcal{P})$ and $\hat{\pi}(\mathcal{N}(u,v))$ to within a factor of $c^{1/4}$. Thus, overall we obtain a c-approximation of the ratio $\pi(\mathcal{P})/\pi(\mathcal{N}(u,v))$.

First we sketch how to set δ so that $\hat{\pi}$ is within a factor of $c^{1/4}$ of π. Recall that the distribution π is defined by a weight function w which is a 2-approximation of the ideal weights w^*. Thus, $4/(nm) \geq \pi(\mathcal{P}), \pi(\mathcal{N}(u,v)) \geq 1/(4nm)$ and we can set $\delta = \Theta(1/(nm))$ so that $\hat{\pi}(\mathcal{P}), \hat{\pi}(\mathcal{N}(u,v)) = \Theta(1/(nm))$ and $\hat{\pi}(\mathcal{P})$ and $\hat{\pi}(\mathcal{N}(u,v))$ are $c^{1/4}$-approximations of $\pi(\mathcal{P})$ and $\pi(\mathcal{N}(u,v))$.

To obtain a $c^{1/4}$ approximation of $\hat{\pi}(\mathcal{P})$ we approximate $E[X]$ within a factor of $c^{1/4}$ by averaging s random variables $X_1, \ldots, X_s$. By the Chernoff bounds, since $E[X] = \Theta(1/(nm))$, it suffices to take $s = O(nm \log \zeta^{-1})$ samples to approximate $E[X] = \hat{\pi}(\mathcal{P})$ with probability $\geq 1 - \zeta$. Analogous arguments hold for $E[X_{u,v}]$.

Putting it all together, the average of the X_i's estimates $\hat{\pi}(\mathcal{P})$ within a factor of $c^{1/4}$ with probability $\geq 1-\zeta$ and $\hat{\pi}(\mathcal{P})$ is within a factor of $c^{1/4}$ of $\pi(\mathcal{P})$. Thus, we obtain estimates of $\pi(\mathcal{P})$ to within a factor of $c^{1/2}$ with probability $\geq 1-\zeta$. Therefore, with probability at least $1 - (nm+1)\zeta$ we obtain $c^{1/2}$ approximations of all $\pi(\mathcal{P})$, $\pi(\mathcal{N}(u,v))$, resulting in factor of c approximations of $w^*(u,v)$ for every u, v. Since we do the bootstrapping for every λ_i, if the number of phases is ℓ, the overall probability of success is $\geq 1 - (nm+1)\ell\zeta$ which we want to be, say, 4/5. It suffices to set $\zeta = \Theta(1/((nm+1)\ell))$.

3.7.1.1 Warm Starts

For a fixed λ the improved approximation of the ideal weights includes running the Markov chain $s = O(nm \log \zeta^{-1}) = O(nm \log(nm))$ times. By Theorem 3.10 the mixing time of the Markov chain started at graph G is $O(nmD^2 d_{max}(\ln(1/\pi(G)) + \log \delta^{-1}))$. The term $\log \pi(G)^{-1}$ comes from the fact that the starting distribution is concentrated on the state G. The graph G^* seems to be a good starting point since $\lambda(G^*) = 1$ and thus $\log \pi(G^*)^{-1} = O(D \log(nm))$. If we start the chain at G^* we need to take $O(nmD^2 d_{max}(\ln(1/\pi(G^*)) + \log \delta^{-1}))$ steps of the chain per sample. The standard method of warm starts can be used to avoid the $\log \pi(G^*)^{-1}$ term in the running time. The idea is to obtain the first sample by taking $O(nmD^2 d_{max}(\ln(1/\pi(G^*)) + \log \delta^{-1}))$ steps, but all subsequent samples are obtained by running the Markov chain started at the previous sample. This way, the chain is effectively started from a distribution close to the stationary distribution and thus the subsequent samples each take only $O(nmD^2 d_{max}(\log \delta^{-1}))$ steps. The same idea is used in [47].

3.7.2 Total Number of Phases

We specify a sequence $\varepsilon(nm)^{-D} = \lambda_1 \leq \cdots \leq \lambda_\ell = 1$ such that

$$\frac{1}{\sqrt{2}} \leq \frac{w^*_{\lambda_i}(u,v)}{w^*_{\lambda_{i+1}}(u,v)} \leq \sqrt{2} \qquad \text{for every } i \in [\ell-1] \text{ and every } u, v$$

Then, if $w_{\text{new}}(u,v)$ is a $\sqrt{2}$-approximation (remember that we are free to choose any constant $c \in (1,2)$) of $w^*_{\lambda_i}(u,v)$, then by the above, $w_{\text{new}}(u,v)$ is also a 2-approximation of $w^*_{\lambda_{i+1}}(u,v)$. Therefore we can increase λ from λ_i to λ_{i+1} and still be able to use Theorem 3.10.

We obtain the above sequence of λ's by reversing the output produced by the algorithm of [8] for computing the cooling schedule λ. It constructs a λ-sequence of length $\ell = O(D \log D \log(nm))$ with the additional property that $\lambda_{i+1}(\mathcal{P})$ is within a factor of $2^{1/4}$ of $\lambda_i(\mathcal{P})$ and $\lambda_{i+1}(\mathcal{N}(u,v))$ is within a factor of $2^{1/4}$ of $\lambda_i(\mathcal{N}(u,v))$ for every u, v and $i \in [\ell-1]$ (see Lemmas 2 and 3 of [8], notice that in our case $s = D$ and $\gamma = (nm)^D$ since we may assume that $\varepsilon \geq 1/(nm)^D$). Thus, $1/\sqrt{2} \leq w^*_{\lambda_i}(u,v)/w^*_{\lambda_{i+1}(u,v)} \leq \sqrt{2}$, as required.

3.8 Counting by Sampling

In this section, we sketch a standard reduction from counting to sampling. Our goal is to estimate $|\mathcal{P}|$. For any sequence $\lambda_1, \ldots, \lambda_\ell$,

$$|\mathcal{P}| = \frac{|\mathcal{P}|}{w_{\lambda_\ell}(\Omega)} \frac{w_{\lambda_\ell}(\Omega)}{w_{\lambda_{\ell-1}}(\Omega)} \frac{w_{\lambda_{\ell-1}}(\Omega)}{w_{\lambda_{\ell-2}}(\Omega)} \cdots \frac{w_{\lambda_2}(\Omega)}{w_{\lambda_1}(\Omega)} w_{\lambda_1}(\Omega)$$

Let us fix the λ-sequence from the previous section. We first estimate

$$w_{\lambda_1}(\Omega) = \lambda_1(\mathcal{P}) + \sum_{u,v} w_{\lambda_1}(u,v)\lambda_1(\mathcal{N}(u,v))$$

where $1 \leq \lambda_1(\mathcal{P}) \leq 1+\varepsilon$ and $x_{u,v} \leq \lambda_1(\mathcal{N}(u,v)) \leq (1+\varepsilon)x_{u,v}$ for every u, v, see Theorem 3.3. Since $w_{\lambda_1}(u,v) = 1/x_{u,v}$, we get that $nm+1$ is a $(1+\varepsilon)$ approximation of $w_{\lambda_1}(\Omega)$. We define $s_* := |\mathcal{P}|/w_{\lambda_\ell}(\Omega)$ and $s_i := w_{\lambda_i}(\Omega)/w_{\lambda_{i-1}}(\Omega)$ and we will use samples of the Markov chain to estimate each s_i within a factor of $e^{\varepsilon/(2\ell)}$ and s_* within a $e^{\varepsilon/2}$ factor. Then, if s'_* and s'_i denote the estimates for s_* and the s_i, the quantity $(nm+1)s'_*s'_2 \ldots s'_\ell$ estimates $|\mathcal{P}|$ within a factor $(1+\varepsilon)e^{\varepsilon} = 1+\varepsilon'$, as required.

Recall that with probability $\geq 4/5$ the weights w_{λ_i} correctly approximate the ideal weights $w^*_{\lambda_i}$ for every i, see Section 3.7.1. In what follows we will assume that the weights w are correct estimates of w^* for every λ_i.

Notice that since $\lambda_\ell = 1$ and each $w(u,v)$ is within a factor of 2 of $w^*(u,v)$, we have that $|\mathcal{P}| = w_{\lambda_\ell}(\mathcal{P})$ is within a constant factor of $w_{\lambda_\ell}(\Omega)/(nm+1)$. Thus, $s_* = \Theta(1/(nm))$. By a similar argument, $s_i = \Theta(1)$ for each i. Therefore we can estimate the s_i as follows. We take a random sample X of the Markov chain for λ_i and consider the value of $\mathrm{est}(X) := w_i(M)/w_{i-1}(M)$. The expectation of this value is exactly s_i. Then we take $O(\ell\varepsilon^{-2})$ samples of the Markov chain for λ_i with the variation distance $\delta = O(\varepsilon/\ell)$ and average their $\mathrm{est}_i(X)$ values, obtaining est_i. By the Chebyshev's inequality, $\prod_{i=2}^{\ell} \mathrm{est}_i$ estimate $\prod_{i=2}^{\ell} s_i$ within an $e^{\varepsilon/2}$ factor with a probability $\geq 11/12$ (for suitable constants within the O notation).

Similarly, we sample X by the Markov chain for λ_ℓ and $\delta = O(\varepsilon)$ and define $\mathrm{est}_*(X)$ to be indicator variable for the event $X \in \mathcal{P}$. Then the expectation of $\mathrm{est}_*(X)$ is s_* and we average the values of $O(nm\varepsilon^{-2})$ samples to get within a factor $e^{\varepsilon/2}$ of s_* with probability $\geq 11/12$. Then, $(nm+1)\mathrm{est}_* \prod_{i=2}^{\ell} \mathrm{est}_i$ approximates $|\mathcal{P}|$ within a factor of $(1+\varepsilon)e^{\varepsilon}$ with probability $\geq 5/6$.

Thus, with probability $\geq 4/5$ we have correct estimates w of the ideal weights w^* and conditioned on the correct weight estimates, the algorithm outputs a $(1+\varepsilon')$-approximation of $|\Omega|$ with probability $\geq 5/6$. Unconditionally, with probability $\geq 2/3$, the algorithm produces an answer within $(1+\varepsilon')$ factor of $|\Omega|$.

See [47] and [8] for details of this computation.

3.9 Proof of Correctness of the Algorithm

We now recall the statement of our main theorem, and conclude its proof.

Theorem 3.2. *For any bipartite graph $G = (U \cup V, E)$ where $U = \{u_1, \ldots, u_n\}$ and $V = \{v_1, \ldots, v_n\}$, any degree sequence $r(1), \ldots, r(n); c(1), \ldots, c(m)$, any $0 < \varepsilon, \eta < 1$, we can approximate the number of subgraphs of G with the desired degree sequence (i.e., u_i has degree $r(i)$ and v_j has degree $c(j)$, for all i, j) in time $O((nm)^2 D^3 d_{\max} \log^5(nm/\varepsilon)\varepsilon^{-2}\log(1/\eta))$ where $D = \sum_i r(i) = \sum_j c(j)$ is the total degree and $d_{\max} = \max\{\max_i r(i), \max_j c(j)\}$ is*

the maximum degree. And, the approximation is guaranteed to be within a multiplicative factor $(1 \pm \varepsilon)$ *of the correct answer with probability* $\geq 1 - \eta$.

Proof. The theorem states that we can approximately count the number of bipartite graphs with a given degree sequence which are subgraphs of any given bipartite graph G. In the previous sections we dealt with the case when $G = K_{n,m}$, the complete bipartite graph on $n+m$ vertices. If G is not complete, we can perform the annealing algorithm in two stages. In the first stage, we run the simulated annealing algorithm described previously. Thus, we estimate the ideal weights for $\lambda = 1$ for the complete graph at the end of the first stage. In the second stage, we do the simulated annealing starting with the weights at the end of the first stage (notice that now all edge activities are 1). However, the annealing will *decrease* the activities of edges *not present in* G from 1 to $\lambda \approx 0$ (hence, we may be decreasing the activities of different edges than the ones whose activities were previously increased). The analysis of the annealing algorithm and the mixing time of the Markov chain remain the same. Thus, the two stage process only doubles the running time.

Now we break up the running time in the first stage. Initially, we spend $O((nmd_{\max})^2)$ time to construct the Greedy graph G^* and to approximate the initial weights, see Theorem 3.3. We need $\ell = O(D \log^2(nm))$ intermediate temperatures for the simulated annealing (Section 3.7.2). As discussed in Section 3.7.1, at each temperature we need to generate $O(nm \log(nm))$ samples from the stationary distribution of the Markov chain in order to do the bootstrapping. By Theorem 3.10, see also Section 3.7.1.1, each sample takes $O(D^2 nmd_{\max} \log(nm))$ steps of the Markov chain (recall that we set $\delta = \Theta(1/(nm))$, Section 3.7.1). Thus, as discussed in Section 3.7.1, with probability $\geq 4/5$ in time $O((nm)^2 D^3 d_{\max} \log^4(nm))$ we compute correct approximations of the ideal weights w^* for $\lambda = 1$. Therefore, we can generate a random bipartite graph with the desired degree sequence, from a distribution within variation distance $\leq \delta$ of uniform, in time $O((nm)^2 D^3 d_{\max} \log^4(nm/\delta))$. The computation of the initial weights is absorbed by this quantity.

For the counting, see Section 3.8, we use $O(nm\varepsilon^{-2})$ samples of the Markov chain to approximate s^* and for every intermediate temperature we need $O(\ell\varepsilon^{-2}) = O(D \log^2(nm)\varepsilon^{-2})$

samples to approximate the corresponding s_i. Taking into account the mixing time of the Markov chain, the counting phase takes time $O(D^4 nmd_{\max} \log^5(nm/\varepsilon)\varepsilon^{-2})$. Thus, the final running time of the algorithm including the weight estimation phase is $O(D^3(nm)^2 d_{\max} \log^5(nm/\varepsilon)\varepsilon^{-2})$. With probability $\geq 2/3$ the algorithm outputs a $(1+\varepsilon)$ approximation of the number of bipartite graphs with the desired degree sequence. This can be boosted to probability $\geq 1-\eta$ by running the algorithm $O(\log \eta^{-1})$ times and outputting the median of the resulting values. □

3.10 Conclusions

We have presented an algorithm for counting and sampling binary contingency tables for arbitrary degree sequences. While our algorithm has many similarities to the permanent algorithm of [47], the new algorithm relies on a surprising combinatorial property of the greedy graph that allows us to start the annealing process.

The Diaconis or "switch" Markov chain does not use the auxiliary states that the Markov chain we analyze uses. An interesting open problem is the efficiency of the Diaconis chain on arbitrary degree sequences. Does there exist a degree sequence for which the chain converges slowly to the stationary distribution, or is the mixing time polynomial for all degree sequences?

CHAPTER IV

SIMULATED TEMPERING MIXES TORPIDLY FOR THE 3-STATE FERROMAGNETIC POTTS MODEL

Simulated tempering, like annealing, attempts to speed up the convergence time of a torpidly mixing Markov chain by defining the chain on an extended space parameterized by temperature. In this chapter, we study the simulated tempering Markov chain for sampling from the Gibbs distribution for the 3-state Potts model. Our results show that in this case, simulated tempering is not successful in overcoming the bottleneck that causes the fixed temperature algorithm to be slow. This suggests a limitation of the method and a need for finding better ways to define the tempering distributions.

4.1 Introduction

Glauber dynamics and local Markov chains like the Metropolis-Hastings algorithm are known to mix torpidly for sampling from spin systems where there are multiple modes in the stationary distribution [67, 85]. The modes usually correspond to different classes of ordered configurations which are predominant in the stationary distribution. The torpid mixing is due to bottlenecks in the state space where the Markov chain takes exponential time to cross from one mode to another since it must pass through a set of configurations enroute that is highly unlikely in the stationary distribution.

Simulated tempering and its variant swapping are Markov chain algorithms that attempt to overcome the bottleneck to rapid mixing at low temperature by allowing the Markov chain to spend some fraction of the time at higher temperatures where the Markov chain is known to mix rapidly. The chain alternates between moves of the local Markov chain at its current (fixed) temperature and randomly changing the temperature by a small amount. The intuition is that it should speed up the mixing time at low temperatures if the chain is "well mixed" on the space at higher temperatures and the intermediate temperatures

interpolate between these two extremes.

In this work, we show that there are natural examples where this intuition can fail. In particular, the Markov chain mixes rapidly at the high temperature distribution, but due to a particular type of phase transition in the system, called a *first order phase transition*, the simulated tempering chain mixes torpidly below a critical temperature.

4.1.1 Phase Transitions and Torpid Mixing

Phase transitions [35] are phenomena studied in statistical physics and thermodynamics and refer to the transformation of a system from one *phase* to another. A phase is characterized by one or more physical properties of the system, for example, the density or specific heat. A *phase transition* is characterized by an abrupt change in the phase of the system when the temperature is changed by a small amount. A physical example of this is the change from liquid to gas when water is boiled.

One way to classify phase transitions is based on the non-analyticity of the derivatives of the *free energy* with respect to the temperature. If $Z(\beta)$ is the partition function at inverse temperature $\beta = 1/kT$, the free energy is defined to be the quantity $-kT \log(Z(\beta))$. The order of the phase transition is given by the lowest order derivative of the free energy with respect to temperature which is discontinuous. Thus, for example, a first order phase transition is one where the first derivative of the free energy is discontinuous. The derivatives of the free energy are relevant because it turns out that they are precisely thermodynamic quantities like internal energy or specific heat for the system.

The order of the phase transition is also characterized by whether there is a *latent heat* involved in the transition. Latent heat is the energy released or absorbed during the transition. First order phase transitions are associated with latent heat in the following sense. At the transition temperature, the heat capacity (which is the second derivative of the free energy) becomes infinite, since the first derivative is discontinuous. This means that heat can be added, but the temperature does not rise, instead the phase change occurs. Once the latent heat has been added, the temperature continues to rise again. This implies that first order phase transitions are associated with *co-existence of phase*. For example,

at the boiling point of water, both liquid water and water vapor co-exist. The co-existence is because the entire system does not complete the phase transition at the same time since the latent heat cannot be transferred to or from the environment instantaneously for the entire system. Mathematically, at a very high level, if the first derivative is discontinuous, it implies that in the distribution, the location of the modes in the ordered phase does not change in a smooth way to the location of the mode in the disordered phase. Rather, there is a critical temperature at which both types of modes, ordered and disordered, must co-exist [35].

Continuous phase transitions or *second order phase transitions*, do not exhibit phase co-existence. An example of this type of phase transition is the ferromagnetic phase transition in iron where the magnetization (the first derivative of the free energy) changes continuously with the field strength, the analogue of temperature.

Systems with phase transitions are known to exhibit torpid mixing at low temperatures. As we will see, the order of the phase transition, or specifically the coexistence of phase can also cause a difference in the behavior of sampling algorithms.

4.1.2 The Mean-field Potts Model

Recall the definition of the q-state Potts model from Section 1.2.2, in Chapter 1 The *mean-field Potts model* refers to the Potts model where G is the complete graph on n vertices. Mean-field models are studied in physics because they often share the same characteristics of systems in high dimensions [35]. Mean-field models are studied in computer science because even in simple cases, the behavior of sampling algorithms for them is not fully understood. The Swendsen-Wang algorithm [83] is an algorithm proposed as an alternative to Glauber dynamics for sampling from configurations of the q-state Potts model. Cooper et al. [18] considered the mean-field Ising model and showed that Swendsen-Wang algorithm mixes rapidly at all temperatures except possibly near the critical point. Gore and Jerrum [39] showed that the Swendsen-Wang algorithm mixes torpidly on the mean-field Potts model for $q \geq 3$ at the critical temperature. The complexity of Swendsen-Wang at the critical point for the Ising model remains unresolved. Interestingly, Madras and Zheng [64] analyzed

simulated tempering for the mean field Ising model and showed that it mixes rapidly at all temperatures, including the critical temperature. In the next section we define the simulated tempering and its variant swapping, and then describe our results for the mean-field Potts model.

4.2 Simulated Tempering and Swapping Algorithms

Recall that for the Potts model, the Gibbs distribution at temperature β is given by

$$\pi_\beta(x) = \frac{e^{\beta H(x)}}{Z(\beta)}$$

where

$$H(x) = \sum_{(i,j)\in E(G)} J \cdot \delta(x(i), x(j))$$

is the Hamiltonian.

We will consider the 3-state ferromagnetic mean-field Potts model, where $q = 3$, $J > 0$ and the underlying graph is complete. Let Ω denote the space of all 3^n configurations and suppose that we want to sample according to the Gibbs distribution at $\beta^* > 0$,

$$\pi_{\beta^*}(x) = \frac{e^{\beta^* H(x)}}{Z(\beta^*)}$$

The local Markov chain underlying the simulated tempering algorithm that we will consider is the Metropolis-Hastings algorithm [70]. The Markov kernel $\mathcal{K}$ is the graph where there is an edge between pairs of configurations $x, y \in \Omega$ which differ in the spin of only one vertex. For $(x, y) \in E(\mathcal{K})$ let

$$P(x, y) = \frac{1}{2\Delta} \min\left(1, \frac{\pi_{\beta^*}(y)}{\pi_{\beta^*}(x)}\right),$$

where Δ is the maximum degree of $\mathcal{K}$. It is easy to verify that π_{β^*} is the stationary distribution. It is well known by methods similar to those in [67, 85] that the Metropolis Markov chain mixes torpidly for low values of β^*. We analyze whether simulated tempering could speed up mixing.

4.2.1 Simulated Tempering Markov Chain

Simulated tempering was defined by Marinari and Parisi, and Geyer and Thompson [37, 65]. For sampling from the distribution π_β above, the algorithm is as follows. Let $0 = \beta_0 < \cdots < \beta_M = \beta^*$ be a set of inverse temperatures. Let $\pi_{\beta_0}, \cdots, \pi_{\beta_M} = \pi_{\beta^*}$ be the corresponding distributions over Ω. The state space of the simulated tempering chain at β_M with intermediate temperatures $\beta_1, \cdots \beta_{M-1}$ is $\widehat{\Omega} = \Omega \times \{0, \cdots, M\}$, which we can think of as the union of $M+1$ copies of the original state space Ω, each corresponding to a different inverse temperature. The choice of $\beta_0 = 0$ corresponds to infinite temperature where the Metropolis algorithm converges rapidly to the stationary (uniform) distribution, while β_M is the inverse temperature at which we wish to sample. The distributions π_{β_i} interpolate between the extremes.

The stationary distribution of the tempering chain $\widehat{\pi}$, is chosen to be uniform over temperatures, and the conditional distributions are the fixed temperature Gibbs distributions:

$$\widehat{\pi}(x, i) = \frac{1}{M+1}\pi_{\beta_i}(x), \qquad x \in \Omega.$$

The tempering Markov chain consists of two types of moves: *level moves*, which update the configuration while keeping the temperature fixed, and *temperature moves,* which update the temperature while remaining at the same configuration. In each step of the chain, with probability 1/2 we choose one of the types of moves to perform.

• A **level move** connects (x, i) and (x', i), where x and x' are connected by one-step transitions of the Metropolis algorithm on Ω at inverse temperature β_i. The move $\widehat{P}((x, i), (x', i))$ is accepted with probability

$$P_i(x, x') \;=\; \frac{1}{2\Delta} \min\left(1, \frac{\pi_{\beta_i}(x')}{\pi_{\beta_i}(x)}\right).$$

Here $P_i(x, x')$ is the Metropolis probability of going from x to x' according to the stationary probability π_{β_i}.

• A **temperature move** connects (x, i) to $(x, i \pm 1)$. For a temperature move, we randomly choose to move the temperature up or down, and the move $\widehat{P}((x, i), (x, i \pm 1))$ is accepted

with probability

$$\frac{1}{2}\min\left(1, \frac{\widehat{\pi}(x, i\pm 1)}{\widehat{\pi}(x,i)}\right) = \frac{1}{2}\min\left(1, \frac{Z(\beta_i)}{Z(\beta_{i\pm 1})} e^{(\beta_{i\pm 1}-\beta_i)H(x)}\right).$$

We will fix M, the number of temperatures to be a polynomial growing at least as $\Omega(n)$. Since M is a polynomial, for every $0 \leq i \leq M$, $\widehat{\pi}(\Omega, i)$ is at least an inverse polynomial fraction. It can be verified that the lower bound on M ensures that the transition probabilities are not too small, by bounding the size of the ratio $\frac{Z(\beta_i)}{Z(\beta_{i\pm 1})}$. Notice that while the exponential factor is simple to calculate given x and i, it is not clear that we can compute the ratio of partition functions, which we need in order to compute the transition probabilities. The swapping algorithm, also an aggregate chain using these temperatures, circumvents this difficulty in implementing temperature moves.

4.2.2 Swapping Markov Chain

The swapping algorithm, also sometimes known as Metropolis Coupled Markov Chain Monte Carlo was introduced by Geyer [36].

The state space is the product space $\widehat{\Omega} = \Omega^{(M+1)}$, the product of $M+1$ copies of the original state space, corresponding to inverse temperatures $\beta_0 < \cdots < \beta_M$.

Let $\pi_M(x) = \pi(x)$ be the distribution from which we wish to sample and let $\pi_0(x) = \frac{1}{|\Omega|}$ (the uniform distribution), for $x \in \Omega$. A configuration in the swapping chain is an $(M+1)$-tuple $x = (x_0, \cdots, x_M) \in \widehat{\Omega}$, where each component represents a configuration chosen from the i^{th} distribution. The probability distribution $\widehat{\pi}$ is the product measure

$$\widehat{\pi}(x) = \prod_{i=0}^{M} \pi_{\beta_i}(x_i).$$

The swapping chain also consists of two types of moves:

• A **level move** connects $x = (x_0, \cdots, x_i, \cdots, x_M)$ and $x' = (x_0, \cdots, x_i', \cdots, x_M)$ if x and x' agree in all but the i^{th} components, and x_i and x_i' are connected by one-step transitions of the Metropolis algorithm on Ω. The move $\widehat{P}(x, x')$ is accepted with probability

$$\frac{1}{2(M+1)} P_i(x, x') \;=\; \frac{1}{2(M+1)} \min\left(1, \frac{\pi_{\beta_i}(x')}{\pi_{\beta_i}(x)}\right).$$

• A **swap move** connects $x = (x_0, \cdots, x_i, x_{i+1}, \cdots, x_M)$ to $x' = (x_0, \cdots, x_{i+1}, x_i, \cdots, x_M)$, i.e., it interchanges the i^{th} and $i+1^{st}$ components, with the appropriate Metropolis probabilities on $\widehat{\pi}$. In particular,

$$\begin{aligned}
\widehat{P}(x, x') &= \frac{1}{2(M+1)} \min\left(1, \frac{\widehat{\pi}(x')}{\widehat{\pi}(x)}\right) \\
&= \frac{1}{2(M+1)} \min\left(1, \frac{\pi_{i+1}(x_i)\pi_{\beta_i}(x_{i+1})}{\pi_{\beta_i}(x_i)\pi_{i+1}(x_{i+1})}\right) \\
&= \frac{1}{2(M+1)} \min\left(1, e^{(\beta_{i+1}-\beta_i)(H(x_i)-H(x_{i+1}))}\right).
\end{aligned}$$

Notice that now the normalizing constants cancel out. Hence, implementing a move of the swapping chain is straightforward, unlike tempering where good approximations for the partition functions are required. Zheng proved that fast mixing of the swapping chain implies fast mixing of the tempering chain [91], although the converse is unknown.

For both tempering and swapping, it is important that successive distributions π_{β_i} and $\pi_{\beta_{i+1}}$ have sufficiently small variation distance so that temperature moves are accepted with nontrivial probability. Hence, M must be sufficiently large. However, M must be small enough so that it does not cause the running time to grow too much. Setting M to be a polynomial which is $\Omega(n)$ ensures both these constraints are satisfied.

4.2.3 Importance Sampling

As an alternative to simulated tempering, Madras and Piccioni [61] proposed importance sampling using the following distribution over Ω:

$$\pi_{av}(x) = \sum_{i=0}^{M} \frac{\pi_{\beta_i}(x)}{M+1}$$

They showed that the simulated tempering Markov chain is identical in distribution to the Metropolis Markov chain defined according to the distribution π_{av}. In some sense, simulated tempering is doing importance sampling using the average of the simulated tempering distributions. As described in Section 3.2.2, we would like to choose the importance sampling distribution so as to ensure the corresponding unbiased estimator has low variance. At a high level, our slow mixing result can be viewed as saying that at low enough temperature, the average of the tempering distributions is not a good choice for the importance sampling distribution for the 3-state Potts model.

4.3 Summary of Results.

We show that the simulated tempering and swapping Markov chains require exponential time to converge for sampling from the 3 state mean-field ferromagnetic Potts model. We show that no matter how the interpolating temperatures for the algorithms are chosen, the chain will still take exponential time to converge.

Theorem 4.1. *Let $\beta_c = \frac{2\ln 2}{n}$. There is a constant $c_1 > 0$ such that for any set of intermediate temperatures, the tempering chain on Ω at temperature β_c has mixing time $\tau(\varepsilon) \geq e^{c_1 n} \ln(1/\varepsilon)$.*

The torpid convergence of the tempering chain is caused by a first-order phase transition in the 3-state ferromagnetic Potts model. In the Potts model, there is a critical temperature which exhibits coexistence of the ordered and disordered phases. In contrast, the Ising model has a second-order phase transition, and there is no phase coexistence, and this distinguishes why simulated tempering mixes rapidly for the Ising model [64] and not the Potts model.

The second result that we show is that there are even cases when the mixing time of the tempering algorithm will be significantly slower than that of the fixed temperature Metropolis algorithm. This disproves the conventional wisdom that tempering can be in the worst case slower by a factor polynomial in the number of temperatures. Let Ω_{RGB} denote the subset of Ω consisting of configurations where the majority of vertices are red. On the restricted space Ω_{RGB}, we show that tempering can slow down the Metropolis algorithm at a fixed temperature by an exponential multiplicative factor.

Theorem 4.2. *Let $\frac{2\ln 2}{n} < \beta^* < \frac{3}{2n}$. There are constants c_2, c_3 (which may depend on β^*) such that $0 < c_2 < c_3$ and the Metropolis chain on Ω_{RGB} at β^* has mixing time $\tau(\varepsilon) \leq e^{c_2 n} \ln(1/\varepsilon)$ while the mixing time of the tempering chain is bounded by $\tau(\varepsilon) \geq e^{c_3 n} \ln(1/\varepsilon)$.*

4.4 Torpid mixing of Simulated Tempering

In this section we show the main result, Theorem 4.1, which states that there is a temperature β_c at which simulated tempering for the 3-state ferromagnetic Potts model mixes torpidly regardless of the intermediate temperatures we choose.

We prove the lower bound on the mixing time of the tempering chain by bounding the conductance. The state space of the tempering chain is $\Omega \times [M+1]$. To show torpid mixing, it is enough to exhibit a cut in the state space whose conductance is small. The cut we construct depends only on the *number* of red, blue and green vertices in the configuration. Hence, for the purpose of defining the cut, it is convenient to view the state space of configurations as equivalence classes of colorings according to the number of vertices of each color. Furthermore, the cut we define will induce the same cut on Ω at each temperature.

It is convenient for the exposition to make the following change of variable using the fact that for the 3-state ferromagnetic Potts model, the underlying graph is complete. Let x_1, x_2, and x_3 be the number of vertices assigned red, green and blue respectively in the configuration $x \in \Omega$. The Gibbs distribution at $\widetilde{\beta}$ with Hamiltonian $\widetilde{H}$ is given by

$$\pi_{\widetilde{\beta}}(x) = \pi_{\widetilde{\beta}}(x_1, x_2, x_3) = \frac{e^{\widetilde{\beta}\widetilde{H}(x)}}{\sum_y e^{\widetilde{\beta}\widetilde{H}(y)}},$$

For the complete graph, $\widetilde{H}(x) = \frac{1}{2}(x_1(x_1-1) + x_2(x_2-1) + x_3(x_3-1))$. Note that since $x_1 + x_2 + x_3 = n$, the linear terms will cancel from both the numerator and denominator. Setting $\beta = \widetilde{\beta}J/2$ and $H(x) = x_1^2 + x_2^2 + x_3^2$, we will work with the following expression for the distribution

$$\pi_\beta(x) = \frac{e^{\beta H(x)}}{Z(\beta)},$$

where $Z(\beta) = \sum_x e^{\beta H(x)}$.

Henceforth, we will use this modified Gibbs distribution since we will always work on the complete graph.

To define the cut, we partition Ω into sets Ω_σ, where $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ is partition of n into a triple, i.e., $\sigma_1 + \sigma_2 + \sigma_3 = n$. The set Ω_σ contains all colorings with σ_1, σ_2 and σ_3 vertices colored red, green and blue, respectively. The set Ω_σ corresponds to $\binom{n}{\sigma_1,\sigma_2,\sigma_3}$ different configurations in Ω and hence the Gibbs distribution on partitions σ at the temperature β_i (that is, the stationary distribution of the tempering chain, conditioned on being at the temperature β_i) is given by

$$\pi_{\beta_i}(\Omega_\sigma) = \binom{n}{\sigma_1, \sigma_2, \sigma_3} \frac{e^{\beta_i(\sigma_1^2+\sigma_2^2+\sigma_3^2)}}{Z(\beta_i)} \tag{11}$$

The idea for defining the cut with small conductance comes from the following properties of the stationary distribution conditioned on the sets Ω_σ. There is a critical temperature β_c where the Gibbs distribution exhibits the coexistence of two modes. There is a "disordered" mode in the distribution at $\left(\frac{n}{3}, \frac{n}{3}, \frac{n}{3}\right)$; this mode is present because though these configurations have small energy, the number of configurations (given by the multinomial term in Equation (11)) is large. At β_c, there are also "ordered" modes at $\left(\frac{2n}{3}, \frac{n}{6}, \frac{n}{6}\right)$, $\left(\frac{n}{6}, \frac{2n}{3}, \frac{n}{6}\right)$, $\left(\frac{n}{6}, \frac{n}{6}, \frac{2n}{3}\right)$. These modes are present because configurations with a predominant number vertices having the same color (red, or green or blue) are favored in the Gibbs distribution, though there are not as many of these configurations. The ordered and disordered modes are separated by a region whose density is exponentially smaller than both the modes, where neither the multinomial nor the energy term dominates. As the inverse temperature is decreased below β_c, the size of the disordered mode grows while the sizes of the ordered modes decrease. However, the region of exponentially small density remains small at every temperature. The cut in the state space of the simulated tempering chain at β_c is to take a region surrounding the ordered mode at each temperature. The conductance of this cut, up to a polynomial (in M) is bounded by the conductance at the critical temperature where the modes coexist. This is because in the stationary distribution, the chance of being at each temperature is equally likely.

In contrast, for the Ising model, there is no temperature at which the ordered and disordered modes coexist. It is due to the coexistence of the ordered and disordered phases that simulated tempering can be torpidly mixing for the Potts model while for the Ising model, it mixes rapidly [64].

We first present a straightforward upper bound on the conductance of the tempering chain at β_c.

Theorem 4.3. *Let $\beta_c = \frac{2\ln 2}{n}$. There is a constant $c_4 > 0$ such that the conductance Φ of the simulated tempering chain for the 3-state mean-field ferromagnetic Potts model at β_c, for any set of interpolating temperatures $\{\beta_i\}$, for $i \in I \subseteq [M]$ is at most $e^{-c_4 n}$.*

As a corollary, by Theorem 2.2, the lower bound on mixing time by the inverse of

conductance, this implies Theorem 4.1. Later, we will refine this bound in order to compare it to the upper bound on the mixing time of the Metropolis chain at a fixed temperature to show Theorem 4.2.

Let $A \subset \Omega$ be the set of configurations x such that $x_1, x_2, x_3 \leq n/2$. Let $S = \{(x, i) \mid x \in A,\ \beta_0 \leq \beta_i \leq \beta_c\}$. Let $B = \{x \in A \mid \exists\, x' \in \Omega \setminus A,\ P(x, x') \neq 0\}$ be the boundary of A (the set of configurations with at least one of x_1, x_2 or x_3 equal to $n/2$). Our aim is to show that the conductance of the set S is bounded. Note that it is not true that $\pi(S) \leq 1/2$ and hence a bound on Φ_S does not immediately imply a bound on Φ. Instead, we will show that the coexistence of the ordered and disordered phases implies that $\Phi \leq \text{poly}(n)\Phi_S$. We start by bounding Φ_S.

$$\begin{aligned} \Phi_S = \frac{F_S}{C_S} &= \frac{\displaystyle\sum_{i \in I} \sum_{x \in B} \pi_{\beta_i}(x) \sum_{x' \in \overline{A}} P(x, x')}{\displaystyle\sum_{i \in I} \sum_{x \in A} \pi_{\beta_i}(x)} \\ &\leq \frac{\displaystyle\sum_{i \in I} \sum_{x \in B} \pi_{\beta_i}(x)}{\displaystyle\sum_{i \in I} \sum_{x \in A} \pi_{\beta_i}(x)} \end{aligned} \tag{12}$$

The last expression above is the ratio of the sum over temperatures of the stationary probabilities of configurations in the set B (the boundary of the set A) to the sum over temperatures of the stationary probabilities of the configurations in the set A. In order to bound this quantity, we will need several technical lemmas which we state in the course of the proof but prove later to maintain the flow of the argument. The proofs of these lemmas are gathered in Section 4.4.1.

For $0 \leq \alpha \leq 1$ let $\Omega_{\alpha n}$ denote the set of configurations Ω_σ where $\sigma_1 = \alpha n$ and $\sigma_2 = \sigma_3 = (1 - \alpha)n/2$. In the next step, we show that by losing only a polynomial factor, the numerator of (12) can be bounded by the sums of the probabilities of the configurations $\Omega_{n/2}$ (the set of configurations on the boundary B with equal numbers of green and blue vertices), while the denominator is certainly is as large as the weight of the configurations in $\Omega_{n/3}$ (the set of configurations with equal numbers of red, blue and green vertices). In particular, we want to show that for some constant $c_5 > 0$,

$$\frac{\sum_{i\in I}\sum_{x\in B}\pi_{\beta_i}(x)}{\sum_{i\in I}\sum_{x\in A}\pi_{\beta_i}(x)} \leq c_5 n \frac{\sum_{i\in I}\pi_{\beta_i}(\Omega_{n/2})}{\sum_{i\in I}\pi_{\beta_i}(\Omega_{n/3})} \tag{13}$$

We use the following lemma, which says that along the line where the number of red vertices is $n/2$, the distribution at every temperature has a unique maximum at the configurations where the number of green vertices is equal to the number of blue vertices. Let $\overline{\pi}_i(x) = \pi_{\beta_i}\left(\frac{n}{2}, xn, \frac{n}{2} - xn\right)$ be the continuous function where x is allowed to vary continuously.

Lemma 4.4. *For n sufficiently large, the function $\overline{\pi}_i(x)$ has a unique maximum in the range $0 < x < \frac{1}{2}$ and attains its maximum at $x = \frac{1}{4}$ for all i such that $\beta_i \leq \beta_c$.*

This implies the inequality (13). Next, we'll show that Φ_S is essentially determined by the conductance of the cut induced at the highest inverse temperature β_M.

Lemma 4.5. *For every inverse temperature $\beta_i \leq \beta_M$, $\frac{\pi_i(\Omega_{n/2})}{\pi_i(\Omega_{n/3})} \leq \frac{\pi_M(\Omega_{n/2})}{\pi_M(\Omega_{n/3})}$.*

Proof. Note that only the exponential term in $\frac{\pi_i(\Omega_{n/2})}{\pi_i(\Omega_{n/3})}$ varies with β_i. Thus, for some function $h(n)$, we have

$$\begin{aligned}
\frac{\pi_i(\Omega_{n/2})}{\pi_i(\Omega_{n/3})} &= h(n)^{\beta_i n^2 (H(\frac{1}{2},\frac{1}{4},\frac{1}{4}) - H(\frac{1}{3},\frac{1}{3},\frac{1}{3}))} \\
&= h(n) e^{\beta_i n^2 (1/24)} \\
&\leq h(n) e^{\beta_c n^2 (1/24)} = \frac{\pi_{\beta_c}(\Omega_{n/2})}{\pi_{\beta_c}(\Omega_{n/3})}.
\end{aligned}$$

□

This implies that the ratio on the RHS of (13) can be bounded as follows

$$\frac{\sum_{i\in I}\pi_{\beta_i}(\Omega_{n/2})}{\sum_{i\in I}\pi_{\beta_i}(\Omega_{n/3})} \leq c_6 n \frac{\pi_{\beta_c}(\Omega_{n/2})}{\pi_{\beta_c}(\Omega_{n/3})}. \tag{14}$$

for some constant $c_6 > 0$.

There are two final steps to bounding the conductance. Firstly, we show that at β_c, $\frac{\pi_{\beta_c}(\Omega_{n/2})}{\pi_{\beta_c}(\Omega_{n/3})}$ is exponentially small. Secondly, we show that $\Phi \leq \text{poly}(n)\Phi_S$. These facts follow from properties of the stationary distribution proved in Lemmas 4.6 and 4.7.

The following lemma demonstrates that there is a critical temperature at which $\Omega_{n/3}$ and $\Omega_{2n/3}$ both have large weight compared to $\Omega_{n/2}$. Also, the configurations $\Omega_{n/3}$ have a weight that is at least a polynomial fraction of the stationary weight of Ω at β_c.

Lemma 4.6. *At* $\beta_c = \frac{2ln2}{n}$,

(i) $\pi_{\beta_c}(\Omega_{n/3}) = \pi_{\beta_c}(\Omega_{2n/3}) + o(1)$.

(ii) $\frac{\pi_{\beta_c}(\Omega_{n/2})}{\pi_{\beta_c}(\Omega_{n/3})} \leq e^{-\Omega(n)}$

(iii) $\pi_{\beta_c}(\Omega_{n/3}) \geq \frac{\pi_{\beta_c}(\Omega)}{n^2}$

Putting together the bound on Φ_S from inequality (14) and part ii) of Lemma 4.6, we obtain that for some constant $c_7 > 0$,

$$\Phi_S \leq e^{-c_7 n}$$

Lastly, we show the bound on the conductance. We need the following lemma, which says that the stationary weight of the configurations on either side of the cut are within a polynomial factor.

Lemma 4.7. *The stationary weight in the tempering chain of the set* S *is bounded as* $\widehat{\pi}(S) \leq \text{poly}(n)\widehat{\pi}(\overline{S})$.

Proof.

$$\begin{aligned}
\widehat{\pi}(\overline{S}) &= \frac{1}{M+1}\sum_{i\in I}\sum_{x\in\Omega\setminus A}\pi_{\beta_i}(x) \\
&\geq \frac{1}{M+1}\pi_{\beta_c}(\Omega_{2n/3}) \\
&= \frac{1}{M+1}\pi_{\beta_c}(\Omega_{n/3}) \\
\text{(By Lemma 4.6 iii))} \quad &\geq \frac{1}{4n^2}\frac{1}{M+1}\pi_{\beta_c}(\Omega) \\
&\geq \frac{1}{4n^2}\frac{1}{(M+1)^2}\sum_{i}\sum_{x\in A}\pi_i(x) \\
&= \frac{1}{4n^2}\frac{1}{M+1}\widehat{\pi}(S)
\end{aligned}$$

□

With this in hand, we can bound the conductance of the tempering Markov chain at the temperature β_c.

$$\begin{aligned} \Phi_{\overline{S}} &= \frac{F_{\overline{S}}}{C_{\overline{S}}} \\ \text{(By Lemma 4.7)} \quad &\leq \text{poly}(n)\frac{F_{\overline{S}}}{C_S} \\ \text{(By reversibility)} \quad &= \text{poly}(n)\frac{F_S}{C_S} \\ &\leq \text{poly}(n)e^{-c_7 n} \end{aligned}$$

This bounds the conductance since $\Phi \leq \max(\Phi_S, \Phi_{\overline{S}}) \leq e^{c_4 n}$ for some $c_4 > 0$. Finally, note that if the tempering Markov chain is defined with temperatures $\beta_c > \beta_{i_1} > \cdots > \beta_{i_j}$ for any $\{i_1, \cdots i_j\} \subseteq \{0, \cdots M-1\}$, the same arguments show that the conductance of the chain will still be exponentially small. This completes the proof of Theorem 4.3.

Zheng [91] has shown that rapid mixing of the swapping Markov chain implies rapid mixing of the tempering chain. Thus Theorem 4.1 implies that the swapping chain for the mean-field Potts model mixes exponentially torpidly.

4.4.1 Proofs of Technical Lemmas

Recall that $\overline{\pi}_i(x) = \pi_{\beta_i}\left(\frac{n}{2}, xn, \frac{n}{2} - xn\right)$ is the continuous function where x is allowed to vary continuously.

Lemma 4.4. For n sufficiently large, the function $\overline{\pi}_i(x)$ has a unique maximum in the range $0 < x < \frac{1}{2}$ and attains its maximum at $x = \frac{1}{4}$ for all i such that $\beta_i \leq \beta_c$.

Proof. Examining $\overline{\pi}_i$ on this line, we find

$$\overline{\pi}_i(x) = \binom{n}{\frac{n}{2}, xn, \frac{n}{2} - xn} \frac{e^{\beta_i n^2\left(\left(\frac{1}{2}\right)^2 + x^2 + \left(\frac{1}{2} - x\right)^2\right)}}{Z(\beta_i)}.$$

Neglecting factors not dependent on x and simplifying using Stirling's formula, we need to check for the stationary points of the function

$$\frac{f(x)}{g(x)} = \frac{e^{\beta_i n(x^2 + (\frac{1}{2} - x)^2)}}{(x(\frac{1}{2} - x))^{\frac{1}{2n}} x^x (\frac{1}{2} - x)^{(\frac{1}{2} - x)}}.$$

To test the sign of the derivative $\left(\frac{f(x)}{g(x)}\right)'$, we compare the quantities $\frac{f'}{f}$ and $\frac{g'}{g}$, where $\frac{f'}{f} = \beta_i n(4x-1)$ and $\frac{g'}{g} = \ln(\frac{x}{\frac{1}{2}-x}) + \frac{1}{2n}\frac{1-4x}{1-2x}$. At $x = \frac{1}{4}$ we have $\frac{f'}{f} = 0 = \frac{g'}{g}$, and $g \neq 0$, and it can be checked that this is the unique stationary point. Since $\beta_c = 2\ln(2)/n$, $\beta_i n \leq 2\ln(2)$. For $n \geq 100$,

$$\begin{aligned} \beta_c n(4x-1) &> \frac{g'}{g},\ x \in \left(0, \frac{1}{4}\right), \\ \beta_c n(4x-1) &< \frac{g'}{g},\ x \in \left(\frac{1}{4}, \frac{1}{2}\right). \end{aligned}$$

As β_i is decreased, the slope of the line $\frac{f'}{f}$ decreases from the (positive) slope of the line $\beta_c n(4x-1)$. Thus, for a maximum at $x = \frac{1}{4}$, it is sufficient to check that the above inequalities hold at β_c to prove the lemma for $\beta_i < \beta_c$ since $\frac{g'}{g}$ is independent of β_i. □

Lemma 4.6 At $\beta_c = \frac{2\ln 2}{n}$

(i) $\pi_{\beta_c}(\Omega_{n/3}) = \pi_{\beta_c}(\Omega_{2n/3}) + o(1)$.

(ii) $\frac{\pi_{\beta_c}(\Omega_{n/2})}{\pi_{\beta_c}(\Omega_{n/3})} \leq e^{-\Omega(n)}$.

(iii) $\pi_{\beta_c}(\Omega_{n/3}) \geq \frac{\pi_{\beta_c}(\Omega)}{n^2}$.

Proof. (i) We solve for β_c. Let $\pi_{\beta_i}(\Omega_{n/3}) = \pi_{\beta_i}(\Omega_{2n/3})$. Then,

$$\binom{n}{\frac{2n}{3}, \frac{n}{6}, \frac{n}{6}} \frac{e^{\beta_i(\frac{4n^2}{9} + \frac{n^2}{18})}}{Z(\beta_i)} = \binom{n}{\frac{n}{3}, \frac{n}{3}, \frac{n}{3}} \frac{e^{\beta_i(n^2/3)}}{Z(\beta_i)}.$$

This implies

$$\begin{aligned} e^{\beta_i n^2(1/6)} &= \frac{\left(\frac{2n}{3}!\right)\left(\frac{n}{6}!\right)\left(\frac{n}{6}!\right)}{\left(\frac{n}{3}!\right)\left(\frac{n}{3}!\right)\left(\frac{n}{3}!\right)} \\ &= \frac{\left(\frac{2}{3}\right)^{\frac{2n}{3}}\left(\frac{1}{6}\right)^{\frac{n}{3}}}{\left(\frac{1}{3}\right)^{n}} \left(\frac{1}{\sqrt{2}}\right)\left(1 + O(n^{-1})\right) \\ &= \frac{2^{\frac{n}{3}}}{\sqrt{2}}\left(1 + O(n^{-1})\right), \end{aligned}$$

which occurs when

$$\beta_i = \frac{2\ln(2)}{n} + \frac{2}{\sqrt{2}n^2}\ln\left(1 + O(n^{-1})\right).$$

Setting β_c to $\frac{2\ln(2)}{n}$ gives the desired result.

(ii) Let $\beta_c = \frac{2\ln(2)}{n}$. Then we have

$$\begin{aligned}
\frac{\pi_{\beta_c}(\Omega_{n/2})}{\pi_{\beta_c}(\Omega_{n/3})} &= \frac{\binom{n}{\frac{n}{2},\frac{n}{4},\frac{n}{4}} e^{\beta_c(3n^2/8)}}{\binom{n}{\frac{n}{3},\frac{n}{3},\frac{n}{3}} e^{\beta_c(n^2/3)}} \\
&= \frac{\left(\frac{n}{3}!\right)^3}{\left(\frac{n}{2}!\right)\left(\frac{n}{4}!\right)^2} e^{\beta_c n^2/24} \\
&= \sqrt{\frac{27}{32}} \left(\frac{8}{9}\right)^{\frac{n}{2}} e^{\ln(2)n/12} \left(1 + O(n^{-1})\right) \\
&= \sqrt{\frac{27}{32}}\, e^{-\frac{n}{12}\ln\left(\frac{3^{12}}{2^{13}}\right)} \left(1 + O(n^{-1})\right) \leq e^{-\Omega(n)}.
\end{aligned}$$

(iii) Let $\beta_c = \frac{2\ln(2)}{n}$. Consider any general point in the space, which is of the form $(x, y, 1-x-y)$ for $0 \leq x + y \leq 1$. It can be verified by calculation that the function

$$h(x,y) = \frac{f(x,y)}{g(x,y)} = \frac{e^{\beta_c n(x^2+y^2+(1-x-y)^2)}}{(xy(1-x-y))^{\frac{1}{2n}} x^x y^y (1-x-y)^{(1-x-y)}}$$

has a global maximum at $(1/3, 1/3)$, i.e. $h(x,y) \leq h(1/3,1/3)$ for all x, y such that $0 \leq x + y \leq 1$. This can be shown by checking that h is maximized at $(1/3,1/3)$ over all stationary points of $h(x,y)$. This implies that $\pi_{\beta_c}(\Omega_{n/3}) \geq \frac{\pi_{\beta_c}(\Omega)}{n^2}$.

□

4.5 Tempering Can Slow Down Fixed Temperature Algorithms

We have shown that simulated tempering can mix torpidly. In fact, tempering can be slower than the fixed temperature algorithm by more than a polynomial factor. In this section we show that just above the critical inverse temperature, on a restricted part of the state space Ω, simulated tempering can be slower than the fixed temperature Metropolis chain by an exponential factor. The idea is that although exponential, the mixing time of the Metropolis chain at β^* is bounded by the size of the cut at β_*, while the mixing time of the simulated tempering chain can be an exponential multiplicative factor worse because the conductance of the same cut at the higher temperatures is much smaller. Intuitively, on average, the chain is spends even less time mixing on both sides of the cut at the higher temperatures than at β^*.

The precise theorem we show is the following. Recall that $\Omega_{RGB} = \{x \in \Omega \ : \ x_1 \geq x_2, x_3\}$.

Theorem 4.8. *Let $\beta_c = \frac{2\ln(2)}{n} < \beta^* < \frac{3}{2n}$. Assume that the number of distributions for tempering is $M = \Theta(n)$. Then, there are constants $\delta > 0$ and $\alpha < 0$ (which may depend on β^*) such that the simulated tempering algorithm on Ω_{RGB} at β^* mixes only after time $\Omega(e^{(\delta-\alpha)n})$. The Metropolis algorithm at temperature β^* mixes in time $O(e^{-\alpha n+o(1)})$*

4.5.1 Torpid Mixing of Tempering for $\frac{2\ln(2)}{n} < \beta^* < \frac{3}{2n}$

We start by proving the first part of the theorem above by showing the following bound on the conductance of the simulated tempering chain. Let Φ_{RGB} denote the conductance of the tempering chain on Ω_{RGB} at inverse temperature β^*.

Theorem 4.9. *Let $\beta_c = \frac{2\ln(2)}{n} < \beta^* < \frac{3}{2n}$. Then, there exists $\alpha < 0$ and $\delta > 0$ such that $\Phi_{RGB} \leq e^{(\alpha-\delta)n+o(n)}$.*

Define the set $K_{RGB} = \{\sigma = (\sigma_1, \sigma_2, \sigma_3) \text{ where } \sigma_1 \geq \sigma_2 \geq \sigma_3, \ \sum_i \sigma_i = n\}$. Thus K_{RGB} is the set of partitions σ corresponding to the configurations in Ω_{RGB}. For $\sigma \in K_{RGB}$, the Gibbs distribution is given by

$$\pi_{\beta_i}(\sigma) = \binom{n}{\sigma_1, \sigma_2, \sigma_3} \frac{e^{\beta_i(\sigma_1^2+\sigma_2^2+\sigma_3^2)}}{Z_{RGB}(\beta_i)}$$

where $Z_{RGB}(\beta_i)$ is the normalizing constant.

Denote by ℓ_λ, the set of points $\sigma_\lambda = \left(\lambda n, \frac{(1-\lambda)n}{2}, \frac{(1-\lambda)n}{2}\right)$, for $\frac{1}{3} \leq \lambda \leq 1$ i.e., the subset of K_{RGB} with equal numbers of blue and green points (see Figure 12). There exists a constant λ_{min} (which can be found by differentiating the function, as usual), a value of λ between the ordered and disordered modes where $\pi_{\beta^*}(\sigma_\lambda)$ is minimized along the line ℓ_λ. Let $\Omega_{\lambda_{min}}$ be the corresponding set of spin configurations. Let $\beta_M = \beta^* = \frac{\mu}{n}$ where μ is a constant such that $2\ln(2) < \mu < \frac{3}{2}$. Let $A \subseteq \Omega_{RGB}$ be the set of configurations x with $x_1 \leq \lambda_{min}$. Let Let $S = \{(x, i) \mid x \in A, \ \beta_0 \leq \beta_i \leq \beta_c\}$. Let $B = \{x \in A \mid \exists\, x' \in \Omega \setminus A, \ P(x, x') \neq 0\}$ be

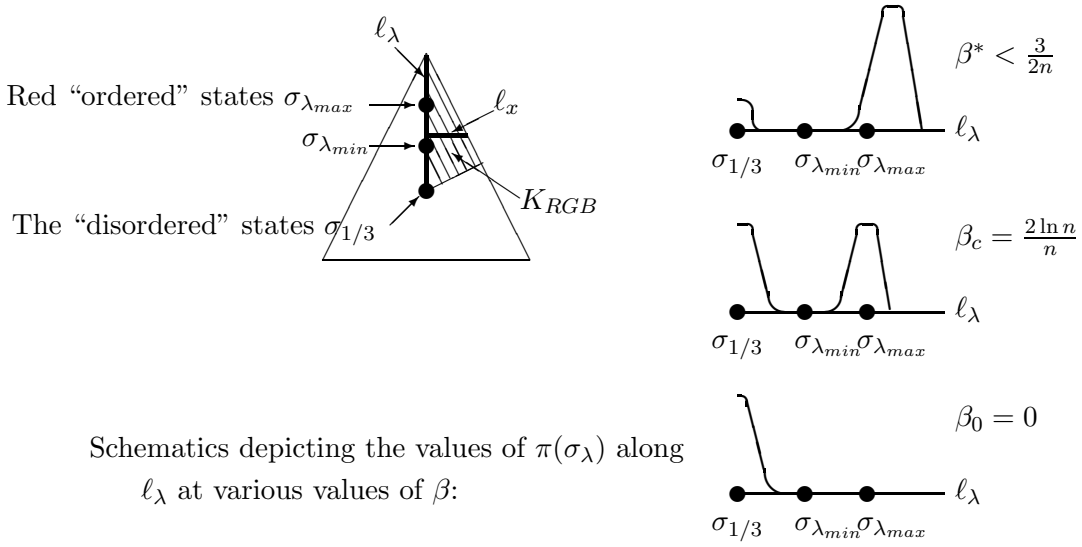

Figure 12: The profile of the probability density function over K_{RGB}

the boundary of A. Then, as before, we can bound the conductance of the set S as follows.

$$\Phi_S \leq \frac{\sum_{i=0}^{M}\sum_{x\in B}\pi_{\beta_i}(x)}{\sum_{i=0}^{M}\sum_{x\in A}\pi_{\beta_i}(x)} \leq O(n)\frac{\sum_{i=0}^{M}\pi_{\beta_i}(\Omega_{\lambda_{min}})}{\sum_{i=0}^{M}\pi_{\beta_i}(\Omega_{1/3})} \tag{15}$$

The second inequality above follows from the fact that the distribution at every temperature is unimodal and is maximized at $\Omega_{\lambda_{min}}$.

Lemma 4.10. *Let $\frac{2\ln(2)}{n} < \beta_* < \frac{3}{2n}$. For n sufficiently large, the continuous function $\overline{\pi}_{\beta_i}(x) = \pi_{\beta_i}(\lambda_{min}n, (1-\lambda_{min}-x)n, xn)$ has a unique maximum in the range $0 \leq x \leq 1-\lambda_{min}$ at $x = \frac{1-\lambda_{min}}{2}$ for all $i \in \{1, \cdots, M\}$.*

Rewriting the last expression in (15), we have

$$\Phi_S \leq O(n)\frac{\pi_{\beta_M}(\Omega_{\lambda_{min}})}{\pi_{\beta_M}(\Omega_{1/3})}\frac{\left(\left(\frac{\pi_{\beta_M}(\Omega_{\lambda_{min}})}{\pi_{\beta_M}(\Omega_{\lambda_{min}})}\right)+\left(\frac{\pi_{\beta_{M-1}}(\Omega_{\lambda_{min}})}{\pi_{\beta_M}(\Omega_{\lambda_{min}})}\right)+\cdots+\left(\frac{\pi_{\beta_0}(\Omega_{\lambda_{min}})}{\pi_{\beta_M}(\Omega_{\lambda_{min}})}\right)\right)}{\left(\left(\frac{\pi_{\beta_M}(\Omega_{1/3})}{\pi_{\beta_M}(\Omega_{1/3})}\right)+\left(\frac{\pi_{\beta_{M-1}}(\Omega_{1/3})}{\pi_{\beta_M}(\Omega_{1/3})}\right)+\cdots+\left(\frac{\pi_{\beta_0}(\Omega_{1/3})}{\pi_{\beta_M}(\Omega_{1/3})}\right)\right)} \tag{16}$$

We use the following properties of the stationary distribution to bound the conductance. The first fact is that the stationary weight of the disordered mode conditioned on being at a particular temperature is non-decreasing as we decrease β.

Lemma 4.11. *For $i \in \{1, \cdots, M\}$, we have $\pi_{\beta_i}(\Omega_{1/3}) \leq \pi_{\beta_{i-1}}(\Omega_{1/3})$.*

Proof. This follows from the fact that $H(\sigma)$ is minimized at $\sigma_1 = \sigma_2 = \sigma_3 = 1/3$. □

Next, we observe that the height of the disordered mode increases faster than the height at $\Omega_{\lambda_{min}}$.

Lemma 4.12. *There is a constant $d > 1$ such that $\frac{\pi_{\beta_{i-1}}(\Omega_{1/3})}{\pi_{\beta_i}(\Omega_{1/3})} > d \cdot \frac{\pi_{\beta_{i-1}}(\Omega_{\lambda_{min}})}{\pi_{\beta_i}(\Omega_{\lambda_{min}})}$.*

Proof. Expanding the terms reveals that

$$\frac{\pi_{\beta_{i-1}}(\Omega_{1/3})/\pi_{\beta_i}(\Omega_{1/3})}{\pi_{\beta_{i-1}}(\Omega_{\lambda_{min}})/\pi_{\beta_i}(\Omega_{\lambda_{min}})} = \frac{\pi_{\beta_{i-1}(\Omega_{1/3})}/\pi_{\beta_{i-1}}(\Omega_{\lambda_{min}})}{\pi_{\beta_i(\Omega_{1/3})}/\pi_{\beta_i}(\Omega_{\lambda_{min}})} = e^{(\beta_i-\beta_{i-1})(H(\Omega_{\lambda_{min}})-H(\Omega_{1/3}))}$$

Recall that $\beta_i - \beta_{i-1} = O(\frac{1}{nM})$ while $H(\sigma_{\lambda_{min}}) - H(\sigma_{1/3}) = \Omega(n^2)$, since λ_{min} is a constant. The claim follows since $M = \Theta(n)$. □

Lemma 4.13. *The density $\pi_{\beta^*}(\Omega_{1/3})$ is exponentially smaller than $\pi_{\beta^*}(\Omega_{2/3})$*

Proof. The claim follows by the setting $\beta^* > \frac{2\ln(2)}{n}$

$$\frac{\pi_{\beta^*}(\Omega_{1/3})}{\pi_{\beta^*}(\Omega_{2/3})} = \frac{e^{\beta^* \frac{n^2}{3} - n\ln\frac{n}{3}}}{e^{\beta^* \frac{n^2}{2} - \left(\frac{2n}{3}\ln\frac{2n}{3} + \frac{n}{3}\ln\frac{n}{6}\right)}} = e^{-\beta^* \frac{n^2}{6} + \frac{n}{3}\ln(2)}.$$

□

A corollary of the above lemma is that at the inverse temperature β^* the weight of the set $\Omega_{1/3}$ is an exponentially small fraction of the total weight. On the other hand, we know that at $\beta_0 = 0$, the weight is at least a polynomial fraction of the total weight. Therefore, by Lemma 4.13, the sequence in the denominator of (16) grows from 1 to at least d_1^n for some constant $d_1 > 1$ in $M = O(n)$ terms. Let d_2 be the smallest constant by which any two consecutive terms of the sequence in the denominator differ. By Lemma 4.11, and the previous statement, $d_2 > 1$.

By Lemma 4.12, the rate of increase of terms in the series in the denominator of (4) is at least a constant, $d > 1$, times the rate of increase of terms in the series in numerator. Hence, for some constant $d_3 > 0$, (15) implies

$$\begin{aligned}
\Phi_\lambda &\leq O(n)\frac{\pi_{\beta_M}(\Omega_{\lambda_{min}})}{\pi_{\beta_M}(\Omega_{1/3})}\frac{\left(1 + \left(\frac{d_2}{d}\right) + \cdots + \left(\frac{d_2}{d}\right)^{d_3 n}\right)}{1 + d_2 + \cdots + d_2^{d_3 n}} \\
&\leq O(n)\frac{\pi_{\beta_M}(\Omega_{\lambda_{min}})}{\pi_{\beta_M}(\Omega_{1/3})}(\min(d_2, d))^{-d_3 n}.
\end{aligned}$$

Theorem 4.9 now follows by setting $\delta = \frac{d}{3}\ln(\min(d_2, d))$.

4.5.2 Proof of the Technical Lemma 4.10

Lemma 4.10 Let $\frac{2\ln(2)}{n} < \beta_* < \frac{3}{2n}$. For n sufficiently large, the continuous function $\overline{\pi}_{\beta_i}(x) = \pi_{\beta_i}(\lambda_{min}n, (1-\lambda_{min}-x)n, xn)$ has a unique maximum in the range $0 \leq x \leq 1-\lambda_{min}$ at $x = \frac{1-\lambda_{min}}{2}$ for all $i \in \{1, \cdots, M\}$.

Proof.

$$\overline{\pi}_i(x) = \binom{n}{\lambda_{min}n, (1-\lambda_{min}-x)n, xn} \frac{e^{\beta_i n^2\left({\lambda_{min}}^2+x^2+(1-\lambda_{min}-x)^2\right)}}{Z(\beta_i)}.$$

Neglecting factors not dependent on x, we need to check for the stationary points of the function

$$\frac{f(x)}{g(x)} = \frac{e^{\beta_i n(x^2+(1-\lambda_{min}-x)^2)}}{(x(1-\lambda_{min}-x))^{\frac{1}{2n}} x^x (1-\lambda_{min}-x)^{1-\lambda_{min}-x}}.$$

Differentiating, we have

$$\begin{aligned} \frac{f'}{f} &= \beta_i n(4x - 2(1-\lambda_{min})) \\ \frac{g'}{g} &= \ln(\frac{x}{1-\lambda_{min}-x}) + \frac{1}{2n}\frac{1-\lambda_{min}-2x}{x(1-\lambda_{min}-x)} \end{aligned}$$

At $x = \frac{1-\lambda_{min}}{2}$ we have $\frac{f'}{f} = 0 = \frac{g'}{g}$, and $g \neq 0$, giving a stationary point. Let $\beta^* = \frac{2\ln(2)+\varepsilon}{n}$ for some $0 < \varepsilon \leq 3/2 - \ln(2)$, and thus $\beta_i n \leq 2\ln(2) + \varepsilon$. For $n \geq 100$,

$$\begin{aligned} \beta^* n(4x - 2(1-\lambda_{min})) &> \frac{g'}{g},\ x \in \left(0, \frac{1-\lambda_{min}}{2}\right), \\ \beta^* n(4x - 2(1-\lambda_{min})) &< \frac{g'}{g},\ x \in \left(\frac{1-\lambda_{min}}{2}, 1-\lambda_{min}\right). \end{aligned}$$

As β_i is decreased, the slope of the line $\frac{f'}{f}$ decreases from the (positive) slope of the line $\beta^* n(4x - 2(1-\lambda_{min}))$. Thus, it is sufficient that the above inequalities hold at $\beta^* = \frac{3}{2n}$ to prove the lemma for $\beta_i < \beta^*$ since $\frac{g'}{g}$ is independent of β_i. □

4.5.3 Upper bound for the Mixing Time of the Metropolis Algorithm on Ω_{RGB}.

The Metropolis Markov chain on Ω at is known to have exponential mixing time and the same argument also holds on Ω_{RGB}. We would now like to derive a good upper bound on this mixing time so that we can compare it to the tempering chain. However, bounding

the conductance and applying Theorem 2.2 will not be sufficient as the square of the conductance gives too weak a bound. Instead, to obtain the best possible lower bound on the spectral gap of the Metropolis chain, we appeal to the comparison theorem [23]. We use this technique to obtain a tight exponential upper bound for the mixing time. Let P be the Metropolis chain on Ω_{RGB} with stationary distribution $\pi = \pi_{\beta^*}$. Then, the second part of Theorem 4.8 is as follows.

Theorem 4.14. *Let $\beta_c < \beta^* < \frac{3}{2n}$ and let $\alpha = \ln\left(\pi_{\beta^*}(\Omega_{\lambda_{min}})/\pi_{\beta^*}(\Omega_{1/3})\right) < 0$. The Markov chain P mixes in time $O(e^{-\alpha n + o(1)})$.*

The comparison theorem of Diaconis and Saloff-Coste is also useful in bounding the mixing time of a Markov chain when the mixing time of a related chain on the same space, but with possibly a different stationary distribution is known. Let $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be two Markov chains on Ω. Let P_1 and π_1 be the transition matrix and stationary distributions of $\mathfrak{M}_1$ and let P_2 and π_2 be those of $\mathfrak{M}_2$. Let $E(P_1) = \{(x, y) : P_1(x, y) > 0\}$ and $E(P_2) = \{(x, y) : P_2(x, y) > 0\}$ be sets of directed edges. For $x, y \in \Omega$ such that $P_2(x, y) > 0$, define a *path* γ_{xy}, a sequence of states $x = x_0, \cdots, x_k = y$ such that $P_1(x_i, x_{i+1}) > 0$. Finally, let $\Gamma(z, w) = \{(x, y) \in E(P_2) : (z, w) \in \gamma_{xy}\}$ denote the set of endpoints of paths that use the edge (z, w).

Theorem 4.15 ([23]). *Let $a = \min_x \left(\frac{\pi_2(x)}{\pi_1(x)}\right)$. Then*

$$Gap(P_1) \geq \frac{a}{A} \cdot Gap(P_2),$$

where

$$A = \max_{(z,w)\in E(P_1)} \left\{ \frac{1}{\pi_1(z)P_2(z, w)} \sum_{\Gamma(z,w)} |\gamma_{xy}|\pi_2(x)P_2(x, y) \right\}.$$

The idea behind showing the mixing time claimed in Theorem 4.14 is to define a new distribution $\widetilde{\pi}$ on K_{RGB} by essentially eliminating the disordered mode. The Metropolis chain $\widetilde{P}$ is defined on Ω_{RGB} with stationary distribution $\widetilde{\pi}$. We will show that the mixing time of $\widetilde{P}$ is at most a polynomial. The comparison theorem then gives the required upper bound on the mixing time of the Metropolis chain P in Theorem 4.14. For $\sigma \in K_{RGB}$ define

$$\widetilde{\pi}(\Omega_\sigma) = \begin{cases} \pi_{\beta^*}(\Omega_{\lambda_{min}}) Z_{RBG}(\beta^*)/\widetilde{Z} & \text{if} \quad \sigma_1 < t_* n \quad \text{and} \quad \pi_{\beta^*}(\Omega_\sigma) \geq \pi_{\beta^*}(\Omega_{\lambda_{min}}), \\ \pi_{\beta^*}(\Omega_\sigma) Z_{RGB}(\beta^*)/\widetilde{Z} & \text{otherwise,} \end{cases}$$

where

$$\widetilde{Z} = Z_{RGB}(\beta^*) \left(\sum_{\sigma \in K_1} \pi_{\beta^*}(\Omega_{\lambda_{min}}) + \sum_{\sigma \in K_2} \pi_{\beta^*}(\Omega_\sigma) \right)$$

is the normalizing partition function and the sets K_1, K_2 partition K_{RGB} into the flattened and unchanged configurations respectively.

For a configuration $x \in \Omega_{RGB}$, we define $\widetilde{\pi}(x)$ to be uniform over all the configurations in the same equivalence class, i.e., if x is in the equivalence class σ

$$\widetilde{\pi}(x) = \binom{n}{\sigma_1 \sigma_2 \sigma_3}^{-1} \widetilde{\pi}(\Omega_\sigma).$$

The first step is to show that $\widetilde{P}$, the Metropolis chain on the flattened distribution, mixes in polynomial time. This will follow from an application of the decomposition theorem [66]. The second step will be to use this bound and the comparison theorem to bound the mixing time of the chain on the original unflattened space. This mixing time of $\widetilde{P}$ will be a lower order term when we compare it to the mixing time of P, which is exponential. Thus, any polynomial bound on the mixing rate of $\widetilde{P}$ will suffice.

Theorem 4.16. *The Markov chain $\widetilde{P}$ with stationary distribution $\widetilde{\pi}$ mixes in polynomial time.*

To apply the decomposition theorem, we partition the space Ω_{RGB} according to the equivalence classes of configurations, i.e. into the space K_{RGB}. Informally, the decomposition theorem states that the mixing rate of a Markov chain on Ω is at most the product of the mixing rate of the chain restricted to Ω_σ (the *restrictions*) and the mixing rate of the chain on the quadratic sized set K_{RGB} (the *projection*).

Instead of $\widetilde{P}$, it will be simpler to bound the mixing time of $Q = \widetilde{P}^2$, the two step transition matrix that allows moves of length 0, 1 or 2. We can then infer the polynomial mixing of $\widetilde{P}$ from the polynomial mixing of Q. It is easy to see that Q is rapidly mixing when restricted to Ω_σ, for any σ, because two-step moves permute the colors on the vertices

without changing the total number of each. Hence, we focus on showing the bound on projection Markov chain $\overline{Q}$. We will use the canonical path method.

Theorem 4.17. *The Markov chain $\overline{Q}$ on the projection of K_{RGB} is rapidly mixing.*

Proof. Define canonical paths in the chain $\overline{Q}$, $\{\gamma_{\sigma\tau}\}$ as follows:
Let $\sigma = (t_1, b_1, g_1)$ and $\tau = (t_2, b_2, g_2)$. We assume that $t_1 \geq t_2$. If not, the path from σ to τ consists of the same vertices as the path from τ to σ but with all edges directed oppositely.

We define the canonical path for t_1 odd and t_2, the other case only needs a minor technical modification due to parity issues. Assume (without loss of generality by the symmetry of the colors blue and green) $b_1 \leq g_1$ and $b_2 \geq g_2$. The path $\gamma_{\sigma\tau}$ is defined to be $(t_1, b_1, g_1), (t_1, b_1 + 1, g_1 - 1), \cdots, (t_1, \frac{n-t_1-1}{2}, \frac{n-t_1+1}{2}), (t_1 - 1, \frac{n-t_1+1}{2}, \frac{n-t_1+1}{2}), (t_1 - 3, \frac{n-t_1+3}{2}, \frac{n-t_1+3}{2}), \cdots (t_2, \frac{n-t_2}{2}, \frac{n-t_2}{2}), \cdots, (t_2, b_2 - 1, g_2 + 1), (t_2, b_2, g_2)$. It can be shown that along the path, the distribution is "unimodal", i.e.,

Lemma 4.18. *For each $\sigma = (t_1, b_1, g_1), \tau = (t_2, b_2, g_2) \in K_{RGB}$, the distribution $\widetilde{\pi}$ attains a unique maximum on the path $\gamma_{\sigma,\tau}$.*

We defer the proof till the end of this argument. Assuming the lemma, the congestion of the paths can be bounded as follows.

$$\begin{aligned} A &= \max_{(\alpha,\beta)\in E(\overline{Q})} \left\{ \frac{1}{\widetilde{\pi}(\Omega_\alpha) P^2(\Omega_\alpha, \Omega_\beta)} \sum_{\Gamma(\alpha,\beta)} |\gamma_{\sigma\tau}| \min(\widetilde{\pi}(\Omega_\sigma), \widetilde{\pi}(\Omega_\tau)) \right\} \\ &= \max_{(\alpha,\beta)\in E(\overline{Q})} \left\{ \frac{1}{\min(\widetilde{\pi}(\Omega_\alpha), \widetilde{\pi}(\Omega_\beta))} \sum_{\Gamma(\alpha,\beta)} |\gamma_{\sigma\tau}| \min(\widetilde{\pi}(\Omega_\sigma), \widetilde{\pi}(\Omega_\tau)) \right\} \end{aligned}$$

Since along every canonical path the distribution is unimodal, and the length of any path is at most linear in n, and there are at most polynomially many paths $\Gamma(\alpha, \beta)$ using the edge (α, β), A is at most a polynomial in n. □

Corollary 4.19. *The Markov chain $\widetilde{P}$ on K_{RGB} is rapidly mixing.*

Proof of Lemma 4.18: Let ℓ_t denote the set of $\sigma \in K_{RGB}$ such that $\sigma_1 = t$. Let $\ell_{b=g}$ denote the set consisting of configurations where the number of green and blue vertices are equal.

Since the space is discrete, because of parity considerations, the canonical paths cannot simply go along the line ℓ_{t_1}, then along the line $\ell_{b=g}$ and finally along ℓ_{t_2}, except in the case that t_1 and t_2 are both even. For this case, it is sufficient to show that firstly, for all $1/3 \leq t \leq 1$, along the lines ℓ_t, the maximum is at the intersection with $\ell_{b=g}$ and secondly, along the line $\ell_{b=g}$, the distribution is unimodal. The observation is that the second fact implies that on the portion of the canonical path along $\ell_{b=g}$, the distribution is either

i) non-increasing

ii) non-decreasing

iii) non-decreasing and then non-increasing

but not decreasing and then increasing. Then in any of the three case above, it can be verified that there is a unique local maximum along the path.

In the other cases, when either both t_1 and t_2 are odd, or one is odd and the other even, the canonical path makes a "diagonal" move to switch parity and we have to argue that the property of being unimodal is not violated. It turns out that this is implied by the unimodality of the continuous function $\widetilde{\pi}$ on the lines ℓ_t and $\ell_{b=g}$. We first show that along the lines ℓ_t and ℓ_λ the distribution $\widetilde{\pi}$ is unimodal.

Claim 4.20. *Let $\beta_c < \beta^* < \frac{3}{2n}$ and $L_t = \{\sigma | \sigma \in \Omega_{RGB}, \sigma_1 = t\}$. Then there exists a constant n_0, such that $\forall n \geq n_0$ the function $\widetilde{\pi}(\sigma)$ when restricted to L_t is maximized at $\sigma_2 = \sigma_3 = \frac{n-t}{2}$ and is non-increasing as σ_3 decreases, $\forall t$ such that $L_t \subseteq \Omega_{RGB}$.*

Proof. Consider the original distribution π

$$\pi_{\beta^*}(tn, xn, (1-t-x)n) = \binom{n}{tn, xn, (1-t-x)n} \frac{e^{\beta^* n^2((t)^2+x^2+(1-t-x)^2)}}{Z(\beta_i)}$$

Neglecting factors not dependent on x, the expression can be simplified using Stirling's formula, to check for the stationary points of the function

$$\frac{f(x)}{g(x)} = \frac{e^{2\beta^* n(x^2-(1-t)x))}}{(x(1-t-x))^{\frac{1}{2n}} x^x (1-t-x)^{(1-t-x)}}$$

To test the sign of the derivative we compare the quantities $\frac{f'}{f}$ and $\frac{g'}{g}$, at β^* where $\frac{f'}{f} = 2\beta^* n(2x-(1-t))$ and $\frac{g'}{g} = \ln(\frac{x}{1-t-x}) + \frac{1}{2n}\frac{1-t-2x}{x(1-t-x)}$.
At $x = \frac{1-t}{2}$ we have $\frac{f'}{f} = 0 = \frac{g'}{g}$, and $g \neq 0$, giving a stationary point. For $n \geq 100$, $\frac{1-t}{2} < x < 1-t$, we have $\frac{f'}{f} < \frac{g'}{g}$, with $g \neq 0$. This can be seen by comparing the growth rate of these functions in the specified interval, given that they take the same value at $x = \frac{1-t}{2}$, and is true for $\beta_c < \beta^* < \frac{3}{2n}$. The proof of the lemma now follows from the definition of $\widetilde{\pi}_R$. □

Claim 4.21. *Let $\beta_c < \beta^* < \frac{3}{2n}$. For n sufficiently large, $\widetilde{\pi}_{\beta^*}(\Omega_\lambda)$ has a unique maximum λ_{max} and is non-increasing on either side of it.*

Proof. We examine the continuous extension $\overline{\pi}$ of the original distribution π.

$$\overline{\pi}_{\beta^*}\left(\lambda n, \frac{(1-\lambda)n}{2}, \frac{(1-\lambda)n}{2}\right) = \binom{n}{\lambda n, \frac{(1-\lambda)n}{2}, \frac{(1-\lambda)n}{2}} \frac{e^{\beta^* n^2\left(\lambda^2 + 2\left(\frac{1-\lambda}{2}\right)^2\right)}}{Z(\beta^*)}$$

Neglecting factors not explicitly dependent on λ, asymptotically, we obtain the function

$$e^{\frac{\beta^* n^2}{2}(3\lambda^2 - 2\lambda) - \lambda n \ln(\lambda) - (1-\lambda)n\ln\left(\frac{1-\lambda}{2}\right)}$$

The claim can be verified by differentiating it, solving for the stationary point λ_{max}, and checking the second derivative. By construction, $\widetilde{\pi}_{\beta^*}$ is non-increasing on either side of λ_{max} for $\frac{1}{3} \leq \lambda \leq 1$. □

Finally, along the "diagonal" portions of the path the change in the value of the distribution will be the net change if we were to move in a continuous fashion horizontally and then vertically. Since along both these segments the change will be of the same sign if the segments on either end are of the same type (increasing or decreasing), by the two claims above, the net change will be positive or negative as required by unimodality. □

The Metropolis chain at β^* mixes torpidly, and by the above lemmas we can bound the mixing time. Note that the proof uses a stronger version of the Comparison Theorem.

To use the comparison theorem to infer a bound on the mixing time of P from that of $\widetilde{P}$ we need good bounds on the parameters A and a. It turns out that A is the insignificant factor in the mixing time, rather, a determines the mixing time of P. In contrast, most

previous applications of the comparison theorem consider chains with identical stationary distributions, so typically the parameter $a = 1$.

Proof of Theorem 4.14. We will use the refined comparison theorem of Diaconis and Saloff-Coste, Theorem 4.15. Note that the two Markov kernels are identical, but their stationary distributions are very different near the disordered state. Since the kernels are identical, we can simply define trivial canonical paths, i.e., when we decompose a step in the unknown chain $\overline{Q}$ with stationary distribution π_{β^*} into a path using steps from the known chain $\overline{Q}$ with distribution $\widetilde{\pi}$, these paths all have length 1. It can be verified that the Metropolis transition probabilities on the two chains are always within a polynomial factor of each other and $\max_x(\widetilde{\pi}(x)/\pi(x))$ is at most a polynomial since flattening the distribution has a negligible effect on the partition function.

Claim 4.22. *For* $2\ln(2)/n < \beta^* < 3/3n$,

$$Z_{RGB}(\beta^*)/\mathrm{poly}(n) \leq \widetilde{Z} \leq Z_{RBG}(\beta^*).$$

Proof. The upper bound is easy to see by the definition of $\widetilde{Z}$. By the construction of the flattened distribution, $\widetilde{Z} \leq Z_{RGB}(\beta^*)$. For the lower bound, we have

$$\begin{aligned}
\widetilde{Z} &= Z_{RGB}(\beta^*)\left(\sum_{\sigma\in K_1}\pi_{\beta^*}(\Omega_{\lambda_{min}}) + \sum_{\sigma\in K_2}\pi_{\beta^*}(\Omega_\sigma)\right) \\
&\geq Z_{RGB}(\beta^*)\left(\sum_{\sigma\in K_2}\pi_{\beta^*}(\Omega_\sigma)\right) \\
&\geq Z_{RGB}(\beta^*)/\mathrm{poly}(n)
\end{aligned}$$

The last inequality follows because for $\beta^* > \beta_c$, the stationary probability on K_2 is at least $1/\mathrm{poly}(n)$ of the total measure. □

Hence the parameter A is bounded by a polynomial.

Finally, we can compare the largest variation in the distributions π and $\widetilde{\pi}$ to bound a. Let x be any configuration in $\sigma_{1/3}$, any x_* a configuration in $\sigma_{\lambda_{min}}$ we have

$$a = \frac{\widetilde{\pi}_{\beta^*}(x)}{\pi_{\beta^*}(x)} = \frac{\pi_{\beta^*}(\Omega_{\lambda_{min}})Z_{RGB}/\widetilde{Z}}{\pi_{\beta^*}(\Omega_{1/3})} \geq \frac{\pi_{\beta^*}(\Omega_{\lambda_{min}})}{\pi_{\beta^*}(\Omega_{1/3})}\frac{1}{\mathrm{poly}(n)}$$

Putting these bounds into the comparison theorem (Theorem 4.15) then implies Theorem 4.14.

4.6 Speeding up Simulated Tempering

The slow mixing results of the previous sections give insight into how to speed up the mixing time of the simulated tempering Markov chain in the special case of the complete graph. It turns out that in this case, the barrier to mixing at the critical inverse temperature β_c is essentially the persistence of the disordered mode at the highest temperature.

For every inverse temperature $\beta = \frac{\mu}{n}$ for some constant μ, we define a modified sequence of simulated tempering distributions to sample from configurations of the 3-state ferromagnetic Potts model at that temperature. Set M be a polynomial that grows as $\Omega(n)$. We define the modified simulated tempering distributions as follows. Let $\beta_i = \beta\frac{i}{M}$ and for $x \in \Omega$ let

$$\rho_M(x) = \pi_{\beta_M}(x) = \frac{e^{\beta_M H(x)}}{Z(\beta_M)},$$

where

$$Z(\beta_M) = \sum_{\sigma} \binom{n}{\sigma_1\ \sigma_2\ \sigma_3} e^{\beta_M(\sigma_1^2+\sigma_2^2+\sigma_3^2)}.$$

For $0 \le i \le M-1$, the modified distributions are defined as follows

$$\rho_i(x) = \frac{\binom{n}{x_1\ x_2\ x_3}^{\frac{i-M}{M}} \rho_M^{\frac{i}{M}}(x)}{Z_i}$$

where

$$Z_i = \sum_{\sigma} \rho_M^{\frac{i}{M}}(\Omega_\sigma).$$

Note that

$$\rho_i(\Omega_\sigma) = \frac{\rho_M^{\frac{i}{M}}(\Omega_\sigma)}{\sum_{\sigma} \rho_M^{\frac{i}{M}}(\Omega_\sigma)}.$$

Theorem 4.23. *Let $\beta = \frac{\mu}{n}$ for a constant $\mu > 0$. Then, for some constant $c_8 > 0$ the simulated tempering Markov chain $\widehat{P}$ with the distributions $\rho_0, \cdots, \rho_M$ defined above mixes in time $O(n^{c_8})$.*

Proof. The proof makes use of the decomposition theorem. The strategy is to partition the state space of the tempering chain $\widehat{\Omega}$ into the sets (Ω_σ, i) (abbreviated (σ, i)) for each equivalence class of configurations σ and inverse temperature β_i.

The restrictions (the sets (Ω_σ, i)) are not connected by the chain $\widehat{P}$ since it only moves between configurations which differ in the spin at exactly one vertex. We can get around this technicality by first bounding the mixing time of the 2-step chain $\widehat{P}^2$.

For $\widehat{P}^2$, it is easy to see that each restriction mixes in polynomial time since the acceptance probabilities at a fixed temperature are always at least inverse polynomial, by the choice of β.

We analyze the projection by comparison to the complete graph on the states of the projection $\{(\sigma, i)\}$. For every pair of states $(sigma, i)$ and (σ', j), we define a path using edges of $\widehat{P}^2$ and show that the congestion of these paths is at most a polynomial.

Assume without loss of generality that $i \leq j$. The path between $(sigma, i)$ and (σ', j) is defined to be the sequence of states $(\sigma, i), (\sigma, i-1), \cdots, (\sigma, 0), \tau(\sigma, \sigma'), (\sigma', 0), \cdots, (sigma', j)$. Here $\tau(\sigma, \sigma')$ is a sequence of $O(n)$ states that is the set of vertices along a shortest path using edges of the projection chain in $(\Omega, 0)$ from Ω_σ to $\Omega_{\sigma'}$, not including the endpoints.

The observations we use to bound the congestion of the paths by a polynomial is as follows.

i) Let $\sigma_{\max}$ be an equivalence class of configurations maximizing $\rho_M(\Omega_\sigma)$. For any i,

$$\rho_M^{\frac{i}{M}}(\Omega_{\sigma_{\max}}) \leq Z_i \leq \text{poly}(n)\rho_M^{\frac{i}{M}}(\Omega_{\sigma_{\max}}).$$

ii) For any edge in the kernel of the Markov chain, the number of paths which are routed through it is at most $O(n^4M^2) \leq \text{poly}(n)$, taking into account the possible starting and ending states.

Then, the congestion of the paths can be bounded as follows. We divide into two cases. The first where an edge corresponds to a change in the temperature and is of the form $(\sigma, i'), (\sigma, i'-1)$ for some $i' \leq i$ (or $(\sigma, j'), (\sigma, j'+1)$ for some$j' < j$). The second is an edge corresponding to a pair of adjacent states at the inverse temperature β_0.

- Assume that the edge is of the form $(\sigma, i'), (\sigma, i'-1)$ for some $i' \leq i$. By the observations i) and ii) above,

$$\begin{aligned} A &\leq \text{poly}(n) \frac{\min\left(\frac{\rho_M^{i/M}(\Omega_\sigma)}{\rho_M^{i/M}(\Omega_{\sigma_{\max}})}, \frac{\rho_M^{j/M}(\Omega_{\sigma'})}{\rho_M^{j/M}(\Omega_{\sigma_{\max}})}\right)}{\min\left(\frac{\rho_M^{i'/M}(\Omega_\sigma)}{\rho_M^{i'/M}(\Omega_{\sigma_{\max}})}, \frac{\rho_M^{(i'-1)/M}(\Omega_{\sigma'})}{\rho_M^{(i'-1)/M}(\Omega_{\sigma_{\max}})}\right)} \\ &\leq \text{poly}(n) \left[\frac{\rho_M(\Omega_\sigma)}{\rho_M(\Omega_{\sigma_{\max}})}\right]^{\frac{i-i'}{M}} \leq \text{poly}(n) \end{aligned}$$

  The other case is analogous.

- Suppose that the edge is a pair of adjacent states at β_0. Since for every σ, $\rho_0(\Omega_\sigma) = \Theta(n^{-2})$, we have

$$A \leq \text{poly}(n) \frac{\min\left(\frac{\rho_M^{i/M}(\Omega_\sigma)}{\rho_M^{i/M}(\Omega_{\sigma_{\max}})}, \frac{\rho_M^{j/M}(\Omega_{\sigma'})}{\rho_M^{j/M}(\Omega_{\sigma_{\max}})}\right)}{n^{-2}} \leq \text{poly}(n)$$

Finally, by applying the comparison theorem, the polynomial mixing time of $\widehat{P}^2$ implies that the mixing time of $\widehat{P}$ is at most a polynomial. This follows since for any two adjacent states of $\widehat{P}$, the ratio of the stationary probabilities is at least an inverse polynomial. Moreover, for any edge of the 1-step chain, there are at most a polynomial number of possibilities for the other step. □

Remarks: Though the above modified tempering algorithm is very specific to the mean-field Potts model, it shows that there can be a lot of flexibility in deciding the tempering distributions. Secondly, preliminary calculations indicate that this argument extends to the case of the q-state Potts model for $q > 3$. In practice, it is usually the swapping algorithm and not tempering which is implemented, since one can avoid computing the normalizing constants. By the slow mixing results of the previous section, the swapping algorithm can be shown to mix torpidly as well. However, it is possible to define a modified swapping algorithm for mean-field models and for this algorithm, and in [9] it is shown that for bimodal mean-field models, it mixes rapidly. Preliminary computations indicate that the modified swapping Markov chain mixes rapidly for the mean-field ferromagnetic 3-state Potts model.

CHAPTER V

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we have examined applications of simulated annealing and tempering to counting and sampling problems. The annealing algorithm for binary contingency tables suggests that one could perhaps exploit the combinatorial structure of other problems in order to apply these techniques when efficient algorithms for sampling are not known. Our negative results give insight into the mechanisms that can cause tempering and annealing algorithms to fail. It points to the need for a better understanding of how to design these algorithms. Below we summarize some problems for which finding algorithms or showing hardness could give more insight into the use of these methods.

5.1 Matchings and Related Problems

Perfect Matchings: One of the most important open problems in the field of approximate counting is counting the number of perfect matchings in a graph. The state of the art for this problem are FPRAS's in the cases when the graph is bipartite [47], or when the graph is *dense*, meaning each vertex is of degree at least $n/2$ [44]. An annealing algorithm analogous to the one defined by Jerrum, Sinclair and Vigoda for bipartite graphs can be defined for this problem as well. The difficulty is in showing the rapid mixing of the Markov chain. Is there a way to refine the weights that need to be computed to take into account the presence of odd cycles? A first problem to attempt might be when there are only $O(\ln n)$ odd cycles in the graph.

Graphs With Given Degrees: The problem of counting the number of graphs with a given degree sequence can be reduced to the problem of counting perfect matchings, just as in the bipartite case. The reverse reduction is not known. Hence it may be an easier problem to extend the Markov chain approach to this problem first. At this time, an algorithm

is known only in the case that the degrees are all nearly equal [45, 19] or when the degrees are bounded [4, 54]. As in the case of the permanent algorithm of [47], our analysis here of the convergence of the Markov chain on perfect and near-perfect graphs uses bipartiteness crucially.

Integer Contingency Tables: A related outstanding open problem is counting integer contingency tables. Morris proves [68] that if the row and column sums satisfy $r_i = \Omega(n^{3/2} m \ln m)$ and $c_i = \Omega(m^{3/2} n \ln n)$, then sampling contingency tables can be reduced to sampling from a convex body, for which there is a large body of work (see for instance [59]). Cryan et al., in [20] show rapid mixing for a Markov chain that samples tables if the number of rows is any constant. Dyer [26] gives an algorithm using dynamic programming, also in the case where the number of rows is constant. Integer contingency tables with fixed row and column sums are bipartite multigraphs with given degrees. Can the annealing approach be extended for sampling from integer contingency tables? Though an analogue of the greedy graph can be constructed in this case, the analysis of the convergence of the Markov chain doesn't go through. This is because the weights of the (multi)graphs must be defined to take into account the multiplicities of edges and the corresponding combinatorial inequalities no longer hold. Lastly, can any hardness of approximate counting be shown for this problem?

5.2 Complexity of Simulated Annealing and Tempering

What is the relative complexity of methods related to annealing? Zheng [91] has shown that if the swapping Markov chain converges, then so does simulated tempering, but the converse is not known. Another result of this flavor is by Madras and Piccioni [61] showing that the simulated tempering Markov chain is equivalent to importance sampling with the average of the tempering distributions.

Is it clear that if annealing converges then tempering with the same underlying Markov chain converges as well, i.e., is there any advantage to randomizing the temperature parameter? It would be interesting to construct natural examples where annealing fails but

tempering succeeds in sampling. Are there examples where annealing converges, but tempering fails?

5.3 Hardness of Approximate Counting

Independent Sets: It is known that it is NP-hard to approximate the number independent sets even in constant degree graphs [60, 27]. An open question is whether this remains true if the graph is bipartite. Dyer, Frieze and Jerrum, in [27] show that there is a bipartite graph of maximum degree 6 such that any local Markov chain for sampling independent sets of the graph will mix torpidly. In Section 1.3.1, we presented a simple argument, with a worse degree bound, showing torpid mixing of Glauber dynamics for sampling independent sets of a bipartite graph. Recently, Mossel, Weitz and Wormald [69] showed that for λ greater than a critical $\lambda_c(d)$, with high probability over d-regular bipartite graphs, Glauber dynamics (or any local Markov chain) will mix torpidly for sampling independent sets with activity λ. The two results above match the upper bound of Weitz in [89] showing that Glauber dynamics mixes rapidly for sampling from independent sets in graphs of maximum degree 5, or respectively, sampling independent sets in graphs of maximum degree d with activity $\lambda < \lambda_c(d)$. Mossel et al. [69] conjecture that $\lambda_c(d)$ is in fact the exact threshold for this computational problem, i.e., that for $\lambda > \lambda_c(d)$ it is NP-hard to approximate the partition function $Z_G(\lambda)$ for a graph G of maximum degree d in time that is polynomial in the size of G.

An intriguing question is whether for bipartite graphs torpid mixing indicates that the problem is computationally hard or whether there could be other methods used for sampling and counting independent sets. The complexity of counting the number of independent sets in a bipartite graph (#Bis) was first studied in [29] where they show that #Bis is a complete problem for a class of counting problems in #P. Interestingly, approximating #Bis is known to be *equivalent* to a number of other approximate counting problems [29] such as

- Counting the number of antichains in a poset.
- Computing the partition function of the Ising model when the external field is not uniform. In the most general case for the Ising model where pairs of particles have

interaction energy J_{ij}, and the magnetic field at a vertex is μ_i, the Hamiltonian is given by

$$H(x) = \sum_{(i,j)\in E(G)} J_{ij}x(i)x(j) + \sum_{i\in[n]} \mu_i x(i)$$

and the partition function is

$$Z(\beta) = \sum_{x\in\{+1,-1\}^n} \exp(\beta H(x)).$$

Jerrum and Sinclair [46] gave an FPRAS for the case that all the magnetic fields are of the same sign.

- Counting the number of stable marriages for a set of preferences.

Can hardness of approximate counting be shown for any of these problems? There are no sampling schemes known for any of the above problems, though perhaps intricate methods like annealing have not been much explored. Even showing that there are instances of these problems where annealing fails would be interesting. In [27] the instance for which Glauber dynamics mixes torpidly was the basis of for showing that the number of independent sets is hard to approximate in graphs of maximum degree at least 25. Thus understanding instances for which Markov chains fail may bring us closer to showing the hardness of approximate counting.

REFERENCES

[1] Aldous, D., "Some inequalities for reversible Markov chains," *Journal of the London Mathematical Society*, vol. 25, pp. 564–576, 1982.

[2] Alon, N., "Eigenvalues and expanders," *Combinatorica*, vol. 6, pp. 83–96, 1986.

[3] Alon, N. and Milman, V., "λ_1, isoperimetric inequalities for graphs and superconcentrators," *Journal of Combinatorial Theory Series B*, vol. 38, pp. 73–88, 1985.

[4] Bayati, M., Kim, J., and Saberi, A., "Fast generation of random graphs via sequential importance sampling," *To appear in the Proceedings of the 11th International Workshop on Randomization and Computation, RANDOM*, 2007.

[5] Bender, A. and Canfield, R., "The asymptotic number of labeled graphs with given degree sequences," *Journal of Combinatorial Theory Series A*, vol. 24, pp. 296–307, 1978.

[6] Besag, J. and Clifford, P., "Generalized Monte Carlo significance tests," *Biometrika*, vol. 76, pp. 633–642, 1989.

[7] Bezáková, I., Bhatnagar, N., and Vigoda, E., "Sampling binary contingency tables with a greedy start," *Random Structures and Algorithms*, vol. 30, pp. 168–205, 2007.

[8] Bezáková, I., Štefankovič, D., Vazirani, V., and Vigoda, E., "Accelerating simulated annealing for combinatorial counting," *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pp. 900–907, 2006.

[9] Bhatnagar, N. and Randall, D., "Torpid mixing of simulated tempering on the Potts model," *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, pp. 278–287, 2004.

[10] Blanchet, J., "Efficient importance sampling for counting," *Preprint*, 2007.

[11] Bollobás, B., "A probabilistic proof of an asymptotic formula for the number of labeled random graphs," *European Journal of Combinatorics*, vol. 1, pp. 311–316, 1980.

[12] Borgs, C., Chayes, J., Frieze, A., Kim, J., Tetali, P., Vigoda, E., and Vu, V., "Torpid mixing of some MCMC algorithms in statistical physics," *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pp. 218–229, 1999.

[13] Brin, S. and Page, L., "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, pp. 107–117, 1998.

[14] Carson, T. and Impagliazzo, R., "Hill-climbing finds random planted bisections," *Proceedings of the 12th Annual Symposium on Discrete Algorithms*, pp. 903–909, 2001.

[15] Černý, V., "A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 45–51, 1985.

[16] Chen, Y., Diaconis, P., Holmes, S., and Liu, J., "Sequential Monte Carlo methods for statistical analysis of tables," *Journal of the American Statistical Association*, vol. 100, pp. 109–120, 2005.

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

[18] Cooper, C., Dyer, M., Frieze, A., and Rue, R., "Mixing properties of the Swendsen-Wang process on the complete graph and narrow grids," *Journal of Mathematical Physics*, vol. 41, pp. 1499–1527, 2000.

[19] Cooper, C., Dyer, M., and Greenhill, C., "On Markov chains for random regular graphs," *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 980–988, 2005.

[20] Cryan, M., Dyer, M., Goldberg, L., Jerrum, M., and Martin, R., "Rapidly mixing Markov chains for sampling contingency tables with a constant number of rows," *SIAM Journal on Computing*, vol. 36, pp. 247–278, 2005.

[21] Diaconis, P. and Efron, B., "Testing for independence in a two-way table: New interpretations of the chi-square statistic," *Annals of Statistics*, vol. 13, pp. 845–874, 1985.

[22] Diaconis, P. and Gangolli, A., *Rectangular Arrays with Fixed Margins*. Discrete Probability and Algorithms, ed. D. Aldous et al., New York, 1995.

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

[24] Diaconis, P. and Strook, "Geometric bounds for eigenvalues of Markov chains," *Annals of Applied Probability*, vol. 1, pp. 36–61, 1991.

[25] Durrett, R., *Probability Models for DNA Sequence Evolution*. Springer, 2002.

[26] Dyer, M., "Approximate counting by dynamic programming," *Proceedings of the 35th ACM Symposium on Theory of Computing*, pp. 693–699, 2003.

[27] Dyer, M., Frieze, A., and Jerrum, M., "On counting independent sets in sparse graphs," *SIAM Journal on Computing*, vol. 31, pp. 1527–1541, 2002.

[28] Dyer, M., Frieze, A., and Kannan, R., "A random polynomial time algorithm for approximating the volume of convex bodies," *Journal of the Association for Computing Machinery*, vol. 38, pp. 1–17, 1991.

[29] Dyer, M., Goldberg, L., Greenhill, C., and Jerrum, M., "On the relative complexity of approximate counting problems," *Algorithmica*, vol. 38, pp. 471–500, 2003.

[30] Feller, W., *An Introduction to Probability Theory and its Applications*. Wiley, 1968.

[31] Fisher, M., "Statistical mechanics of dimers on a plane lattices," *Physics Review*, vol. 124, pp. 1664–1672, 1961.

[32] Fowler, R. and Rushbrooke, G., "Statistical theory of perfect solutions," *Transactions of the Faraday Society*, vol. 33, pp. 1272–1294, 1937.

[33] Gale, D., "A theorem on flows in networks," *Pacific Journal of Mathematics*, vol. 7, pp. 1073–1082, 1957.

[34] Galvin, D. and Tetali, P., "Slow mixing of Glauber dynamics for the hard-core model on regular bipartite graphs," *Random Structures and Algorithms*, vol. 28, no. 4, pp. 427–443, 2006.

[35] Georgii, H.-O., *Gibbs measures and phase transitions*. de Gruyter, Berlin, 1988.

[36] Geyer, C., "Markov chain Monte Carlo maximum likelihood," *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pp. 156–163, 1991.

[37] Geyer, C. and Thompson, E., "Annealing Markov chain Monte Carlo with applications to ancestral inference," *Journal of the American Statistical Association*, vol. 90, pp. 909–920, 1995.

[38] Good, I. and Crook, J., "The enumeration of arrays and a generalization related to contingency tables," *Discrete Mathematics*, vol. 19, pp. 23–65, 1977.

[39] Gore, V. and Jerrum, M., "The Swendsen-Wang process does not always mix rapidly," *Journal of Statistical Physics*, vol. 97, pp. 67–86, 1995.

[40] Hajek, B., "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, pp. 311–329, 1988.

[41] Hardy, G. and Wright, E., *An Introduction to the Theory of Numbers, 5th edition*. Oxford Press, 1979.

[42] Jerrum, M., "Two-dimensional monomer-dimer systems are computationally intractable," *Journal of Statistical Physics*, vol. 48, pp. 121–134, 1987.

[43] Jerrum, M., "Counting, sampling and integrating: Algorithms and complexity," *Lectures in Mathematics ETH Zurich*, 2003.

[44] Jerrum, M. and Sinclair, A., "Approximating the permanent," *SIAM Journal on Computing*, vol. 18, pp. 1149–1178, 1989.

[45] Jerrum, M. and Sinclair, A., "Fast uniform generation of regular graphs," *Theoretical Computer Science*, vol. 73, pp. 91–100, 1990.

[46] Jerrum, M. and Sinclair, A., "Polynomial-time approximation algorithms for the Ising model," *SIAM Journal on Computing*, vol. 22, no. 5, pp. 1087–1116, 1993.

[47] Jerrum, M., Sinclair, A., and Vigoda, E., "A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries," *Journal of the ACM*, vol. 51, no. 4, pp. 671–697, 2004.

[48] Jerrum, M., Valiant, L., and Vazirani, V., "Random generation of combinatorial structures from a uniform distribution," *Theoretical Computer Science*, vol. 43, pp. 169–188, 1986.

[49] Jerrum, M. and Sinclair, A., "Approximate counting, uniform generation and rapidly mixing Markov chains," *Information and Computation*, vol. 82, pp. 93–133, 1989.

[50] Jerrum, M., Son, J.-B., Tetali, P., and Vigoda, E., "Elementary bounds on Poincare and log-Sobolev constants for decomposable Markov chains," *Annals of Applied Probability*, vol. 14(4), pp. 1741–1765, 2004.

[51] Jerrum, M. and Sorkin, G., "Simulated annealing for graph bisection," *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pp. 94–103, 1993.

[52] Kannan, R., Tetali, P., and Vempala, S., "Simple Markov chain algorithms for generating bipartite graphs and tournaments," *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 193–200, 1997.

[53] Kasteleyn, P., "The statistics of dimers on a lattice I. the number of dimer arrangements on a quadratic lattice," *Physica*, vol. 27, pp. 1209–1125, 1961.

[54] Kim, J. and Vu, V., "Generating random regular graphs," *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pp. 213–222, 2003.

[55] Kirchoff, G., "Über die Auflösung der Gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer Ströme geführt wird," *Annals of Physical Chemistry*, vol. 72, pp. 497–508, 1847. English translation in *IRE Transactions in Circuit Theory*, vol. 5, pp. 4-8, 1958.

[56] Kirkpatrick, S., Gellatt, L., and Vecchi, M., "Optimization by simulated annealing," *Science*, vol. 220, pp. 498–516, 1983.

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

[58] Liu, J., *Markov Chain Strategies in Scientific Computing*. Springer Series in Statistics, 2001.

[59] Lovász, L. and Vempala, S., "Simulated annealing in convex bodies and an $o^*(n^4)$ volume algorithm," *Proceedings of the 44th IEEE Foundations of Computer Science*, pp. 650–660, 2003.

[60] Luby, M. and Vigoda, E., "Approximately counting up to four," *Proceedings of the 29th annual ACM Symposium on Theory of Computing*, pp. 682–687, 1997.

[61] Madras, N. and Piccioni, M., "Importance sampling for families of distributions," *Annals of applied probability*, vol. 9, pp. 1202–1225, 1999.

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

[63] Madras, N. and Slade, G., *The self-avoiding walk: Probability and its applications*. Birkhäuser, Boston, 1996.

[64] Madras, N. and Zheng, Z., "On the swapping algorithm," *Random Structures and Algorithms*, vol. 22, pp. 66–97, 2003.

[65] Marinari, E. and Parisi., G., "Simulated tempering: a new Monte Carlo scheme," *Europhysics Letters*, vol. 19, pp. 451–458, 1992.

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

[67] Martinelli, F., "Lectures on Glauber dynamics for discrete spin models," *Lecture Notes on Probability Theory and Statistics (Saint-Flour)*, 1999.

[68] Morris, B., "Improved bounds for sampling contingency tables," *Random Structures and Algorithms*, vol. 21, pp. 135–146, 2002.

[69] Mossel, E., Weitz, D., and Wormald, N., "On the hardness of sampling independent sets beyond the tree threshold," *ArXiv Mathematics e-prints*, 2007. Available at `http://arxiv.org/abs/math/0701471`.

[70] N., M., Rosenbluth, A., Rosenbluth, M., Teller, A., and E., T., "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[71] Papadimitriou, C., *Computational Complexity*. Addison Wesley, 1993.

[72] Potts, R., "Some generalized order-disorder transformations," *Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 106–109, 1952.

[73] Randall, D., "Mixing," *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*, pp. 4–15, 2003.

[74] Randall, D., "Slow mixing of Glauber dynamics via topological obstructions," *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 870–879, 2006.

[75] Reingold, O., Vadhan, S., and Wigderson, A., "Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors," *Annals of Mathematics*, vol. 155, pp. 157–187, 2002.

[76] Ryser, H., "Combinatorial properties of matrices of zeroes and ones," *Canadian Journal of Mathematics*, vol. 9, pp. 371–377, 1957.

[77] Sanderson, J., "Testing ecological patterns," *American Scientist*, vol. 88, pp. 332–339, 2000.

[78] Schweinsberg, J., "An $o(n^2)$ bound for the relaxation time of a Markov chain on cladograms," *Random Structures and Algorithms*, vol. 20, pp. 59–70, 2002.

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

[80] Sinclair, A., *Algorithms for Random Generation and Counting: a Markov Chain Approach*. Birkhäuser, 1993.

[81] Stanley, R., *Enumerative Combinatorics*. Cambridge University Press, 1997.

[82] Steger, A. and Wormald, N., "Generating random regular graphs quickly," *Combinatorics, Probability and Computing*, vol. 8, pp. 377–396, 1999.

[83] Swendsen, R. H. and Wang, J.-S., "Nonuniversal critical dynamics in Monte Carlo simulations," *Physics Review Letters*, vol. 58, no. 2, pp. 86–88, 1987.

[84] Temperley, H. and Fisher, M., "Dimer problem in statistical mechanics - an exact result," *Philosophical Magazine*, vol. 6, pp. 1061–1063, 1961.

[85] Thomas, L., "Bound on the mass gap for finite volume stochastic Ising models at low temperature," *Communications in Mathematical Physics*, vol. 126, pp. 1–11, 1989.

[86] Tutte, W., "A short proof of the factor theorem for finite graphs," *Canadian Journal of Mathematics*, vol. 6, pp. 347–352, 1954.

[87] Valiant, L., "The complexity of computing the permanent," *Theoretical Computer Science*, vol. 8, pp. 189–201, 1979.

[88] Vigoda, E., "Improved bounds for sampling colorings," *Journal of Mathematical Physics*, vol. 41(3), pp. 1555–1569, 2000.

[89] Weitz, D., "Counting independent sets up to the tree threshold," *Proceedings of the 38th annual ACM Symposium on Theory of Computing*, pp. 140–149, 2006.

[90] West, D., *Introduction to graph theory*. Prentice-Hall, 1996.

[91] Zheng, Z., "Analysis of swapping and tempering Monte Carlo algorithms," *Dissertation, York University*, 1999.