Accelerating Simulated Annealing for the Permanent and Combinatorial Counting Problems

Ivona Bezáková* Daniel Štefankovič* Vijay V. Vazirani† Eric Vigoda†

July 15, 2005

Abstract

We present an improved "cooling schedule" for simulated annealing algorithms for combinatorial counting problems. Under our new schedule the rate of cooling accelerates as the temperature decreases. Thus, fewer intermediate temperatures are needed as the simulated annealing algorithm moves from the high temperature (easy region) to the low temperature (difficult region). We present applications of our technique to colorings and the permanent (perfect matchings of bipartite graphs). Moreover, for the permanent, we improve the analysis of the Markov chain underlying the simulated annealing algorithm. This improved analysis, combined with the faster cooling schedule, results in an $O(n^7 \log^4 n)$ time algorithm for approximating the permanent of a 0/1 matrix.

1 Introduction

Simulated annealing is an important algorithmic approach for counting and sampling combinatorial structures. Two notable combinatorial applications are estimating the partition function of statistical physics models, and approximating the permanent of a non-negative matrix. For combinatorial counting problems, the general idea of simulated annealing is to write the desired quantity, say Z, (which is, for example, the number of colorings or matchings of an input graph) as a telescoping product:

$$Z = \frac{Z_\ell}{Z_{\ell-1}} \frac{Z_{\ell-1}}{Z_{\ell-2}} \cdots \frac{Z_1}{Z_0} Z_0, \tag{1}$$

*Department of Computer Science, University of Chicago, Chicago, IL 60637. Email: {ivona,stefanko}@cs.uchicago.edu. I.B. was supported by NSF grant CCR-0237834.

†College of Computing, Georgia Institute of Technology, Atlanta, GA 30332. Email: {vazirani,vigoda}@cc.gatech.edu. V.V. is supported by NSF grants CCR-0311541 and CCR-0220343. E.V. is supported by NSF grant CCR-0237834.

where $Z_\ell = Z$ and Z_0 is trivial to compute. By further ensuring that each of the ratios Z_i/Z_{i-1} is bounded, a small number of samples (from the probability distribution corresponding to Z_{i-1}) suffices to estimate the ratio. These samples are typically generated from an appropriately designed Markov chain.

Each of the quantities of interest corresponds to the counting problem at a different temperature. The final quantity $Z = Z_\ell$ corresponds to zero-temperature, whereas the trivial initial quantity Z_0 is infinite temperature. The temperature slowly decreases from high temperature (easy region) to low temperature (difficult region). A notable application of simulated annealing to combinatorial counting was the algorithm of Jerrum, Sinclair and Vigoda [8] for approximating the permanent of a non-negative matrix. In their algorithm, the cooling schedule is uniform: the rate of cooling was constant.

Our first main result is an improved cooling schedule. In contrast to the previous cooling schedule for the permanent, our schedule is accelerating (the rate of cooling accelerates as the temperature decreases). Consequently, fewer intermediate temperatures are needed, and thus fewer Markov chain samples overall suffice. It is interesting to note that our schedule is similar to the original proposal of Kirkpatrick et al [11], and is related to schedules used recently in geometric settings by Lovász and Vempala [13] and Kalai and Vempala [9].

We illustrate our new cooling schedule in the context of colorings, which corresponds to the anti-ferromagnetic Potts model from statistical physics. We present general results defining a cooling schedule for a broad class of counting problems. These general results seem applicable to a wide range of combinatorial counting problems, such as the permanent, and binary contingency tables [1].

The permanent of an $n \times n$ matrix A is defined as

$$\mathrm{per}(A) = \sum_\sigma \prod_{i=1}^n a_{i,\sigma(i)},$$

where the sum goes over all permutations σ of $[n]$. The permanent of a $0/1$ matrix A is the number of perfect matchings in the bipartite graph with bipartite adjacency matrix A. In addition to traditional applications in statistical physics [10], the permanent has recently been used in a variety of areas, e. g., computer vision [15], and statistics [14]. Jerrum, Sinclair, and Vigoda presented a simulated annealing algorithm [8] for the permanent of non-negative matrices with running time $O(n^{10} \log^3 n)$ for $0/1$ matrices.

Our cooling schedule reduces the number of intermediate temperatures in the simulated annealing for the permanent from $O(n^2 \log n)$ to $O(n \log^2 n)$. We also improve the analysis of the Markov chain used for sampling. The improved analysis comes from several new inequalities about perfect matchings in bipartite graphs. The consequence of the new analysis and improved cooling schedule is an $O(n^7 \log^4 n)$ time algorithm for estimating the permanent of an $0/1$ $n \times n$ matrix. Here is the formal statement of our result:

Theorem 1. *For all $\epsilon > 0$, there exists a randomized algorithm to approximate, within a factor $(1 \pm \epsilon)$, the permanent of a 0/1 $n \times n$ matrix A in time $O(n^7 \log^4(n) + n^6 \log^5(n)\epsilon^{-2})$. The algorithm extends to arbitrary matrices with non-negative entries.*

The remainder of the paper is organized as follows. In Section 2 we present our new cooling schedule, motivated by its application to colorings. We then focus on the permanent in Section 3. We begin by presenting the simulated annealing algorithm for the permanent in Section 3. In Section 4 we explain the background techniques for analyzing the Markov chain. We present our new inequalities in Section 5. Finally, in Section 6 we use these new inequalities for bounding the mixing time of the Markov chain. We then conclude the analysis of the permanent algorithm for 0/1 matrices in Sections 7 and 8, and present the extension to non-negative matrices in Section 9.

2 Improved Cooling Schedule

We begin by motivating the simulated annealing framework in the context of colorings. We then present a general method for obtaining improved cooling schedules and show how it can be applied to colorings. We conclude with the proofs of technical lemmas for improved cooling schedules.

2.1 Counting Colorings

Our focus in this section is counting all valid k-colorings of a given graph G. Let $G = (V, E)$ be the input graph and k be the number of colors. A (valid) k-coloring of G is an assignment of colors from $[k]$ to the vertices of G such that no two adjacent vertices are colored by the the same color (i. e., $\sigma(u) \neq \sigma(v)$ for every $(u, v) \in E$). Let $\Omega = \Omega(G)$ denote the set of all k-colorings of G. For input parameters ϵ, δ, our goal is to approximate $|\Omega|$ within a multiplicative factor $1 \pm \epsilon$ with probability $\geq 1 - \delta$.

Before we present our new reduction, it is worth illustrating the standard reduction (see e.g., [6]). Let $E_0 = E = \{e_1, ..., e_m\}$ (ordered arbitrarily), and, for $1 \leq i \leq m$, let $E_i = E_{i-1} \setminus e_i$ and $G_i = (V, E_i)$. Then the number of k-colorings of G can be written as a telescoping product:

$$|\Omega(G)| = k^n \prod_i \frac{|\Omega(G_{i-1})|}{|\Omega(G_i)|}$$

For $k \geq \Delta + 2$ where Δ is the maximum degree of G, it is possible to verify the following bound on the i-th ratio:

$$\frac{1}{2} \leq \frac{|\Omega(G_{i-1})|}{|\Omega(G_i)|} \leq 1,$$

Therefore we can estimate the i-th ratio by generating random k-colorings of G_i and counting the proportion that are also valid k-colorings of G_{i-1}. This reduces the approximate counting problem of estimating the cardinality of $\Omega(G)$ to m random sampling problems, see Jerrum [6] for details on the reduction as a function of the error parameter ϵ and confidence parameter δ.

We instead look at a continuous version of the problem, the anti-ferromagnetic Potts model from Statistical Physics, which allows more flexibility in how we remove edges. In addition to the underlying graph G and the number of partitions k, the Potts model is also specified by an activity[1] λ. The configuration space of the Potts model, denoted $[k]^V$, is the set of all labelings $\sigma : V \to [k]$. The partition function of the Potts model counts the number of configurations weighted by their "distance" from a valid k-coloring. The "distance" is measured in terms of the activity λ and we will specify it shortly. As the activity goes to zero, the partition function limits to $|\Omega|$.

Our reduction from approximating $|\Omega|$ to sampling from Ω, works by specifying a sequence of activities for the anti-ferromagnetic Potts model, so that the partition functions do not change by more than a constant factor between successive activities. This allows us to reduce the activity to an almost zero value while being able to estimate the ratios of two consecutive partition functions. Then, as before, we can approximate $|\Omega|$. The advantage of the new reduction lies in using fewer random sampling problems, namely instead of m problems we now need to consider only $O(n \log n)$ sampling problems to estimate $|\Omega|$.

For $\lambda > 0$, the partition function of the Potts model is

$$Z(\lambda) = \sum_{\sigma \in [k]^V} \lambda^{M(\sigma)}$$

where $M(\sigma) = M_G(\sigma) = |(u,v) \in E : \sigma(u) = \sigma(v)|$ is the number of monochromatic edges of the labeling σ.

The partition function can be viewed as a polynomial in λ. Notice that its absolute coefficient equals $|\Omega|$, the number of k-colorings of G. Moreover, $Z(1) = |\Omega(G_m)| = k^n$ is the sum of the coefficients of Z. It can be shown that for $k > \Delta$ the number of k-colorings of G is bounded from below by $(k/e)^n$ (i. e., $|\Omega| \geq (k/e)^n$). For completeness, we prove this lower bound in the Appendix in Corollary 20 of Section 11. If we used the trivial lower bound of $|\Omega| \geq 1$, we would introduce an extra factor of $O(\log k)$ in the final running time. Observe that the value of the partition function at $\lambda = 1/e^n$ is at most $2|\Omega|$:

$$|\Omega| \leq Z(1/e^n) \leq |\Omega| + Z(1)(1/e^n) \leq |\Omega| + k^n/e^n \leq 2|\Omega|. \tag{2}$$

This will be sufficiently close to $|\Omega|$ so that we can obtain an efficient estimator for $|\Omega|$.

[1] The activity corresponds to the temperature of the system. Specifically, the temperature is $1/\ln \lambda$, thus $\lambda = 1$ corresponds to the infinite temperature and $\lambda = 0$ corresponds to the zero temperature.

We will define a sequence

$$\lambda_0 = 1, \lambda_1, \ldots, \lambda_\ell \le 1/e^n, \lambda_{\ell+1} = 0,$$

where $\ell = O(n \log n)$, and, for all $0 \le i \le \ell$,

$$\frac{1}{2} \le \frac{Z(\lambda_{i+1})}{Z(\lambda_i)} \le 1.$$

We estimate the number of k-colorings of G via the telescoping product:

$$|\Omega| = k^n \prod_{0 \le i \le \ell} \alpha_i,$$

where $\alpha_i = Z(\lambda_{i+1})/Z(\lambda_i)$. We will estimate α_i by sampling from the probability distribution corresponding to Z_i. Before we describe how to estimate these ratios, we first specify the cooling schedule (i.e., the sequence of activities).

2.2 Intuition for Accelerating Cooling Schedule for Colorings

We need to ensure that for consecutive λ_i, λ_{i+1} the ratio $Z(\lambda_{i+1})/Z(\lambda_i)$ is in the interval $[\frac{1}{2}, 1]$. The polynomial Z has degree m since any labeling has at most $m = |E|$ monochromatic edges. Hence it suffices to define $\lambda_{i+1} = 2^{-1/m}\lambda_i$, then $Z(\lambda_{i+1}) \ge (2^{-1/m})^m Z(\lambda_i) \ge Z(\lambda_i)/2$. This specifies a uniform cooling schedule with a rate of decrease $2^{-1/m}$.

If we had $Z(\lambda) = k^n \lambda^m$ we could not decrease λ faster than $2^{-1/m}$. Fortunately, in our case the absolute coefficient of $Z(\lambda)$ is at least $|\Omega| \ge (k/e)^n$. To illustrate the idea of non-uniform decrease, let $f_i(\lambda) = \lambda^i$. The polynomial f_m will always decrease faster than Z. At first (for values of λ close to 1) this difference will be small, however, as λ goes to 0, the rate of decrease of Z slows down because of its absolute term. Thus, at a certain point f_{m-1} will decrease faster than Z. Once λ reaches this point, we can start decreasing λ by a factor of $2^{-1/(m-1)}$. As time progresses, the rate of Z will be bounded by the rate of polynomials f_m, then f_{m-1}, f_{m-2}, $\ldots$, all the way down to f_1 for λ close to 0. When the polynomial f_i "dominates" we can decrease λ by a factor of $2^{-1/i}$. Note that the rate of decrease increases with time, i. e., the schedule is accelerating.

2.3 General Cooling Schedule

Now we formalize the accelerated cooling approach. We state our results in a general form which proves useful in other contexts, e. g., for the permanent later in this paper, and binary contingency tables [1].

Let $Z(\lambda)$ be the partition function polynomial. Let s be the degree of $Z(\lambda)$ (note that $s = m$ for colorings). Our goal is to find $1 = \lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_\ell$ such that

$Z(\lambda_i)/Z(\lambda_{i+1}) \leq c$ (e.g., for colorings we took $c = 2$). The important property of $Z(\lambda)$ for colorings is $Z(0) \geq (k/e)^n$ (i.e., $Z(\lambda)$ has large constant coefficient). For some applications it will not be possible to make the absolute coefficient large, instead we will show that a coefficient a_D of λ^D is large (for some small D). Finally, let γ be an upper bound on $Z(1)/a_D$. For colorings we can take $\gamma = e^n$. The γ measures how small λ needs to get for $Z(\lambda)$ to be within constant factor of $Z(0)$. Now we present a general algorithm in terms of parameters s, c, γ, D.

Algorithm for computing the cooling schedule λ, given parameters s, c, γ, and D:

> Set $\hat{\lambda}_0 = 1$, $i = s$ and $j = 0$.
> While $\hat{\lambda}_j > 1/\gamma$ do
>   Set $\hat{\lambda}_{j+1} = c^{-1/i}\hat{\lambda}_j$.
>   If $i > D + 1$ and $\hat{\lambda}_{j+1} < (s/\gamma)^{1/(i-D)}$,
>   Set $\hat{\lambda}_{j+1} = (s/\gamma)^{1/(i-D)}$ and decrement $i = i - 1$.
>   Increment $j = j + 1$.
> Set $\ell = j$.

The following lemma estimates the number of intermediate temperatures in the above cooling schedule, i.e., the length ℓ of the $\hat{\lambda}$ sequence.

Lemma 2. *Let $c, \gamma > 0, D \geq 0$ and let $\hat{\lambda}_0, \ldots, \hat{\lambda}_\ell$ be the sequence computed by the above algorithm. Then $\ell = O([(D+1)\log(s-D) + s/(s-D)]\log_c \gamma)$. If c and D are constants independent of s, then $\ell = O(\log s \log \gamma)$.*

We will prove the lemma in Section 2.5. Note that for colorings $\ell = O(n \log n)$.

The following lemma shows that for the sequence of the λ_i the value of $Z(\lambda)$ changes by a factor $\leq c$ for consecutive λ_i and λ_{i+1}. We postpone the proof to Section 2.5.

Lemma 3. *Let $c, \gamma, D \geq 0$ and let $Z_1, \ldots, Z_q$ be a collection of polynomials of degree s. Suppose that for every $i \in [q]$, the polynomial Z_i satisfies the following conditions:*

i) Z_i has non-negative coefficients,

ii) there exists $d \leq D$ such that the coefficient of x^d in Z_i is at least $Z_i(1)/\gamma$.

Let $\hat{\lambda}_0, \hat{\lambda}_1, \ldots, \hat{\lambda}_\ell$ be the sequence constructed by the above algorithm. Then

$$Z_i(\hat{\lambda}_j) \leq c Z_i(\hat{\lambda}_{j+1}) \qquad \text{for every } i \in [q] \text{ and } j \in [\ell].$$

2.4 Applying the Improved Cooling Schedule to Colorings

Before applying these general results to colorings, we quickly review how to approximate the ratios α_i in the telescoping product (1) (see [6] for details). We can approximate α_i using the following unbiased estimator. Let $X_i \sim \pi_i$ denote a random labeling chosen from the distribution π_i defined by $Z(\lambda_i)$, (i. e., the probability of a labeling σ is $\pi_i(\sigma) = {\lambda_i}^{M(\sigma)}/Z(\lambda_i)$). Let $Y_i = (\lambda_{i+1}/\lambda_i)^{M(X_i)}$. Then Y_i is an unbiased estimator for α_i:

$$\mathbf{E}\left(Y_i\right) = \mathbf{E}_{X_i\sim\pi_i}\left((\lambda_{i+1}/\lambda_i)^{M(X_i)}\right) = \sum_{\sigma\in[k]^V} \frac{(\lambda_{i+1})^{M(\sigma)}}{Z(\lambda_i)} = \frac{Z(\lambda_{i+1})}{Z(\lambda_i)} = \alpha_i. \tag{3}$$

Assume that we have an algorithm for generating labelings X_i' from a distribution that is within variation distance $\leq \varepsilon/\ell$ of π_i. We draw $\Theta(\ell/\varepsilon^2)$ samples of X_i' and take the mean $\overline{Y}_i$ of their corresponding estimators Y_i'. Then the expected value of $\overline{Y}_i$ is $E[\overline{Y}_i](1 \pm \varepsilon/\ell)$ and the variance can be bounded as $V[\overline{Y}_i] = O(\varepsilon^2/\ell)E[\overline{Y}_i]^2$. Therefore, by the Chebyshev's inequality $k^n \prod_{i=0}^{\ell-1} \overline{Y}_i$ equals $|\Omega|(1 \pm 2\varepsilon)$ with probability $\geq 3/4$.

If the algorithm generates a sample X_i' from a distribution within variation distance $\leq \varepsilon/\ell$ of π_i in time $T(\varepsilon/\ell)$, then the computation of $k^n \prod_{i=0}^{\ell-1} \overline{Y}_i$ takes time $O(\ell^2/\varepsilon^2 T(\varepsilon/\ell))$.

Now we return to colorings, and conclude the final running time of the algorithm. Recall that $\ell = O(n \log n)$. For $k > 2\Delta$, it is possible to generate a labeling within variation distance $\leq \varepsilon'$ of π_i in time $T(\varepsilon') = \frac{k}{k-2\Delta} n \log(n/\varepsilon')$ [3, 6]. Hence one can approximate $|\Omega|$ within a multiplicative factor $1\pm\varepsilon$ with probability $\geq 3/4$ in $O(\frac{k}{k-2\Delta}\frac{n^3 \log^2 n}{\varepsilon^2} \ln(n/\varepsilon))$ time. In contrast, for $k > 2\Delta$ using the standard counting to sampling reduction, Jerrum states a running time of $O(\frac{k}{k-2\Delta}\frac{nm^2}{\varepsilon^2} \ln(n/\varepsilon))$ where m is the number of edges. For $k \leq 2\Delta$ results on mixing time are known for certain classes of graphs [5]. These are proved for k-colorings, but most likely they can be extended to the non-zero temperature.

2.5 Proof of Lemmas 2 and 3

The rest of this section is devoted to the proof of Lemmas 2 and 3.

Proof of Lemma 2. We define intervals:

$$I_s = [(s/\gamma)^{1/(s-D)}, \infty),$$

for $i = D+2, \ldots, s-1$,

$$I_i = [(s/\gamma)^{1/(i-D)}, (s/\gamma)^{1/(i+1-D)}],$$

and finally,

$$I_{D+1} = (0, (s/\gamma)^{1/2})].$$

Let ℓ_i be the number of $\hat{\lambda}$ values lying in the interval I_i. For $i \in \{D+2, \ldots, s-1\}$ we have the estimate:

$$\ell_i \le \log_c \left(\frac{[(s/\gamma)^{1/(i+1-D)}]^i}{[(s/\gamma)^{1/(i-D)}]^i} \right) \le \frac{D+1}{i-D} \log_c \gamma.$$

Similarly,

$$\ell_s \le \log_c \left(\frac{\gamma}{[(s/\gamma)^{1/(s-D)}]^s} \right) \le \frac{2s-D}{s-D} \log_c \gamma,$$

and

$$\ell_{D+1} \le \log_c \left(\frac{[(s/\gamma)^{1/2}]^{D+1}}{[1/\gamma]^{D+1}} \right) \le \frac{D+1}{2} \log_c \gamma.$$

Putting it all together, we get the bound

$$\ell \le \sum_{i=D+1}^{s} \ell_i \le \left((D+1)H_{s-D} + \frac{2s-D}{s-D} + \frac{D+1}{2} \right) \log_c \gamma,$$

where $H_i = \sum_{j=1}^{i} 1/j = O(\log i)$ is the harmonic sum. Therefore

$$\ell = O([(D+1)\log(s-D) + s/(s-D)] \log_c \gamma).$$

□

The *log-derivative* of a function f is $(\log f)' = f'/f$. The log-derivative measures how quickly a function increases.

Definition 4. *We say that a polynomial f is* dominant *over a polynomial g on an interval I if $f'(x)/f(x) \ge g'(x)/g(x)$ for every $x \in I$.*

Lemma 5. *Let $f, g : I \to \mathbf{R}^+$ be two non-decreasing polynomials. If f dominates over g on I, then $f(y)/f(x) \ge g(y)/g(x)$ for every $x, y \in I$, $x \le y$.*

We partition the interval $(0, \infty)$ into subintervals $I_{D+1}, \ldots, I_s$ such that x^i dominates over every Z-polynomial on the interval I_i. The $\hat{\lambda}_j$ in I_i will be such that x^i decreases by a factor c between consecutive $\hat{\lambda}$. Therefore the Z-polynomials decrease by at most a factor of c.

Lemma 6. *Let $g(x) = \sum_{j=0}^{s} a_j x^j$ be a polynomial with non-negative coefficients. Then x^s dominates over g on the interval $(0, \infty)$.*

Proof. It suffices to verify that $(x^s)'/x^s \ge g'(x)/g(x)$ for every $x > 0$. □

Lemma 7. *Let $g(x) = \sum_{j=0}^{s} a_j x^j$ be a polynomial with non-negative coefficients such that $g(1) \leq \gamma$ and at least one of $a_0, a_1, \ldots, a_D$ is ≥ 1. Then for any $i \geq D+1$ the polynomial x^i dominates g on the interval $(0, (s/\gamma)^{1/(i+1-D)}]$.*

Proof. The logarithmic derivative of x^i is i/x. Hence we need to prove that $ig(x) \geq xg'(x)$ for $x \leq (s/\gamma)^{1/(i+1-D)}$.

Let d be the smallest integer such that $a_d \geq 1$. From the assumptions of the lemma $d \leq D$. For $x \leq (s/\gamma)^{1/(i+1-D)}$ the following holds

$$\sum_{j=i+1}^{s} j a_j x^{j-d} \leq \sum_{j=i+1}^{s} s a_j x^{j-D} \leq \sum_{j=i+1}^{s} s a_j \left(\frac{s}{\gamma}\right)^{(j-D)/(i+1-D)} \leq \sum_{j=i+1}^{s} s a_j \left(\frac{s}{\gamma}\right) \leq 1.$$

Since $i > d$, for $x \leq (s/\gamma)^{1/(i+1-D)}$ we have

$$xg'(x) = \sum_{j=0}^{i} j a_j x^j + \sum_{j=i+1}^{s} j a_j x^j \leq \sum_{j=d}^{i} j a_j x^j + a_d x^d \leq \sum_{j=d}^{i} i a_j x^j = ig(x).$$

$\square$

Proof of Lemma 3. Let $I_{D+1}, \ldots, I_s$ be as in the proof of Lemma 2. Let $Q_q(\lambda) = \gamma Z_q(\lambda)/Z_q(1)$. Notice that the Q_q satisfy the conditions required of g by Lemma 7. Therefore x^i dominates over every Q_q (and hence also Z_q) on the interval I_i for $i < s$. Moreover, Lemma 7 and Lemma 5 imply that x^s dominates over every Q_q (and hence Z_q) on the interval I_s. Notice that if $\hat{\lambda}_j, \hat{\lambda}_{j+1} \in I_i$, then $c\hat{\lambda}_{j+1}^i \geq \hat{\lambda}_j^i$ (where inequality happens only if $\hat{\lambda}_{j+1} = (s/\gamma)^{1/(i-D)}$). Therefore all of the Z_q-polynomials decrease by a factor at most c between consecutive values of $\hat{\lambda}$. $\square$

3 Permanent Algorithm

Here we describe the simulated annealing algorithm for the permanent. We show the application of our improved cooling schedule, and our improvement in the mixing time bound for the Markov chain underlying the simulated annealing algorithm. We present the new inequalities which are key to the improved mixing time result. This analysis is more difficult than the earlier work of [8].

3.1 Preliminaries

Let $G = (V_1, V_2, E)$ be a bipartite graph with $|V_1| = |V_2| = n$. We will let $u \sim v$ denote the fact that $(u, v) \in E$. For $u \in V_1, v \in V_2$ we will have a positive real number $\lambda(u, v)$ called the *activity* of (u, v). If $u \sim v$, $\lambda(u, v) = 1$ throughout the algorithm, and otherwise, $\lambda(u,v)$ starts at 1 and drops to $1/n!$ as the algorithm evolves. The activities allow us to work on the complete graph on V_1 and V_2.

Let $\mathcal{P}$ denote the set of perfect matchings (recall that we are working on the complete graph now), and let $\mathcal{N}(u,v)$ denote the set of near-perfect matchings with holes (or unmatched vertices) at u and v. Similarly, let $\mathcal{N}(x,y,w,z)$ denote the set of matchings that have holes only at the vertices x,y,w,z. Let $\mathcal{N}_i$ denote the set of matchings with exactly i unmatched vertices. The set of states of the Markov chain is $\Omega = \mathcal{P} \cup \mathcal{N}_2$. For any matching M, denote its activity as

$$\lambda(M) := \prod_{(u,v)\in M} \lambda(u,v).$$

For a set S of matchings, let $\lambda(S) := \sum_{M\in S} \lambda(M)$. For $u \in V_1, v \in V_2$ we will have a positive real number $w(u,v)$ called the *weight* of the hole pattern u,v. Given weights w, the weight of a matching $M \in \Omega$ is

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

The weight of a set S of matchings is

$$w(S) := \sum_{M\in S} w(M).$$

For given activities, the *ideal weights* on hole patterns are the following:

$$w^*(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))} \tag{4}$$

Note that for the ideal weights all the $\mathcal{N}(u,v)$ and $\mathcal{P}$ have the same weight. Hence, $w^*(\Omega) = (n^2+1)\lambda(\mathcal{P})$.

For the purposes of the proof, we need to extend the weights to 4-hole matchings. Let

$$w^*(x,y,w,z) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(x,y,w,z))}$$

and for $M \in \mathcal{N}(x,y,w,z)$, let

$$w^*(M) = \lambda(M)w^*(x,y,w,z).$$

3.2 Markov chain definition

At the heart of the algorithm lies a Markov chain MC, which was used in [8], and a slight variant was used in [2, 7]. Let $\lambda : V_1 \times V_2 \to \mathbb{R}_+$ be the activities and $w : V_1 \times V_2 \to \mathbb{R}_+$ be the weights. The state space is Ω, the set of all perfect and near-perfect matchings of the complete bipartite graph on V_1, V_2. The stationary distribution π is proportional to w, i. e., $\pi(M) = w(M)/Z$ where $Z = \sum_{M\in\Omega} w(M)$.

The transitions $M_t \to M_{t+1}$ of the Markov chain MC are defined as follows:

1. If $M_t \in \mathcal{P}$, choose an edge e uniformly at random from M_t. Set $M' = M_t \setminus e$.
2. If $M_t \in \mathcal{N}(u,v)$, choose vertex x uniformly at random from $V_1 \cup V_2$.
   (a) If $x \in \{u, v\}$, let $M' = M \cup (u, v)$.
   (b) If $x \in V_2$ and $(w, x) \in M_t$, let $M' = M \cup (u, x) \setminus (y, x)$.
   (c) If $x \in V_1$ and $(x, z) \in M_t$, let $M' = M \cup (x, v) \setminus (x, z)$.
   (d) Otherwise, let $M' = M_t$.
3. With probability $\min\{1, w(M')/w(M_t)\}$, set $M_{t+1} = M'$; otherwise, set $M_{t+1} = M_t$.

Note, cases 1 and 2a move between perfect and near-perfect matchings, whereas cases 2b and 2c move between near-perfect matchings with different hole patterns.

The key technical theorem is that the Markov chain quickly converges to the stationary distribution π if the weights w are close to the ideal weights w^*. The mixing time $\tau(\delta)$ is the time needed for the chain to be within variation distance δ from the stationary distribution.

Theorem 8. *Assuming the weight function w satisfies inequality*

$$w^*(u, v)/2 \le w(u, v) \le 2w^*(u, v) \tag{5}$$

for every $(u, v) \in V_1 \times V_2$ with $\mathcal{M}(u, v) \neq 0$, then the mixing time of the Markov chain MC is bounded above by $\tau(\delta) = O(n^4(\ln(1/\pi(M)) + \log \delta^{-1}))$.

This theorem improves the mixing time bound by $O(n^2)$ over the corresponding result in [8]. The theorem will be proved in Section 6.

3.3 Bootstrapping Ideal Weights

We will run the chain with weights w close to w^*, and then we can use samples from the stationary distribution to redefine w so that they are arbitrarily close to w^*. For the Markov chain run with weights w, note that

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

Rearranging, we have

$$w^*(u,v) = \frac{\pi(\mathcal{P})}{\pi(\mathcal{N}(u,v))} w(u,v) \tag{6}$$

Given weights w which are a rough approximation to w^*, identity (6) implies an easy method to recalibrate weights w to an arbitrarily close approximation to w^*. We generate many samples from the stationary distribution, and observe the number of perfect matchings in our samples versus the number of near-perfect matchings with holes u, v. By generating sufficiently many samples, we can estimate $\pi(\mathcal{P})/\pi(\mathcal{N}(u,v))$ within an arbitrarily close factor, and hence we can estimate $w^*(u,v)$ (via (6)) within an arbitrarily close factor.

More precisely, recall that for $w = w^*$, the stationary distribution of the chain satisfies $\pi(\mathcal{N}(u,v)) = 1/(n^2+1)$. For weights w that are within a factor of 2 of the ideal weights w^*, it follows that $\pi(\mathcal{N}(u,v)) \geq 1/4(n^2+1)$. Then, by Chernoff bounds, $O(n^2 \log(1/\hat{\eta}))$ samples of the stationary distribution of the chain suffice to approximate $\pi(\mathcal{P})/\pi(\mathcal{N}(u,v))$ within a factor $\sqrt{2}$ with probability $\geq 1 - \hat{\eta}$. Thus, by (6) we can also approximate w^* within a factor $\sqrt{2}$ with the same bounds.

Theorem 8 (with $\delta = O(1/n^2)$) implies that $O(n^4 \log n)$ time is needed to generate each sample. To be precise this requires the use of "warm start" samples. We refer the reader to [8] for details of this aspect.

3.4 Simulated Annealing with New Cooling Schedule

In this section we present an $O^*(n^7)$ algorithm for estimating the ideal weights w^*. The algorithm will be used in Section 8 to approximate the permanent of a 0-1 matrix. The algorithm can be generalized to compute the permanent of general non-negative matrices, the necessary modifications are described in Section 9.

The algorithm runs in phases, each characterized by a parameter $\hat{\lambda}$. In every phase,

$$\lambda(e) = \begin{cases} 1 & \text{for } e \in E \\ \hat{\lambda} & \text{for } e \notin E \end{cases} \tag{7}$$

We start with $\hat{\lambda} = 1$ and slowly decrease $\hat{\lambda}$ until it reaches its target value $1/n!$.

At the start of each phase we have a set of weights within a factor 2 of the ideal weights, for all u, v, with high probability. Applying Theorem 8 we generate many samples from the stationary distribution. Using these samples and (6), we refine the weights to within a factor $\sqrt{2}$ of the ideal weights:

$$\frac{w^*(u,v)}{\sqrt{2}} \leq w(u,v) \leq \sqrt{2} w^*(u,v) \tag{8}$$

This allows us to decrease $\hat{\lambda}$ so that the current estimates of the ideal weights for $\hat{\lambda}_i$ are within a factor 2 of the ideal weights for $\hat{\lambda}_{i+1}$.

In [8], $O(n^2 \log n)$ phases are required. A straightforward way to achieve this is to decrease $\hat{\lambda}$ by a factor $(1 - 1/3n)$ between phases. This ensures that the weight of any matching changes by at most a factor $(1 - 1/3n)^n \leq \exp(1/3) < \sqrt{2}$.

We use only $O(n \log^2 n)$ phases by progressively decreasing $\hat{\lambda}$ by a larger amount per phase. Initially we decrease $\hat{\lambda}$ by a factor of roughly $(1 - 1/3n)$ per phase, but during the final phases we decrease $\hat{\lambda}$ by a constant factor per phase.

Here is the pseudocode of our algorithm. The algorithm outputs w which is a 2-approximation of the ideal weights w^* with probability $\geq 1 - \eta$. The quantities S and T satisfy $S = O(n^2(\log n + \log \eta^{-1}))$ and $T = O(n^4 \log n)$.

Algorithm for approximating ideal weights of 0-1 matrices:

Initialize $\hat{\lambda} = 1$ and $i = n$.
Initialize $w(u,v) \leftarrow n$ for all $(u,v) \in V_1 \times V_2$.
While $\hat{\lambda} > 1/n!$ do:
  Take S samples from MC with parameters λ, w, using a warm start simulation
    (i. e., inital matchings for the simulation are the final matchings from
    the previous simulation). We use T steps of the MC per sample,
    except for the first sample which needs $O(Tn \log n)$ steps.
  Use the samples to obtain estimates $w'(u,v)$ satisfying
    condition (8), for all u, v. The algorithm fails
    (i. e., (8) is not satisfied) with small probability.
  Set $\hat{\lambda} = 2^{-1/(2i)}\hat{\lambda}$.
  If $i > 2$ and $\hat{\lambda} < (n-1)!^{-1/(i-1)}$,
    Set $\hat{\lambda} = (n-1)!^{-1/(i-1)}$ and decrement $i = i - 1$.
  If $\hat{\lambda} < 1/n!$, set $\hat{\lambda} = 1/n!$.
  Set $w(u,v) = w'(u,v)$ for all $u \in V_1, v \in V_2$.
Output the final weights $w(u,v)$.

By Lemma 2, the above algorithm consists of $O(n \log^2 n)$ phases. This follows from setting $s = n$, $c = \sqrt{2}$, $\gamma = n!$, and $D = 1$ (the choice of D becomes clear in Section 7). In Section 7 we show that Lemma 3 implies that our weights at the start of each phase satisfy (5) assuming that the estimates w' satisfied condition (8) throughout the execution of the algorithm. Therefore the total running time is $O(STn \log^2 n) = O(n^7 \log^4 n)$. In Section 8 we show how to use the (constant factor) estimates of the ideal weights to obtain a $(1 \pm \epsilon)$-approximation of the permanent.

4 Canonical Paths for Proving Theorem 8

We bound the mixing time by the canonical paths method. For $(I, F) \in \Omega \times \mathcal{P}$, we will define a *canonical path* from I to F, denoted, $\gamma(I, F)$, which is of length $\leq n$. The path is along transitions of the Markov chain. We then bound the weighted sum of canonical paths (or "flow") through any transition. More precisely, for a transition $T = M \rightarrow M'$, let

$$\rho(T) = \sum_{\substack{(I,F)\in\Omega\times\mathcal{P}: \\ T\in\gamma(I,F)}} \frac{\pi(I)\pi(F)}{\pi(M)P(M, M')},$$

denote the *congestion* through the transition T, where $P(M, M')$ denotes the probability of the transition T. Let

$$\rho = \max_T \rho(T).$$

Then (see [16, 4]) for any initial state M_0, the mixing time is bounded as

$$\tau_{M_0}(\delta) \leq \frac{7n\rho}{\pi(\mathcal{P})} \left(\ln \pi(M_0)^{-1} + \ln \delta^{-1}\right)$$

The factor $1/\pi(\mathcal{P})$ comes from restricting to $F \in \mathcal{P}$, see Lemma 9 in [8]. When the weights w satisfy (5), we have $\pi(\mathcal{P}) = \Omega(1/n^2)$. Thus, to prove Theorem 8 we need to prove $\rho(T) = O(n)$ for every transition T.

We define the canonical paths now, and defer the bound on the congestion to Section 6, after presenting some combinatorial lemmas in Section 5. We will assume that the vertices of G are numbered. If $I \in \mathcal{P}$, then $I \oplus F$ consists of even length cycles. Let us assume that the cycles are numbered according to the smallest numbered vertex contained in them. The path $\gamma(I, F)$ "corrects" these cycles in order. Let $v_0, v_1, \ldots, v_{2k}$ be a cycle C, where v_0 is the smallest numbered vertex in C and $(v_0, v_1) \in I$. The path starts by unmatching edge (v_0, v_1) and successively interchanging edge (v_{2i-1}, v_{2i}) for edge (v_{2i}, v_{2i+1}). Finally it adds edge (v_{2k-1}, v_{2k}) to the matching.

If $I \in \mathcal{N}(w, z)$, then there is an augmenting path from w to z in $I \oplus F$. The canonical path starts by augmenting I along this path by first exchanging edges and finally adding the last edge. It then "corrects" the even cycles in order.

For a transition $T = M \rightarrow M'$, we need to bound the number of canonical paths passing thru T. We partition these paths into $2n^2 + 1$ sets,

$$cp_T = \{(I, F) \in \mathcal{P} \times \mathcal{P} : \gamma(I, F) \ni T\},$$

And, for all w, z,

$$cp_T^{w,z} = \{(I, F) \in \mathcal{N}(w, z) \times \mathcal{P} : \gamma(I, F) \ni T\}.$$

5 Key Technical Lemmas

The following lemma contains the new combinatorial inequalities which are the key to our improvement of $O(n^2)$ in Theorem 8. They will be used to analyze the congestion through a transition. In [8] weaker inequalities were proved without the sum in the left-hand side, and were a factor of 2 smaller in the right-hand side. The proof of Lemma 10 below improves on Lemma 7 in [8] by constructing more efficient mappings. We first present our mappings in the simpler setting of Lemma 9 and later use them to prove Lemma 10. Using these new inequalities to bound the congestion requires more work than the analysis of [8].

Lemma 9. *Let $u, w \in V_1$, $v, z \in V_2$ be distinct vertices. Then,*

1. $$\sum_{x,y:(u,y),(x,v)\in E} |\mathcal{N}(u,v)||\mathcal{N}(x,y)| \leq 2|\mathcal{P}|^2.$$

2. $$\sum_{x:(x,v)\in E} |\mathcal{N}(u,v)||\mathcal{N}(x,z)| \leq 2|\mathcal{N}(u,z)||\mathcal{P}|.$$

3. $$\sum_{x,y:(u,y),(v,x)\in E} |\mathcal{N}(u,v)||\mathcal{N}(x,y,w,z)| \leq 2|\mathcal{N}(w,z)||\mathcal{P}|.$$

The basic intuition for the proofs of these inequalities is straightforward. For example consider the first inequality. Take matchings $M \in \mathcal{N}(u,v), M' \in \mathcal{N}(x,y)$. The set $M \cup M' \cup (u,y) \cup (v,x)$ consists of a set of alternating cycles. Hence, this set can be broken into a pair of perfect matchings. One of the perfect matchings contains the edge (u,y) and one matching contains the edge (v,x). Hence, given the pair of perfect matchings, we can deduce the original unmatched vertices (by guessing which of the two edges incident to u), and thereby reconstruct M and M'. This outlines the approach for proving Lemma 9.

Proof. 1. We will construct a one-to-one map:

$$f_1 : \mathcal{N}(u,v) \times \bigcup_{x,y:(u,y),(v,x)\in E} \mathcal{N}(x,y) \to \mathcal{P} \times \mathcal{P} \times b,$$

where b is a bit, i.e., b is 0/1.

Let $L_0 \in \mathcal{N}(u,v)$ and $L_1 \in \cup_{x,y:(u,y),(v,x)\in E}\mathcal{N}(x,y)$. In $L_0 \oplus L_1$ the four vertices u, v, x, y each have degree one, and the remaining vertices have degree zero or two. Hence these four vertices are connected by two disjoint paths. Now there are three possibilities:

- If the paths are u to x and v to y, they must both be even.
- If the paths are u to v and x to y, they must both be odd.
- The third possibility, u to y and v to x is ruled out since such paths start with an L_0 edge and end with an L_1 edge and hence must be even length; on the other hand, they connect vertices across the bipartition and hence must be of odd length.

Now, the edges (u, y) and (v, x) are in neither matching, and so $(L_0 \oplus L_1) \cup \{(u, y), (v, x)\}$ contains an even cycle, say C, containing (u, y) and (v, x). We will partition the edges of $L_0 \cup L_1 \cup \{(u, y), (v, x)\}$ into two perfect matchings as follows. Let M_0 contain the edges of L_0 outside of C and alternate edges of C starting with edge (u, y). M_1 will contain the remaining edges. Bit b is set to 0 if $(x, v) \in M_0$ and to 1 otherwise. This defines the map f_1.

Next, we show that f_1 is one-to-one. Let M_0 and M_1 be two perfect matchings and b be a bit. If u and v are not in one cycle in $M_0 \oplus M_1$ then (M_0, M_1, b) is not mapped onto by f_1. Otherwise, let C be the common cycle containing u and v. Let y be the vertex matched to u in M_0. If $b = 0$, denote by x the vertex that is matched to v in M_0; else denote by x the vertex that is matched to v in M_1. Let L_0 contain the edges of M_0 outside C and let it contain the near-perfect matching in C that leaves u and v unmatched. Let L_1 contain the edges of M_1 outside C and let it contain the near-perfect matching in C that leaves x and y unmatched. It is easy to see that $f_1(L_0, L_1) = (M_0, M_1, b)$.

2. We will construct a one-to-one map:

$$f_2 : \mathcal{N}(u, v) \times \bigcup_{x:(v,x)\in E} \mathcal{N}(x, y) \to \mathcal{N}(u, y) \times \mathcal{P} \times b.$$

Let $L_0 \in \mathcal{N}(u, v)$ and $L_1 \in \cup_{x:(v,x)\in E}\mathcal{N}(x, y)$. As before, u, v, x, y are connected by two disjoint paths of the same parity in $L_0 \oplus L_1$ and $(v, x) \notin L_0 \cup L_1$. Hence, $L_0 \cup L_1 \cup \{(v, x)\}$ contains an odd path from u to y, say P. Construct $M_0 \in \mathcal{N}(u, y)$ by including all edges of L_0 not on P and alternate edges of P, leaving u, y unmatched. Let $M_1 \in \mathcal{P}$ consist of the remaining edges of $L_0 \cup L_1 \cup \{(v, x)\}$. Let $b = 0$ if $(v, x) \in M_0$, and to 1 otherwise. Clearly, path P appears in $M_0 \oplus M_1$, and as before, L_0 and L_1 can be retrieved from (M_0, M_1, b).

3. We will construct a one-to-one map:

$$f_3 : \mathcal{N}(u, v) \times \bigcup_{x,y:(u,y),(v,x)\in E} \mathcal{N}(x, y, w, z) \to \mathcal{N}(w, z) \times \mathcal{P} \times b.$$

Let $L_0 \in \mathcal{N}(u, v)$ and $L_1 \in \cup_{x,y:(u,y),(v,x)\in E}\mathcal{N}(x, y, w, z)$. Consider $L_0 \oplus L_1$. There are two cases. If there are two paths connecting the four vertices u, v, x, y (and a separate path connecting w and z), then the mapping follows using the construction given in 1. Otherwise, by parity considerations the only possibilities are:

- u to w and v to y even paths and x to z odd path
- u to x and v to z even paths and w to y odd path
- u to w and v to z even paths and x to y odd path
- u to v, x to z, and w to y odd paths

Now, $L_0 \cup L_1 \cup \{(u,y),(v,x)\}$ contains an odd path, say P, from w to z. Now, the mapping follows using the construction given in 2.

□

The following lemma is an extension of the previous lemma, which serves as a warm-up. This lemma is used to bound the congestion.

Lemma 10. *Let $u,w \in V_1$, $v,z \in V_2$ be distinct vertices. Then,*

1. $$\sum_{x\in V_1, y\in V_2} \lambda(u,y)\lambda(x,v)\lambda(\mathcal{N}(u,v))\lambda(\mathcal{N}(x,y)) \le 2\lambda(\mathcal{P})^2.$$

2. $$\sum_{x\in V_1} \lambda(x,v)\lambda(\mathcal{N}(u,v))\lambda(\mathcal{N}(x,z)) \le 2\lambda(\mathcal{N}(u,z))\lambda(\mathcal{P}).$$

3. $$\sum_{x\in V_1, y\in V_2} \lambda(u,y)\lambda(x,v)\lambda(\mathcal{N}(u,v))\lambda(\mathcal{N}(x,y,w,z)) \le 2\lambda(\mathcal{N}(w,z))\lambda(\mathcal{P}).$$

Proof. We will use the mappings f_1, f_2, f_3 constructed in Lemma 9. Observe that since mapping f_1 constructs matchings M_0 and M_1 using precisely the edges of L_0, L_1 and the edges $(u,y),(v,x)$, it satisfies

$$\lambda(u,y)\lambda(x,v)\lambda(L_0)\lambda(L_1) = \lambda(M_0)\lambda(M_1).$$

Summing over all pairs of matchings in

$$\mathcal{N}(u,v) \times \bigcup_{x,y:(u,y),(v,x)\in E} \mathcal{N}(x,y)$$

we get the first inequality. The other two inequalities follow in a similar way using mappings f_2 and f_3.

□

6 Bounding Congestion: Proof of Theorem 8

We bound the congestion separately for transitions which move between near-perfect matchings (Cases 2b and 2c), and transitions which move between a perfect and near-perfect matching. Our goal for this section will be to prove for every transition $T = M \to M'$,

$$\sum_{\substack{(I,F)\in\Omega\times\mathcal{P}:\\ T\in\gamma(I,F)}} \frac{w^*(I)w^*(F)}{w^*(M)} = O(w^*(\Omega)). \tag{9}$$

At the end of the section we will prove that this easily implies the desired bound on the congestion.

The following lemma converts into a more manageable form, the weighted sum of I, F pairs which contain a transition of the first type.

Lemma 11. *Let $T = M \to M'$ be a transition which moves between near-perfect matchings (i.e., Case 2b or 2c). Let $M \in \mathcal{N}(u,v), M' \in \mathcal{N}(u,v')$, $u \in V_1, v, v' \in V_2$, and $M' = M \setminus (v', x) \cup (v, x)$ for some $x \in V_1$. Then, the following hold:*

1.
$$\sum_{(I,F)\in cp_T} \lambda(I)\lambda(F) \le \sum_{y\in V_2} \lambda(\mathcal{N}(x,y))\lambda(u,y)\lambda(x,v)\lambda(M)$$

2. *For all $z \in V_2$,*
$$\sum_{(I,F)\in cp_T^{u,z}} \lambda(I)\lambda(F) \le \lambda(\mathcal{N}(x,z))\lambda(v,x)\lambda(M)$$

3. *For all $w \in V_1, w \neq u$ and $z \in V_2, z \neq v, v'$,*
$$\sum_{(I,F)\in cp_T^{w,z}} \lambda(I)\lambda(F) \le \sum_{y\in V_2} \lambda(\mathcal{N}(w,z,x,y))\lambda(u,y)\lambda(v,x)\lambda(M)$$

Proof. 1. We will first construct a one-to-one map

$$\eta_T : cp_T \to \bigcup_{x,y:(u,y),(v,x)\in E} \mathcal{N}(x,y).$$

Let $I, F \in \mathcal{P}$ and $(I,F) \in cp_T$. Let S be the set of cycles in $I \oplus F$. Order the cycles in S using the convention given in Section 4. Clearly, u, v, x lie on a common cycle, say $C \in S$, in $I \oplus F$. Since T lies on the canonical path from I to F, M has already corrected cycles before C and not yet corrected cycles after C in S. Let y be a neighbor of u on C. Define $M'' \in \mathcal{N}(x,y)$ to be the near-perfect matching that picks edges as follows: outside

C, it picks edges $(I \cup F) - M$, and on C it picks the near perfect-matching leaving x, y unmatched. Define $\eta_T(I, F) = M''$.

Clearly, $(M \oplus M'') \cup \{(u, v), (x, y)\}$ consists of the cycles in S, and I and F can be retrieved from M, M'' by considering the order defined on S. This proves that the map constructed is one-to-one. Since the union of edges in I and F equals the edges in $M \cup M'' \cup \{(u, v), (x, y)\}$,

$$\lambda(I)\lambda(F) = \lambda(M)\lambda(M'')\lambda(u, y)\lambda(x, v).$$

Summing over all $(I, F) \in cp_T$ we get the first inequality.

2. For all $z \in V_2$, we will first construct a one-to-one map

$$\eta_T^{u,z} : cp_T^{u,z} \to \mathcal{N}(x, z).$$

Let $I \in \mathcal{N}(u, z), F \in \mathcal{P}$ and $(I, F) \in cp_T^{u,z}$. Let S be the set of cycles and P be the augmenting path from u to z in $I \oplus F$. Clearly, v, x lie on P. M has "corrected" part of the path P and none of the cycles in S. It contains the edges of I from z to v and the edges of F from x to u. Also, it contains the edges of I from the cycles in S, as well as the edges in $I \cap F$.

Construct matching $M'' \in \mathcal{N}(x, z)$ as follows. It contains the edges of F from the cycles in S, the edges $I \cap F$ and $(P - \{(x, v)\} - M)$. Define $\eta_T^{u,z}(I, F) = M''$. It is easy to see that $M \cup M'' = I \cup F \cup \{(x, v)\}$. Therefore,

$$\lambda(I)\lambda(F) = \lambda(M)\lambda(M'')\lambda(x, v).$$

Furthermore, I, F can be retrieved from M, M''. Hence, summing over all $(I, F) \in cp_T^{u,z}$ we get the second inequality.

3. For all $w \in V_1, w \neq u$ and $z \in V_2, z \neq v, v'$, we will first construct a one-to-one map

$$\eta_T^{w,z} : cp_T^{w,z} \to \bigcup_{y: u \sim y} \mathcal{N}(w, z, x, y).$$

Let $I \in \mathcal{N}(w, z), F \in \mathcal{P}$ and $(I, F) \in cp_T^{w,z}$. Let S be the set of cycles and P be the augmenting path from w to z in $I \oplus F$. Clearly, u, v, x lie on a common cycle, say $C \in S$, in $I \oplus F$. and M has already "corrected" P and so looks like F on P. Construct $M'' \in \mathcal{N}(w, z, x, y)$ as follows. On P, it looks like I. Outside $P \cup C$, it picks edges $(I \cup F) - M$, and on C it picks the near perfect-matching leaving x, y unmatched. Define $\eta_T^{w,z}(I, F) = M''$. It is easy to see that $M \cup M'' = I \cup F \cup \{(u, y), (x, v)\}$. Therefore,

$$\lambda(I)\lambda(F) = \lambda(M)\lambda(M'')\lambda(u, y)\lambda(x, v).$$

Furthermore, I, F can be retrieved from M, M''. Hence, summing over all $(I, F) \in cp_T^{w,z}$ we get the third inequality.

$\square$

We now prove (9) for the first type of transitions. The proof applies Lemma 11 and then Lemma 10. We break the statement of (9) into two cases depending on whether I is a perfect matching or a near-perfect matching.

Lemma 12. *For a transition $T = M \to M'$ which moves between near-perfect matchings (i.e., Case 2b or 2c), the congestion from $(I, F) \in \mathcal{P} \times \mathcal{P}$ is bounded as*

$$\sum_{(I,F)\in cp_T} \frac{w^*(I)w^*(F)}{w^*(M)} \leq \frac{2w^*(\Omega)}{n^2} \tag{10}$$

And, the congestion from $(I, F) \in \mathcal{N}_2 \times \mathcal{P}$ is bounded as

$$\sum_{w\in V_1, z\in V_2} \sum_{(I,F)\in cp_T^{w,z}} \frac{w^*(I)w^*(F)}{w^*(M)} \leq 3w^*(\Omega) \tag{11}$$

Proof. The transition T is sliding an edge, let x denote the pivot vertex, let $M \in \mathcal{N}(u, v), M' \in \mathcal{N}(u, v')$, $u \in V_1, v, v' \in V_2$. Thus, $M' = M \setminus (v', x) \cup (v, x)$ for some $x \in V_1$. The encodings from Lemma 11 will always contain x as a hole.

We begin with the proof of (10).

$$\begin{aligned}
&\sum_{(I,F)\in cp_T} \frac{w^*(I)w^*(F)}{w^*(M)} \\
&= \sum_{(I,F)\in cp_T} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{P})} \\
&\leq \sum_{y\in V_2} \frac{\lambda(\mathcal{N}(x,y))\lambda(u,y)\lambda(x,v)\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{P})} \qquad \text{by Lemma 11} \\
&\leq 2\lambda(\mathcal{P}) \qquad \text{by Lemma 10} \\
&= \frac{2w^*(\Omega)}{n^2+1}
\end{aligned}$$

This completes the proof of (10). We now prove (11) in two parts. This first bound covers the congestion due to the first part of the canonical paths from a near-perfect matching to a perfect matching – unwinding the augmenting path. The second bound covers the second part of these canonical paths when we unwind the alternating cycle(s). During the unwinding of the augmenting path, one of the holes of the transition is the same as one of the holes of the initial near-perfect matching. This is what characterizes the first versus the second part of the canonical path.

$$\begin{aligned}
&\sum_{z\in V_2}\sum_{(I,F)\in cp_T^{u,z}} \frac{w^*(I)w^*(F)}{w^*(M)} \\
&\quad= \sum_{z\in V_2}\sum_{(I,F)\in cp_T^{u,z}} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{N}(u,z))} \\
&\quad\leq \sum_{z\in V_2} \frac{\lambda(\mathcal{N}(x,z))\lambda(v,x)\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(u,z))} \qquad \text{by Lemma 11} \\
&\quad\leq \sum_{z\in V_2} 2\lambda(\mathcal{P}) \qquad \text{by Lemma 10} \\
&\quad= \frac{2n}{n^2+1} w^*(\Omega) \\
&\quad\leq w^*(\Omega)
\end{aligned}$$

Finally, bounding the congestion from the unwinding of the alternating cycle(s) on canonical paths from near-perfect matchings to perfect matchings,

$$\begin{aligned}
&\sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}}\sum_{(I,F)\in cp_T^{w,z}} \frac{w^*(I)w^*(F)}{w^*(M)} \\
&\quad= \sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}}\sum_{(I,F)\in cp_T^{w,z}} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{N}(w,z))} \\
&\quad\leq \sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}}\sum_{y\in V_2} \frac{\lambda(\mathcal{N}(w,z,x,y))\lambda(u,y)\lambda(v,x)\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(w,z))} \qquad \text{by Lemma 11} \\
&\quad\leq \sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}} 2\lambda(\mathcal{P}) \qquad \text{by Lemma 10} \\
&\quad\leq 2w^*(\Omega)
\end{aligned}$$

□

We now follow the same approach as Lemmas 11 and 12 to prove (9) for transitions moving between a perfect and near-perfect matching. The proofs in this case are easier.

Lemma 13. *For a transition $T = M \to M'$ which adds or subtracts an edge (i.e., Case 1 or 2a), let N denote the near-perfect matching of M and M'. Then,*

$$\sum_{(I,F)\in cp_T} \lambda(I)\lambda(F) \leq \lambda(\mathcal{P})\lambda(u,v)\lambda(N).$$

And, for all $w \in V_1, z \in V_2$,

$$\sum_{(I,F)\in cp_T^{w,z}} \lambda(I)\lambda(F) \leq \lambda(\mathcal{N}(w,z))\lambda(u,v)\lambda(N).$$

Proof. Let P denote the perfect matching of M and M'. Define $\eta = \eta_T^{w,z} : cp_T^{w,z} \to \mathcal{N}(w,z)$ as

$$\eta(I,F) = I \cup F \setminus P.$$

The mapping satisfies $\lambda(I)\lambda(F) = \lambda(P)\lambda(\eta(I,F))$. Note, $\lambda(P) = \lambda(N)\lambda(u,v)$. Since the mapping is one-to-one, summing over all $N' \in \mathcal{N}(w,z)$ proves the lemma for all w, z. The proof is identical for cp_T with the observation that when $I \in \mathcal{P}$, we have $I \cup F \setminus P$ is in $\mathcal{P}$. $\square$

Lemma 14. *For a transition $T = M \to M'$ which adds or subtracts an edge (i.e., Case 1 or 2a), the congestion from $(I,F) \in \Omega \times \mathcal{P}$ is bounded as*

$$\sum_{w,z} \sum_{(I,F)\in cp_T^{w,z}} \frac{w^*(I)w^*(F)}{w^*(M)} \leq w^*(\Omega) \tag{12}$$

$$\sum_{(I,F)\in cp_T} \frac{w^*(I)w^*(F)}{w^*(M)} \leq \frac{w^*(\Omega)}{n^2} \tag{13}$$

Proof. Let $M \in \mathcal{N}(u,v)$ and $M' \in \mathcal{P}$, thus the transition adds the edge (u,v). The proof for the transition which subtracts the edge will be analogous. The proof is a simplified version of Lemma 12, since the encoding is simpler in this case (see Lemma 13 versus Lemma 11).

Observe that for any x, y,

$$\lambda(x,y)\lambda(\mathcal{N}(x,y)) \leq \lambda(\mathcal{P}) \tag{14}$$

We begin with the proof of (12).

$$\begin{aligned}\sum_{w,z} \sum_{(I,F)\in cp_T^{w,z}} \frac{w^*(I)w^*(F)}{w^*(M)} &= \sum_{w,z} \sum_{(I,F)\in cp_T^{w,z}} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{N}(w,z))} \\ &\leq \sum_{w,z} \lambda(u,v)\lambda(\mathcal{N}(u,v)) \qquad \text{by Lemma 13} \\ &\leq w^*(\Omega) \qquad \text{by (14)}\end{aligned}$$

We now prove (13).

$$\begin{aligned}\sum_{(I,F)\in cp_T} \frac{w^*(I)w^*(F)}{w^*(M)} &= \sum_{(I,F)\in cp_T} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{P})} \\ &\le 2\lambda(u,v)\lambda(\mathcal{N}(u,v)) \qquad \text{by Lemma 13} \\ &\le \lambda(\mathcal{P}) \qquad \text{by (14)}\end{aligned}$$

□

Proof of Theorem 8. Inequality (5) implies for any set of matchings $S \subset \Omega$, the stationary distribution $\pi(S)$ under w is within a factor 4 of the distribution under w^*. Therefore, to prove Theorem 8 it suffices to consider the stationary distribution with respect to w^*. In other words, we need to prove, for every transition T, $\rho(T) = O(n)$ where, for $M \in \Omega$, $\pi(M) = w^*(M)/w^*(\Omega)$. Then for weights satisfying (5) the congestion increases by at most a constant factor. Thus, we need to prove

$$\sum_{\substack{(I,F)\in\Omega\times\mathcal{P}:\\ T\in\gamma(I,F)}} \frac{w^*(I)w^*(F)}{w^*(M)P(M,M')} = O(nw^*(\Omega)).$$

Recall that the transitions $M_t \to M_{t+1}$ of our Markov chain are according to the Metropolis filter. From M_t, a new matching N is proposed with probability $1/4n$, and then the proposed new matching is accepted with probability $\min\{1, w^*(N)/w^*(M_t)\}$. Hence, for the transition $T = M \to M'$,

$$w^*(M)P(M,M') = \frac{1}{4n}\min\{w^*(M), w^*(M')\}.$$

Since the chain is reversible for every transition $T = M \to M'$, there is a reverse transition $T' = M' \to M$. To prove Theorem 8, it suffices to prove that for every transition $T = M \to M'$,

$$\sum_{\substack{(I,F)\in\Omega\times\mathcal{P}:\\ T\in\gamma(I,F)}} \frac{w^*(I)w^*(F)}{w^*(M)} = O(w^*(\Omega)). \tag{15}$$

Lemmas 12 and 14 imply (15) which completes the proof of the Theorem. □

7 Phases in the Permanent Algorithm

In this section we show that the choice of $\hat{\lambda}$ from the weight-estimating algorithm ensures that (5) is satisfied in each phase. Recall that we can obtain a refined estimate of the ideal weights in each phase, see (8). We need to guarantee that the weights of two consecutive phases do not differ too much. Namely, if they are within a $\sqrt{2}$ factor of each other, together with (8) we have (5) for the next phase. As we will see shortly, for our choice of activities the ideal weights $w^*(u,v)$ are a ratio of two polynomials of degree $\leq n$ evaluated at $\hat{\lambda}$. This observation will allow us to use Lemma 3.

Definition 15. *We say that a matching $M \in \mathcal{P}$ of a complete bipartite graph* covers k *edges of a graph G if the size of $M \cap E(G)$ is k. Let*

$$R_G(x) = \sum_{k=0}^{n} p_k x^{n-k},$$

where p_k is the number of matchings in $\mathcal{P}$ covering k edges of G.

Note that the ideal weights w^*, defined by (4), for activities given by (7) can be expressed as follows

$$w^*_{\hat{\lambda}}(u,v) = \frac{R_G(\hat{\lambda})}{R_{G\setminus\{u,v\}}(\hat{\lambda})}. \tag{16}$$

First we observe that every R-polynomial has a positive low-degree coefficient. In particular, the coefficient of either x^0 or x^1 is positive in each of the polynomials R_G, $R_{G\setminus\{u,v\}}$, for every $u \in V_1$, $v \in V_2$. This follows from the fact that G contains a perfect matching. Let M be a perfect matching of G. The existence of M implies that the absolute coefficient in R_G is positive. Similarly, if $(u,v) \in M$, then the absolute coefficient in $R_{G\setminus\{u,v\}}$ is positive because $M \setminus \{(u,v)\}$ is a perfect matching in $G \setminus \{u,v\}$. If $(u,v) \notin M$, let u', resp. v' be the vertices matched to u and v in M, and let $M' = M \cup \{(v',u')\} \setminus \{(u,u'),(v,v')\}$. Depending on (v',u') being an edge in G, the cardinality of M' is either $n-1$ or $n-2$. Therefore either the coefficient of x^0 or x^1 in $R_{G\setminus\{u,v\}}$ is positive.

Now we are ready to apply Lemma 3. Let $c = \sqrt{2}$, $\gamma = n!$, $D = 1$, $s = n$, $Q_1 = R_G$ and the polynomials $Q_2, \ldots, Q_{n^2+1}$ are the $R_{G\setminus\{u,v\}}$ polynomials for $u \in V_1, v \in V_2$. Let $\hat{\lambda}_0, \ldots, \hat{\lambda}_\ell$ be the sequence obtained from the algorithm in Section 2.3. Notice, that we obtain the same sequence in the algorithm for estimating weights of the permanent. Then

$$\begin{array}{ll} R_G(\hat{\lambda}_k) \geq R_G(\hat{\lambda}_{k+1}) \geq R_G(\hat{\lambda}_k)/\sqrt{2}, & \text{and} \\ R_{G\setminus\{u,v\}}(\hat{\lambda}_k) \geq R_{G\setminus\{u,v\}}(\hat{\lambda}_{k+1}) \geq R_{G\setminus\{u,v\}}(\hat{\lambda}_k)/\sqrt{2} & \text{for every } u,v. \end{array} \tag{17}$$

To shorten the notation, we will use w_k to denote $w_{\hat{\lambda}_k}$. Equations (16) and (17) imply the w^*_k and w^*_{k+1} are within a constant factor. Moreover if the weight-estimating algorithm does not fail, i. e., the w_k satisfy (8), then w_k and w_{k+1} are within a constant factor as well.

The following corollaries are used in Section 8 for approximating the permanent once a good approximation of the ideal weights is obtained.

Corollary 16. *For every u, v,*

$$\frac{1}{\sqrt{2}} w^*_{k+1}(u,v) \le w^*_k(u,v) \le \sqrt{2} w^*_{k+1}(u,v). \tag{18}$$

If the w_k satisfy (8) then for every u, v,

$$\frac{1}{2\sqrt{2}} w_{k+1}(u,v) \le w_k(u,v) \le 2\sqrt{2} w_{k+1}(u,v). \tag{19}$$

Note that

$$w_k(\Omega) = R_G(\hat{\lambda}_k) + \sum_{u,v} R_{G\setminus\{u,v\}}(\hat{\lambda}_k) w_k(u,v).$$

Corollary 16 and (17) imply the following result.

Corollary 17. *If the weight-estimating algorithm does not fail then*

$$\frac{w_k(\Omega)}{2\sqrt{2}} \le w_{k+1}(\Omega) \le 2\sqrt{2} w_k(\Omega).$$

Let $M \in \Omega$ be a matching. Note that $\hat{\lambda}_{k+1} \le \hat{\lambda}_k$ and hence $\lambda_{k+1}(M) \le \lambda_k(M)$. For $M \in P$ we have $w_{k+1}(M) \le w_k(M)$. If $M \in \mathcal{N}(u,v)$ then, assuming that the weight-estimating algorithm did not fail we have $w_{k+1}(M) = w_{k+1}(u,v)\lambda_{k+1}(M) \le 2\sqrt{2} w_k(u,v)\lambda_k(M) = 2\sqrt{2} w_k(M)$. Hence we have the following observation.

Corollary 18. *Assume that the weight-estimating algorithm does not fail. Then for any matching $M \in \Omega$*

$$w_{k+1}(M) \le 2\sqrt{2} w_k(M).$$

8 Reduction from Counting to Sampling

In this section we show how to approximate the permanent of a matrix once we have good approximations of the ideal weights. For simplicity we consider the case of 0/1 matrix. The argument follows the argument of Section 5 from [8].

We want to estimate the number of perfect matchings $|\mathcal{P}_G|$ in a bipartite graph G. Let $\hat{\lambda}_0 = 1 > \hat{\lambda}_1 > \cdots > \hat{\lambda}_\ell = 1/n!$, $\ell = O(n \log^2 n)$ be the sequence of $\hat{\lambda}$ used in the weight-estimating algorithm. Assume that the algorithm did not fail, i. e., the hole weights $w_0, \dots, w_\ell$ computed by the algorithm are within a factor of $\sqrt{2}$ from the ideal weights $w^*_0, \dots, w^*_\ell$. Recall that $w_0(\Omega) = n!(n^2+1)$. We can express $|\mathcal{P}_G|$ as a telescoping product:

$$|\mathcal{P}_G| = \frac{|\mathcal{P}_G|}{w_\ell(\Omega)} \frac{w_\ell(\Omega)}{w_{\ell-1}(\Omega)} \frac{w_{\ell-1}(\Omega)}{w_{\ell-2}(\Omega)} \cdots \frac{w_1(\Omega)}{w_0(\Omega)} w_0(\Omega) = n!(n^2+1)\alpha^* \prod_{0 \le k < \ell} \alpha_k, \tag{20}$$

where $\alpha^* = |\mathcal{P}_G|/w_\ell(\Omega)$ and $\alpha_k = w_{k+1}(\Omega)/w_k(\Omega)$. Note that corollary 17 implies $\alpha_k = \Theta(1)$. The quantity $w_\ell(\Omega)$ is within a constant factor of $(n^2+1)|\mathcal{P}_G|$ and hence $\alpha^* = \Theta(1/n^2)$.

Let $X_k \sim w_k$ denote a random matching chosen from the distribution defined by w_k, i.e., the probability of a matching M is $w_k(M)/w_k(\Omega)$. Let $Y_k = w_{k+1}(X_k)/w_k(X_k)$. Then Y_k is an unbiased estimator for α_k:

$$\mathbf{E}\left(Y_k\right) = \mathbf{E}_{X_k \sim w_k}\left(\frac{w_{k+1}(X_k)}{w_k(X_k)}\right) = \sum_{M \in \Omega} \frac{w_k(M)}{w_k(\Omega)} \frac{w_{k+1}(M)}{w_k(M)} = \frac{w_{k+1}(\Omega)}{w_k(\Omega)} = \alpha_k. \tag{21}$$

For $k = \ell$ let $Y_\ell = 1_{X_\ell \in \mathcal{P}_G}$, where $1_{M \in \mathcal{P}_G}$ is the indicator function which takes value 1 if M is a perfect matching of G, and 0 otherwise. Then Y_ℓ is an unbiased estimator for α^*:

$$\mathbf{E}\left(Y_\ell\right) = \mathbf{E}_{X_\ell \sim w_\ell}\left(1_{X_k \in \mathcal{P}_G}\right) = \sum_{M \in \Omega} \frac{w_\ell(M)}{w_\ell(\Omega)} 1_{M \in \mathcal{P}_G} = \frac{|\mathcal{P}_G|}{w_\ell(\Omega)} = \alpha^*. \tag{22}$$

Corollary 18 implies that $0 \le Y_k \le 2\sqrt{2}$ and hence $\mathbf{Var}\left(Y_k\right) = O(1)$. Thus the mean $\overline{Y}_k$ of $\Theta(\ell\epsilon^{-2})$ samples of Y_k has $\mathbf{Var}\left(\overline{Y}_k\right) = O(\epsilon^2/\ell)$. Therefore

$$\frac{\mathbf{E}\left(\overline{Y}_k^2\right)}{\mathbf{E}\left(\overline{Y}_k\right)^2} = 1 + \frac{\mathbf{Var}\left(\overline{Y}_k\right)}{\mathbf{E}\left(\overline{Y}_k\right)^2} = 1 + O(\epsilon^2/\ell),$$

since $\mathbf{E}\left(\overline{Y}_k\right) = \mathbf{E}\left(Y_k\right) = \Theta(1)$.

Let $Z = \prod_{k=0}^{\ell-1} \overline{Y}_k$. We have

$$\frac{\mathbf{E}\left(Z^2\right)}{\mathbf{E}\left(Z\right)^2} = (1 + O(\epsilon^2/\ell))^\ell = 1 + O(\epsilon^2),$$

and hence $\mathbf{Var}\left(Z\right)/(\mathbf{E}\left(Z\right))^2 = O(\epsilon^2)$. Thus by the Chebychev inequality Z is within a factor $1 \pm \epsilon/6$ from $\mathbf{E}\left(Z\right) = \prod_{k=0}^{\ell-1} \alpha_k$ with probability $\ge 11/12$ for appropriately chosen constants within the O notation.

Even though we cannot sample from w_k exactly, it suffices to sample the X_k (and hence Y_k) from a distribution within variation distance $O(\epsilon/\ell)$ from w_k. The expectation of Z will be within factor $1 \pm \epsilon/6$ from $\prod_{k=0}^{\ell-1} \alpha_k$ and the above variance argument remains unchanged.

Similarly to estimate α^* it is enough to take the mean of $O(n^2\epsilon^{-2})$ values of Y_ℓ with X_ℓ from a distribution within variation distance $O(\epsilon)$ from w_ℓ. The result is an estimate of α^* within a factor of $1 \pm \epsilon/3$ with probability at least 11/12.

Therefore, $n!(n^2+1)\prod_{k=0}^{\ell} \overline{X}_k$ estimates $|\mathcal{P}_G|$ within a factor of $1 \pm \epsilon$ with probability $\geq 5/6$. The total running time of the reduction from counting to sampling is $O(\ell^2/\epsilon^2 n^4 \log n) = O(n^6 \log^5 n\epsilon^{-2})$. See [8] for details.

9 Non-negative Matrices

A slight modification of our algorithm can be used to compute the permanent of a matrix $A = (a_{i,j})_{n\times n}$ with non-negative entries $a_{i,j}$. Suppose $per(A) > 0$. (The question $per(A) = 0$ can be decided in deterministic polynomial time by finding the maximum matching in the corresponding weighted bipartite graph. The permanent is positive if and only if there exists a matching of nonzero weight.) Let $a_{\max} = \max_{i,j} a_{i,j}$ and $a_{\min} = \min_{i,j:a_{i,j}>0} a_{i,j}$. As before, we will have a uniform activity $\hat{\lambda}$ which will decrease from 1 to a value close to 0. The activity of an edge (u,v) is a function of x:

$$\lambda_{u,v}(x) = \frac{a_{u,v} + x(a_{\max} - a_{u,v})}{a_{\min}}$$

The activities and weights of matchings (of the underlying complete bipartite graph) are defined analogously to the 0/1 case. Now the R-polynomials are:

$$R_A(x) = \sum_{M\in\mathcal{P}} \prod_{(u,v)\in M} \lambda_{u,v}(x)$$

Notice that for the 0/1 case we get the R_G polynomial. The weights are

$$w(u,v) = \frac{R_A(\hat{\lambda})}{R_{A\setminus\{u,v\}}(\hat{\lambda})},$$

where $A\setminus\{u,v\}$ is the $(n-1)\times(n-1)$ matrix obtained from A by removing the u-th row and the v-th column. Since $a_{\min}^n \leq per(A) \leq a_{\max}^n n!$ we get that $R(1) = per(A)/a_{\min}^n \leq (a_{\max}/a_{\min})^n n!$. Moreover, since for $a_{u,v} > 0$ the activity $\lambda_{u,v}(x) \geq 1$ for $x = 0$, the absolute coefficient of R_G is at least 1. By an argument analogous to the 0/1 case, either the absolute coefficient or the coefficient of the linear term will be ≥ 1 in every $R_{A\setminus\{u,v\}}$. Therefore, as before, we may use Lemma 2 and Lemma 3 to get the following algorithm for estimating the weights (recall that $S = O(n^2(\log n + \log \eta^{-1}))$ and $T = O(n^4 \log n)$).

Algorithm for approximating ideal weights of non-negative matrices:

Initialize $\hat{\lambda} := 1$ and $i := n$.

Initialize $w(u, v) \leftarrow n a_{\max}/a_{\min}$ for all $(u, v) \in V_1 \times V_2$.
While $\hat{\lambda} > (a_{\min}/a_{\max})^n/n!$ do:
Take S samples from MC with parameters λ, w, using a warm start simulation (i. e., inital matchings for the simulation are the final matchings from the previous simulation). We use T steps of the MC per sample, except for the first sample which needs $O(Tn \log n)$ steps.
Use the samples to obtain estimates $w'(u, v)$ satisfying condition (8), for all u, v, with high probability.
Set $\hat{\lambda} = 2^{-1/(2i)}\hat{\lambda}$.
If $i > 2$ and $\hat{\lambda} < ((a_{\max}/a_{\min})^n(n-1)!)^{-1/(i-1)}$,
Set $\hat{\lambda} := ((a_{\max}/a_{\min})^n(n-1)!)^{-1/(i-1)}$ and decrement i by 1.
If $\hat{\lambda} < (a_{\min}/a_{\max})^n/n!$, set $\hat{\lambda} := (a_{\min}/a_{\max})^n/n!$.
Set $w(u, v) := w'(u, v)$ for all $u \in V_1, v \in V_2$.
Output the final weights $w(u, v)$.

The correctness of the algorithm follows from Lemma 3 for parameters $s = n$, $\gamma = (a_{\max}/a_{\min})^n n!$, $D = 1$, and $c = \sqrt{2}$. By Lemma 2, we need $O(n \log n(\log n + \log(a_{\max}/a_{\min})))$ different values of $\hat{\lambda}$. Therefore the running time of the algorithm is $O(ST(n \log n(\log n + \log \frac{a_{\max}}{a_{\min}}))) = O(n^7 \log^3 n(\log n + \log \frac{a_{\max}}{a_{\min}}))$.

As discussed in [8], this can be converted to a strongly polynomial time algorithm for approximating the permanent by first applying the algorithm of Linial, Samorodnitsky and Wigderson [12], which converts our input matrix into a nearly doubly stochastic matrix. See Section 7 of [8] for details.

10 Discussion

With the improvement in running time of the approximation algorithm for the permanent, computing permanents of $n \times n$ matrices with $n \approx 100$ now appears feasible. Further improvement in running time is an important open problem.

Some avenues for improvement are the following. We expect that the mixing time of the underlying chain is better that $O(n^4)$. Some slack in the analysis is in the application of the new inequalities to bound the congestion. In their application we simply use a sum over y, whereas the inequalities hold for a sum over x and y as stated in Lemma 10. Another direction is the number of samples needed per phase. It is possible that fewer samples are needed at each intermediate activity for estimating the ideal weights w^*. Perhaps the w^* satisfy relations which allow for fewer samples to infer them.

References

[1] I. Bezáková, N. Bhatnagar, and E. Vigoda, Sampling Binary Contingency Tables with a Greedy Start. To appear in *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), 2006.

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

[3] R. Bubley and M. Dyer, Path coupling: A technique for. proving rapid mixing in Markov chains. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 223–231, 1997.

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

[5] T. Hayes, E. Vigoda, A Non-Markovian Coupling for Randomly Sampling Colorings. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 618–627, 2003.

[6] M. Jerrum, A Very Simple Algorithm for Estimating the Number of k-Colorings of a Low-Degree Graph. *Random Struct. Algorithms*, 7(2):157–166, 1995.

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

[8] M.R. Jerrum, A. Sinclair and E. Vigoda, A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Journal of the Association for Computing Machinery*, 51(4):671-697, 2004.

[9] A. Kalai and S. Vempala, Simulated Annealing for Convex Optimization, Preprint, 2004.

[10] P. W. Kasteleyn, The statistics of dimers on a lattice, I., The number of dimer arrangements on a quadratic lattice, *Physica*, 27:1664-1672, 1961.

[11] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, Optimization by Simulated Annealing, *Science*, 220:671-680, 1983.

[12] N. Linial, A. Samorodnitsky and A. Wigderson, A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents, *Combinatorica* 20: 545–568, 2000.

[13] L. Lovász and S. Vempala, Simulated Annealing in Convex Bodies and an $O^*(n^4)$ Volume Algorithm. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 650–659, 2003.

[14] P. McCullagh, On prediction and density estimation, Preprint, 2004.

[15] H. Pasula, S.J. Russell, M. Ostland, Y. Ritov, Tracking Many Objects with Many Sensors, *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1160–1171, 1999.

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

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

APPENDIX

11 A Lower Bound of the Number of k-colorings

We will use $a^{\underline{b}}$ to denote $a!/(a-b)!$. Let G be a graph and for each vertex i of G let L_i be a list of colors. A valid list-coloring of G is a coloring such that each i has a color from L_i, and no two neighbors have the same color.

Lemma 19. *Let G be a graph with n vertices. Let d_j be the number of vertices of degree j in G. Let s be an integer, $s \geq 1$. Let $L_1, \dots, L_n$ be sets such that $|L_i| \geq s + \deg i$ for each vertex i of G. Let Ω be the set of valid list-colorings of G. Then*

$$|\Omega| \geq \prod_{j=0}^{n} c_j^{d_j},$$

where $c_j = ((s+j)^{\underline{j+1}})^{1/(j+1)}$.

Proof. We will use induction on $d_0 + \cdots + d_k$. Let v be the vertex of minimum degree ℓ. We have $|L_v| \geq \ell + s$. Let $j_1, \dots, j_\ell$ be the degrees of the neighbors of v. Note that $j_i \geq \ell$ for $i = 1, \dots, \ell$.

By the induction hypothesis

$$|\Omega| \geq (\ell+s)\left(\prod_{j=0}^{n} c_j^{d_j}\right)\frac{1}{c_\ell}\left(\prod_{i=1}^{\ell}\frac{c_{j_i-1}}{c_{j_i}}\right) \geq (\ell+s)\left(\prod_{j=0}^{n} c_j^{d_j}\right)\frac{1}{c_\ell}\left(\frac{c_{\ell-1}}{c_\ell}\right)^{\ell} = \prod_{j=0}^{n} c_j^{d_j}, \quad (23)$$

where in the second inequality we used the inequality $c_j/c_{j+1} \geq c_{j-1}/c_j$, which we prove next.

Let $T = (s+j)^{\underline{j+1}}$. We want to show

$$T^{2/(j+1)} \geq \left(\frac{T}{s+j}\right)^{1/j} (T(s+j+1))^{1/(j+2)}.$$

After raising both sides to $-j(j+1)(j+2)/2$ and multiplying by $T^{\binom{j+1}{2}+\binom{j+2}{2}}$ we obtain an equivalent inequality

$$T \leq \frac{(s+j)^{\binom{j+2}{2}}}{(s+j+1)^{\binom{j+1}{2}}}. \tag{24}$$

Using the inequality between arithmetic and geometric mean

$$\left((s+j)^{\underline{j+1}}(s+j+1)^{\binom{j+1}{2}}\right)^{1/\binom{j+2}{2}} \leq s+j,$$

which implies (24). Therefore $c_j^2 \geq c_{j+1}c_{j-1}$ and hence the induction step (23) is proved. □

For k-colorings we obtain the following result.

Corollary 20. *Let G be a graph with n vertices and maximum degree Δ. Let $k > \Delta$. Let Ω be the set of valid k-colorings of G. Then*

$$|\Omega| \geq \left(k^{\underline{\Delta+1}}\right)^{n/(\Delta+1)} \geq \left(k - \Delta\left(1 - \frac{1}{e}\right)\right)^n.$$

Proof. Let $s = k - \Delta$. The first inequality follows from Lemma 19 with the $L_i = [k]$.

The second inequality is equivalent to

$$(s+\Delta)^{\underline{\Delta+1}} \geq (s+\Delta/e)^{\Delta+1}. \tag{25}$$

The inequality (25) is true for $\Delta = 0$ and hence from now on we assume $\Delta \geq 1$.

We first show that (25) holds with strict inequality for $s = 1$. We want to show $n! > (1+(n-1)/e)^n$. The inequality $n! > \sqrt{2\pi n}(n/e)^n$ implies that it is enough to show $2\pi n \geq ((n+e-1)/n)^{2n}$, which (using $1+x \leq e^x$) is implied by $2\pi n \geq e^{2(e-1)}$. Hence we proved (25) for $s = 1$ and $n \geq 5$. For $n \leq 4$ and $s = 1$ the (strict version of) inequality (25) is easily verified by hand.

Let $f(s,\Delta) = \sum_{i=0}^{\Delta} \ln \frac{s+i}{s+\Delta/e}$. Inequality (25) is equivalent to $f(s,\Delta) \geq 0$. From previous paragraph we have

$$f(1,\Delta) > 0. \tag{26}$$

We also have

$$\lim_{s\to\infty} f(s,\Delta) = 0. \tag{27}$$

Note that

$$f'(s,\Delta) = \frac{\partial f}{\partial s}(s,\Delta) = \frac{1}{s+\Delta/e}\sum_{i=0}^{\Delta}\frac{\Delta/e - i}{s+i}.$$

From $\Delta(\Delta+1)/e < \Delta(\Delta+1)/2$ it follows that for every Δ there exists s_Δ such that

$$f'(s,\Delta) < 0 \text{ for all } s > s_\Delta. \tag{28}$$

Let $g(s,\Delta,y) = \sum_{i=0}^{\Delta}\frac{y-i}{s+i}$. We have $g(s,\Delta,y) = 0$ iff

$$y = y_\Delta(s) = \left(\sum_{i=0}^{\Delta}\frac{i}{s+i}\right) / \left(\sum_{i=0}^{\Delta}\frac{1}{s+i}\right).$$

We will show that $y_\Delta(s)$ is an increasing function of s. This will imply that the equation $\Delta/e = y_\Delta(s)$ has at most one solution for any fixed Δ. Note that $f'(s,\Delta) = g(s,\Delta,\Delta/e)$. Hence we will obtain that $f'(s,\Delta) = 0$ has at most one solution for any fixed Δ. This together with (26), (27), (28) implies $f(s,\Delta) \geq 0$.

It remains to show

$$(\partial y_\Delta/\partial s)(s) > 0. \tag{29}$$

The sign of $(\partial y_\Delta/\partial s)(s)$ is the same as the sign of

$$h(s,\Delta) := \left(\sum_{i=0}^{\Delta}\frac{i}{s+i}\right)\left(\sum_{i=0}^{\Delta}\frac{1}{(s+i)^2}\right) - \left(\sum_{i=0}^{\Delta}\frac{1}{s+i}\right)\left(\sum_{i=0}^{\Delta}\frac{i}{(s+i)^2}\right).$$

For $\Delta = 0$ we have $h(s,\Delta) = 0$. To show (29) it is enough to show that for $\Delta \geq 1$ the following quantity is positive.

$$h'(s,\Delta) := h(s,\Delta) - h(s,\Delta-1) = \frac{1}{s+\Delta}\left(\sum_{i=0}^{\Delta}\frac{\Delta-i}{(s+i)^2}\right) + \frac{1}{(s+\Delta)^2}\left(\sum_{i=0}^{\Delta}\frac{i-\Delta}{s+i}\right).$$

For $\Delta = 0$ we have $(s+\Delta)^2 h'(s,\Delta) = 0$. To show $h'(s,\Delta) > 0$ for $\Delta \geq 1$ it is enough to show that for $\Delta \geq 1$ the following quantity is positive

$$h''(s,\Delta) := (s+\Delta)^2 h'(s,\Delta) - (s+\Delta-1)^2 h'(s,\Delta-1) = \sum_{i=0}^{\Delta-1}\frac{2\Delta-2i-1}{(s+i)^2}.$$

We have that $h''(s,\Delta)$ is a sum of positive numbers and hence $h''(s,\Delta) > 0$ for $\Delta \geq 1$. This implies $h'(s,\Delta) > 0$ for $\Delta > 0$ and this in turn implies $h(s,\Delta) > 0$ for $\Delta \geq 1$. We just proved (29) which was the only thing remaining to be proved. $\square$