ACCELERATING SIMULATED ANNEALING FOR THE PERMANENT AND COMBINATORIAL COUNTING PROBLEMS*

IVONA BEZÁKOVÁ†, DANIEL ŠTEFANKOVIČ‡, VIJAY V. VAZIRANI§, AND ERIC VIGODA§

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.

Key words. Markov chain Monte Carlo, simulated annealing, cooling schedule, approximate counting problems

AMS subject classifications. 68W20, 68W25, 68W40

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:

$$Z = \frac{Z_{\ell+1}}{Z_\ell} \frac{Z_\ell}{Z_{\ell-1}} \cdots \frac{Z_1}{Z_0} Z_0, \tag{1.1}$$

where $Z_{\ell+1} = Z$ 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 $Z = Z_{\ell+1}$ 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 is the algorithm of Jerrum, Sinclair and Vigoda [10] for approximating the permanent of a non-negative matrix. In their algorithm, the cooling schedule is uniform: the rate of cooling was constant.

---

*An extended abstract of this paper appeared in *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), 900-907, 2006.

†Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623. Email: ib@cs.rit.edu. I.B. was supported by NSF grant CCR-0237834.

‡Department of Computer Science, University of Rochester, Rochester, NY 14627. Email: stefanko@cs.rochester.edu.

§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-0455666.

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. [13], and is related to schedules used recently in geometric settings by Lovász and Vempala [14] and Kalai and Vempala [11].

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 [12], the permanent has recently been used in a variety of areas, e. g., computer vision [16], and statistics [15]. Jerrum, Sinclair, and Vigoda presented a simulated annealing algorithm [10] 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 relating sets of 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 1.1. *For all $\varepsilon > 0$, there exists a randomized algorithm to approximate, within a factor $(1 \pm \varepsilon)$, the permanent of a 0/1 $n \times n$ matrix A in time $O(n^7 \log^4(n) + n^6 \log^5(n)\varepsilon^{-2})$. The algorithm extends to arbitrary matrices with non-negative entries.*

The remainder of the paper is organized as follows. We introduce some basic machinery and definitions in the following section. In Section 3 we present our new cooling schedule, motivated by its application to colorings. We then focus on the permanent in Section 4. We begin by presenting the simulated annealing algorithm for the permanent in Section 4. In Section 5 we explain the background techniques for analyzing the Markov chain. We present our new inequalities in Section 6. Finally, in Section 7 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 Section 8.

2. Preliminaries.

2.1. Colorings and Potts model. 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 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 $\geq 1 - \delta$. This is commonly known as a fully-polynomial randomized approximation scheme (fpras) for counting colorings.

The colorings problem corresponds to the zero-temperature version of the anti-ferromagnetic Potts model. In addition to the underlying graph G and the number of colors 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 k-coloring. More precisely, for activity $\lambda \geq 0$, 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. For $\lambda = 0$ we also define $0^0 = 1$, and thus $Z(0) = |\Omega|$. In Section 3.1 we will consider simulated annealing algorithms for estimating the partition function of the Potts model.

An elementary component of the upcoming algorithms is the following approach for estimating the ratio of the partition function at a pair of temperatures. In particular, for a sequence $1 = \lambda_0 > \lambda_1 > \cdots > \lambda_\ell > \lambda_{\ell+1} = 0$ we will estimate the ratios $\alpha_i := Z(\lambda_{i+1})/Z(\lambda_i)$ for all $0 \leq i \leq \ell$. Assuming $1/2 \leq \alpha_i \leq 1$, we can approximate α_i efficiently 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:

$$\mathbf{E}\left(Y_i\right) = \mathbf{E}_{X_i \sim \pi_i}\left((\lambda_{i+1}/\lambda_i)^{M(X_i)}\right) = \sum_{\sigma \in [k]^V} \frac{(\lambda_{i+1})^{M(\sigma)}}{Z(\lambda_i)} = \frac{Z(\lambda_{i+1})}{Z(\lambda_i)} = \alpha_i. \tag{2.1}$$

The expected value of $Y = Y_0 Y_1 \ldots Y_\ell$ is

$$\mathbf{E}\left(Y\right) = \prod_{i=0}^{\ell} \mathbf{E}\left(Y_i\right) = \frac{Z(0)}{Z(1)},$$

where $Z(1)$ is easy to compute. Thus, our goal of estimating $Z(0) = |\Omega|$ can reduced to estimating $\mathbf{E}\left(Y\right)$.

Assume that we have an algorithm for generating labelings X_i from π_i. We draw $64(\ell+1)/\varepsilon^2$ samples of X_i and take the mean $\overline{Y}_i$ of their corresponding estimators Y_i. We have

$$\frac{\mathbf{Var}\left(\overline{Y}_i\right)}{\mathbf{E}\left(\overline{Y}_i\right)^2} = \frac{\varepsilon^2}{64(\ell+1)} \frac{\mathbf{Var}\left(Y_i\right)}{\mathbf{E}\left(Y_i\right)^2} \leq \frac{\varepsilon^2}{16(\ell+1)}.$$

Hence for $\overline{Y} = \overline{Y}_0 \overline{Y}_1 \ldots \overline{Y}_\ell$ we have

$$\frac{\mathbf{Var}\left(\overline{Y}\right)}{\mathbf{E}\left(\overline{Y}\right)^2} = \left(1 + \frac{\mathbf{Var}\left(\overline{Y}_0\right)}{\mathbf{E}\left(\overline{Y}_0\right)^2}\right) \cdots \left(1 + \frac{\mathbf{Var}\left(\overline{Y}_\ell\right)}{\mathbf{E}\left(\overline{Y}_\ell\right)^2}\right) - 1 \leq \mathrm{e}^{\varepsilon^2/16} - 1 \leq \varepsilon^2/8,$$

[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.

where in the last two inequalities we used $1+x \le \mathrm{e}^x$ (true for all x), and $\mathrm{e}^x - 1 \le 2x$ (true for $x \in [0,1]$). Now, by Chebyshev's inequality, with probability at least $7/8$ we have that the value of $\overline{Y}$ is in the interval $[(1-\varepsilon)\mathbf{E}(Y), (1+\varepsilon)\mathbf{E}(Y)]$.

Of course, we will not be able to obtain perfect samples from π_i. Assume now that we have X_i' which are from a distribution with a variation distance $\le \varepsilon^2/(512(\ell+1)^2)$ of π_i (we choose the variation distance to be 1/8-th of the reciprocal of the number of all samples). Let $\overline{Y}'$ be defined as $\overline{Y}$ above, but instead of X_i we will use X_i'. If we couple X_i with X_i' optimally, then with probability $\ge 7/8$ we have $\overline{Y} = \overline{Y}'$. Hence, $\overline{Y}'$ is in the interval $[(1-\varepsilon)\mathbf{E}(Y), (1+\varepsilon)\mathbf{E}(Y)]$ with probability $\ge 3/4$.

2.2. Markov chain basics. For a pair of distributions μ and π on a finite space Ω we will measure their distance using *variation distance*, defined to be

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

For an ergodic Markov chain with finite state space Ω, transition matrix P, and unique stationary distribution π, we are interested in the *mixing time*, defined to be

$$\tau(\delta) = \max_{x\in\Omega} \tau_x(\delta),$$

where

$$\tau_x(\delta) = \min\{t : d_{\mathrm{TV}}(P^t(x,\cdot),\pi) \le \delta\}.$$

In the case of the permanent, we will bound the mixing time by the canonical paths method. For some $S \subseteq \Omega$, for each $(I,F) \in \Omega\times S$, we will define a *canonical path* from I to F, denoted, $\gamma(I,F)$, which is of length $\le \ell$. The path is along transitions of the Markov chain (i. e., along pairs $(x,y) \in \Omega^2$ where $P(x,y) > 0$). We then bound the weighted sum of canonical paths (or "flow") through any transition. More precisely, for a transition $T = x \to y$, let

$$\rho(T) = \sum_{\substack{(I,F)\in\Omega\times\mathcal{P}:\\ T\in\gamma(I,F)}} \frac{\pi(I)\pi(F)}{\pi(x)P(x,y)},$$

denote the *congestion* through the transition T.

Let

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

Then (see [17, 4]), for any initial state $x \in \Omega$, the mixing time is bounded as

$$\tau_x(\delta) \le \frac{7\ell\rho}{\pi(S)}\left(\ln \pi(x)^{-1} + \ln \delta^{-1}\right).$$

The factor $1/\pi(S)$ comes from restricting to $F \in S$, see Lemma 9 in [10].

3. 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.

3.1. Basic Cooling Schedule for Counting Colorings. Our focus in this section is obtaining an fpras for counting all k-colorings of a given graph G. Let $\Omega = \Omega(G)$ denote the set of k-colorings of G. We are of course only considering cases where $|\Omega| \geq 1$. There are various situations where a polynomial-time algorithm for approximating $|\Omega|$ exists, e. g., see [5] for a survey, and [6] for a more recent result when $k = \Omega(\Delta/\log\Delta)$ for planar graphs. Our aim is to improve the running time of these approximate counting algorithms.

For the purposes of reducing the approximation of $|\Omega|$ to sampling from Ω, we will define a sequence of activities of the anti-ferromagnetic Potts model (defined in Section 2.1). We will express $|\Omega|$ as a telescoping product over instances of the Potts model where we slowly move from the original k-colorings (corresponding to activity $\lambda = 0$) to a trivial instance of the Potts model, namely $\lambda = 1$ since $Z(1) = k^n$. We specify a sequence of activities, 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.

The partition function $Z(\lambda)$ can be viewed as a polynomial in λ. Notice that its constant 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 11.2 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 $2|\Omega|$:

$$|\Omega| \leq Z(1/e^n) \leq |\Omega| + Z(1)(1/e^n) \leq |\Omega| + k^n/e^n \leq 2|\Omega|. \tag{3.1}$$

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

We will define a sequence, called a *cooling schedule*,

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

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

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

Notice that for $i = \ell$ the inequality follows from (3.1), so we need to take care of $i < \ell$. We estimate the number of k-colorings of G via the telescoping product:

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

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 as described in Section 2.1.

A straightforward cooling schedule sets $\lambda_{i+1} = 2^{-1/m}\lambda_i$. Then,

$$Z(\lambda_{i+1}) \geq (2^{-1/m})^m Z(\lambda_i) = Z(\lambda_i)/2,$$

as required. This specifies a uniform cooling schedule with a rate of decrease $2^{-1/m}$. Note, once $\lambda_\ell \leq k^{-n}$ we can set $\lambda_{\ell+1} = 0$ since we then have $Z(\lambda_\ell) \leq |\Omega| + 1 \leq$

2Ω assuming $|\Omega| \geq 1$. Therefore, this uniform cooling schedule is of length $\ell = O(nm \log k)$. We present a new cooling schedule which is of length only $O(n \log n)$.

Our goal is to obtain a general cooling schedule which applies to all instances of the colorings problem and will also apply to many other combinatorial problems. If we restrict attention to certain regions of k versus Δ, sometimes a straightforward telescoping product is more efficient for colorings. In particular, assuming $k \geq (1+\varepsilon)\Delta$, where $\varepsilon > 0$ is a constant, then the removal of all at most Δ edges adjacent to one of the vertices increases the number of colorings by a factor at most $k/(k-\Delta) \leq (1+\varepsilon)/\varepsilon$. Hence, in this case one can obtain a schedule of length $O(n)$, but such a schedule does not apply, for example, to the previously mentioned results of [6] which hold for $k = \Omega(\Delta/\log \Delta)$ for planar graphs.

3.2. New Cooling Schedule. Note if we had $Z(\lambda) = k^n \lambda^m$ we could not decrease λ faster than $2^{-1/m}$. Fortunately, in our case the constant term of $Z(\lambda)$ is at least one. To illustrate the idea of non-uniform decrease, let $f_i(\lambda) = \lambda^i$. As we decrease λ, the polynomial f_m will always decrease faster (in a relative sense) 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 constant 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}, $\dots$, 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.

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 $1 = \lambda_1 > \lambda_2 > \cdots > \lambda_\ell > \lambda_{\ell+1} = 0$ 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 $Z(0) \geq 1$ (e. g., $Z(\lambda)$ has positive constant term). In fact, when $k > \Delta$, we have $Z(0) \geq (k/e)^n$, which will save a factor of $O(\log k)$ in the final result. For completeness, we prove this lower bound in the Appendix in Corollary 11.2 of Section 11.

For some applications it will not be possible to make the constant term positive, 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 $\lambda_0 = 1$, $i = s$ and $j = 0$.
While $\lambda_j > 1/\gamma$ do
  Set $\lambda_{j+1} = c^{-1/i}\lambda_j$.
  If $i > D + 1$ and $\lambda_{j+1} < (s/\gamma)^{1/(i-D)}$,
    Set $\lambda_{j+1} = (s/\gamma)^{1/(i-D)}$ and decrement $i = i - 1$.
  Increment $j = j + 1$.
Set $\ell = j$.

The following lemmas prove that the above algorithm produces a short schedule. We prove the lemmas in Section 3.3. The first lemma bounds the number of intermediate temperatures in the above cooling schedule, i. e., the length ℓ of the λ sequence.

LEMMA 3.1. *Let $c, \gamma > 0, D \geq 0$ and let $\lambda_0, \dots, \lambda_\ell$ be the sequence computed by the above algorithm. Then $\ell = O([(D+1)\log(s-D) + s/(s-D)]\log_c(s\gamma))$. If c and D are constants independent of s, then $\ell = O((\log s)\log(s\gamma))$.*

Note that for colorings $\ell = O(n(\log n)\log k)$ (assuming $Z(0) \geq 1$), and when $k > \Delta$ we have $\ell = O(n \log n)$ since $Z(0) \geq (k/e)^n$.

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}. For the later application to the permanent, we will need to simultaneously consider a collection of polynomials. Therefore, we state the following lemma in this more general context.

LEMMA 3.2. *Let $c, \gamma, D \geq 0$ and let $Z_1, \dots, Z_q$ 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 $\lambda_0, \lambda_1, \dots, \lambda_\ell$ be the sequence constructed by the above algorithm. Then

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

Recall from Section 2.1, that to estimate $\prod_{i=0}^{\ell} \alpha_i$ within a factor $(1 \pm \varepsilon)$ with probability $\geq 3/4$ we need to generate $O(\ell/\varepsilon^2)$ samples from within variation distance $O(\varepsilon^2/\ell^2)$ of π_i, for all $i = 0, \dots, \ell$. To illustrate the application of the shorter cooling schedule, recall that for colorings, when $k > \Delta$ we have $\ell = O(n \log n)$. Hence, we need a total of $O(n^2\varepsilon^{-2}\log^2 n)$ samples. For $k > 2\Delta$, for all activities $1 \leq \lambda \leq 0$, there is a Markov chain with mixing time $T(\varepsilon) = \frac{k}{k-2\Delta} n \log(n/\varepsilon)$ [3, 8]. Consequently, we 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.

For $k \leq 2\Delta$ there are a variety of results showing fast convergence of Markov chains for generating a random k-coloring [5]. These results are proved for k-colorings, but most likely they can be extended to the non-zero temperature. One particular example is the previously mentioned result of [6] which for planar graphs shows $O^*(n)$ mixing time of a Markov chain when $k = \Omega(\Delta/\log \Delta)$ for all activities. (The $O^*()$ notation hides logarithmic factors and the dependence on ϵ.) Consequently, in this case we again obtain an $O^*(n^3)$ time algorithm for approximating the number of k-colorings.

3.3. Proof of Lemmas 3.1 and 3.2. The rest of this section is devoted to the proof of Lemmas 3.1 and 3.2.

Proof. [Proof of Lemma 3.1] We define intervals:

$$I_i = \begin{cases} [(s/\gamma)^{1/(s-D)}, \infty) & \text{for } i = s, \\ [(s/\gamma)^{1/(i-D)}, (s/\gamma)^{1/(i+1-D)}] & \text{for } D+1 < i < s, \\ (0, (s/\gamma)^{1/2}] & \text{for } i = D+1. \end{cases}$$

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

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

Similarly,

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

and

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

Recall that $s \geq 1$. Putting it all together, we get the bound

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

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

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

□

We now present a few preliminary lemmas before proving Lemma 3.2. The *log-derivative* of a function f is $(\log f)' = f'/f$. The log-derivative measures how quickly a function increases.

Definition 3.3. *We say that a polynomial f is* dominant *over a polynomial g on an interval I if $f'(x)/f(x) \geq g'(x)/g(x)$ for every $x \in I$.*

Lemma 3.4. *Let $f, g : I \to \mathbf{R}^+$ be two non-decreasing polynomials. If f dominates over g on I, then $f(y)/f(x) \geq g(y)/g(x)$ for every $x, y \in I$, $x \leq y$.*

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

Lemma 3.5. *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 $(x^s)'/x^s \geq g'(x)/g(x)$ for every $x > 0$. □

Lemma 3.6. *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 $a_0, a_1, \dots, a_D$ is ≥ 1. Then for any $i \geq D+1$ the polynomial x^i dominates over 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

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

□

Proof. [Proof of Lemma 3.2] Let $I_{D+1}, \ldots, I_s$ be as in the proof of Lemma 3.1. Let $Q_q(\lambda) = \gamma Z_q(\lambda)/Z_q(1)$. Notice that the Q_q satisfy the conditions required of g by Lemma 3.6. Therefore, x^i dominates over every Q_q (and hence also Z_q) on the interval I_i for $i < s$. Moreover, Lemma 3.6 and Lemma 3.4 imply that x^s dominates over every Q_q (and hence Z_q) on the interval I_s. Notice that if $\lambda_j, \lambda_{j+1} \in I_i$, then $c\lambda_{j+1}^i \geq \lambda_j^i$ (where inequality happens only if $\lambda_{j+1} = (s/\gamma)^{1/(i-D)}$). Therefore, all of the Z_q-polynomials decrease by a factor at most c between consecutive values of λ. □

4. Permanent Algorithm. Here we describe the simulated annealing algorithm for the permanent. For simplicity we consider the case of a 0/1 matrix A. The generalization to non-negative matrices proceeds as in [10]. We assume $per(A) > 0$, i. e., there is at least one perfect matching.

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 in Section 6.

4.1. 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. Once non-edges have activity $\leq 1/n!$ we have that the total activity of perfect matchings containing at least one non-edge is < 1, and hence they alter the permanent by at most a factor 2. The activities allow us to work on the complete bipartite graph between 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:

$$w^*(u,v) := \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}. \tag{4.1}$$

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

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

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

4.2. Markov chain definition. At the heart of the algorithm lies a Markov chain MC, which was used in [10], and a slight variant was used in [2, 9]. 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 $M' = M \cup (u,x) \setminus (w,x)$.
   - (c) If $x \in V_1$ and $(x,z) \in M_t$, let $M' = M \cup (x,v) \setminus (x,z)$.
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. Case 3 applies the Metropolis filter which ensures that the stationary distribution of the Markov chain is proportional to w.

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 4.1. *Assuming the weight function w satisfies inequality*

$$w^*(u,v)/2 \le w(u,v) \le 2w^*(u,v) \tag{4.2}$$

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

This theorem improves the mixing time bound by $O(n^2)$ over the corresponding result in [10]. The theorem will be proved in Section 7.

4.3. 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

$$w^*(u,v) = \frac{\pi(\mathcal{P})}{\pi(\mathcal{N}(u,v))}w(u,v). \tag{4.3}$$

Given weights w which are a rough approximation to w^*, identity (4.3) 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 (4.3)) 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, $S = O(n^2 \log(1/\hat{\eta}))$ 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 (4.3) we can also approximate w^* within a factor $\sqrt{2}$ with the same bounds.

Theorem 4.1 (with $\delta = \Theta(1/n^2)$) implies that $T = O(n^4 \log n)$ time is needed to generate each sample (we will choose $\hat{\eta}$ so that the failure probability of the entire algorithm is small, e. g., $\hat{\eta} = \Theta(1/n^4)$ suffices). To be precise this requires the use of "warm start" samples in which the initial matching for the Markov chain simulation is a reasonable approximation to the stationary distribution. In particular, after the initial sample from (close to) the stationary distribution, the initial matching for each simulation is the final matching from the previous simulation. (The application of warm starts in our work is identical to their use in [10], hence we refer the interested reader to [10] for further details.)

4.4. 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 4.5 to approximate the permanent of a 0-1 matrix. The algorithm can be generalized to compute the permanent of general non-negative matrices, see Section 9.

The algorithm runs in phases, each characterized by a parameter λ. In every phase,

$$\lambda(e) := \begin{cases} 1 & \text{for } e \in E, \\ \lambda & \text{for } e \notin E. \end{cases} \tag{4.4}$$

We start with $\lambda = 1$ and slowly decrease λ 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 4.1 we generate many samples from the stationary distribution. Using these samples and (4.3), we refine the weights to within a factor $\sqrt{2}$ of the ideal weights:

$$\frac{w^*(u,v)}{\sqrt{2}} \leq w(u,v) \leq \sqrt{2}w^*(u,v) \tag{4.5}$$

This allows us to decrease λ so that the current estimates of the ideal weights for λ_i are within a factor 2 of the ideal weights for λ_{i+1}.

In [10], $O(n^2 \log n)$ phases are required. A straightforward way to achieve this is to decrease λ by a factor $2^{-1/n}$ between phases as considered in Section 3.1 for colorings.

We use only $\ell = O(n \log^2 n)$ phases by progressively decreasing λ by a larger amount per phase. Initially we decrease λ by a factor $2^{-1/n}$ per phase, but during the final phases we decrease λ 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 $\geq 1-\eta$. Recall from the last paragraphs of the previous section that $S = O(n^2(\log n + \log \eta^{-1}))$ since $\eta = O(\ell\hat{\eta})$, and $T = O(n^4 \log n)$.

Algorithm for approximating ideal weights of 0-1 matrices:

Initialize $\lambda = 1$ and $i = n$.
Initialize $w(u,v) \leftarrow n$ for all $(u,v) \in V_1 \times V_2$.
While $\lambda > 1/n!$ do:
  Take S samples from MC with parameters λ, w, using a warm start simulation (in particular, initial 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 (4.5), for all u, v. The algorithm fails (i. e., (4.5) is not satisfied) with small probability.
  Set $\lambda = 2^{-1/(2i)}\lambda$.
  If $i > 2$ and $\lambda < (n-1)!^{-1/(i-1)}$,
    Set $\lambda = (n-1)!^{-1/(i-1)}$ and decrement $i = i - 1$.
  If $\lambda < 1/n!$, set $\lambda = 1/n!$.
  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 3.1, 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 8). In Section 8 we show that Lemma 3.2 implies that our weights at the start of each phase satisfy (4.2) assuming that the estimates w' satisfied condition (4.5) throughout the execution of the algorithm. Therefore, the total running time is $O(STn \log^2 n) = O(n^7 \log^4 n)$.

4.5. Reduction from Counting to Sampling. Let $\lambda_0 = 1 > \lambda_1 > \cdots > \lambda_\ell = 1/n!$, $\ell = O(n \log^2 n)$, be the sequence of λ used in the weight-estimating algorithm from the previous section. Assume that the algorithm did not fail, i. e., the hole weights $w_0, \ldots, w_\ell$ computed by the algorithm are within a factor of $\sqrt{2}$ from the ideal weights $w_0^*, \ldots, w_\ell^*$.

It remains to use these (constant factor) estimates of the ideal weights to obtain a $(1 \pm \varepsilon)$-approximation of the permanent. This is done by expressing the permanent as a telescoping product as was done for colorings in Section 3.1. We refer the interested

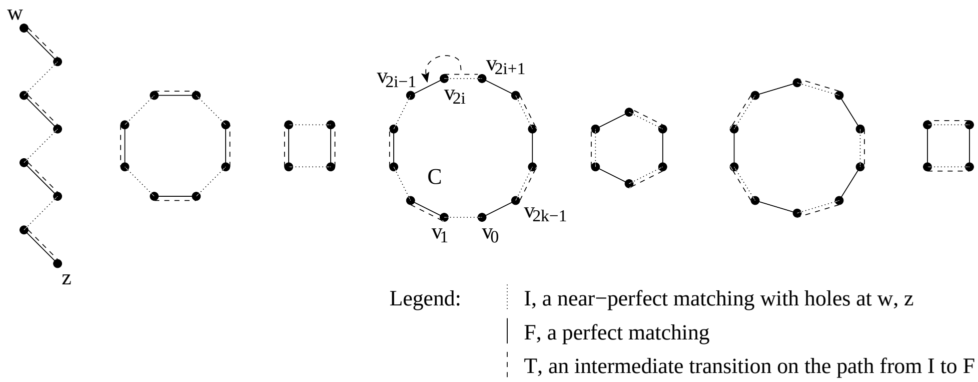

FIG. 5.1. *This figure illustrates a near-perfect matching I and a perfect matching F, along with a transition T on the canonical path from I to F. The transition is "sliding" an edge in the cycle denoted as C. The components of $I \oplus F$ are shown in increasing order (from left to right). The alternating path and the cycles to the left of C have been already "corrected," whereas the cycles on the right still need to be "corrected." The unfinished cycle is partially corrected: from v_0 to v_{2i-1} the cycle is the same as F, whereas from v_{2i+1} to v_{2k-1} the cycle is the same as I. (A similar picture arises on the canonical path from a perfect matching to a perfect matching, except in that case there is no alternating path.)*

reader to Section 5 from [10] for details of the argument. The only difference from [10] is that the number of intermediate temperatures is $\ell = O(n \log^2 n)$ as opposed to $O(n^2 \log n)$. The total running time of this part of the algorithm is $O(\ell^2/\varepsilon^2 n^4 \log n) = O(n^6 \log^5 n\varepsilon^{-2})$. This completes the description of the algorithm for $0/1$ matrices.

5. Canonical Paths for Proving Theorem 4.1. Recall the canonical paths method from Section 2.2. We will use this approach with $S = \mathcal{P}$. To prove Theorem 4.1 we need to define canonical paths $\gamma(I, F)$ for all initial $I \in \Omega$ and final $F \in \mathcal{P}$. These paths will have length $\ell \leq n$, and hence we need to show that the congestion satisfies $\rho(T) = O(n)$ for every transition T. The canonical paths we use are identical to those considered in [10] (and the earlier work of [9]).

We define the canonical paths now, and defer the bound on the congestion to Section 7, after presenting some combinatorial lemmas in Section 6. We will assume that the vertices of G are numbered. If $I \in \mathcal{P}$, then $I \oplus F$ consists of even length cycles, where $\oplus$ denotes the symmetric difference. 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 $v_0, v_1, \ldots, v_{2k-1}$ 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}, v_{2i+1}) for edge (v_{2i-1}, v_{2i}) for $1 \leq i \leq k-1$. Finally it adds edge (v_{2k-1}, v_0) 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. Figure 5.1 shows an intermediate transition on the canonical path from I to F.

This completes the definition of the canonical paths and it remains to bound their associated congestion. To this end, in the following section we present several inequalities which are used to improve the analysis of the congestion.

6. Key Technical Lemmas. The following lemma contains the new combinatorial inequalities which are the key to our improvement of $O(n^2)$ in Theorem 4.1.

These inequalities will be used to bound the total weight of (I, F) pairs whose canonical path passes through a specified transition. In [10] 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 6.2 below improves on Lemma 7 in [10] by constructing more efficient mappings. We first present our mappings in the simpler setting of Lemma 6.1 and later use them to prove Lemma 6.2. Using these new inequalities to bound the congestion requires more work than the analysis of [10].

LEMMA 6.1. *Let $u, w \in V_1$, $v, z \in V_2$ be distinct vertices. Then,*

1.

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

2.

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

3.

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

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 $M \cup M' \cup (u,y) \cup (x,v)$ 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 (x,v). Hence, given the pair of perfect matchings, we can deduce the original unmatched vertices (by guessing which of the two edges is incident to u), and thereby reconstruct M and M'. This outlines the approach for proving Lemma 6.1.

Proof.

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

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

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

Let $L_0 \in \mathcal{N}(u,v)$ and $L_1 \in \cup_{x,y:(u,y),(x,v)\in E}\mathcal{N}(x,y)$. 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 length, as seen in Figure 6.1(a).
- If the paths are u to v and x to y, they must both be odd length as seen in Figure 6.1(b).
- 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.

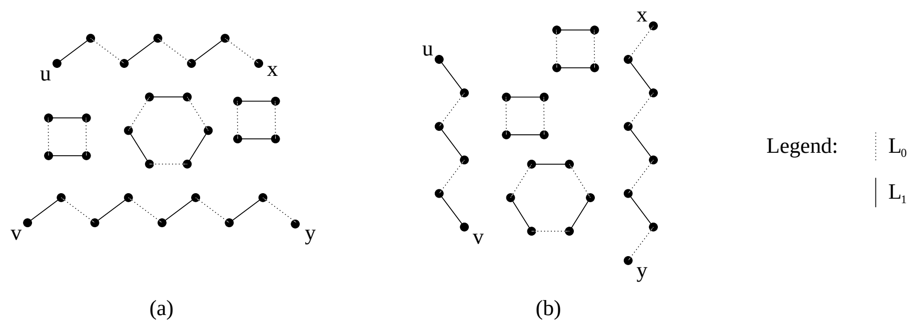

FIG. 6.1. *Proof of Lemma 6.1, part 1, two different possibilities for* $L_0 \oplus L_1$.

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 (x, v). 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:

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

Let $L_0 \in \mathcal{N}(u, v)$ and $L_1 \in \cup_{x:(x,v)\in E}\mathcal{N}(x, z)$. As before, u, v, x, z 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, $L_0 \cup L_1 \cup \{(x, v)\}$ contains an odd length path from u to z, say P. Construct $M_0 \in \mathcal{N}(u, z)$ by including all edges of L_0 not on P and alternate edges of P, leaving u, z unmatched. Let $M_1 \in \mathcal{P}$ consist of the remaining edges of $L_0 \cup L_1 \cup \{(x, v)\}$. Let $b = 0$ if $(v, x) \in M_0$, and $b = 1$ 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:

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

Let $L_0 \in \mathcal{N}(u, v)$ and $L_1 \in \cup_{x,y:(u,y),(x,v)\in E}\mathcal{N}(x, y, w, z)$. 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 case 1. Otherwise, by parity considerations the only possibilities are:

- u to w and v to y are even length paths and x to z is an odd length path;
- u to x and v to z are even length paths and w to y is an odd length path;
- u to w and v to z are even length paths and x to y is an odd length path;
- u to v, x to z, and w to y are odd length paths.

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

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

Lemma 6.2. *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 the proof of Lemma 6.1. 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 $(u,y),(x,v)$, 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

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

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

7. Bounding Congestion: Proof of Theorem 4.1. We bound the congestion separately for transitions which move between near-perfect matchings (Cases 2b and 2c in the definition of chain MC in Section 4.2), and transitions which move between a perfect and near-perfect matching (Case 1). Our goal for this section will be to prove for every transition $T = M \to M'$,

$$\sum_{\substack{(I,F)\in\Omega\times\mathcal{P}:\\ T\in\gamma(I,F)}} \frac{w^*(I)w^*(F)}{w^*(M)} = O(w^*(\Omega)). \tag{7.1}$$

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

For a transition $T = M \to M'$, 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,

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

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 7.1. *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 $M' = M \setminus (x, v') \cup (x, v)$ 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 $z \in V_2$,

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

3. For all $w \in V_1, w \neq u$ and $z \in V_2, z \neq v, v'$,

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

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

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

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 5. 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. Figure 7.1 shows the definition of M''. 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.

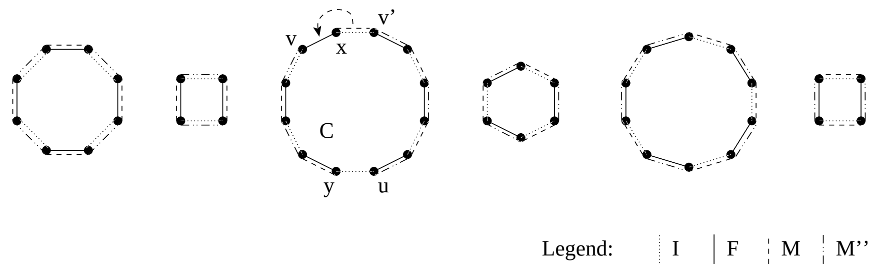

FIG. 7.1. *Proof of Lemma 7.1, part 1.*

2. For all $z \in V_2$, we will again 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, x, v 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 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 $I \oplus F$. Therefore, M has already "corrected" P and so it 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. □

We now prove (7.1) for the first type of transitions. The proof applies Lemma 7.1 and then Lemma 6.2. We break the statement of (7.1) into two cases depending on whether I is a perfect matching or a near-perfect matching.

LEMMA 7.2. *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*

$$\sum_{(I,F)\in cp_T} \frac{w^*(I)w^*(F)}{w^*(M)} \leq \frac{2w^*(\Omega)}{n^2}. \tag{7.2}$$

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

$$\sum_{w\in V_1, z\in V_2} \sum_{(I,F)\in cp_T^{w,z}} \frac{w^*(I)w^*(F)}{w^*(M)} \leq 3w^*(\Omega). \tag{7.3}$$

Proof. The transition T is sliding an edge, let x denote the pivot vertex, let $M \in \mathcal{N}(u, v)$ and $M' \in \mathcal{N}(u, v')$ where $u \in V_1, v, v' \in V_2$. Thus, $M' = M \setminus (v', x) \cup (x, v)$ for some $x \in V_1$.

We begin with the proof of (7.2).

$$\begin{aligned}
&\sum_{(I,F)\in cp_T} \frac{w^*(I)w^*(F)}{w^*(M)} \\
&= \sum_{(I,F)\in cp_T} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{P})} \\
&\leq \sum_{y\in V_2} \frac{\lambda(\mathcal{N}(x,y))\lambda(u,y)\lambda(x,v)\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{P})} \qquad \text{by Lemma 7.1} \\
&\leq 2\lambda(\mathcal{P}) \qquad \text{by Lemma 6.2} \\
&= \frac{2w^*(\Omega)}{n^2+1}
\end{aligned}$$

Note the application of Lemma 6.2 only uses the summation over y and does not require the summation over x. We have now completed the proof of (7.2). We now prove (7.3) 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.

$$\begin{aligned}
&\sum_{z\in V_2} \sum_{(I,F)\in cp_T^{u,z}} \frac{w^*(I)w^*(F)}{w^*(M)} \\
&= \sum_{z\in V_2} \sum_{(I,F)\in cp_T^{u,z}} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{N}(u,z))} \\
&\leq \sum_{z\in V_2} \frac{\lambda(\mathcal{N}(x,z))\lambda(x,v)\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(u,z))} \qquad \text{by Lemma 7.1} \\
&\leq \sum_{z\in V_2} 2\lambda(\mathcal{P}) \qquad \text{by Lemma 6.2} \\
&= \frac{2n}{n^2+1} w^*(\Omega) \\
&\leq w^*(\Omega)
\end{aligned}$$

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

$$\begin{aligned}
&\sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}} \sum_{(I,F)\in cp_T^{w,z}} \frac{w^*(I)w^*(F)}{w^*(M)} \\
&= \sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}} \sum_{(I,F)\in cp_T^{w,z}} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{N}(w,z))} \\
&\le \sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}} \sum_{y\in V_2} \frac{\lambda(\mathcal{N}(w,z,x,y))\lambda(u,y)\lambda(x,v)\lambda(\mathcal{N}(u,v))}{\lambda(\mathcal{N}(w,z))} \qquad \text{by Lemma 7.1} \\
&\le \sum_{\substack{w\in V_1, z\in V_2:\\ w\neq u}} 2\lambda(\mathcal{P}) \qquad \text{by Lemma 6.2} \\
&\le 2w^*(\Omega).
\end{aligned}$$

□

We now follow the same approach as Lemmas 7.1 and 7.2 to prove (7.1) for transitions moving between a perfect and near-perfect matching. The proofs in this case are easier.

Lemma 7.3. *For a transition $T = M \to M'$ which removes an edge (i. e., Case 1) or adds an edge (i. e., Case 2a), let (u,v) be the removed/added edge, and let N be the near-perfect matching from the pair M, M' (i. e., if adding an edge $N = M$, and if removing an edge $N = M'$). Then,*

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

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

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

Proof. Let P denote the perfect matching from the pair M, 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 $\mathcal{P}$. □

Lemma 7.4. *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*

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

and

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

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 7.2, using Lemma 7.3.

Observe that for any x, y,

$$\lambda(x,y)\lambda(\mathcal{N}(x,y)) \le \lambda(\mathcal{P}) \tag{7.4}$$

We begin with the proof of (7.4).

$$\begin{aligned}\sum_{w,z}\sum_{(I,F)\in cp_T^{w,z}} \frac{w^*(I)w^*(F)}{w^*(M)} &= \sum_{w,z}\sum_{(I,F)\in cp_T^{w,z}} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{N}(w,z))} \\ &\le \sum_{w,z}\lambda(u,v)\lambda(\mathcal{N}(u,v)) \qquad \text{by Lemma 7.3} \\ &\le w^*(\Omega) \qquad \text{by (7.4)}\end{aligned}$$

We now prove (7.4).

$$\begin{aligned}\sum_{(I,F)\in cp_T} \frac{w^*(I)w^*(F)}{w^*(M)} &= \sum_{(I,F)\in cp_T} \lambda(I)\lambda(F)\frac{\lambda(\mathcal{N}(u,v))}{\lambda(M)\lambda(\mathcal{P})} \\ &\le 2\lambda(u,v)\lambda(\mathcal{N}(u,v)) \qquad \text{by Lemma 7.3} \\ &\le \lambda(\mathcal{P}) \qquad \text{by (7.4)}\end{aligned}$$

□

We now recall Theorem 4.1 and then present its proof.

THEOREM 4.1. *Assuming the weight function w satisfies inequality*

$$w^*(u,v)/2 \le w(u,v) \le 2w^*(u,v) \tag{4.2}$$

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

Proof. [Proof of Theorem 4.1] Inequality (4.2) 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 4.1 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 (4.2) 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 Case 3 in the definition of the Markov chain MC (Section 4.2) where the Metropolis filter is applied.

In particular, 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 $T = M \to M'$,

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

The chain is reversible, thus for every transition $T = M \to M'$, there is a reverse transition $T' = M' \to M$. Hence, to prove Theorem 4.1, it suffices to prove that for every transition $T = M \to M'$,

$$\sum_{\substack{(I,F)\in\Omega\times\mathcal{P}:\\ T\in\gamma(I,F)}} \frac{w^*(I)w^*(F)}{w^*(M)} = O(w^*(\Omega)). \tag{7.5}$$

Lemmas 7.2 and 7.4 imply (7.5) which completes the proof of the Theorem. □

8. Phases in the Permanent Algorithm. In this section we show that the choice of λ from the weight-estimating algorithm ensures that (4.2) is satisfied in each phase. Recall that we can obtain a refined estimate of the ideal weights in each phase, see (4.5). 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 (4.5) we have (4.2) 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 λ. This observation will allow us to use Lemma 3.2.

DEFINITION 8.1. *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 (4.1), for activities given by (4.4) can be expressed as follows

$$w^*_\lambda(u,v) = \frac{R_G(\lambda)}{R_{G\setminus\{u,v\}}(\lambda)}. \tag{8.1}$$

First we observe that every R-polynomial has a positive low-degree coefficient (and consequently in the application of Lemma 3.2 we will have that D is small). 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 constant term in R_G is positive. Similarly, if $(u,v) \in M$, then the constant term 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 3.2. Let $c = \sqrt{2}$, $\gamma = n!$, $D = 1$, $s = n$, $Z_1 = R_G$ and the polynomials $Z_2, \ldots, Z_{n^2+1}$ are the $R_{G\setminus\{u,v\}}$ polynomials for $u \in V_1, v \in V_2$. Let $\lambda_0, \ldots, \lambda_\ell$ be the sequence obtained from the algorithm in Section 3.2.

Notice, that we obtain the same sequence in the algorithm for estimating weights of the permanent. Then

$$\begin{array}{ll} R_G(\lambda_k) \geq R_G(\lambda_{k+1}) \geq R_G(\lambda_k)/\sqrt{2}, & \text{and} \\ R_{G\setminus\{u,v\}}(\lambda_k) \geq R_{G\setminus\{u,v\}}(\lambda_{k+1}) \geq R_{G\setminus\{u,v\}}(\lambda_k)/\sqrt{2} & \text{for every } u,v. \end{array} \tag{8.2}$$

Equations (8.1) and (8.2) imply the $w^*_{\lambda_k}$ and $w^*_{\lambda_{k+1}}$ are within a $\sqrt{2}$ factor. Moreover, if the weight-estimating algorithm does not fail, i. e., the w_{λ_k} satisfy (4.5), then w_{λ_k} also satisfy (4.2), as required by Theorem 4.1.

9. Non-negative Matrices. The extension to non-negative matrices follows identically as in Section 7 of [10], hence we refer the interested reader to [10].

10. Discussion.

10.1. Recent Improvements. In this paper, we have presented a near optimal cooling schedule subject to the constraint that each of the ratios Z_i/Z_{i-1} in (1.1) is bounded. However, in order to estimate the ratio efficiently, it suffices to have an unbiased estimator with bounded variance and the ratio itself might be large. A recent paper [18] presents a general cooling schedule achieving the bounded variance property. As a consequence, for many combinatorial problems, such as colorings or matchings, [18] achieves a cooling schedule of length $O^*(\sqrt{n})$, whereas in this paper we present a schedule of length $O^*(n)$. Therefore, the improved schedule of [18] reduces (compared with this paper) the overall running time by a factor $O^*(n)$ for many combinatorial counting problems. For the permanent, the improved cooling schedule of [18] does not appear to apply, since the algorithm for approximating the permanent needs to consider multiple polynomials simultaneously.

10.2. Permanent Application. 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 the 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 6.2.

Another direction is reducing 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. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), 414-423, 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), 50-58, 1986. Erratum in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 551, 1988.

[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), 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] A. Frieze and E. Vigoda, A Survey on the use of Markov Chains to Randomly Sample Colorings. *Combinatorics, Complexity and Chance*, eds. Grimmett and McDiarmid, Oxford University Press, 2007.

[6] T. Hayes, J. Vera, and E. Vigoda, Randomly coloring planar graphs with fewer colors than the maximum degree. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing* (STOC), 450-458, 2007.

[7] T. Hayes and 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.

[8] 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.

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

[10] M. 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.

[11] A. Kalai and S. Vempala, Simulated Annealing for Convex Optimization, *Mathematics of Operations Research*, 31(2):253-266, 2006.

[12] 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.

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

[14] 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.

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

[16] 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.

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

[18] D. Štefankovič, S. Vempala, and E. Vigoda, Adaptive Simulated Annealing: A Near-optimal Connection between Sampling and Counting. To appear in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science* (FOCS), 2007. Preprint available from arXiv at http://arxiv.org/abs/cs.DS/0612058

[19] 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 11.1. *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

$$|\Omega| \geq (\ell+s)\left(\prod_{j=0}^{n} c_j^{d_j}\right)\frac{1}{c_\ell}\left(\prod_{i=1}^{\ell}\frac{c_{j_i-1}}{c_{j_i}}\right) \geq (\ell+s)\left(\prod_{j=0}^{n} c_j^{d_j}\right)\frac{1}{c_\ell}\left(\frac{c_{\ell-1}}{c_\ell}\right)^{\ell} = \prod_{j=0}^{n} c_j^{d_j}, \tag{11.1}$$

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

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

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

$$T \leq \frac{(s+j)^{\binom{j+2}{2}}}{(s+j+1)^{\binom{j+1}{2}}}. \tag{11.2}$$

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 (11.2). Therefore, $c_j^2 \geq c_{j+1}c_{j-1}$ and hence the induction step (11.1) is proved. □

For k-colorings we obtain the following result.

Corollary 11.2. *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*

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

Proof. Let $s = k - \Delta$. The first inequality follows from Lemma 11.1 with the $L_i = [k]$.

The second inequality is equivalent to

$$(s+\Delta)^{\underline{\Delta+1}} \geq (s+\Delta/e)^{\Delta+1}. \tag{11.3}$$

The inequality (11.3) is true for $\Delta = 0$ and hence from now on we assume $\Delta \geq 1$.

Let $f(s,\Delta) = \sum_{i=0}^{\Delta} \ln\frac{s+i}{s+\Delta/e}$. Inequality (11.3) is equivalent to $f(s,\Delta) \geq 0$.

Claim:

$$f(1,\Delta) > 0 \tag{11.4}$$

Proof. [Proof of the claim] We need to show that (11.3) holds with strict inequality for $s = 1$. Let $n = \Delta + 1$. 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 (11.3) for $s = 1$ and

$n \geq 5$. For $n \leq 4$ and $s = 1$ the (strict version of) inequality (11.3) is easily verified by hand. □

Each term in the definition of f goes to zero as s goes to infinity. Hence we have

$$\lim_{s\to\infty} f(s,\Delta) = 0. \tag{11.5}$$

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

$$f'(s,\Delta) < 0 \text{ for all } s > s_\Delta. \tag{11.6}$$

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

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

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 (11.4), (11.5), (11.6) implies $f(s,\Delta) \geq 0$.

It remains to show

$$(\partial y_\Delta/\partial s)(s) > 0. \tag{11.7}$$

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 (11.7) 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 (11.7) which was the only thing remaining to be proved. □