Journal of Computational Physics **149,** 128–147 (1999)
Article ID jcph.1998.6149, available online at http://www.idealibrary.com on IDEAL®

Approximating the Permanent via Importance Sampling with Application to the Dimer Covering Problem

Isabel Beichl* and Francis Sullivan†

**NIST, Gaithersburg, Maryland 20899; †IDA/Center for Computing Sciences, Bowie, Maryland 20715*
E-mail: isabel@cam.nist.gov, fran@super.org

Received April 16, 1998; revised November 3, 1998

We estimate the asymptotic growth rate of the number of dimer covers of a cubic lattice. Our estimate, $\lambda_3 = 0.4466 \pm 0.0006$ is consistent with the lower bounds obtained by Hammersley and (later) Schrijver and the more recent improved upper bound obtained by Ciucu. Obtaining this estimate is an important step toward approximating the partition function of the cubic monomer–dimer system. From the partition function, all of the standard thermodynamic quantities can be evaluated. It is well known that computing λ_3 is equivalent to computing the permanent of a certain 0–1 matrix. We describe an extremely efficient Monte Carlo algorithm for approximating the permanent. Previous work on Monte Carlo approaches includes the pioneering results of Jerrum and Sinclair, who use a rapidly mixing random walk. Our method is inspired by results of Soules on convergence of Sinkhorn balancing to obtain a maximum entropy, doubly stochastic matrix. We use the Sinkhorn balanced matrix to generate an importance function that allows us to do direct random sampling, rather than a random walk that converges to a limiting distribution.

1. INTRODUCTION

We use a new method for approximating the permanent of a 0–1 matrix, based on the Monte Carlo technique of importance sampling to solve the dimer covering problem in two and three dimensions, that is, to estimate the asymptotic behavior of the number of dimer coverings of a regular rectangular lattice. This is an important step toward approximating the partition function of the cubic monomer–dimer system. From the partition function, all of the standard thermodynamic quantities can be obtained [7]. We use an excellent "importance function" that is readily available and fairly easy to compute. The importance function is obtained from doubly stochastic matrices that are the result of applying Sinkhorn

0021-9991/99 $30.00

balancing to a particular 0–1 matrix and to some of its minors. It is known that Sinkhorn balancing converges quickly for a completely supported matrix [26, 17] and, as we shall see, it is also easy to ensure that all matrices encountered are, in fact, completely supported. (All terminology is defined in Sections 2 and 3.)

The dimer covering problem is a special case of the more general monomer–dimer problem, where one wishes to count the number of ways of covering a lattice with both monomers and dimers. An extension of our method applies to the more general monomer–dimer setting, but the computations are much more elaborate. We hope to report on this in a later paper. For the present paper, our specific goal is to estimate the asymptotic value,

$$\lambda_d = \lim_{m\to\infty} \frac{\log(\mathrm{perm}(A_d(m)))}{m^d},$$

where d is the dimension (d is 2 or 3 in our case), and $\mathrm{perm}(A_d(m))$ is the permanent of a particular $(m^d/2) \times (m^d/2)$ matrix, to be described shortly. All logs in this paper are to the base e. Our calculation of λ_2 reproduces known analytic results. Our calculation for λ_3 gives $\lambda_3 = 0.4466 \pm 0.0006$. It is worth noting that existing analytic results give extremely tight bounds on the possible values for λ_3. In fact, by combining the results of Schrijver and Ciucu, we have that

$$0.440076 \le \lambda_3 \le 0.463107.$$

The organization of the rest of the paper is as follows: in Section 2 we give a history of the problem; in Section 3 we sketch the main ideas of importance sampling; in Section 4 we explain Sinkhorn balancing and comment on its relationship with the permanent; Section 5 gives the details of our permanent approximation algorithm with subsections discussing the variance and some nonstandard programming details; and finally our results for λ_2 and λ_3 are given in Section 6.

2. BACKGROUND AND HISTORY OF THE PROBLEM

We define a *brick* to be a d-dimensional ($d \ge 2$) rectangular parallelepiped with sides whose lengths are integers. A dimer is a brick whose volume is 2. An m-brick is a brick with m units on each side. The number of different ways to fill an m-brick with dimers (with no holes) we call F_d. It is well known that F_d grows exponentially with the volume of the m-brick, that is with m^d in all dimensions d, and it was proved in [9] that

$$\lim_{m\to\infty} \frac{\log(F_d)}{m^d}$$

exists. The limit will be denoted λ_d. Figure 1 shows one way to fill a 6-brick with dimers in 2D. To estimate F_d, we would need to find how many ways this is possible for all m-bricks.

There is extensive literature concerning the calculation of λ_d. For $d = 2$, Temperley and Fisher [29] and Kastelyn [15] gave an analytic solution,

$$\lambda_2 = 0.29156090\cdots.$$

Their method, based on finding a Pfaffian orientation for the lattice [16], does not extend to dimension 3. The early paper by Fowler and Rushbrooke [6] gave rigorous bounds,

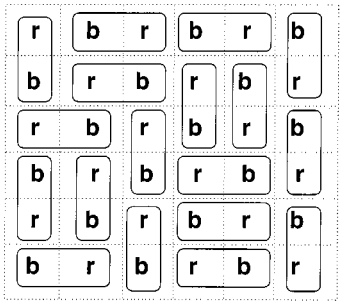

FIG. 1. Example of one dimer covering of red (r) and black (b) sites.

$0 \le \lambda_3 \le 0.54931$. The upper bound was improved by Minc [19] to 0.54827 and recently by Ciucu [5] to 0.463107 by using an elegant application of group theory.

Because λ_d is a nondecreasing function of d, a lower bound for λ_3 is $\lambda_2 = 0.29156$. This was improved by Hammersley [8] to 0.418347 and by Priezzhev [22] to 0.419989. A conjecture by Schrijver and Valiant [28] on lower bounds for permanents would imply, as noted by Minc [20], that $\lambda_3 \ge 0.440075$. Recently, Schrijver [27] proved that the lower bound is indeed 0.440076.

It was known [8] that computing λ_d is equivalent to finding the permanent of a certain matrix A. Here is what this means: Let us consider $d = 2$ first. Think of the m-brick as a checkerboard with m squares on a side. We label the squares red and black separately and arbitrarily as in Fig. 1. There are $m^2/2$ red squares and $m^2/2$ black squares (m must be even). We then form an incidence matrix A_m of size $m^2/2 \times m^2/2$ with a 1 in position (i, j) if red square i touches black square j, and 0 otherwise. One dimer covering of the checkerboard means a "path" through A, that is, a selection of exactly one nonzero element in each row and column. The dimer covering problem then translates into finding the number of paths through the matrix A. We assume a periodic boundary condition (=toroidal) so that every square has exactly four neighbors (north, south, east, and west) and thus, the matrix A would have exactly 4 ones in every row and column. When $d = 3$, the m-brick is a cube with side m, volume m^3, and the matrix A is $m^3/2 \times m^3/2$ with exactly 6 ones in every row and column, because each red or black cube has neighbors on the top and bottom in addition to north, south, east, and west. We will let $a_{i,j}$ denote the (i, j)th element of A. Unless noted otherwise, we will assume that $d = 3$ for the rest of this paper.

The *permanent* of A, which we will also write as $|A|$ is

$$\sum_{\sigma} \prod_{i=1}^{m^3/2} a_{i,\sigma(i)}.$$

Here σ ranges over $\mathcal{S}_{m^3/2}$, the set of all permutations on $m^3/2$ letters. Note that when A is a 0–1 matrix, the only terms that contribute to the sum are for those permutations σ, where all $a_{i,\sigma(i)}$ are nonzero, which is exactly when the set of those $a_{i,\sigma(i)}$ are a path through the matrix. So the number of paths through a 0–1 matrix, A, is $|A|$. Note also, that the permanent of a matrix is similar to the determinant but it lacks the alternating signs of the terms. (In earlier notation, $F_d = |A_m|$.)

At first glance, both the permanent and the determinant seem to require $O(n!)$ operations. However, the determinant is an alternating multilinear form and classic methods of linear

algebra apply, giving algorithms to do the evaluation in $O(n^3)$ operations. Computing the permanent, however, really *is* difficult. The best known algorithm for computing the permanent exactly, due to Ryser, [23] requires $O(n2^n)$ operations. This is not a practical method for this problem because the size of the matrices necessary to find λ_3 are greater than 2000×2000 for those m's necessary to obtain statistically reliable estimates of the limit.

The matrix A arising from the monomer–dimer problem has a special, highly regular structure. Fisher and Temperley [29] and, independently, Kastelyn [15] used the special structure, find a Pfaffian and thus reduce evaluating the permanent to linear algebra for the 2D case. It is known, however, that the 3D case cannot yield to the linear algebra approach. For details see Hammersley [8] and Kenyon, Randall, and Sinclair [16].

Approximating the permanent also has been studied by many authors. Karmarkar, Karp, Lipton, Lovasz, and Luby [14] gave a permanent approximation algorithm whose runtime grows exponentially with matrix size. More recently, Barvinok [3] proposed a technique based on "measure concentration." In Section 5.2 we give some data comparing these methods to ours for a few sample examples. Jerrum and Sinclair [13] developed a random-walk algorithm that runs in polynomial time for some important classes of matrices. An excellent source of results using random walks can be found in the paper by Jerrum and Sinclair [13]. In [16], possible application is described. These methods could ultimately give an independent approximation for λ_3.

3. IMPORTANCE SAMPLING

We use direct random sampling using an importance function rather than the more widely used Markov chain random walk approach. Importance sampling is a form of Monte Carlo sampling designed to reduce the variance of the estimators for a given size sample [10].

One formulation of importance sampling is as follows: we wish to estimate a sum

$$\mathcal{F}(N) = \sum_{j=1}^{N} f(\sigma_j),$$

where f is a known function, the σ_j belong to some set of size N, and N is very large. In our case, for an $n \times n$ matrix, the $\sigma_j \epsilon S_n$, the permutations on n letters, so $N = n!$ and

$$f(\sigma) = \prod_{i=1}^{n} a_{i,\sigma(i)}.$$

A simple Monte Carlo method would be to choose $M \ll N$ samples, σ_j, uniformly and compute

$$\left(\frac{\sum_{j=1}^{M} f(\sigma_j)}{M} \right) N$$

to get an average value of f. In other words, take as a "typical" value for $f(\sigma)$ the sample mean

$$\left(\frac{\sum_{j=1}^{M} f(\sigma_j)}{M} \right)$$

and then scale for the size of the sample space, N. Notice that the probability, $p(\sigma)$ of choosing any particular σ is $1/N$ so that our sum can be written

$$\left(\frac{\sum_{j=1}^{M} f(\sigma_j)N}{M}\right) = \frac{1}{M}\left(\sum_{j=1}^{M} \frac{f(\sigma_j)}{p(\sigma_j)}\right).$$

The technique of importance sampling is to use a nonuniform probability, $p(\sigma)$ that is somehow better than uniform, in order to reduce the variance. Notice that as M gets large

$$\sum_{j=1}^{M} \frac{f(\sigma_j)}{p(\sigma_j)}\frac{1}{M} \to \sum_{j=1}^{M} \frac{f(\sigma_j)}{p(\sigma_j)}\, p(\sigma_j) = F(N)$$

and that this limit will hold for *any* probability distribution $p(\sigma)$. The ideal choice for $p(\sigma)$ is

$$p(\sigma) = \frac{f(\sigma)}{F(N)}.$$

That is, the weight assigned to σ is equal to its relative contribution to the desired sum. This choice is ideal because it eliminates the variance,

$$\text{var}^2 = \frac{1}{M}\left(\sum_{j=1}^{M}\left(\frac{f(\sigma_j)}{p(\sigma_j)}\right)^2\right) - F^2 \to \sum_{j=1}^{M} \frac{f^2(\sigma)}{p(\sigma)^2}\, p(\sigma) - F^2$$

$$= \sum_{j=1}^{M} \frac{f^2(\sigma)}{p(\sigma)} - F^2 = F^2 - F^2 = 0.$$

Of course, this $p(\sigma)$ requires prior knowledge of F, the answer! Our aim is to get close to the ideal importance function, $p(\sigma)$.

We choose a permutation by choosing one element from successive rows, using an estimate of the percentage of the paths that go through that element (i.e. the probability that a path goes through that matrix location). We notice that

$$\frac{a_{i,j}|A_{i,j}|}{|A|}$$

is the fraction of paths passing through location (i, j) and that if, for each A, we could evaluate the following matrix, which we will call the *matrix balance*, m-bal(A), then we would have a perfect importance function:

$$m\text{-bal}(A) = \begin{bmatrix} \frac{a_{1,1}|A_{1,1}|}{|A|} & \frac{a_{1,2}|A_{1,2}|}{|A|} & \cdots & \frac{a_{1,n}|A_{1,n}|}{|A|} \\ \frac{a_{2,1}|A_{2,1}|}{|A|} & \frac{a_{2,2}|A_{2,2}|}{|A|} & \cdots & \frac{a_{2,n}|A_{2,n}|}{|A|} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{a_{n,1}|A_{n,1}|}{|A|} & \frac{a_{n,2}|A_{n,2}|}{|A|} & \cdots & \frac{a_{n,n}|A_{n,n}|}{|A|} \end{bmatrix}.$$

Here $A_{i,j}$ denotes the minor of A obtained by deleting row i and column j from the original matrix A.

Clearly, the (i, j)th element contains the percentage of paths that go through location (i, j). The map $A \to m\text{-bal}(A)$ is known as the Bregman map and has been studied by Bregman [4], Bapat [2], and Linial, Smorodnitsky, and Wigderson [17]. Note that m-bal(A) is *doubly-stochastic*; that is, its entries are all nonnegative and all the rows sum to 1, as do all the columns. We choose a "perfect" path through the matrix as follows:

(1) In row 1, we select column j with probability

$$\frac{a_{1,j}|A_{1,j}|}{|A|}.$$

Notice that if we do select column j, then not only must $a_{1,j} \neq 0$ but also the permanent of the minor $|A_{1,j}|$ must be nonzero and so we know that there *must* be a path in the minor that along with $a_{1,j}$ gives a path in the matrix A. We say that $a_{i,j}$ is *supported* if there exists a path in A that goes through position (i, j). So $a_{i,j}$ is supported if and only if $m\text{-bal}(A)_{i,j} \neq 0$.

(2) Look at the minor $A_{1,j}$ obtained by deleting row 1 and column j from the original matrix A and matrix-balance $A_{1,j}$. In the first row (obtained from the second row of the original A matrix), an element will look like

$$\frac{a_{2,k}\left|A_{(1,j),(2,k)}\right|}{|A_{1,j}|},$$

where $|A_{(1,j),(2,k)}|$ is the permanent of the $(n-2) \times (n-2)$ minor of A obtained by deleting row 1, column j and row 2, column k. Select one particular column k of row 2 with probability

$$\frac{a_{2,k}\left|A_{(1,j),(2,k)}\right|}{|A_{1,j}|}.$$

(3) Again take the minor of A deleting rows 1 and 2 and columns j and k. Minor-balance that minor. Select another column, as above, etc.

$\vdots$

(n) Continue in this way, until there is a 1×1 matrix left. This final matrix must be a nonzero because at the previous (and every) stage of this procedure we select a nonzero element with nonzero probability. But that element is an element of A (which thus must be nonzero) times a permanent of a minor which is also nonzero.

So, if we were able to get the minor-balance of a matrix easily we would have $|A|$ exactly, namely the product of the inverse probabilities chosen at each stage. That is,

$$|A| = \frac{|A|}{\left|A_{(1,j_1)}\right|} \frac{\left|A_{(1,j_1)}\right|}{\left|A_{(1,j_1)(2,j_2)}\right|} \frac{\left|A_{(1,j_1)(2,j_2)}\right|}{\left|A_{(1,j_1)(2,j_2)(3,j_3)}\right|} \cdots \left|A_{(1,j_1)\cdots(n-1,j_{n-1})}\right|.$$

Unfortunately, knowing m-bal(A) is equivalent to knowing $|A|$ and so computing it is intractable [12].

There is an approximation to m-bal(A) available in the Sinkhorn balance of A [25, 26]. This is the importance function that we use to calculate the dimer constants. In principle, one can Sinkhorn-balance a matrix in polynomial time [17].

4. SINKHORN BALANCING

The Sinkhorn balance of a matrix, $B = s\text{-bal}(A)$, is a doubly stochastic matrix obtained from A by multiplying by diagonal matrices D and E, $B = DAE$. The (i, j)th entry of B we call $b_{i,j}$. Sinkhorn [25] discusses why this is possible and Soules [26] gives convergence information. For more information on the mathematics of Sinkhorn balancing see [1, 18]. In practice, instead of determining D and E directly, we Sinkhorn balance by first dividing all rows of A by their sum. This makes A row stochastic but possibly not column-stochastic. So divide all columns by their sum. This makes A column stochastic, but possibly not row stochastic. Continuing alternatively with rows and columns converges to s-bal(A). Sinkhorn balancing is thus an application of the method of iterating projections in Hilbert space [17, 24].

Soules [26] shows that Sinkhorn balancing converges linearly when all elements are supported. We take advantage of this by predetermining the unsupported elements and setting the Sinkhorn balance of those elements to zero (to which unsupported elements would eventually converge) before doing the balancing computation. By an unsupported element we mean an element (i, j) of the matrix that is on no path. Recall that a path in a matrix is a selection of one nonzero element from every row in the matrix, so that every column occurs exactly once. Hopcroft and Karp [11] have an $O(n^{5/2})$ algorithm for finding paths through a matrix. It is possible to alter their algorithm so that after finding one path in the matrix deciding if another element is on a path is $O(n)$.

4.1. *Relationships between* s-bal *and* m-bal

The purpose of this section is to give a heuristic explanation of why the algorithm works as well as it does. Recall from Section 3 that ideally s-bal$(A) = m$-bal(A), that is, $b_{i,j} = |A_{i,j}|/|A|$. Unfortunately this is not the case. In fact, for some matrices, the ratio of individual terms can be exponential in the size of the matrix, although these never occur for the dimer problem, except for very small minors. However, it is the case that the Sinkhorn balance is, in a loose sense, "as good as it can be." In particular, because the Sinkhorn balance maximizes entropy for a given zero pattern in the set of doubly stochastic matrices, it tends to minimize the permanent, and a minimum permanent matrix for a given zero pattern would have the ideal property that s-bal$(A) = m$-bal(A). We present some empirical evidence for the relationship between maximum entropy and minimum permanent and also give a novel proof of the maximum entropy property. In addition we prove that the effect of importance sampling is to choose a path such that the expected value of the ratio of $|B_{i,j}|$ to $|B|$ equals one. We also use the properties of importance sampling to calculate the expected value of individual probabilities and their relation to row sums.

To explain in a little more detail, we first show that

$$\frac{b_{i,j}|B_{i,j}|}{|B|} = \frac{|A_{i,j}|}{|A|}$$

for $b_{i,j} \neq 0$ (supported $a_{i,j}$). This is Theorem 1. For s-bal$(A) = m$-bal(A), we would like $|B_{i,j}|/|B|$ to equal one. This is unfortunately not the case. It is only true that $|B_{i,j}|/|B| = 1$ when all the row sums of A are equal. This is in Ando [1]. Ando also shows that A has all row and column sums equal if and only if its Sinkhorn balance is a matrix with minimum permanent among doubly stochastic matrices with a given zero pattern. The entropy

of a doubly stochastic matrix B is $-\sum_{i,j} b_{i,j} \log(b_{i,j})$. The Sinkhorn balance maximizes entropy. This is Theorem 2. When the permanent is minimized, the entropy of that matrix is maximized, if all the row and column sums are equal. However, if row sums are not all equal, it is possible that there is no matrix that minimizes the permanent for a given zero pattern. As a worst case we look at the upper triangular matrix with a 1 in position $(n, 1)$. For $n = 4$ this is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

We will refer to this as the "bad" matrix. It is as far from having equal row and column sums as possible; $|B_{i,j}|/|B| = e^{\pm n/2}$ for this matrix.

Figure 2 shows iterations of the Bregman map on this matrix. We plot permanent versus entropy. The point at the top of the curve, maximizing entropy, is the Sinkhorn balance, B. By iterating the Bregman map on B, the points to the right are obtained. In this case, we can compute the inverse of the Bregman map to obtain the points to the left of B. Even in this worst case, the maximum entropy matrix is not that bad an approximation to the minimum permanent matrix for the purposes of this problem because of Theorem 3, which says that the expected value of $|B_{i,j}|/|B|$ is 1.

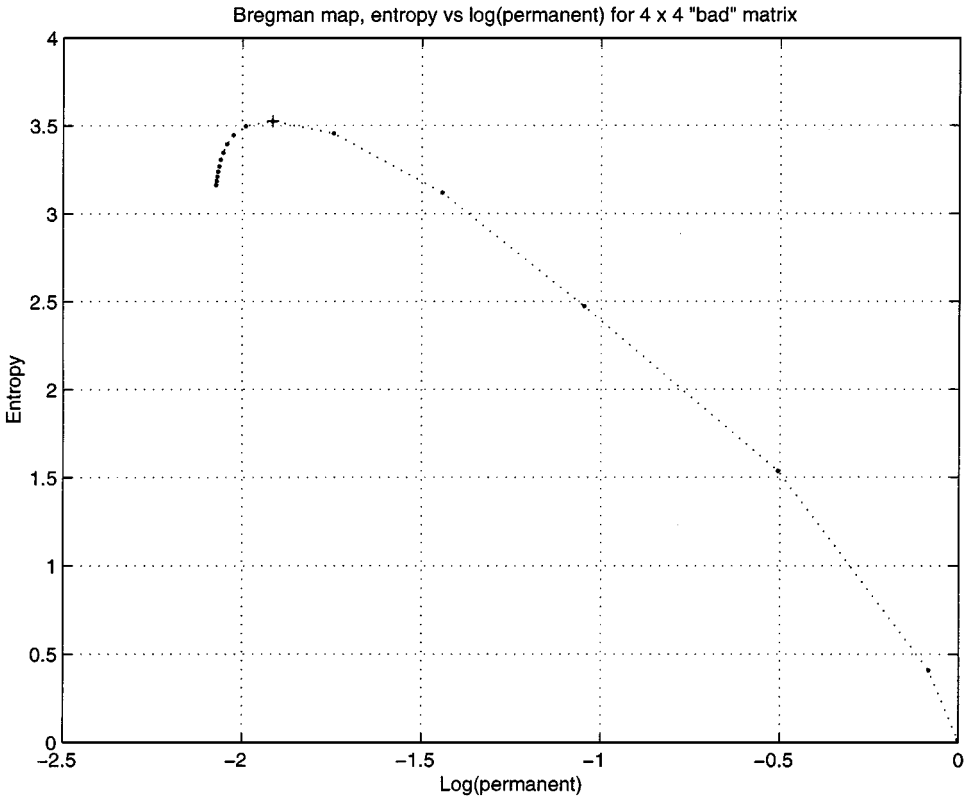

FIG. 2. Iterations of the Bregman map on the "bad" matrix.

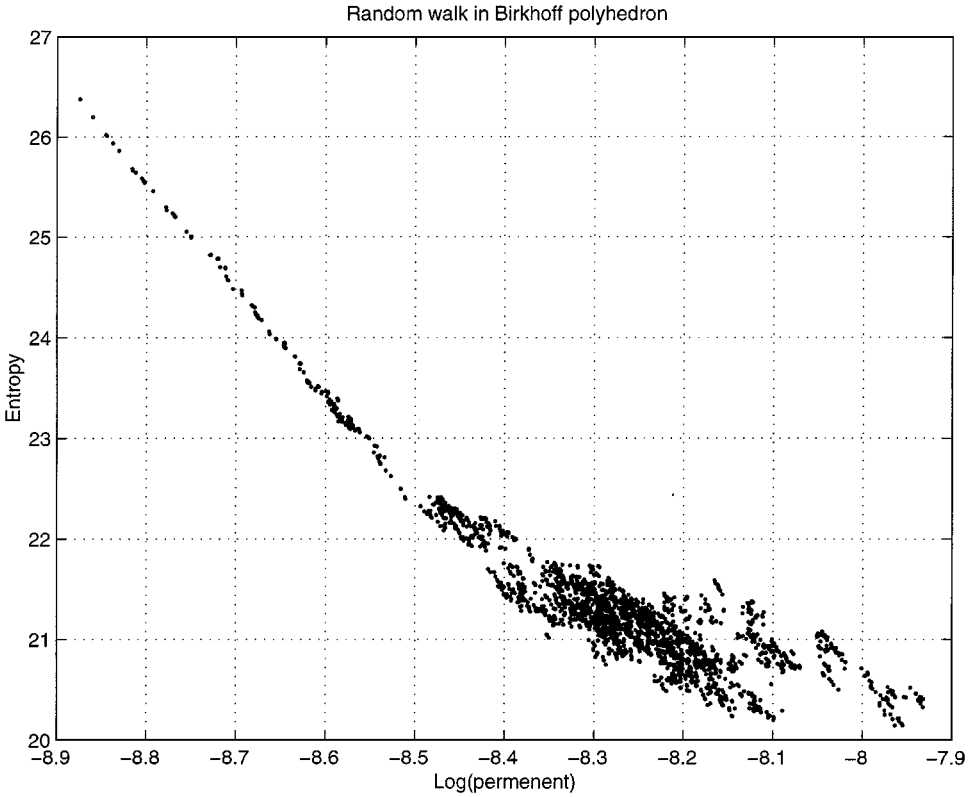

FIG. 3. Random walk through Birkhoff polyhedron.

A random walk on the *Birkhoff polyhedron*, the set of doubly stochastic matrices, also supports the observation that the maximum entropy matrix is not a bad approximation to the minimum permanent matrix. We illustrate this in Figs. 3 and 4. The random walk starts at the Sinkhorn balance of the 11×11 "bad" matrix and we plot permanent versus entropy. Figure 4 is a three-dimensional version of Fig. 3, where we plot elapsed time as the z-axis.

The proof of the theorems follows.

LEMMA 1. *The product along any path through B equals $|B|/|A|$.*

Proof. $B = s\text{-bal}(A)$, so $B = DAE$, where D and E are diagonal. So, $|B| = |D||A||E|$ and, hence, for any path σ

$$\frac{|B|}{|A|} = |D||E| = \prod_{1 \le i \le n} d_i \prod_{1 \le j \le n} e_j = \prod_{1 \le i \le n} d_i e_{\sigma(i)}.$$

On the other hand,

$$\prod_{1 \le i \le n} b_{i,\sigma(i)} = \prod_{1 \le i \le n} d_i a_{i,\sigma(i)} e_{\sigma(i)} = \prod_{1 \le i \le n} d_i e_{\sigma(i)},$$

where the last equality holds because A is a 0–1 matrix. ■

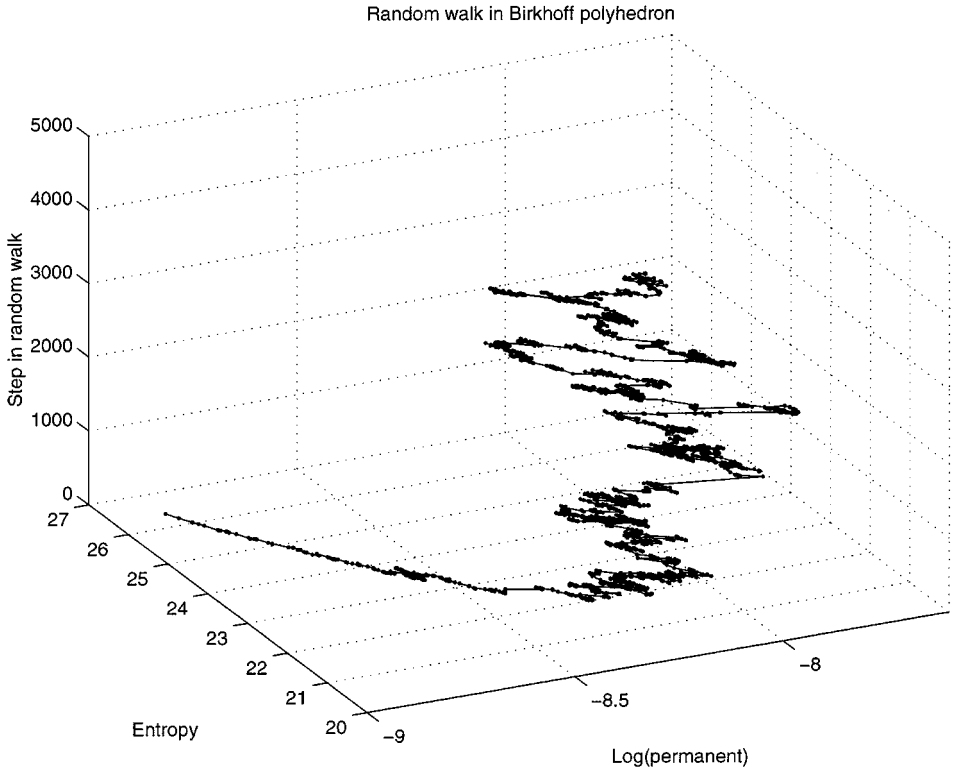

FIG. 4. Random walk through Birkhoff polyhedron with time as the third dimension.

THEOREM 1. *For all* i, j,

$$\frac{|A_{i,j}|}{|A|} = b_{i,j}\frac{|B_{i,j}|}{|B|}$$

Proof.

$$b_{i,j}|B_{i,j}| = b_{i,j}\left(\sum_{\bar{\sigma}}\prod_{k\neq i} b_{k,\bar{\sigma}(k)}\right),$$

where the sum ranges over all $(n-1)$-paths $\bar{\sigma}$, omitting i and j. Hence,

$$b_{i,j}|B_{i,j}| = \sum_{\sigma}\prod_{k=1}^{n} b_{k,\sigma(k)},$$

where the σ range over all permutations where $\sigma(i) = j$. Therefore,

$$b_{i,j}|B_{i,j}| = |A_{i,j}|\frac{|B|}{|A|}$$

because there are $|A_{i,j}|$ such product terms and by the previous lemma, each one is equal to $|B|/|A|$. ■

LEMMA 2. *If B is the Sinkhorn balance of a completely supported, 0–1 matrix, A, then for any doubly stochastic matrix, D, whose support is contained in the support of A we have that*

$$\sum_{i,j} d_{i,j} \log(b_{i,j}) = \log\left(\frac{|B|}{|A|}\right).$$

Proof. Because D is doubly stochastic with support contained in the support of A, we may write

$$D = \sum_{\sigma} \lambda_\sigma P_\sigma,$$

where the P_σ are permutation matrices whose nonzero elements occur at nonzero locations in B and the λ_σ are positive with

$$\sum_{\sigma} \lambda_\sigma = 1.$$

For each i, j, we have that

$$d_{i,j} = \sum \lambda_{\sigma_{i,j}},$$

where the sum is over those permutations, $\sigma_{i,j}$ that are nonzero at the i, j element of the matrix D. Because B is the Sinkhorn balance of A, we have for each σ

$$\prod b_{i,\sigma(i)} = \frac{|B|}{|A|}$$

by our first theorem. Therefore,

$$\sum_{i} \log\left(b_{i,\sigma(i)}\right) = \log\left(\frac{|B|}{|A|}\right)$$

and, hence,

$$\sum_{\sigma} \lambda_\sigma \sum_{i} \log\left(b_{i,\sigma(i)}\right) = \log\left(\frac{|B|}{|A|}\right).$$

Re-arranging this double sum to collect the coefficient of $\log b_{i,j}$ gives

$$\sum_{i,j} \Big(\sum \lambda_{\sigma_{i,j}} \Big) \log(b_{i,j}) = \log\left(\frac{|B|}{|A|}\right).$$

The result now follows from the expression for $d_{i,j}$ as sums of the $\lambda_{\sigma_{i,j}}$.

THEOREM 2 (Maximum entropy). *If D is any doubly stochastic matrix with support contained in the support of B, then*

$$-\sum_{i,j} d_{i,j} \log(d_{i,j}) \le -\sum_{i,j} b_{i,j} \log(b_{i,j}).$$

Proof. By the generalized arithmetic–geometric mean inequality (see, for example, [21]),

$$-\sum_{i,j} d_{i,j} \log(d_{i,j}) \le -\sum_{i,j} d_{i,j} \log(b_{i,j})$$

and the result follows from Lemma 2. ■

Note that we may choose

$$d_{i,j} = \frac{|A_{i,j}|}{|A|}.$$

THEOREM 3. *The expected value is*

$$\mathcal{E}\left(\frac{|B_{i,j}|}{|B|}\right) = 1.$$

Proof.

$$\mathcal{E}\left(\frac{|B_{i,j}|}{|B|}\right) = \left(\sum_{k=1}^{N} \frac{|B_{i,j}|}{|B|}\right) \Big/ N,$$

where N is the number of samples. Fixing i, the probability that $|B_{i,j}|/|B|$ is chosen on row i is $b_{i,j}$. So in the long run,

$$\frac{\sum_{k=1}^{N} |B_{i,j}|/|B|}{N} \to \sum_{j=1}^{n} b_{i,j} \frac{|B_{i,j}|}{|B|} = \frac{|B|}{|B|} = 1. \ ■$$

THEOREM 4. *The expected value is*

$$\mathcal{E}\left(\frac{1}{b_{i,j}}\right) = m_i,$$

where m_i is the number of 1*'s in row i.*

Proof. Suppose row i is fixed:

$$\mathcal{E}\left(\frac{1}{b_{i,j}}\right) = \sum_{k=1}^{N} \frac{1}{b_{i,j}} \Big/ N.$$

The probability that j is chosen in row i is $b_{i,j}$. Thus,

$$\sum_{k=1}^{N} \frac{1}{b_{i,j}} \Big/ N = \sum_{j=1}^{N} b_{i,j} \frac{1}{b_{i,j}} = m_i. \ ■$$

5. PROCEDURE

Here then is the procedure:

(1) Sinkhorn balance A. This produces a doubly stochastic matrix B with (i, j)th element $b_{i,j}$. Because row 1 sums to 1, we can select a column, j_1, with probability b_{1,j_1}. We will write this simply as b_{j_1} because at each stage we will always select from the first row of the Sinkhorn balanced matrix.

(2) Sinkhorn balance the $(n-1)\times(n-1)$ minor A_{1,j_1} giving matrix $B^{(1)}$ with elements $b^{(1)}_{i,j}$, select a column j_2, with probability $b^{(1)}_{j_2}$, where we use the first row of $B^{(1)}$ as probabilities.

$\vdots$

(k) Sinkhorn balance the $(n-k+1)\times(n-k+1)$ minor of A obtained by removing the first $(k-1)$ rows and columns $j_1, \ldots, j_{k-1}$ from A. Call the resulting matrix $B^{(k-1)}$. Select a column j_k from the first row of $B^{(k-1)}$ with probability $b^{(k-1)}_{j_k}$.

$\vdots$

($n-1$) Continue until there is only a 1×1 matrix left.

Then $(1, j_1), (2, j_2), \ldots, (n, j_n)$ is a path of 1's in A. The value of $|A|$ is then approximated by the mean of terms like

$$\frac{1}{b_{j_1}}\frac{1}{b^{(1)}_{j_2}}\frac{1}{b^{(2)}_{j_3}}\cdots\frac{1}{b^{(n-2)}_{j_{n-1}}}.$$

Notice that this is the same as

$$\frac{a_{1,j_1}}{b_{j_1}}\frac{a_{2,j_2}}{b^{(1)}_{j_2}}\frac{a_{3,j_3}}{b^{(2)}_{j_3}}\cdots\frac{a_{n-1,j_{n-1}}}{b^{(n-2)}_{j_{n-1}}}=\frac{1}{p(\sigma)}.$$

5.1. *Our Method as Importance Sampling*

In our case, the sample space is $\mathcal{S}_n$ the set of permutations on n letters, where n is the size of the matrix we need to evaluate. For dimension 3, n is $m^3/2$ and so, for example, for an $18\times 18\times 18$ cube the matrix will have size 2916×2916 and $\mathcal{S}_n=\mathcal{S}_{2916}$ will contain 2916! elements. Our estimator $f(\sigma)/p(\sigma)$ is the characteristic function of a matrix path of A in $\mathcal{S}_n$ multiplied by $1/p(\sigma)$, which is obtained from the product of estimates of the relative number of paths at each stage of the selection. In other words $f(\sigma)=1$ if $a_{1,\sigma(1)}, a_{2,\sigma(2)}, \ldots, a_{n,\sigma(n)}$ are all 1's in the matrix A and $p(\sigma)=\prod b^{(k)}_{\sigma(k)}$. $\sigma(1)=j_1, \sigma(2)=j_2, \sigma(3)=j_3, \ldots$. Note that we compute the importance function as we proceed.

This is a subtle but important point. The Sinkhorn balance is not used as the estimator but rather as the importance factor for selecting and scaling the estimate. The mean converges to the permanent because the naive method converges to the permanent. How good or bad the Sinkhorn balance is as an estimate of the minor balance does not affect what the approximation converges to but rather how fast it converges. (In our case it converged fast enough to get extremely good error bars.)

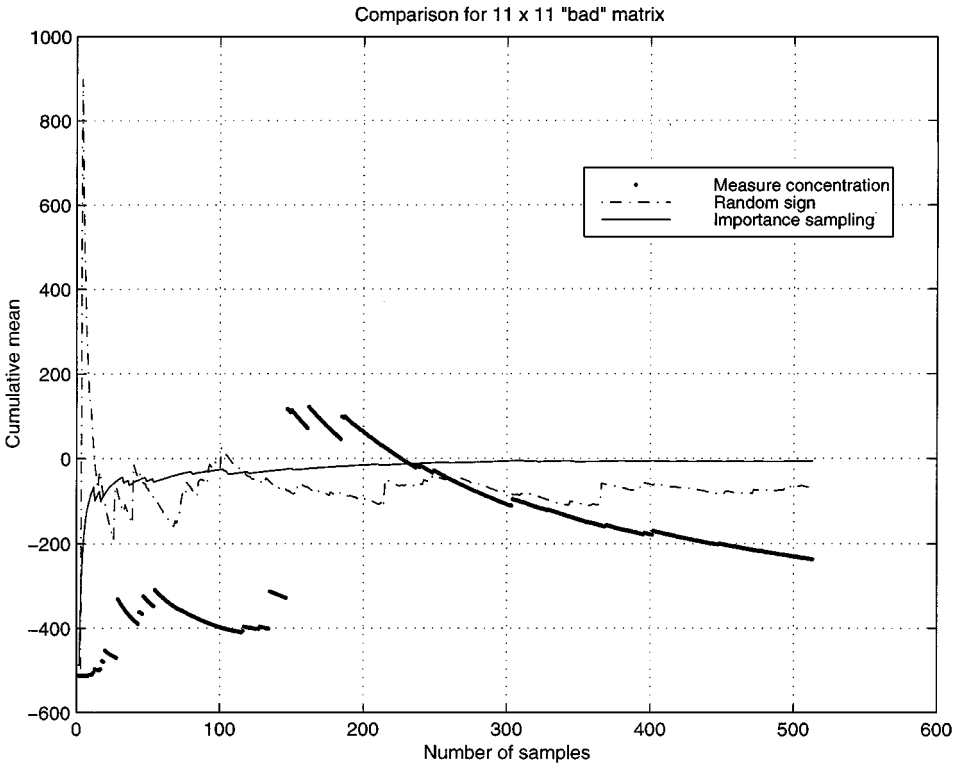

FIG. 5. Comparison of convergence rates of permanent approximation algorithms, showing error. This is the 11×11 "bad" matrix.

5.2. *Performance*

We compare the rates of convergence for the known permanent algorithms in Figs. 5 and 6. Figure 5 is the 11×11 "bad" matrix. Figure 6 is the 32×32 dimer matrix and we plot the error. We compare our technique with the methods of Barvinok [3] and Karmarkar, Karp, Lipton, Lovasz, and Luby [14]. The matrix used is the 11×11 "bad" matrix, that is, in a sense, a worst case for our algorithm.

5.3. *Variance*

The variance, var, in this calculation is by definition

$$\text{var}^2 = \frac{1}{N} \sum_{\sigma} (s_\sigma^2 - |A|^2),$$

where N is the number of samples and s is the value obtained from a single sample, namely $p(\sigma)^{-1}$ where the σ is a permutation on n letters whose selection is described in Section 5. So

$$s_\sigma^2 = \left(\frac{1}{p(\sigma)} \right)^2$$

and

$$\text{var}^2 = \frac{1}{N} \sum_{\sigma} \left(\frac{1}{p(\sigma)} \right)^2 - |A|^2.$$

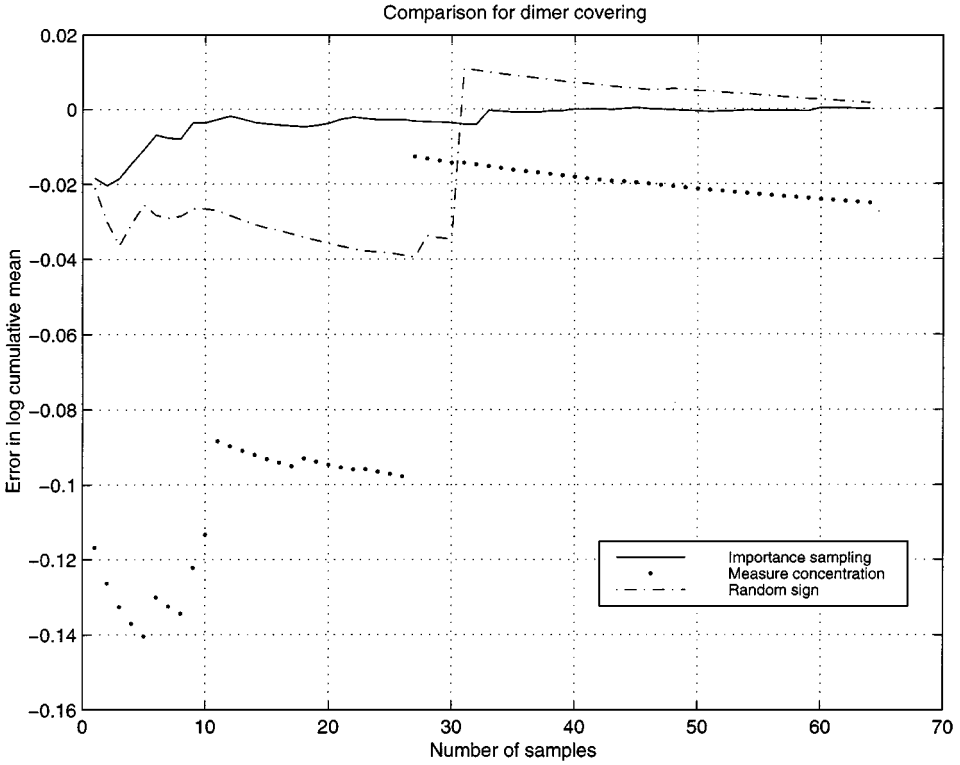

FIG. 6. Comparison of convergence rates of permanent approximation algorithms, showing the error of the log of the mean. This is the 32×32 dimer matrix.

Because of importance sampling, the probability that a particular path σ will be chosen is $p(\sigma)$. So,

$$\text{var}^2 \to \sum_{\sigma} \frac{1}{(p(\sigma))^2} p(\sigma) - |A|^2,$$

where σ ranges over all supported paths. Simplifying, this gives in the limit

$$\text{var}^2 \to \sum_{\sigma} \frac{1}{p(\sigma)} - |A|^2 = |A| \left(\frac{1}{|A|} \sum_{\sigma} \frac{1}{p(\sigma)} - |A| \right).$$

But this is

$$|A| \left(\left\langle \frac{1}{p(\sigma)} \right\rangle - |A| \right)$$

because there are $|A|$ paths. Here $\langle \cdot \rangle$ denotes the uniform average over *all* paths, not just paths chosen by importance sampling. Thus we have

$$\text{var}^2 = |A|^2 \left(\frac{1}{|A|} \left\langle \frac{1}{p(\sigma)} \right\rangle - 1 \right)$$

and the relative variance is

$$\frac{\text{var}^2}{|A|^2} = \frac{1}{|A|}\left\langle \frac{1}{p(\sigma)} \right\rangle - 1.$$

If we think of $p(\sigma)$ as a probability distribution on the paths σ, we note that the relative variance is determined by the degree to which $p(\sigma)$ approximates the "perfect" distribution which would give each path weight equal to $1/|A|$.

It is also worth noting what the variance would be if we just chose $1/b_j$ uniformly. Without importance sampling var^2 would be

$$\frac{1}{|A|}\left\langle \frac{1}{p(\sigma)^2} \right\rangle - 1,$$

a very large quantity.

5.4. *Data Structures Used in Permanent Computation*

The matrix A used in this calculation is sparse. It contains $6n$ nonzero elements where A is $n \times n$. We make use of the sparsity by maintaining, instead of the matrix A, a row-matrix and a column-matrix, which contain respectively for a given row, the column numbers of the next nonzero element and for a given column the row numbers for the next nonzero element. The actual values of the elements in the Sinkhorn balancing are kept as one array with pointers into it from the row-matrix and the column-matrix.

Sinkhorn balancing a minor iterates in two steps, row balancing and column balancing. For our data structure row balancing is straightforward. Column balancing is more elaborate. Because we need to keep a record of which columns of A are to be used in the minor we wish to balance, we maintain this information by using a linked list of the active columns. When we delete a column (by choosing it in a path) we do so by marking it as deleted and then on the next traversal of the active-column list we do the delete from the list.

5.5. *Computing* $\log(|A|)$ *when* $|A|$ *is not Representable*

One of the challenges in this calculation is that $|A|$ is not representable in floating point when the matrix size is large. For example, $\lambda_3^{(14)}$ is around 0.45, so $\log(|A_{14}|) = 0.45 * 14^2/2 \approx 617$. Thus $|A| \approx e^{617}$. However, the logs of the individual samples *are* representable and we must use these, instead of actual estimates of the permanent. We want to estimate

$$\lambda_3^{(m)} = \log(\text{Perm})/m^3,$$

where

$$\text{Perm} \approx \left\langle \frac{1}{p} \right\rangle = \frac{1}{N}\sum_{i=1}^{N}\prod_j \frac{1}{b_j}.$$

But

$$\prod \frac{1}{b_j} = \exp\left(\log \prod \frac{1}{b_j}\right) = \exp\left(\sum (-1)\log(b_j)\right).$$

All our samples of $\sum(-1)\log(b_j)$ are approximately the same size. Let C be the smallest integer contained in all of these terms. That is, C is the minimum over all samples of

$$\left\lfloor \sum(-1)\log(p_i) \right\rfloor.$$

So,

$$\prod \frac{1}{b_j} = \exp(C + R_i),$$

where R_i is the remainder after C is subtracted from the log of the estimate of sample i. So,

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}\prod_j \frac{1}{b_j} &= \frac{1}{N}\left(\sum_{i=1}^{N}\exp(C)\exp(R_i)\right)\\ &= \exp(C)\frac{1}{N}\sum_i \exp(R_i)\\ &= \exp(C)\,\mathcal{E}(\exp(R_i)).\end{aligned}$$

Thus,

$$\begin{aligned}\lambda_3^{(m)} &= \log(|A|)/m^3\\ &\approx \log(\exp(C)\,\mathcal{E}(\exp(R_i)))/m^3\\ &= C + \log(\mathcal{E}(\exp(R_i)))/m^3.\end{aligned}$$

Thus, it is possible to get the log of the average permanent even though each permanent is not representable by using the average of exp of the remainder which *is* representable.

6. OBSERVATIONS AND RESULTS

It is interesting to observe that on the whole, for different paths, the same probabilities occur and in approximately the same proportions. In other words, the probability distribution $p(\sigma)$ is observed to concentrate near the uniform distribution $1/|A|$.

Figure 7 shows calculations for λ_2, and Fig. 8 shows calculations for λ_3. The error bars in these figures were obtained by taking twice the standard deviation over $\sqrt{N}$, where N is the number of samples. For both two and three dimensions we fit the input points with a quadratic, $y = \alpha + \beta/x^2$, where x are our values m and the corresponding y are the values $\log(|A_m|)/m^d$, $d = 2, 3$. The α's, the limiting values, are our approximations to λ_2 and λ_3. We use $1/x^2$ instead of x because it is more stable numerically to find α by assuming $1/x^2 = 0$ rather than taking a limit as x gets large. The error bars on the limiting value α were obtained by doing a similar regression on the error obtained from the first fit.

For λ_2, our result 0.291 agrees extremely well with the known analytic value, 0.29156090. For λ_3 then we get 0.4466 ± 0.0006.

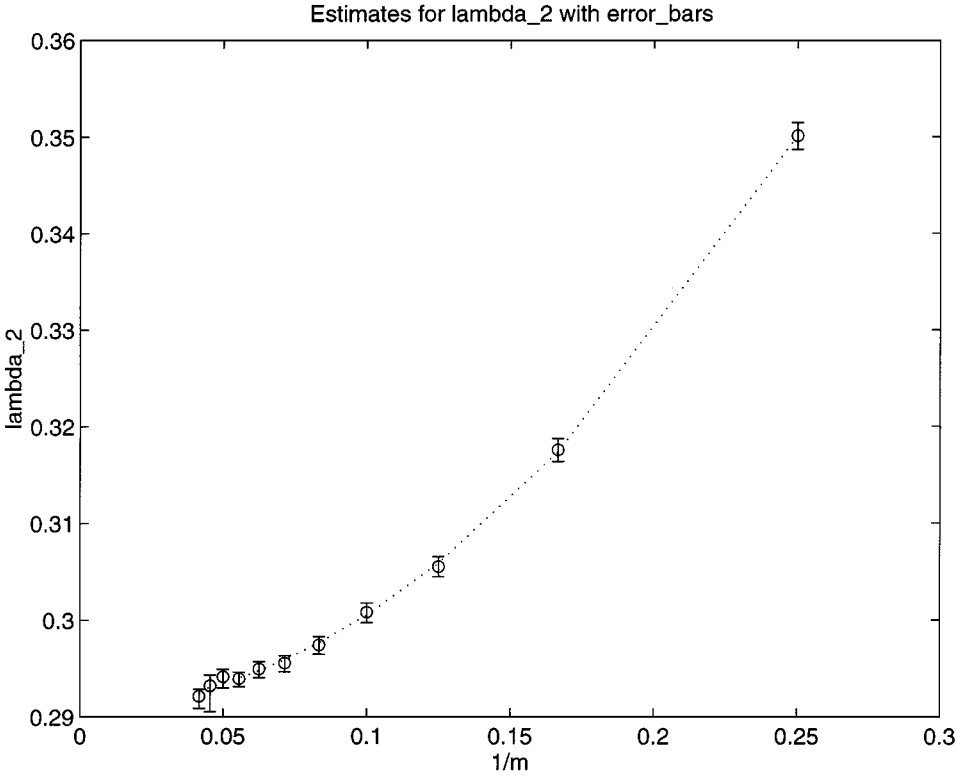

FIG. 7. $\lambda_2^{(m)}$ values fit with $y = \alpha + \beta/x^2$.

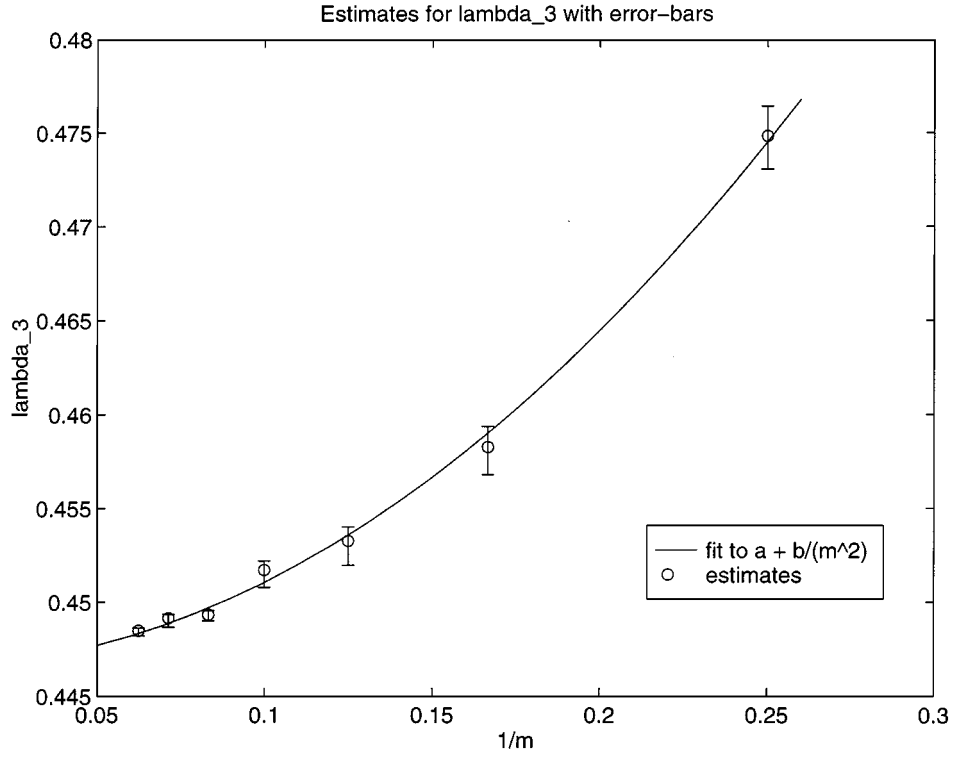

FIG. 8. $\lambda_3^{(m)}$ values fit with $y = \alpha + \beta/x^2$.

ACKNOWLEDGMENTS

We thank Dr. Eleazer Bromberg for many useful discussions and insights on the permanent. We also thank Dr. George Soules for introducing us to the importance of importance sampling.

REFERENCES

1. T. Ando, Majorization, doubly stochastic matrices and comparison of eigenvalues, *Linear Algebra Appl.* **118**, 163 (1989).
2. R. B. Bapat, Applications of an inequality in information theory to matrices, *Linear Algebra Appl.* **78**, 107 (1986).
3. A. Barvinok, Computing mixed discriminants, mixed volumes and permanents, *Discrete Comput. Geom.* **18**, 205 (1997). [Mixed discriminants within a simply exponential factor, to appear]
4. L. M. Bregman, Proof of convergence of Sheleikhovskii's method for a problem with transportation constraints, *Zh. Vychsl. Mat. Mat. Fiz.* **7**, 147 (1967).
5. M. Ciucu, An improved upper bound for the three-dimensional dimer problem, to appear.
6. R. H. Fowler and G. S. Rushbrooke, Statistical theory of perfect solutions, *Trans. Faraday Soc.* **33**, 1271 (1937).
7. D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic Press, San Diego, 1996).
8. J. M. Hammersley, An improved lower bound for the multidimensional dimer problem, *Proc. Camb. Phil. Soc.* **64**, 455 (1968).
9. J. M. Hammersley, Existence theorems and Monte Carlo methods for the monomer–dimer problem, in *Research Papers in Statistics: Festschrift for J. Neyman* (Wiley, New York, 1966), p. 125.
10. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods* (Wiley, New York, 1964).
11. J. Hopcroft and R. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* **2**, 225 (1973).
12. M. R. Jerrum, Two-dimensional monomer–dimer systems are computationally intractible, *J. Stat. Phys.* **48**, 121 (1987).
13. M. R. Jerrum and A. J. Sinclair, Approximating the permanent, *SIAM J. Comput.* **18**, 1149 (1989).
14. N. Karmarkar, R. M. Karp, R. Lipton, L. Lovasz, and M. Luby, A Monte Carlo algorithm for estimating the permanent, *SIAM J. Comput.* **22**, 284 (1993).
15. P. W. Kastelyn, The statistics of dimers on a lattice, *Physica* **27**, 1209 (1961).
16. C. Kenyon, D. Randall, and A. Sinclair, Approximating the number of monomer–dimer coverings of a lattice, *J. Stat. Phys.* **83**, 637 (1996).
17. N. Linial, A. Samorodnitsky, and A. Wigderson, A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents, to appear.
18. D. London, On matrices with double stochastic pattern, *J. Math. Anal. Appl.* **34**, 648 (1971).
19. H. Minc, An upper bound for the multidimensional dimer problem, *Math. Proc. Cambridge Philos. Soc.* **83**, 461 (1978).
20. H. Minc, Review of the paper "On Lower Bounds for Permanents" by A. Schrijver and W. G. Valiant, *Math Rev.* (1982). [article 82a:15004, 63]
21. A. L. Peressini, F. E. Sullivan, and J. J. Uhl, *The Mathematics of Nonlinear Programming* (Springer Verlag, New York/Berlin, 1988).
22. V. B. Priezzhev, The statistics of dimers on a three-dimensional lattice. II. An improved lower bound, *J. Stat. Phys.*, 829 (1981).
23. H. Ryser, *Combinatorial Mathematics*, Carus Monographs, Vol. 14 (Math. Assoc. of America, Washington, DC, 1963).
24. F. Sullivan and B. Atlestam, A descent method with smooth rotund spaces with applications to approximation in L_p, *J. Math. Anal. Appl.* **48**, 155 (1974).
25. R. Sinkhorn, A relationship between arbitrary positive matrices and double stochastic matrices, *Ann. Math. Stat.* **35**, 876 (1964).

26. G. W. Soules, The rate of convergence of Sinkhorn balancing, *Linear Algebra Appl.* **150**, 3 (1991).
27. A. Schrijver, Counting 1-factors in regular bipartite graphs, *J. Combin. Theory B* **72**, 122 (1998).
28. A. Schrijver and W. G. Valiant, On lower bounds for permanents, *Nederl. Akad. Wetensch. Indag. Math.* **42**, 425 (1980).
29. H. N. V. Temperley and M. E. Fisher, Dimer problem in statistical mechanics—An exact result, *Philos. Mag.* **6**, 1061 (1961).