SIAM J. COMPUT.
Vol. 22, No. 2, pp. 284–293, April 1993

005

A MONTE-CARLO ALGORITHM FOR ESTIMATING THE PERMANENT*

N. KARMARKAR†, R. KARP‡, R. LIPTON§, L. LOVÁSZ¶, AND M. LUBY[1]

Abstract. Let A be an $n \times n$ matrix with 0-1 valued entries, and let $\mathrm{per}(A)$ be the permanent of A. This paper describes a Monte-Carlo algorithm that produces a "good in the relative sense" estimate of $\mathrm{per}(A)$ and has running time $\mathrm{poly}(n)2^{n/2}$, where $\mathrm{poly}(n)$ denotes a function that grows polynomially with n.

Key words. permanent, matching, Monte-Carlo algorithm, algorithm, bipartite graph, determinant

AMS(MOS) subject classifications. 05C50, 05C70, 68Q25, 68R05, 68R10

1. Introduction. Let A be an $n \times n$ matrix with 0-1 valued entries, $\det(A)$ denote the determinant of A, and $\mathrm{per}(A)$ denote the permanent of A. The marked contrast between the computational complexity of computing $\det(A)$ versus that of computing $\mathrm{per}(A)$, despite the deceiving similarity between the two tasks, has baffled researchers for years. One of the reasons for interest in computing $\mathrm{per}(A)$ is that A can be viewed as the adjacency matrix of a bipartite graph, $H = (X, Y, E)$, where X corresponds to the rows in A, Y to the columns in A, and $A_{ij} = 1$ if there is an edge between X_i and Y_j. The quantity $\mathrm{per}(A)$ is exactly the number of perfect matchings in H.

It is well known that $\det(A)$ can be computed in $\mathrm{poly}(n)$ time. On the other hand, the fastest algorithm known for computing $\mathrm{per}(A)$ runs in $n2^n$ time [20]. Solid grounds for arguing that computing $\mathrm{per}(A)$ is an inherently difficult problem were first provided in [21], which shows that the problem is $\#P$-complete. One implication of this result is that if $P \neq NP$, then there is no $\mathrm{poly}(n)$ time algorithm for computing $\mathrm{per}(A)$.

Because of the apparent nonexistence of a $\mathrm{poly}(n)$ time algorithm for computing $\mathrm{per}(A)$ exactly, we focus our attention on finding an algorithm that produces a good estimate of $\mathrm{per}(A)$ and has a small running time. An (ϵ, δ) *approximation algorithm* for $\mathrm{per}(A)$ is a Monte-Carlo algorithm that accepts as input A and two positive parameters ϵ and δ. The output of the algorithm is an estimate Y of $\mathrm{per}(A)$, which satisfies

$$\Pr[(1-\epsilon)\mathrm{per}(A) \leq Y \leq (1+\epsilon)\mathrm{per}(A)] \geq 1 - \delta.$$

The papers [9], [15], [16] discuss (ϵ, δ) approximation algorithms for counting problems in greater detail. We develop an (ϵ, δ) approximation algorithm for $\mathrm{per}(A)$, which runs in $2^{n/2}\frac{1}{\epsilon^2}\log(\frac{1}{\delta})\mathrm{poly}(n)$ time. For fixed ϵ and δ, the running time of the approximation algorithm is essentially the square root of the running time for the fastest known algorithm that computes $\mathrm{per}(A)$ exactly.

In [3], [10], [11] (ϵ, δ) approximation algorithms for $\mathrm{per}(A)$ are given which run in $\mathrm{poly}(n)$ time in the special case when each row and column in A contains at least $n/2$ $1's$. The report [13], which appeared during the final revision of this paper, gives an

*Received by the editors November 12, 1990; accepted for publication (in revised form) December 19, 1991.

†AT&T Bell Laboratories, Incorporated, Murray Hill, New Jersey 07974-2010.

‡Computer Science Division, University of California, Berkeley, California 94720, and International Computer Science Institute, Berkeley, California 94704-1105. The research of this author was supported by National Science Foundation grant CCR-9005448.

§Computer Science Department, Princeton University, Princeton, New Jersey 08544-0001.

¶Hungarian Academy of Sciences, 1051 Budapest, Roosevelt-Ter9, Hungary and Computer Science Department, Princeton University, Princeton, New Jersey 08544-0001.

[1]International Computer Science Institute, Berkeley, California 94720. The research of this author was partially supported by National Science Foundation operating grant CCR-9016468 and by grant number 89-00312 from the United States–Israel Binational Science Foundation (BSF), Jerusalem, Israel.

(ϵ, δ) approximation algorithm for per(A), which runs in time $O(c^{\sqrt{n}\log^2(n)}\epsilon^{-2}\log(\frac{1}{\delta}))$. Whether or not there is an (ϵ, δ) approximation algorithm for per(A), which runs in poly(n) time for general A, is still an open problem.

2. Some general considerations regarding (ϵ, δ) approximation algorithms. Suppose we would like to estimate some quantity Q and have available a stochastic experiment whose output is a random variable X such that $\mathrm{E}[X] = Q$ and $\mathrm{E}[X^2]$ is finite. Suppose further that we can repeat this experiment as many times as we wish, and that the outcomes of the successive trials will be independent and identically distributed, with the same distribution as X. Let X_i be the outcome of the ith trial. A straightforward application of Chebyshev's inequality shows that, if we conduct N trials, where $N = (\mathrm{E}[X^2]/\mathrm{E}[X]^2)(1/\epsilon^2\delta)$ then $(\sum_{i=1}^{N} X_i/N)$ gives an (ϵ, δ) approximation to Q.

We can improve the dependence of the number of trials on δ using a well-known trick. Setting $\delta = \frac{1}{4}$, we find that, if $N = (4\mathrm{E}[X^2]/\mathrm{E}[X]^2)(1/\epsilon^2)$, the probability is at least $\frac{3}{4}$ that the average of the N trials will lie within ϵ of Q. To obtain an (ϵ, δ)-approximation algorithm, we repeat such an N-sample experiment K times, where K is an odd integer greater than a suitable constant c times $\log(1/\delta)$, and take as our estimator of Q the median of the estimators produced by the K experiments of N samples each. Let us say that the outcome of a N-sample experiment is *good* if it lies within ϵ of Q. Then the median of the K outcomes will be good whenever the majority of the K outcomes are good. Using the fact that the outcomes are independent and identically distributed, and that each outcome has probability at least $\frac{3}{4}$ of being good, standard bounds on the tail of the binomial distribution [5] reveal that the median is good with probability at least $1 - \delta$. Thus, the number of trials required for an (ϵ, δ) approximation to Q is

$$O\left(\frac{\mathrm{E}[X^2]}{\mathrm{E}[X]^2}\frac{1}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right).$$

3. The Godsil/Gutman estimator of per(A). The discussion of the last section shows how an (ϵ, δ)-algorithm for approximating a quantity Q can be constructed from any computable stochastic experiment whose outcome is a random variable Y such that $\mathrm{E}[Y] = Q$ and $\mathrm{E}[Y^2]$ is finite. The efficiency of the algorithm will be based on the computational difficulty of performing the stochastic experiment, and on the ratio $\mathrm{E}[Y^2]/\mathrm{E}[Y]^2$. The rest of the paper is devoted to studying two particular stochastic experiments for estimating per(A). The first of these, which we call the Godsil/Gutman estimator, was suggested in [8], but no analysis of the number of trials needed for an (ϵ, δ) approximation algorithm was provided; the second one is a variant of the Godsil/Gutman estimator, which has a smaller second moment and thus leads to a more efficient algorithm.

The Godsil/Gutman estimator is defined as follows:

(1) An $n \times n$ matrix B is formed from A as follows:

For all i, j, $1 \le i, j \le n$,

If $A_{ij} = 0$ then $B_{ij} \leftarrow 0$

Elseif $A_{ij} = 1$ then randomly and independently choose $B_{i,j} \in \{-1, 1\}$, each choice with probability $\frac{1}{2}$.

(2) $Y \leftarrow (\det(B))^2$.

This stochastic experiment can be executed in poly(n) time.

In §4 we introduce some technical language that is appropriate for all of the following analysis. Section 5 concerns the Godsil/Gutman estimator; we show that $\mathrm{E}[Y] =$ per(A) and derive an upper bound on $\mathrm{E}[Y^2]$. Then, in §6, we present a refinement of the Godsil/Gutman estimator, show that it is unbiased, and derive an upper bound on its second moment.

4. Terminology. Let P be the set of all $n!$ permutations of $1, \ldots, n$. For all $\sigma \in P$, let $\operatorname{sgn}(\sigma) = -1^t$, where t is the number of transpositions to form σ.

Let $P(A) \subseteq P$ be the set of all permutations σ such that for $i = 1, \ldots, n$, $A_{i,\sigma(i)} = 1$. Then, $\operatorname{per}(A) = \sum_{\sigma \in P(A)} 1 = |P(A)|$. For each $\sigma \in P(A)$, for $i = 1, \ldots, n$, we label $\langle i, \sigma(i) \rangle$ with the symbol σ. Let $P^2(A) = P(A) \times P(A)$. For each $\dot{\sigma} = \langle \sigma_1, \sigma_2 \rangle \in P^2(A)$, let $G(\dot{\sigma})$ be the unlabelled graph where there is an unlabelled node for each $\langle i, j \rangle$ which is labelled with either or both of σ_1, σ_2, and where there is an edge between two distinct nodes $\langle i, j \rangle$ and $\langle i', j' \rangle$ if and only if $i = i'$ or $j = j'$. Each connected component of $G(\dot{\sigma})$ is an isolated node or an even length cycle. For each cycle in $G(\dot{\sigma})$, designate one of the nodes in the cycle as the root of the cycle. Let $D = \{G(\dot{\sigma}) : \dot{\sigma} \in P^2(A)\}$. For each graph $G \in D$, let $c(G)$ be the number of cycles in G. For each $G \in D$, let

$$\dot{\text{eq}}(G) = \{\dot{\sigma} \in P^2(A) : G(\dot{\sigma}) = G\}.$$

PROPOSITION 4.1. $|\dot{\text{eq}}(G)| = 2^{c(G)}$.

Proof. Let $\dot{\sigma} = \langle \sigma_1, \sigma_2 \rangle \in \dot{\text{eq}}(G)$. Each isolated node in G is labelled with both σ_1 and σ_2. Let c be a cycle in G. If the label of the root in c is σ_1, then every node at an even distance from the root in c must be labelled σ_1 and every node at an odd distance from the root in c must be labelled σ_2. The case when the root is labelled σ_2 is symmetric, interchanging the roles of σ_1 and σ_2. Thus, for each cycle there are two possible labellings and the total number of labellings of all cycles is then $2^{c(G)}$. □

Let $D' = \{G \in D : c(G) = 0\}$.

PROPOSITION 4.2.

(1) $G(\langle \sigma_1, \sigma_2 \rangle) \in D' \Leftrightarrow \sigma_1 = \sigma_2$.

(2) $G(\langle \sigma_1, \sigma_1 \rangle) = G(\langle \sigma_2, \sigma_2 \rangle) \Leftrightarrow \sigma_1 = \sigma_2$.

(3) $|D'| = \operatorname{per}(A)$.

(4) *For all* $G \in D'$, $|\dot{\text{eq}}(G)| = 1$.

5. Analysis of the Godsil/Gutman estimator.

THEOREM 5.1. $\mathrm{E}[Y] = \operatorname{per}(A)$.

Proof. For each $\dot{\sigma} = \langle \sigma_1, \sigma_2 \rangle \in P^2(A)$, let

$$x(\dot{\sigma}) = \prod_{k=1}^{2} \left(\operatorname{sgn}(\sigma_k) \prod_{i=1}^{n} B_{i,\sigma_k(i)} \right).$$

Then, since $\det(B) = \sum_{\sigma \in P(A)} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} B_{i,\sigma(i)}$,

$$Y = (\det(B))^2 = \sum_{\dot{\sigma} \in P^2(A)} x(\dot{\sigma}) = \sum_{G \in D} \sum_{\dot{\sigma} \in \dot{\text{eq}}(G)} x(\dot{\sigma})$$

$$= \sum_{\dot{\sigma} = \langle \sigma_1, \sigma_1 \rangle \in P^2(A)} x(\dot{\sigma}) \tag{1}$$

$$+ \sum_{G \in D - D'} \sum_{\dot{\sigma} \in \dot{\text{eq}}(G)} x(\dot{\sigma}). \tag{2}$$

For each $\dot{\sigma} = \langle \sigma_1, \sigma_1 \rangle \in P^2(A)$, $x(\dot{\sigma}) = 1$ independent of the values chosen for B. Thus, part (1) is equal to $\operatorname{per}(A)$ because the number of terms in the sum is $|P(A)| = \operatorname{per}(A)$. We show that the expected value of part (2) is equal to zero as follows. Fix $G \in D - D'$ and $\dot{\sigma} = \langle \sigma_1, \sigma_2 \rangle \in \dot{\text{eq}}(G)$. We show that $\mathrm{E}[x(\dot{\sigma})] = 0$, thus showing that the expected

value of every term in part (2) is zero. Because $G \in D - D'$, G contains at least one cycle. Let $\langle i, j\rangle$ be some node in some cycle of G. Because either $\langle i, j\rangle$ is labelled with σ_1 and not with σ_2 or vice versa, $x(\dot{\sigma})$ can be written as $y(\dot{\sigma})B_{i,j}$, where $y(\dot{\sigma})$ does not contain $B_{i,j}$. Because $B_{i,j}$ is independent of $y(\dot{\sigma})$,

$$\mathrm{E}[x(\dot{\sigma})] = \mathrm{E}[y(\dot{\sigma})]\mathrm{E}[B_{i,j}].$$

Because $\mathrm{E}[B_{i,j}] = 0$, $\mathrm{E}[x(\dot{\sigma})] = 0$. □

THEOREM 5.2.

$$\frac{\mathrm{E}[Y^2]}{\mathrm{E}[Y]^2} = \frac{\sum_{G \in D} 6^{c(G)}}{\sum_{G \in D} 2^{c(G)}}.$$

Proof.

$$\mathrm{E}[Y]^2 = (\mathrm{per}(A))^2 = \sum_{\langle \sigma_1, \sigma_2\rangle \in P^2(A)} 1 = \sum_{G \in D} \sum_{\langle \sigma_1, \sigma_2\rangle \in \ddot{\mathrm{eq}}(G)} 1 = \sum_{G \in D} 2^{c(G)},$$

where the last equality is from Proposition 4.1. Let $P^4(A) = P(A) \times P(A) \times P(A) \times P(A)$. For each $\ddot{\sigma} = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4\rangle \in P^4(A)$, let

$$x(\ddot{\sigma}) = \prod_{k=1}^{4} \left(\mathrm{sgn}(\sigma_k) \prod_{i=1}^{n} B_{i,\sigma_k(i)} \right).$$

Then,

$$Y^2 = (\det(B))^4 = \sum_{\ddot{\sigma} \in P^4(A)} x(\ddot{\sigma})$$

and

$$\mathrm{E}[Y^2] = \sum_{\ddot{\sigma} \in P^4(A)} \mathrm{E}[x(\ddot{\sigma})].$$

Let $\mathrm{ODD} = \{\ddot{\sigma} \in P^4(A) :$ there is some $\langle i, j\rangle$ which is labelled with an odd number of labels from $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}\}$, and let $\mathrm{EVEN} = P^4(A) - \mathrm{ODD}$. For each $\ddot{\sigma} \in \mathrm{ODD}$, there is some $\langle i, j\rangle$ such that $x(\ddot{\sigma})$ can be written as $y(\ddot{\sigma})B_{i,j}$, where $y(\ddot{\sigma})$ does not contain $B_{i,j}$. Thus, $\mathrm{E}[x(\ddot{\sigma})] = \mathrm{E}[y(\ddot{\sigma})]\mathrm{E}[B_{i,j}]$. Because $\mathrm{E}[B_{i,j}] = 0$, $\mathrm{E}[x(\ddot{\sigma})] = 0$. For each $\ddot{\sigma} \in \mathrm{EVEN}$, for each row $i = 1, \ldots, n$, either there is a j such that $\langle i, j\rangle$ is labelled with all four of $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ or there is a j and a $j' \neq j$ such that $\langle i, j\rangle$ is labelled with exactly two of $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, and $\langle i, j'\rangle$ is labelled with the other two. Let $G(\ddot{\sigma})$ be the graph where there is a node for each $\langle i, j\rangle$ such that $\langle i, j\rangle$ is labelled with at least one of $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$. There is an edge between two distinct nodes $\langle i, j\rangle$ and $\langle i', j'\rangle$ if and only if $i = i'$ or $j = j'$. Then, $\{G(\ddot{\sigma}) : \ddot{\sigma} \in \mathrm{EVEN}\} = D$, where D is as previously defined. For each $G \in D$, define

$$\ddot{\mathrm{eq}}(G) = \{\ddot{\sigma} \in \mathrm{EVEN} : G(\ddot{\sigma}) = G\}.$$

We claim that $|\ddot{\mathrm{eq}}(G)| = 6^{c(G)}$. The reasoning is similar to that used for the proof of Proposition 4.1. Let $\ddot{\sigma} = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4\rangle \in \ddot{\mathrm{eq}}(G)$. Each isolated node in G is labelled with all four of $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$. For each cycle c in G, the root of c and every

node at an even distance from the root must be labelled with the same two elements of $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ and every node at an odd distance from the root must be labelled with the remaining two. Thus, there are $\binom{4}{2} = 6$ possible labellings of each cycle, and consequently a total of $6^{c(G)}$ labellings.

It is not hard to see that for each $\ddot{\sigma} \in \text{EVEN}$, $x(\ddot{\sigma}) = 1$ independent of the values chosen for the entries in B. This fact rests on the following two observations:

• For any two permutations σ and τ, $\text{sgn}(\sigma\tau) = \text{sgn}(\sigma) \cdot \text{sgn}(\tau)$ and $\text{sgn}(\sigma^{-1}) = \text{sgn}(\sigma)$. Hence $\prod_{k=1}^{4} \text{sgn}(\sigma_k) = \text{sgn}(\prod_{k=1}^{4} \sigma_k) = \prod_{k=2}^{4} \text{sgn}(\sigma_1^{-1}\sigma_k)$. It follows from the definition of EVEN that, if each of the three permutations $\sigma_1^{-1}\sigma_k, k = 2, 3, 4$ is written in the usual cycle notation as a product of disjoint cycles, then the cycles occurring will correspond to the cycles of even length in $G(\ddot{\sigma})$, and each cycle will occur in exactly two of the three permutations $\sigma_1^{-1}\sigma_k$. Thus, if each of these cycles is expressed in a standard way as a product of transpositions, then each transposition will occur an even number of times. It follows that $\prod_{k=1}^{3} \text{sgn}(\sigma_1^{-1}\sigma_k) = 1$, and hence $\prod_{k=1}^{4} \text{sgn}(\sigma_k) = 1$;

• Each factor in the product $\prod_{k=1}^{4} \prod_{i=1}^{n} B_{i,\sigma_k(i)}$ occurs an even number of times, and thus the product is equal to 1.

Thus, $\text{E}[Y^2] = \sum_{\ddot{\sigma} \in \text{EVEN}} 1 = \sum_{G \in D} 6^{c(G)}$. Since $\text{E}[Y]^2 = \sum_{G \in D} 2^{c(G)}$, the proof is complete. □

COROLLARY 5.3. *The Godsil/Gutman estimator yields an* (ϵ,δ)*- approximation algorithm for estimating* $\text{per}(A)$ *which runs in time* $\text{poly}(n)3^{n/2}\frac{1}{\epsilon^2}\log(\frac{1}{\delta})$.

Proof. Each evaluation of the estimator can be performed in time $\text{poly}(n)$. Also,

$$\frac{\text{E}[Y^2]}{\text{E}[Y]^2} = \frac{\sum_{G \in D} 6^{c(G)}}{\sum_{G \in D} 2^{c(G)}} \leq \max_{G \in D} 3^{c(G)} \leq 3^{n/2},$$

where the previous inequality follows because there are at most $n/2$ cycles in any $G \in D$. □

6. A better estimator and its analysis. We now present a variant of the Godsil/Gutman estimator which yields a more efficient (ϵ, δ)-approximation algorithm for $\text{per}(A)$. Let

$$w_0 = 1, \quad w_1 = -\frac{1}{2} + \frac{\sqrt{3}}{2}i, \quad w_2 = -\frac{1}{2} - \frac{\sqrt{3}}{2}i$$

be the three cube roots of unity. If $y = a + bi$ is a complex number, then $\overline{y} = a - bi$ is the complex conjugate of y.

The estimator is computed as follows.

(1) An $n \times n$ matrix B is formed from A as follows:

For all i, j, $1 \leq i, j \leq n$,

If $A_{i,j} = 0$ then $B_{i,j} \leftarrow 0$

Elseif $A_{i,j} = 1$ then randomly and independently choose $B_{i,j} \in \{w_0, w_1, w_2\}$, each choice with probability $\frac{1}{3}$.

(2) $Z \leftarrow \det(B)\overline{\det(B)}$.

This estimator can be evaluated in $\text{poly}(n)$ time.

THEOREM 6.1.

$$\mathrm{E}[Z] = \mathrm{per}(A).$$

Proof. The proof is similar to the proof of Theorem 5.1. For each $\dot{\sigma} = \langle \sigma_1, \sigma_2 \rangle \in P^2(A)$, let

$$x(\dot{\sigma}) = \mathrm{sgn}(\sigma_1)\mathrm{sgn}(\sigma_2) \prod_{i=1}^{n} B_{i,\sigma_1(i)} \overline{B}_{i,\sigma_2(i)}.$$

Then,

$$Z = \det(B)\overline{\det(B)} = \sum_{\dot{\sigma} \in P^2(A)} x(\dot{\sigma})$$

$$= \sum_{\dot{\sigma} = \langle \sigma_1, \sigma_1 \rangle \in P^2(A)} x(\dot{\sigma}) \tag{3}$$

$$+ \sum_{G \in D - D'} \sum_{\dot{\sigma} \in \dot{\mathrm{eq}}(G)} x(\dot{\sigma}). \tag{4}$$

For each $\dot{\sigma} = \langle \sigma_1, \sigma_1 \rangle \in P^2(A)$, $x(\dot{\sigma}) = 1$ independent of the values chosen for B. Thus, part (3) is equal to $\mathrm{per}(A)$. Showing that the expected value of part (4) is equal to zero is very similar to the corresponding portion of the proof in Theorem 5.1. The observation needed is again that for any $\langle i, j \rangle$, $\mathrm{E}[B_{i,j}] = 0$. □

THEOREM 6.2.

$$\frac{\mathrm{E}[Z^2]}{\mathrm{E}[Z]^2} = \frac{\sum_{G \in D} 4^{c(G)}}{\sum_{G \in D} 2^{c(G)}}.$$

Proof. The proof follows exactly the outline of the proof of Theorem 5.2. We only note the differences here.

For each $\ddot{\sigma} = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4 \rangle \in P^4(A)$, let

$$x(\ddot{\sigma}) = \left(\prod_{k=1}^{2} \mathrm{sgn}(\sigma_k) \prod_{i=1}^{n} B_{i,\sigma_k(i)} \right) \left(\prod_{k=3}^{4} \mathrm{sgn}(\sigma_k) \prod_{i=1}^{n} \overline{B}_{i,\sigma_k(i)} \right).$$

Then, $\mathrm{E}[Z^2] = \sum_{\ddot{\sigma} \in P^4(A)} \mathrm{E}[x(\ddot{\sigma})]$. From the definitions in the proof of Theorem 5.2,

$$P^4(A) = \mathrm{ODD} \cup \mathrm{EVEN} = \mathrm{ODD} \cup \bigcup_{G \in D} \ddot{eq}(G).$$

For $\ddot{\sigma} \in \mathrm{ODD}$ there is some $\langle i, j \rangle$ such that $x(\ddot{\sigma})$ can be written as $y(\ddot{\sigma})B_{i,j}$ or as $y(\ddot{\sigma})\overline{B}_{i,j}$, where in either case $y(\dot{\sigma})$ contains neither $B_{i,j}$ nor $\overline{B}_{i,j}$. Because $\mathrm{E}[B_{i,j}] = \mathrm{E}[\overline{B}_{i,j}] = 0$, $\mathrm{E}[x(\ddot{\sigma})] = 0$. For each $G \in D$ we further partition $\ddot{\mathrm{eq}}(G)$ as follows:

$$\ddot{\mathrm{eq}}'(G) = \{\ddot{\sigma} \in \ddot{\mathrm{eq}}(G) : \text{in each cycle } c \text{ in } G \text{ the root of } c \text{ is labelled with exactly one of } \{\sigma_1, \sigma_2\} \text{ and with exactly one of } \{\sigma_3, \sigma_4\}\}.$$

Because there are four possible labellings for each cycle in G, $|\ddot{e}\ddot{q}'(G)| = 4^{c(G)}$. For each $\ddot{\sigma} \in \ddot{e}\ddot{q}'(G)$, for each node $\langle i,j\rangle$ in G there are an equal number of occurrences of $B_{i,j}$ and $\overline{B}_{i,j}$ in $x(\ddot{\sigma})$. Thus, $x(\ddot{\sigma}) = 1$ independent of the values chosen for B, and $\mathrm{E}[x(\ddot{\sigma})] = 1$. For each $\ddot{\sigma} \in \ddot{e}\ddot{q}(G) - \ddot{e}\ddot{q}'(G)$, there is some node $\langle i,j\rangle$ in G such that $B_{i,j}$ occurs twice in $x(\ddot{\sigma})$ and $\overline{B}_{i,j}$ does not occur at all. Thus, $x(\ddot{\sigma})$ can be written as $y(\ddot{\sigma})B_{i,j}^2$, where $y(\ddot{\sigma})$ contains no occurrences of $B_{i,j}$ or $\overline{B}_{i,j}$. Then, $\mathrm{E}[x(\ddot{\sigma})] = \mathrm{E}[y(\ddot{\sigma})]\mathrm{E}[B_{i,j}^2]$. Because $\mathrm{E}[B_{i,j}^2] = 0$, $\mathrm{E}[x(\ddot{\sigma})] = 0$.

Putting this together, $\mathrm{E}[Z^2] = \sum_{G\in D} 4^{c(G)}$, and thus

$$\frac{\mathrm{E}[Z^2]}{\mathrm{E}[Z]^2} = \frac{\sum\limits_{G\in D} 4^{c(G)}}{\sum\limits_{G\in D} 2^{c(G)}}.$$

□

COROLLARY 6.3. *The estimator Z yields an (ϵ,δ)-approximation algorithm for estimating* $\mathrm{per}(A)$ *which runs in time* $\mathrm{poly}(n)2^{n/2}\frac{1}{\epsilon^2}\log(\frac{1}{\delta})$.

Proof. Each evaluation of the estimator can be performed in time $\mathrm{poly}(n)$. Also,

$$\frac{\mathrm{E}[Z^2]}{\mathrm{E}[Z]^2} = \frac{\sum\limits_{G\in D} 4^{c(G)}}{\sum\limits_{G\in D} 2^{c(G)}} \le \max_{G\in D} 2^{c(G)} \le 2^{n/2},$$

where the previous inequality follows because there are at most $n/2$ cycles in any $G \in D$. □

It is natural to consider a generalization of the above two algorithms, in which $B_{i,j}$ is set equal to zero when $A_{i,j} = 0$, and to a random kth root of unity whenever $A_{i,j} = 1$. The Godsil/Gutman estimator corresponds to the case $k = 2$, and the algorithm of the present section, to the case $k = 3$. Theorem 3 holds for all integers $k \ge 2$, and Theorem 4 holds for all integers $k \ge 3$. However, choosing k greater than 3 appears to give no reduction in the variance of the estimator.

7. Some special cases. We have introduced two unbiased estimators of $\mathrm{per}(A)$: the Godsil/Gutman estimator Y and a second estimator Z which refines the Godsil/Gutman technique by using cube roots of unity. We showed that

$$\frac{\mathrm{E}[Y^2]}{\mathrm{E}[Y]^2} = \frac{\sum\limits_{G\in D} 6^{c(G)}}{\sum\limits_{G\in D} 2^{c(G)}}$$

and

$$\frac{\mathrm{E}[Z^2]}{\mathrm{E}[Z]^2} = \frac{\sum\limits_{G\in D} 4^{c(G)}}{\sum\limits_{G\in D} 2^{c(G)}}.$$

We then obtained an upper bound of $3^{n/2}$ on $\mathrm{E}[Y^2]/\mathrm{E}[Y]^2$ and an upper bound of $2^{n/2}$ on $\mathrm{E}[Z^2]/\mathrm{E}[Z]^2$ using the trivial observation that, for all G, $c(G) \le \frac{n}{2}$. Although this bound seems terribly pessimistic, the following example shows that there are cases

in which it is close to the truth. Let n be even and let A be the $n \times n$ matrix such that for $i = 1, \ldots, \frac{n}{2}$, $A_{2i-1,2i-1} = A_{2i-1,2i} = A_{2i,2i-1} = A_{2i,2i} = 1$ and all other entries in A are zero. For the first algorithm the expected number of trials before there is even one trial where $Y \neq 0$ is $\Omega(2^{n/2})$, and for the second algorithm it is $\Omega((\frac{3}{2})^{n/2})$. There is a lot of room between $3^{n/2}$ and $2^{n/2}$ and between $(\frac{3}{2})^{n/2}$ and $2^{n/2}$; we are not sure which bound is closer to the worst case behavior for the respective algorithms.

Despite this bad example, we suspect that in most situations, these bounds are far too pessimistic, since $c(G)$ will "typically" be much smaller than $n/2$. As one heuristic indication of this phenomenon we note that, if $\dot{\sigma} = \langle \sigma_1, \sigma_2 \rangle$, where σ_1 and σ_2 are independent random permutations, then $c(G(\dot{\sigma}))$ will be close to $\ln(n)$ with very high probability.

In this section we analyze a concrete example in which the bounds of $3^{n/2}$ for the Godsil/Gutman algorithm and $2^{n/2}$ for its refinement are provably too pessimistic: the $n \times n$ matrix C in which every element is equal to 1. The permanent of this matrix is, of course, $n!$.

Let the random variable Y be the estimator produced by the Godsil/Gutman algorithm applied to the matrix C, and let Z be the estimator produced by the refinement given in §6 applied to the matrix C. Our main result is as follows.

THEOREM 7.1. *In the particular case of the $n \times n$ matrix C,*

$$\frac{\mathrm{E}[Y^2]}{\mathrm{E}[Y]^2} \leq \frac{(n+1)(n+2)}{2} \quad \textit{and} \quad \frac{\mathrm{E}[Z^2]}{\mathrm{E}[Z]^2} \leq n+1.$$

Proof. Since the two estimators are unbiased, $\mathrm{E}[Y] = \mathrm{E}[Z] = n!$. In order to discuss the second moments of Y and Z we require a definition: for any permutation σ of $\{1, 2, \ldots, n\}$, let $d(\sigma)$ be the number of cycles of length greater than one in the permutation σ.

We know from Theorems 5.2 and 6.2 that $\mathrm{E}[Y^2] = \sum_G 6^{c(G)}$ and $\mathrm{E}[Z^2] = \sum_G 4^{c(G)}$. We note that, for any $\dot{\sigma} = \langle \sigma_1, \sigma_2 \rangle$, $c(G(\dot{\sigma})) = d(\sigma_1^{-1}\sigma_2)$. Combining this with the fact that $|\dot{\mathrm{eq}}(G)| = 2^{c(G)}$ we find that $\sum_{G \in D} 6^{c(G)} = \sum_{\langle \sigma_1, \sigma_2 \rangle \in P^2(A)} 3^{d(\sigma_1^{-1}\sigma_2)}$. In the case of the complete graph, $P(A)$ is equal to P, the set of all permutations of $\{1, 2, \ldots, n\}$, and thus $\mathrm{E}[Y^2] = \sum_{\langle \sigma_1, \sigma_2 \rangle \in P^2} 3^{d(\sigma_1^{-1}\sigma_2)}$. Since each permutation σ can be written as $\sigma_1^{-1}\sigma_2$ in exactly $n!$ ways, the right-hand side reduces to $n! \sum_{\sigma \in P} 3^{d(\sigma)}$. Similarly, we obtain $\mathrm{E}[Z^2] = n! \sum_{\sigma \in P} 2^{d(\sigma)}$. But it follows from [17] (solution to Exercise 3.12, p. 203) that $\sum_{\sigma \in P} 3^{d(\sigma)} \leq \frac{(n+2)!}{2}$ and $\sum_{\sigma \in P} 2^{d(\sigma)} \leq (n+1)!$. Hence, $\mathrm{E}[Y^2]/\mathrm{E}[Y]^2 \leq (n+2)(n+1)/2$ and $\mathrm{E}[Z^2]/\mathrm{E}[Z]^2 \leq n+1$. □

It follows that, in the special case of the matrix C, a quadratic number of trials of the Godsil/Gutman Monte-Carlo algorithm, or a linear number of trials of our refinement of the Godsil/Gutman algorithm, are sufficient for an (ϵ, δ)-approximation.

In a preliminary version of this paper we conjectured that similar claims hold true for almost all $n \times n$ zero-one matrices. This was established in [9], where it was shown that, for almost every 0, 1 matrix, $O(n\omega(n)\epsilon^{-2})$ independent trials, using the estimator of §6, suffice to obtain an $(\epsilon, \frac{1}{4})$ approximation to the permanent. Here $\omega(n)$ is any function tending to infinity as $n \to \infty$. It was previously known that [12], combined with [7] or [19], yields a polynomial-time (ϵ, δ) approximation algorithm for the permanent of a random zero-one matrix of arbitrary density. However, Jerrum's result, based on the algorithm of §6, gives a better running time than these methods.

8. Comments, generalizations, refinements. For both algorithms presented here, once the values of B have been chosen, the value of the estimator can be computed exactly in $\mathrm{poly}(n)$ time. In the second algorithm, this requires that $\sqrt{3}$ be represented symbolically. The symbol $\sqrt{3}$ will not appear in the final answer, which is an integer.

Both Monte-Carlo algorithms can easily be modified to estimate $\mathrm{per}(A)$ when the entries in A are allowed to be arbitrary positive numbers. However, in this case there is an issue with precision; the estimators cannot be computed exactly in $\mathrm{poly}(n)$ time, although they can be closely approximated. The modification for the Godsil/Gutman estimator is to randomly choose

$$B_{i,j} \in \left\{ \sqrt{A_{i,j}}, -\sqrt{A_{i,j}} \right\},$$

each choice with probability $\frac{1}{2}$. The modification for the second estimator is to randomly choose

$$B_{i,j} \in \left\{ \sqrt{A_{i,j}}w_0, \sqrt{A_{i,j}}w_1, \sqrt{A_{i,j}}w_2 \right\},$$

each choice with probability $\frac{1}{3}$. In both cases, the expected value of the estimator is $\mathrm{per}(A)$ and the upper bounds on the number of trials to guarantee an (ϵ, δ) approximation algorithm stated in Corollaries 5.3 and 6.3 still apply, assuming that at each trial the estimator is computed exactly. The analysis would have to be modified to allow for truncation error.

Each of our Monte-Carlo algorithms consists of $O(\log(\frac{1}{\delta}))$ phases, with each phase consisting of some number N of independent trials. We required that each phase be an $(\epsilon, \frac{1}{4})$ approximation algorithm. Our analysis of the upper bound on the number of trials to guarantee this property was based on Chebyshev's inequality, and thus the analysis still holds if trials are pairwise independent. Consider running the first Monte-Carlo algorithm with a fixed ϵ and δ. As it is written it requires $\Theta(3^{\frac{n}{2}}n^2)$ random bits per phase in total, i.e., n^2 random bits per trial to randomly choose the values for B. This can be reduced to $O(n^3)$ random bits per phase using standard methods of generating pairwise independent unbiased random bits [1], [6], [18].

9. Open questions.

(1) Is there an (ϵ, δ) approximation algorithm for $\mathrm{per}(A)$ which runs in $\mathrm{poly}(n)$ time? One possible approach to solving this problem is based on the observation that the trials within a phase need only be pairwise independent, rather than completely independent. The result of each phase is $\sum_{i=1}^{\ell} Y_i/\ell$ where ℓ, the number of trials, is exponential in n. Pairwise independence permits the ℓ samples to be generated using only $O(n^3)$ random bits. Perhaps the rule for generating the samples from the random bits can be designed so that the quantity $\sum_{i=1}^{\ell} Y_i$ can be computed directly and efficiently from the random bits, without the need to calculate the result of each trial explicitly. Similar ideas have been used successfully in other contexts [1], [2], [6], [4], [14], [18].

(2) Is there a deterministic algorithm with running time $o(2^n)$, which accepts as input A and ϵ and which outputs Y such that

$$(1-\epsilon)\,\mathrm{per}(A) \leq Y \leq (1+\epsilon)\,\mathrm{per}(A)?$$

REFERENCES

[1] W. ALEXI, B. CHOR, O. GOLDREICH, AND C. P. SCHNORR, RSA *Rabin functions: certain parts are as hard as the whole*, SIAM J. Comput., 17 (1988), pp. 194–209.

[2] E. BACH, *Realistic analysis of some randomized algorithms*, in 19th Proceedings of the ACM Symposium on the Theory of Computing, 1987, pp. 453–461.

[3] A. BRODER, *How hard is it to marry at random (on the approximation of the permanent)*, in 19th Proceedings of the ACM Symposium on the Theory of Computing, 1986, pp. 50–58.

[4] J. CARTER AND M. WEGMAN, *Universal class of hash functions*, J. Comput. Systems Sci., 18 (1979), pp. 143–154.

[5] H. CHERNOFF, *A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations*, Ann. Math. Stat., 23 (1952), pp. 493–509.

[6] B. CHOR AND O. GOLDREICH, *On the power of two-points based sampling*, J. Complexity, to appear.

[7] A. M. FRIEZE, *A note on computing random permanents*, manuscript, 1989.

[8] C. D. GODSIL AND I. GUTMAN, *On the matching polynomial of a graph*, Algebraic Methods in Graph Theory, I, L. Lovász and V. T. Sós, eds., Math. Soc. János Bolyai, North-Holland, Amsterdam, 1981, pp. 241–249.

[9] M. JERRUM, *An analysis of a Monte-Carlo algorithm for estimating the permanent*, Tech. Report ECS-LFCS-91-164, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, Edinburgh, Scotland, June, 1991.

[10] M. JERRUM, L. L. VALIANT, AND U. VAZIRANI, *Random generation of combinatorial structures from a uniform distribution*, Theoret. Comput. Sci., 43 (1986), pp. 169–188.

[11] M. JERRUM AND A. SINCLAIR, *Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved*, Proceedings of the ACM Symposium on the Theory of Computing, 1988, pp. 235–243.

[12] ———, *Approximating the permanent*, Internal Report CSR-275-88, Department of Computer Science, University of Edinburgh, Edinburgh, Scotland, 1991; SIAM J. Comput., 18 (1989), pp. 1149–1178.

[13] M. JERRUM AND U. VAZIRANI, *A mildly exponential approximation algorithm for the permanent*, Internal Report ECS-LFCS-91-179, Department of Computer Science, University of Edinburgh, Scotland, 1991.

[14] H. KARLOFF AND P. RAGHAVAN, *Randomized algorithms and pseudorandom numbers*, in Proceedings of the 20th ACM Symposium on the Theory of Computing, 1988, pp. 310–321.

[15] R. KARP AND M. LUBY, *Monte-Carlo algorithms for enumeration and reliability problems*, in 24th Proceedings of the IEEE Foundation of Computer Science, 1983, pp. 56–64.

[16] R. KARP, M. LUBY, AND N. MADRAS, *Monte-Carlo approximation algorithms for enumeration problems*, J. Algorithms, 10 (1989), pp. 429–448.

[17] L. LOVÁSZ, *Combinatorial Problems and Exercises*, North-Holland, Amsterdam, 1979.

[18] M. LUBY, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 15 (1986), pp. 1036–1053.

[19] R. MOTWANI, *Expanding graphs and the average-case analysis of algorithms for matchings and related problems*, in Proceedings of the ACM Symposium on the Theory of Computing, 1989, pp. 550–561.

[20] H. RYSER, *Combinatorial Mathematics*, The Carus Mathematical Monographs, 14, the Mathematical Association of America, Washington, DC, 1963.

[21] L. VALIANT, *The complexity of computing the permanent*, Theoret. Comput. Sci., 8 (1979), pp. 189–201.