An Almost Linear Time Approximation Algorithm for the Permanent of a Random (0-1) Matrix

Martin Fürer* and Shiva Prasad Kasiviswanathan

Computer Science and Engineering, Pennsylvania State University
University Park, PA 16802
{furer, kasivisw}@cse.psu.edu

Abstract. We present a simple randomized algorithm for approximating permanents. The algorithm with inputs A, $\epsilon > 0$ produces an output X_A with $(1-\epsilon)\text{per}(A) \leq X_A \leq (1+\epsilon)\text{per}(A)$ for almost all (0-1) matrices A. For any positive constant $\epsilon > 0$, and almost all (0-1) matrices the algorithm runs in time $O(n^2\omega)$, i.e., almost linear in the size of the matrix, where $\omega = \omega(n)$ is any function satisfying $\omega(n) \to \infty$ as $n \to \infty$. This improves the previous bound of $O(n^3\omega)$ for such matrices. The estimator can also be used to estimate the size of a backtrack tree.

1 Introduction

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

$$\text{per}(A) = \sum_{\pi} \prod_{i} a(i, \pi(i)),$$

where the sum is over all permutations π of $\{1, 2, \ldots, n\}$. The permanent function was first introduced in the memoirs of Cauchy and Binet in 1812 (see [11] for a comprehensive history). The (0-1) permanent also has a simple combinatorial interpretation, $\text{per}(A)$ counts the perfect matchings in the $(n+n)$-vertex bipartite graph whose adjacency matrix is A. The permanent has important applications in physical sciences and plays a central role in many linear algebra and combinatorial enumeration problems.

Despite its syntactic similarity to the determinant, no efficient method for computing the permanent could be found for almost two centuries. This apparent paradox was solved by Valiant [16] who showed in his celebrated paper that $\text{per}(A)$ is #P-complete. The class #P is defined as $\{f : \exists$ a non deterministic polynomial time Turing Machine M such that on input x, M has exactly $f(x)$ accepting leaves$\}$. Thus, it comes as no surprise that there is no polynomial time deterministic algorithm for calculating the permanent exactly. The best known deterministic algorithm has a running time of $O(n2^n)$ [14]. Therefore, the more recent research has focussed on efficient approximation algorithms for

* Research supported in part by NSF Grant CCR-0209099

the permanent with desired performance guarantees. In this paper, we describe a simple estimator for the permanent and prove that it has an overall running time of $O(n^2\omega)$ for random (0-1) matrices from $G(n, 1/2)$, where $\omega = \omega(n)$ is any function satisfying $\omega(n) \to \infty$ as $n \to \infty$. We then generalize our technique to obtain a running time which is polynomial in the size of the input matrix for the case of $G(n, p)$. We also show how the same estimator when applied to another problem, that of estimating the size of a tree, could result in better running times for some kinds of random trees. The estimator is a randomized approximation scheme.

2 Definitions

Let Q be some function from Σ^* to the natural numbers. A fully-polynomial randomized approximation scheme (a.k.a. fpras) for Q is a randomized algorithm that accepts an input $x \in \sum^\star$ together with an accuracy parameter $\epsilon \in (0, 1]$, outputs a number X (a random variable depending on the coin tosses of the algorithm) such that

$$\Pr[(1-\epsilon)Q(x) \leq X \leq (1+\epsilon)Q(x)] \geq \frac{3}{4}$$

and runs in time polynomial in $|x|$ and ϵ^{-1}. The probability of $\frac{3}{4}$ can be boosted to $1-\delta$ for any $0 < \delta < 1$ by outputting the median of $O(\log \delta^{-1})$ independent trials [7].
Suppose we would like to estimate Q and have a probabilistic algorithm running in time polynomial in $|x|$, whose output is a random variable X such that $E[X] = Q(x)$ and $E[X^2]$ is finite. Suppose further that we can repeat this experiment as many times as we wish, and the outcomes of the successive trials are independent and identically distributed. Let X_i be the outcome of the i^{th} trial. A straightforward application of Chebychev's inequality shows that, if we conduct $O(\frac{E[X^2]}{E[X]^2}\epsilon^{-2})$ trials and take the mean, we have an fpras for Q. Together, the complexity of performing the stochastic experiment, and the ratio of $\frac{E[X^2]}{E[X]^2}$ (a.k.a. critical ratio) will determine the efficiency of the algorithm.

3 Related Work

Current research in the area of permanents has been divided into four major categories [2]. They are: elementary recursive algorithms; reductions to determinants; iterative balancing; and Markov chain Monte-Carlo methods. One of the simplest estimators of the permanent using elementary recursive algorithms was proposed by Rasmussen [13]. This estimator has a running time of $O(n^3\omega)$ for almost all (0-1) matrices. We will extend Rasmussen's idea to get a running time of $O(n^2\omega)$ for almost all (0-1) matrices. The more famous K^2L^3 [8] estimator uses reductions to determinants. This estimator, which is based on the

Godsil/Gutman estimator [4], has a running time of poly$(n)2^{n/2}$ for all (0-1) matrices. In 1995, Frieze and Jerrum [3] proved that the $\mathrm{K}^2\mathrm{L}^3$ estimator runs in time $\mathrm{O}(n\mathrm{M}(n)\omega)$ for almost all non-negative matrices, where $\mathrm{M}(n)$ is the time required to perform matrix multiplications. Recently, an fpras for computing the permanent of an arbitrary matrix with non-negative entries was proposed by Jerrum, Sinclair, Vigoda [6]. This is based on the Markov chain Monte-Carlo approach. However, due to their high exponent in the running time, i.e. $\tilde{\mathrm{O}}(n^{10})$, the algorithm is unlikely to be practical [2]. For this reason, it is still worth investigating alternative approaches. The following table summarizes the running times of various estimators of the permanent of a random (0-1) matrix.

Authors	*Year*	*Running Time*
Jerrum and Sinclair [5]	1989	$\mathrm{O}(n^{\mathrm{O}(1)})$
$\mathrm{K}^2\mathrm{L}^3$ [8], Frieze and Jerrum [3]	1993, 1995	$\mathrm{O}(n\mathrm{M}(n)\omega)$
Rasmussen [13]	1994	$\mathrm{O}(n^3\omega)$
Ours	2004	$\mathrm{O}(n^2\omega)$

Fig. 1. Performance of Various Estimators for Random (0-1) matrices of $G(n, 1/2)$

4 The Rasmussen Estimator

The Rasmussen [13] estimator is inspired by Laplace's expansion formula for the permanent:

$$\mathrm{per}(A) = \sum_{j=1}^{n} a_{1j}\mathrm{per}(A_{1j}),$$

where A_{1j} denotes the submatrix obtained from A by removing the 1^{st} row and the j^{th} column, and the permanent of the empty matrix is set to 1. The idea is similar to that of Knuth's estimator [9] for estimating the size of a backtrack tree. Let W=$\{j : a_{1j} = 1\}$ (the set of columns with 1 in the current row). The estimator X_A is defined as the product of the number of 1's in the current row with the estimator $X_{A_{1j}}$ for the remaining submatrix. The column j is chosen uniformly at random from W. The estimator is clearly unbiased ($\mathrm{E}[X] = \mathrm{per}(A)$ and $\mathrm{E}[X^2]$ is finite) and one run of the algorithm can be implemented in $\mathrm{O}(n^2)$. The benefits of such an estimator are its simplicity and wide range of applicability.

5 A Better Estimator

A closer look at the above estimator tells us that this estimator makes most of its mistakes towards the end when the matrix becomes small. This motivates us to increase the frequency of runs as we go down (here run stands for a single application of the estimator algorithm over the remaining submatrix). At every level with height $s^i (i \geq 1)$ (= branching points), we do r (= branching factor)

Random Approximator of the Permanent:

```
RAP(A,n,s,r)
if n = 0 then
      X_A = 1
else
            W = {j : a_1j = 1}
            if W = ∅ then
                  X_A = 0
            else
                  if n = s^i for some i ≥ 1 then
                        K = r
                  else
                        K = 1
                  for ℓ = 1 to K do
                        choose J(ℓ) u.a.r. from W
                        compute X_{A_{1J(ℓ)}} using RAP (A_1j, n − 1, s, r)
                  X_A = |W|(1/K Σ_{ℓ=1}^{K} X_{A_{1J(ℓ)}})
```

Fig. 2. The Algorithm

runs on the submatrix, rather than the usual one. At height 1, we have only one element. The estimator X_A of a (0-1) $n \times n$ matrix $A{=}(a_{ij})(1 \leq i, j \leq n)$ is computed as in Figure 2. We call our algorithm RAP. This idea is similar to one used by Karger and Stein [15] to obtain a faster algorithm for Min-Cut. The computation has a tree structure (Figure 3). Each path from the root to a leaf represents one run of the Rasmussen estimator. We do a bottom-up evaluation. At each branching level, we find the estimator by taking the product of the number of 1's in that row with the mean over the estimators of the previous level. This gives an unbiased estimator for permanent of any (0-1) matrix.

6 The Permanent of a Random Matrix

The two most frequently used models of random graphs are $\mathrm{G}(n,p)$ and $\mathrm{G}(n,m)$. $\mathrm{G}(n,p)$ includes each possible edge with independent probability p and $\mathrm{G}(n,m)$ assigns equal probability to all graphs with m edges (see [1] for an extensive treatment). Following [13] we use $\mathcal{A}(n)$ to represent the set of all $n \times n$ (0-1) matrices, $\mathcal{A}(n,m)$ to represent the set of all $n \times n$ (0-1) matrices with exactly m 1's, E_σ to represent the mean over the coin-tosses of the estimator, $\mathrm{E}_{\mathcal{A}}$ to represent the mean over $\mathcal{A}(n)$, and $\mathrm{E}_{\mathcal{A}(n,m)}$ to represent the mean over $\mathcal{A}(n,m)$. In this section we deal with the more widely used random graph model $G(n,1/2)$. Here at every level with height $2^i (i \geq 1)$ $(s = 2)$, we do $r = 3$ runs on the submatrix.

Theorem 1. *The running time of the algorithm* $\mathrm{RAP}(A,n,2,3)$ *is* $\mathrm{O}(n^2)$.

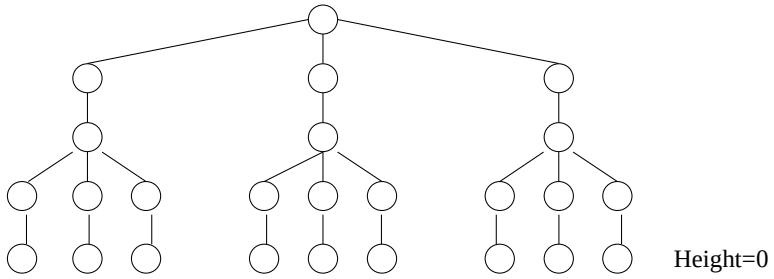

Fig. 3. Run of the Algorithm RAP(A,4,2,3)

Proof. Let $2^{i-1} < n \leq 2^i$. Then the running time: Between top and 1^{st} branching level is $< n^2 \leq 2^{2i}$. Between 1^{st} and 2^{nd} branching level is $< 3(2^{i-1})^2$. Between 2^{nd} and 3^{rd} branching level is $< 9(2^{i-2})^2$. As this forms a geometric series, the total running time is $\mathrm{O}(n^2)$. □

Our result rests on the following results of Frieze and Jerrum [3], Rasmussen [13], which we state here without proof.

Lemma 1. *(**Frieze and Jerrum [3]**) Suppose $m = m(n)$ satisfies $\frac{m^2}{n^3} \to \infty$ as $n \to \infty$, and choose A from $\mathcal{A}(n, m)$ (set of all (0-1) matrices with exactly m 1's). Then*

$$\mathrm{E}[\mathrm{per}(A)^2] = (n!)^2(\frac{m}{n^2})^{2n}\exp(-\frac{n^2}{m} + 1 + \mathrm{O}(\frac{n^3}{m^2}))$$

and

$$\frac{\mathrm{E}[\mathrm{per}(A)^2]}{\mathrm{E}[\mathrm{per}(A)]^2} = 1 + \mathrm{O}(\frac{n^3}{m^2})$$

Lemma 2. *(**Rasmussen [13]**) Let $\omega = \omega(n)$ be any function satisfying $\omega \to \infty$ as $n \to \infty$. Let $\mu(n) = \mathrm{E}_{\mathcal{A}(n)}[\mathrm{per}(A)]$ denote the mean of the permanent of a random matrix. Then for almost all (0-1) matrices A, and any unbiased estimator X_A of the permanent,*

$$\mathrm{per}(A) \geq \frac{\mu(n)}{\omega} \quad \textit{and} \quad \frac{\mathrm{E}_\sigma[X_A^2]}{\mathrm{E}_\sigma[X_A]^2} \leq \omega \frac{\mathrm{E}_\mathcal{A}[\mathrm{E}_\sigma[X_A^2]]}{\mathrm{E}_\mathcal{A}[\mathrm{E}_\sigma[X_A]]^2}$$

For Rasmussen's algorithm the critical factor $\frac{\mathrm{E}_\sigma[X_A^2]}{\mathrm{E}_\sigma[X_A]^2}$ is $\mathrm{O}(n\omega)$. The idea of repeated runs allows us to achieve an asymptotically lower bound on the critical ratio.

Our algorithm RAP (Figure 2) differs from Rasmussen's algorithm on what it does only at branching points. $\widehat{\mathrm{RAP}}$ is an auxiliary random approximator of the permanent. It's only difference from RAP is that K=1 at the start, i.e., there is no branching in the root of the computation tree even if $n = 2^j$. The random variables $\hat{X}$ and X are the outputs of $\widehat{\mathrm{RAP}}$ and RAP respectively. To model the quality of $\widehat{\mathrm{RAP}}$ and RAP, we introduce two terms $\hat{R}(h)$ and $R(h)$. $\hat{R}(h)$ models

the ratio of means of the auxiliary approximator $\widehat{\mathrm{RAP}}$, while $R(h)$ models the ratio of means of RAP until height h.

$$\hat{R}(h) = \frac{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h{}^2]]}{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h]]^2} \quad \text{and} \quad R(h) = \frac{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[X_h^2]]}{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[X_h]]^2}$$

The proofs are organized as follows: We establish the recursive relationship between $R(h)$ and $\hat{R}(h)$ in Theorems 2 and 3. With Theorems 4 and 5, we establish the claimed performance bounds. The following Theorem shows how $\hat{R}(h)$ varies as a function of $R(2^{\lfloor \lg(h-1) \rfloor})$, i.e. R at the previous branching point.

Theorem 2. *Let A_n denote a random matrix from the set $\mathcal{A}(n)$, and let $R(h)$ and $\hat{R}(h)$ be the functions defined as above. Then*

$$\hat{R}(h) = \begin{cases} 2 & \text{for } h = 1 \\ \frac{h+1}{2^{\lfloor \lg(h-1) \rfloor}+1} R(2^{\lfloor \lg(h-1) \rfloor}) & \text{for } h > 1 \end{cases}$$

Proof. The numerator of $\hat{R}(h)$ is

$$\begin{aligned} \mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h^2]] &= \sum_{m=0}^{h} \mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h^2] | M_h = m] \Pr[M_h = m] \\ &= \sum_{m=0}^{h} \mathrm{E}_{\mathcal{A}}[m^2 \mathrm{E}_{\sigma}[(X_{h-1})^2]] \Pr[M_h = m] \\ &= \mathrm{E}_{\mathcal{A}}[M_h^2] \mathrm{E}_{\mathcal{A}}[M_{h-1}^2] \dots \mathrm{E}_{\mathcal{A}}[M_{(2^{\lfloor \lg(h-1) \rfloor}+1)}^2] \, \mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[(X_{2^{\lfloor \lg(h-1) \rfloor}})^2]] \end{aligned}$$

where M_i denotes a binomial variable with parameters i and $p = \frac{1}{2}$ and thus $\sum_{m=0}^{h} m^2 \Pr[M_h = m]$ is equal to $\mathrm{E}_{\mathcal{A}}[M_h^2]$. The denominator of $\hat{R}(h)$ is

$$\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h]]^2 = \mathrm{E}[\mathrm{per}(\mathrm{A}_h)]^2 = \left(\frac{h!}{2^h}\right)^2 = \prod_{i=1}^{h} \mathrm{E}[M_i]^2$$

$$\begin{aligned} \hat{R}(h) &= \frac{\mathrm{E}[M_h^2]}{\mathrm{E}[M_h]^2} \frac{\mathrm{E}[M_{h-1}^2]}{\mathrm{E}[M_{h-1}]^2} \cdots \frac{\mathrm{E}[M_{(2^{\lfloor \lg(h-1) \rfloor}+1)}^2]}{\mathrm{E}[M_{(2^{\lfloor \lg(h-1) \rfloor}+1)}]^2} \frac{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[(X_{2^{\lfloor \lg(h-1) \rfloor}})^2]]}{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[X_{2^{\lfloor \lg(h-1) \rfloor}}]]^2} \\ &= \frac{4h(h+1)}{4h^2} \cdots \frac{4(2^{\lfloor \lg(h-1) \rfloor}+1)(2^{\lfloor \lg(h-1) \rfloor}+2)}{4(2^{\lfloor \lg(h-1) \rfloor}+1)^2} \frac{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[(X_{2^{\lfloor \lg(h-1) \rfloor}})^2]]}{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[X_{2^{\lfloor \lg(h-1) \rfloor}}]]^2} \\ &= \frac{h+1}{2^{\lfloor \lg(h-1) \rfloor}+1} R(2^{\lfloor \lg(h-1) \rfloor}) \end{aligned}$$

□

Before venturing into showing the dependence of $R(h)$ on $\hat{R}(h)$ we establish a few important lemmas. The following lemma estimates a bound of higher moment of the binomial distribution. A lot of similar results have appeared in literature (see [10] for more details).

Lemma 3. *For $n \geq 0$ we have*

$$\frac{1}{2^{n^2}} \sum_{j=0}^{n^2} \binom{n^2}{j} j^{2n} = \mathrm{O}\left(\left(\frac{n^2}{2}\right)^{2n}\right)$$

Proof. The term $\binom{n^2}{j} j^{2n}$ has its maximum value around $j = \frac{n^2}{2} + n$. The idea is to split the summation into three parts around this point and to bound each part. □

In the following lemma we try to establish a relationship which is similar to the one established in Lemma 1. Note that Lemma 1 holds for the random graph model $G(n, m)$, but we are interested in the more commonly used $G(n, p)$ model.

Lemma 4. *Let A_n be a matrix from the set $\mathcal{A}(n)$. Then for some constant c independent of n*

$$\frac{\mathrm{E}[(\mathrm{per}(A_n))^2]}{\mathrm{E}[\mathrm{per}(A_n)]^2} \leq c$$

Proof. We can split the numerator by conditioning it on the number of 1's (M) in the matrix as

$$\Pr[M < \frac{n^2}{4}]\mathrm{E}[(\mathrm{per}(A_n))^2 | M < \frac{n^2}{4}] + \Pr[M \geq \frac{n^2}{4}]\mathrm{E}[(\mathrm{per}(A_n))^2 | M \geq \frac{n^2}{4}]$$

By Chernoff's Bound, we have $\Pr[M < \frac{n^2}{4}] < \exp(\frac{-n^2}{16})$. So for the numerator we have

$$\Pr[M < \frac{n^2}{4}] < \Pr[M \geq \frac{n^2}{4}] \text{ and } \mathrm{E}[(\mathrm{per}(A_n))^2 | M < \frac{n^2}{4}] \leq \mathrm{E}[(\mathrm{per}(A_n))^2 | M \geq \frac{n^2}{4}]$$

The second inequality follows as adding more 1's can't reduce the value of the permanent. This implies

$$\begin{aligned}\mathrm{E}[\mathrm{per}(A_n)^2] &< 2\Pr[M \geq \frac{n^2}{4}]\mathrm{E}[(\mathrm{per}(A_n))^2 | M \geq \frac{n^2}{4}] \\ &\leq 2\mathrm{E}[(\mathrm{per}(A_n))^2 | M \geq \frac{n^2}{4}] = 2 \sum_{j=\frac{n^2}{4}}^{n^2} \mathrm{E}[(\mathrm{per}(A_n))^2 | M = j] \Pr[M = j]\end{aligned}$$

Substituting for the probability of having j 1's and using Lemma 1 for the value of $\mathrm{E}[(\mathrm{per}(A_n))^2]$, we obtain

$$\begin{aligned}\frac{\mathrm{E}[(\mathrm{per}(A_n))^2]}{\mathrm{E}[\mathrm{per}(A_n)]^2} &\leq \frac{(2^{2n+1})}{(n!)^2} \sum_{j=\frac{n^2}{4}}^{n^2} (n!)^2 (\frac{j}{n^2})^{2n} \exp \underbrace{(-\frac{n^2}{j} + 1 + \mathrm{O}(\frac{n^3}{j^2}))}_{\leq c'} \binom{n^2}{j} \left(\frac{1}{2}\right)^{n^2} \\ &< \frac{2^{2n+1}\exp(c')}{n^{4n}} \sum_{j=0}^{n^2} j^{2n} \binom{n^2}{j} 2^{-n^2}\end{aligned}$$

However from Lemma 3, we know that $\sum_{j=0}^{n^2} j^{2n}\binom{n^2}{j} = \mathrm{O}(\left(\frac{n^2}{2}\right)^{2n} 2^{n^2})$. Substituting this result we finish the proof of the Lemma 4. □

We are now prepared to establish the dependence of $R(h)$ on $\hat{R}(h)$. As mentioned earlier $R(h)$ and $\hat{R}(h)$ vary only at the branching points.

Theorem 3. *Let $R(h)$ and $\hat{R}(h)$ be the functions defined as above. Then*

$$R(h) \leq \begin{cases} \frac{\hat{R}(h)}{K} + \frac{(K-1)c}{K} & \text{if } h \text{ is a branching point} \\ \hat{R}(h) & \text{otherwise} \end{cases}$$

where c is an upper bound on $\frac{\mathrm{E}_{\mathcal{A}}[\mathrm{per}(A)^2]}{\mathrm{E}_{\mathcal{A}}[\mathrm{per}(A)]^2}$.

Proof. At all levels other than the branching levels, we have $K = 1$ implying $R(h)$=$\hat{R}(h)$. However, at the branching levels we have:

$$R(h) = \frac{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[X_h^2]]}{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[X_h]]^2} = \frac{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[(\frac{1}{K}\sum_{i=1}^{K}\hat{X}_h^{(i)})^2]]}{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\frac{1}{K}\sum_{i=1}^{K}\hat{X}_h^{(i)}]]^2}.$$

Furthermore since the outcomes of the successive trials $\hat{X}_h^{(i)}$ are independent and identically distributed

$$\mathrm{E}_{\sigma}[(\frac{1}{K}\sum_{i=1}^{K}\hat{X}_h^{(i)})^2] = \frac{\mathrm{E}_{\sigma}[\hat{X}_h^2] + (K-1)\mathrm{E}_{\sigma}[\hat{X}_h]^2}{K}$$

$$R(h) = \frac{\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h^2]]}{K\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h]]^2} + \frac{(K-1)\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h]^2]}{K\mathrm{E}_{\mathcal{A}}[\mathrm{E}_{\sigma}[\hat{X}_h]]^2} = \frac{\hat{R}(h)}{K} + \frac{(K-1)\mathrm{E}_{\mathcal{A}}(\mathrm{per}(A_h)^2]}{K\mathrm{E}_{\mathcal{A}}[\mathrm{per}(A_h)]^2}$$

Using Lemma 4 we complete the proof. □

Substituting $K = 3$, we get $R(h) = \frac{\hat{R}(h)+2\mathrm{O}(1)}{3}$ at the branching points. In the following two Theorems we show that both $R(h)$ and $\hat{R}(h)$ are bound by a constant implying that the critical ratio is $\mathrm{O}(\omega)$ from Lemma 2.

Theorem 4. *Let $\hat{R}(h)$ and $R(h)$ be the functions defined as above. Then for all $h \leq n$ and for c from Theorem 3*

$$\hat{R}(h) \leq \frac{2c(h+1)}{2^{\lfloor \lg(h-1) \rfloor}+1} \quad \text{and} \quad R(h) \leq \begin{cases} 2c & h = \text{branching point} \\ \frac{2c(h+1)}{2^{\lfloor \lg(h-1) \rfloor}+1} & \text{otherwise} \end{cases}$$

Proof. We use induction over h. We know that $c \geq 1$. For $h = 1$, $\hat{R}(1) = 2 \leq 2c$ and $R(1) = 2 \leq 2c$. Assuming the statement is true for h, we prove it for $h + 1$. There are two cases:
Case 1: $h + 1$ is a branching point. From Theorem 2 we get

$$\hat{R}(h+1) = \frac{h+2}{2^{\lfloor \lg(h) \rfloor}+1} R(2^{\lfloor \lg(h) \rfloor}) \leq \frac{2c(h+2)}{2^{\lfloor \lg(h) \rfloor}+1}$$

(where $R(2^{\lfloor \lg(h) \rfloor}) \leq 2c$ is by induction hypothesis). From Theorem 3, we also get $R(h+1) \leq 2c$.
Case 2: $h+1$ is not a branching point.

$$\hat{R}(h+1) = R(2^{\lfloor \lg(h) \rfloor}) \frac{h+2}{2^{\lfloor \lg(h) \rfloor}+1} = \frac{2c(h+2)}{2^{\lfloor \lg(h) \rfloor}+1}$$

From Theorem 3, $R(h+1) = \hat{R}(h+1)$. □

Theorem 5. *Let $\omega = \omega(n)$ be any function satisfying $\omega \to \infty$ as $n \to \infty$. Then for almost all (0-1) matrices A, we have,*

$$\frac{\mathrm{E}_\sigma[X_A^2]}{\mathrm{E}_\sigma[X_A]^2} = \mathrm{O}(\omega)$$

Proof. The factor of $\frac{2h}{2^{\lfloor \lg(h-1) \rfloor + 1}+1}$ is less than 2. Hence, both $R(h)$ and $\hat{R}(h)$ are O(1) (Theorem 4). Using Lemma 2, we bound the critical factor by $\mathrm{O}(\omega)$. □

Each run of the estimator presented here can be performed using $\mathrm{O}(n^2)$ operations, and the number of times we need to repeat the experiment to obtain an fpras is $\mathrm{O}(\omega)$. Thus, we obtain a total running time of $\mathrm{O}(n^2\omega)$ for almost all (0-1) matrices.

7 Arbitrary Probabilities

Till now we have been dealing with the interesting random graph model $\mathrm{G}(n, \frac{1}{2})$. In this section we investigate the performance of the proposed estimator for the case $\mathrm{G}(n,p)$ for arbitrary edge probability p. We propose three different choices of the branching point parameter s (Figure 4) to RAP depending on p. We always branch by a factor of 2 at powers of s (branching point) rounded up to the next integer. One can pretend the powers of s are integers because

1. With rounding up, one gets a result that is not worse (compared to rounding down).
2. The extra cost (of rounding up compared to rounding down) is negligible. It can trivially be bound by a factor of 2.

As before, together the complexity of performing the stochastic experiment, and the ratio of $\frac{\mathrm{E}[X^2]}{\mathrm{E}[X]^2}$ will determine the efficiency of the algorithm. Results are summarized in Figure 4.

7.1 Probability p > $\frac{1}{3}$

Here, the major contribution to the complexity is from the work we do at the top of the computation tree, handling the first $n/2$ rows costs $\mathrm{O}(n^2)$. In the remaining part, the time spent between two consecutive branching levels decrease as geometric series, giving us total running time of $\mathrm{O}(n^2)$. Also as in Section 6, we can show that $R(n) = \mathrm{O}(1)$.

7.2 Probability p = $\frac{1}{3}$

Here we do $O(n^2)$ work between any two branching points, since we have $O(\lg n)$ such branching levels, the complexity of a single run of the experiment is $O(n^2 \lg n)$.

Theorem 6. *Let $h = s^k$ and $h' = s^{k+1}$ be two consecutive branching points. Let 2 be the branching factor. Then*

$$R(h') = \frac{R(h)s^{(\frac{1}{p}-1)} + c}{2}$$

where c is the constant from Theorem 3

Proof. In this probability space where each entry is chosen to be 1 with probability p

$$\frac{\mathrm{E}[M_h^2]}{\mathrm{E}[M_h]^2} = \left(1 + \frac{1-p}{hp}\right)$$

Hence, between h and h' the ratio $\hat{R}$ grows by

$$\prod_{j=s^k+1}^{s^{k+1}} \left(1 + \frac{1-p}{jp}\right) = \exp\left(\ln \prod_{j=s^k+1}^{s^{k+1}} \left(1 + \frac{1-p}{jp}\right)\right) \leq \exp\left(\sum_{j=s^k+1}^{s^{k+1}} \frac{1-p}{jp}\right)$$
$$\leq \exp\left(\frac{1-p}{p} \int_{s^k}^{s^{k+1}} \frac{dx}{x}\right) = s^{(\frac{1}{p}-1)}$$

As in Theorem 2, we can show for this probability space $\hat{R}(h') = s^{(\frac{1}{p}-1)}R(h)$. By using these values in Theorem 3 we complete the proof. □

From Theorem 6, we can see that for $p = 1/3$ the $R(h)$ increases by $\frac{c}{2}$ between two consecutive branching points. Since there are $O(\lg n)$ such branching points, $R(n) = O(\lg n)$.

7.3 Probability p < $\frac{1}{3}$

Here the major contribution to the complexity of the experiment is from the work we do at the leaves which is of order $O(2^{\frac{\lg n}{\lg s}})$ =$O(n^{\frac{1}{\lg s}})$. Again by application of Theorem 6, we can show $R(n)$ is $O\left(n^{\frac{1}{p}-1-\lg s}\right)$.

8 Permanent of Matrices with Arbitrary Entries

Let $S = \{e_1, e_2, \ldots, e_n\}$ be a large set, with element e_i having weight w_i. One could again use the idea (Figure 5) of randomized selection to obtain an estimate of the weight $W = \sum_{i=1}^{n} w_i$ of S. The estimator can easily be shown to be unbiased. This idea can be used to extend our algorithm to deal with an arbitrary matrix A with non-negative entries. When working with some row r containing some vector v of entries. Choose entry a_{rj} with probability $p_v(j)$ and output $a_{rj}/p_v(j)$, where one reasonable choice of probabilities is $p_v(j) = (\sum_i a_{ri})^{-1} a_{rj}$

Probability	*B.P. Selector s*	*Single Run*	*Critical Ratio*	*Total Running Time*
p $> \frac{1}{3}$	$\sqrt{2} < s < 2^{\frac{1}{(p^{-1}-1)}}$	$O(n^2)$	$O(\omega(n))$	$O(n^2\omega(n))$
p $= \frac{1}{3}$	$\sqrt{2}$	$O(n^2 \lg n)$	$O(\lg(n)\omega(n))$	$O(n^2 \lg^2(n)\omega(n))$
p $< \frac{1}{3}$	$2^{\frac{1}{(p^{-1}-1)}} < s < \sqrt{2}$	$O(n^{\frac{1}{\lg s}})$	$O(n^{\frac{1}{p}-1-\frac{1}{\lg s}}\omega(n))$	$O(n^{\frac{1}{p}-1}\omega(n))$

Fig. 4. Performance of the Estimator for different Probabilities

Algorithm to estimate the weight W of S: `Assume` $\sum_{i=1}^{n} p_i \leq 1$
`Let` $q_i = \sum_{j=1}^{i} p_j$
`Select` y `uniformly at random from` $[0,1)$. `If` $q_{i-1} \leq y < q_i$ `then pick` e_i `and output` $X = \frac{w_i}{p_i}$, `else` $(q_n \leq y < 1)$ `don't pick anything and output` $X = 0$.

Fig. 5. Estimator of the weight of S

9 Estimating the size of Tree

One of the chief difficulties involved with the backtracking technique for combinatorial problems has been the inability to predict efficiency of the algorithm. Knuth [9] was the first to present a reasonable estimator for this problem and it was later enhanced by Purdom [12]. Knuth's idea is to estimate the size of a backtrack tree by repeatedly following random paths from the root. We could also apply our method to construct an unbiased estimator for determining the size of backtrack trees.
We conjecture that for certain classes of trees our estimator performs better than Knuth's estimator. One example where we perform better is a random tree model where for every node at depth d we toss $h - d$ coins to generate at most $h - d$ children. This results in height bounded random tree with degree of nodes strictly decreasing as we go down. One could easily see that such a restricted random tree model is essentially what we encounter with permanents, and in previous sections we have shown that our estimator outperforms Knuth's estimator (Knuth's estimator works similar to Rasmussen's estimator).

10 Concluding Remarks

We have presented a very simple, randomized algorithm for approximating the permanent. We have also shown that for almost all matrices the estimator runs in time almost linear in the number of entries. This is the fastest known algorithm for approximating permanents of random (0-1) matrices. To do better than $O(n^2)$, one could think of an estimator that inspects only a fraction of elements in a given row. Indeed, such unbiased estimators can easily be constructed by estimating the number of ones in a row by sampling. However, on the flip side, such a sublinear estimator may have a much higher variance.
We envisage such a scheme to be part of larger general framework which can be

used to solve similar combinatorial problems. We already know that a similar schema gives a good algorithm for finding a Min-Cut [15]. Also in the mean time the same schema has been successfully extended to count matchings in general graphs by the authors.

Acknowledgements

The authors would like to thank the referees for pointing out similarity of our method with [15] and for providing valuable comments on the results.

References

1. B. Bollobás. *Random Graphs*. Academic Press, London, England, 1985.
2. S. Chien, L. Rasmussen, and A. Sinclair. Clifford algebras and approximating the permanent. *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 222–231, 2002.
3. A. Frieze and M. Jerrum. An analysis of a Monte-Carlo algorithm for approximating the permanent. *Combinatorica*, pages 67–83, 1995.
4. C. Godsil and I. Gutman. On the matching polynomial of a graph. *Algebraic Methods in Graph Theory*, pages 241–249, 1981.
5. M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal of Computing*, 18:1149–1178, 1989.
6. 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 ACM*, 51(4), 2004.
7. M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.
8. N. Karmarkar, R. Karp, R. Lipton, L. Lovsáz, and M. Luby. A Monte-Carlo algorithm for estimating the permanent. *SIAM Journal of Computing*, 22:284–293, 1993.
9. D. E. Knuth. Estimating the efficiency of backtrack programs. *Mathematics of Computation*, 29:121–136, 1974.
10. V. D. la Pena and E. Giné. *Decoupling, from Dependence to Independence*. Springer Verlag, New York, 1999.
11. H. Minc. *Permanents, Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company, 1982.
12. P. W. Purdom. Tree size by partial backtracking. *SIAM Journal on Computing*, 7(4):481–491, 1978.
13. L. Rasmussen. Approximating the permanent:A simple approach. *Random Structures and Algorithms*, 5:349–361, 1994.
14. H. Ryser. *Combinatorial Mathematics, The Carus Mathematical Monographs*. The Mathematical Association of America, Washington DC, 1963.
15. C. Stein and D. R. Karger. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, 1996.
16. L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.