

## A Mildly Exponential Approximation Algorithm for the Permanent

by

Mark Jerrum  
Umesh Vazirani

A Mildly Exponential Approximation Algorithm for the Permanent

LFCS Report Series

ECS-LFCS-91-179

LFCS

October 1991

Department of Computer Science  
University of Edinburgh  
The King's Buildings  
Edinburgh EH9 3JZ

Copyright © 1991, LFCS

**Copyright © 1991, Laboratory for Foundations of Computer Science,  
University of Edinburgh. All rights reserved.**

**Reproduction of all or part of this work  
is permitted for educational or research use  
on condition that this copyright notice is  
included in any copy.**

# A Mildly Exponential Approximation Algorithm for the Permanent

Mark Jerrum<sup>†</sup>

*Department of Computer Science, University of Edinburgh  
The King's Buildings, Edinburgh EH9 3JZ, UK*

mrj@dcs.ed.ac.uk

and

Umesh Vazirani<sup>‡</sup>

*Department of Computer Science, University of California  
Berkeley, CA 94720, USA*

vazirani@cs.berkeley.edu

October 1991

**ABSTRACT** A new approximation algorithm for the permanent of an  $n \times n$  0,1-matrix is presented. The algorithm is shown to have worst-case time complexity  $\exp(O(n^{1/2} \log^2 n))$ . Asymptotically, this represents a considerable improvement over the best existing algorithm, which has worst-case time complexity of the form  $e^{\Theta(n)}$ .

---

<sup>†</sup> Supported by SERC grant GR/F 90363; work done in part while visiting DIMACS (Center for Discrete Mathematics and Computer Science).

<sup>‡</sup> Supported by an NSF PYI grant, with matching equipment grant from the AT&T Foundation; work done in part while visiting DIMACS.

# 1. Summary

The *permanent* of an  $n \times n$  matrix  $A = (a_{ij} : 0 \leq i, j \leq n-1)$  is defined by

$$\text{per}(A) = \sum_{\pi} \prod_{i=0}^{n-1} a_{i, \pi(i)},$$

where the sum is over all permutations  $\pi$  of  $[n] = \{0, \dots, n-1\}$ . In this paper,  $A$  will always be a 0,1-matrix, in which case the permanent of  $A$  has a simple combinatorial interpretation: namely,  $\text{per}(A)$  is equal to the number of perfect matchings (1-factors) in the bipartite graph  $G = (V, V', E)$ , where  $V = V' = [n]$ , and  $(i, j') \in E$  iff  $a_{ij} = 1$ . (Primes will be used consistently to distinguish the blocks of the bipartition of  $G$ .) The permanent has been the object of extensive study, since first appearing in 1812 in the work of Cauchy and Binet. Minc [9] has provided an excellent survey of the results of these investigations.

Despite considerable effort, and in contrast with the syntactically very similar determinant, no efficient procedure for computing the permanent is known. Valiant [11] provided convincing evidence for the inherent intractability of the permanent, by demonstrating that it is complete for the class  $\#P$  of enumeration problems, and thus as hard as counting the number of satisfying assignments to a CNF formula, or the number of accepting computations of a polynomial-time-bounded nondeterministic Turing machine. In the light of this result, we should not expect to find a polynomial-time algorithm for the permanent of a 0,1-matrix. Indeed, the fastest known (exact) algorithms for the permanent have time complexity  $\Theta(n2^n)$ ; see, for example, the method of Ryser [9, p. 122].

Recently, Karmarkar, Karp, Lipton, Lovász, and Luby [6] showed that a lower time complexity can be achieved, provided we are content to settle for an algorithm that provides only an *approximation* to  $\text{per}(A)$  within specified relative error. Their algorithm is an example of a ‘randomised approximation scheme’ [7]. Let  $f$  be a function from input strings to the natural numbers. A *randomised approximation scheme* for  $f$  is a probabilistic algorithm that takes as input a string  $x$  and a real number  $0 < \epsilon < 1$ , and produces as output a number  $Y$  (a random variable) such that  $Y$  approximates  $f(x)$  within ratio  $1 + \epsilon$  with high probability.<sup>†</sup> For definiteness we take the phrase “with high probability” to mean with probability at least  $\frac{3}{4}$ . The success probability may be boosted to  $1 - \delta$  for any desired  $\delta > 0$  by running the algorithm  $O(\log \delta^{-1})$  times and taking the median of the results [5, Lemma 6.1].

---

<sup>†</sup> For non-negative real numbers  $a, \hat{a}, \epsilon$ , we say that  $\hat{a}$  *approximates  $a$  within ratio  $1 + \epsilon$*  if  $a(1 + \epsilon)^{-1} \leq \hat{a} \leq a(1 + \epsilon)$ .

The randomised approximation scheme for  $\text{per}(A)$  proposed by Karmarkar et al. has time complexity  $\epsilon^{-2} \text{poly}(n) 2^{n/2}$ . It is a classical Monte Carlo algorithm which performs a sequence of trials using an unbiased estimator for the permanent; this estimator is closely modelled on one given earlier by Godsil and Gutman. The worst-case time complexity of this algorithm is the lowest of any existing approximation scheme for the permanent; nevertheless it shares with the exact algorithm of Ryser the general form  $e^{\Theta(n)}$ .

In this paper we use different techniques to design a randomised approximation scheme for  $\text{per}(A)$  that has time complexity  $\epsilon^{-2} \exp(O(n^{1/2} \log^2 n))$ . The new algorithm is easily sketched at an informal level, though the details need careful working out. The approach is one of ‘divide-and-conquer’, and can best be described using the graph-theoretic formulation introduced earlier. Let  $G$  be a bipartite graph, and suppose that we wish to estimate the number  $N(G)$  of perfect matchings in  $G$ . The progress of the algorithm is controlled by the expansion properties of  $G$ . If  $G$  has sufficiently large expansion ratio then an existing approximation scheme of Jerrum and Sinclair [4] is used to estimate  $N(G)$  directly with acceptable speed. Otherwise a set of vertices  $A \subset V$  is found that has the property that  $A$  is adjacent to rather few vertices in  $V'$ . The set  $A$  forms a ‘constriction’ in  $G$ , which allows an economical decomposition of  $G$  into subproblems that may be solved recursively. The full description and analysis of this approximation scheme form the main body of this note.

Given our current (lack of) knowledge, it is entirely possible that there exists a randomised approximation scheme for  $\text{per}(A)$  that has time complexity  $\text{poly}(n, \epsilon^{-1})$ . (Such an approximation scheme is said to be *fully polynomial*.) Thus, although the approximation scheme presented here is asymptotically much faster than any previously proposed, it may still be very far from optimal. The question of whether there exists a fully-polynomial (randomised) approximation scheme for  $\text{per}(A)$  presents an intriguing challenge. Optimists may take heart from the fact that such algorithms are known for some restricted classes of 0,1-matrices [4, 3]; these classes have the property that the proportion of  $n \times n$  matrices lying within the class tends to 1 as  $n$  tends to infinity.

## 2. The algorithm and its analysis

Let  $G$  be a bipartite graph on vertex set  $V + V'$  and edge set  $E$ , where  $V = V' = [n] = \{0, 1, \dots, n-1\}$ . We shall use primes to distinguish vertices in  $V'$  from those in  $V$ ; thus each vertex  $v$  in  $V$  has a counterpart  $v'$  in  $V'$ , and each subset  $U$  of  $V$  has a counterpart  $U' = \{u' \in V' : u \in U\}$  in  $V'$ . Denote by  $G^*$  the reversal of  $G$ , i.e., the graph with vertex set  $V + V'$  and edge set  $E^* = \{(u, v') \in V \times V' : (v, u') \in E\}$ . For each subset  $A \subseteq V$ , let  $\Gamma_G(A)$  denote the set of all vertices in  $V'$  that are adjacent to some vertex in  $A$ . Suppose  $\alpha$  is in the range  $0 < \alpha \leq 1$ . We say that  $G$  is an  $\alpha$ -*expander* if  $|\Gamma_G(A)| \geq (1 + \alpha)|A|$  for every  $A \subseteq V$  with  $|A| \leq \frac{1}{2}n$ . Let  $N(G)$

denote the number of perfect matchings in  $G$  and  $N^-(G)$  the number of *near-perfect* matchings in  $G$ , i.e., matchings that leave precisely two vertices uncovered.

**Lemma 1.** If  $G$  and  $G^*$  are  $\alpha$ -expanders then  $N^-(G)/N(G) = \exp(O(\alpha^{-1} \log^2 n))$ .

We postpone the proof of Lemma 1 to allow an investigation of its consequences.

**Corollary 2.** There is a randomised approximation scheme MARKOVSIM that meets the following specification:

- (i) the input to MARKOVSIM is a triple  $(G, \alpha, \epsilon)$ , where  $G$  is a bipartite graph,  $\alpha$  is a number which is to be interpreted as a guarantee that  $G$  and  $G^*$  are both  $\alpha$ -expanders, and  $0 < \epsilon < 1$  controls the accuracy of the result;
- (ii) the output is a random variable that with probability at least  $\frac{3}{4}$  approximates  $N(G)$  within ratio  $1 + \epsilon$ ;
- (iii) the execution time of MARKOVSIM is  $\epsilon^{-2} \exp(O(\alpha^{-1} \log^2 n))$ .

**Proof.** The approximation scheme in question is presented as Figure 2 of [4], which should be considered in conjunction with the discussion following Corollary 5.2, and the statement and proof of Theorem 5.3. (The quantity  $q(n)$  that appears in the figure is an upper bound on the ratio  $N^-(G)/N(G)$ .) The algorithm works by simulating a Markov chain whose states are matchings in  $G$ ; hence we dub the procedure here “MARKOVSIM”. (The notion of basing an approximation scheme for the permanent on such a Markov chain was first proposed by Broder [1], but two years were to elapse before a correct analysis was completed [10, 4].) The procedure call “ $\mathcal{G}(G(c), \xi)$ ” which appears in MARKOVSIM invokes a simulation of the Markov chain; this simulation runs until the chain is near to its equilibrium distribution, at which point the current state (matching) is returned as a sample. (The parameter  $\xi$  determines how close the Markov chain must be to equilibrium before the sample is taken.) A single call to the simulation procedure has time complexity  $\log(\epsilon^{-1}) \text{poly}(n, \rho)$ , where  $\rho$  is an upper bound on the ratio  $N^-(G)/N(G)$ ; and the total number of calls to the procedure is bounded by  $\epsilon^{-2} \text{poly}(n)$ . Since these procedure calls dominate the time complexity of MARKOVSIM, the overall time complexity is bounded by  $\epsilon^{-2} \log(\epsilon^{-1}) \text{poly}(n, \rho)$ . Observe that there is no point in taking  $\epsilon^{-1}$  to be greater than  $3n!$ , for this already guarantees an *exact* evaluation of  $N(G)$ ; thus, without loss of generality, we may assume  $\log \epsilon^{-1} = O(n \log n)$ . Further observe that  $\rho \geq N^-(G)/N(G) \geq n$ . These two observations allow us to simplify the expression of the time complexity of MARKOVSIM to  $\epsilon^{-2} \text{poly}(\rho)$ . The corollary follows from Lemma 1, setting  $\rho = \exp(O(\alpha^{-1} \log^2 n))$ .  $\square$

**Proof of Lemma 1.** Let  $M$  be any near-perfect matching in  $G$ . We demonstrate that  $M$  can be transformed to a perfect matching by augmentation along a short alternating path. It follows easily that the number of near-perfect matchings cannot exceed the number of perfect matchings by a large factor.

Define an *alternating path* (in  $G$ ) to be a path that is composed of edges that are alternately elements of  $M$  and  $E - M$ . (The first and last edges of the path may or may not be elements of  $M$ .) Let  $s \in V$  and  $t' \in V'$  be the vertices left uncovered by  $M$ . We shall show that  $s$  is joined to  $t'$  via an alternating path of length at most  $4k + 1$ , where  $k$  is that smallest integer such that  $(1 + \alpha)^k > \frac{1}{2}n$ ; observe that  $k = O(\alpha^{-1} \log n)$ . Let  $A_i$ , for  $0 \leq i \leq k$ , be the set of vertices in  $V$  that are reachable from  $s$  via an alternating path of length at most  $2i$ . The expansion property of  $G$  entails  $|A_{i+1}| \geq (1 + \alpha)|A_i|$  provided  $|A_i| \leq \frac{1}{2}n$ . Thus  $|A_k| \geq \min\{(1 + \alpha)^k, \frac{1}{2}n\} = \frac{1}{2}n$ . Similarly, let  $B'_i$ , for  $0 \leq i \leq k$  be the set of vertices in  $V'$  that are reachable from  $t'$  via an alternating path of length at most  $2i$ . The expansion property of  $G^*$  entails  $|B'_k| \geq \frac{1}{2}n$ . Now there must be an edge joining  $A_k$  to  $B'_k$ , and hence an alternating path from  $s$  to  $t'$  of length at most  $4k + 1$ . (If the edge  $e$  joining  $A_k$  to  $B'_k$  happens to be an element of  $M$ , we apparently do not obtain an alternating path; note, however, that the edge  $e$  must then occur three times in succession, and can be collapsed to a single occurrence.)

We have demonstrated that every near-perfect matching  $M$  may be associated with a perfect matching  $\overline{M}$  by augmentation along an alternating path of length  $O(\alpha^{-1} \log n)$ . This process assigns at most  $\exp(O(\alpha^{-1} \log^2 n))$  near-perfect matchings to any given perfect matching. (To recover  $M$  from  $\overline{M}$  we follow an alternating path of length at most  $4k + 1$ , the selection of which involves a sequence of at most  $2k + 1$  choices from among at most  $n$  possibilities.) Thus  $N^-(G)/N(G) = \exp(O(\alpha^{-1} \log^2 n))$ , as claimed.  $\square$

Suppose the bipartite graph  $G$  contains a perfect matching, say  $M$ . It is clear that  $|\Gamma_G(A)| \geq |A|$  for every set  $A \subseteq V$ ; let us call  $A$  *tight* [8] if equality holds. Note that the collection of all tight sets in  $V$  is closed under union and intersection; in particular for each  $v \in V$  there is a *smallest* tight set containing  $v$ , which we shall denote by  $\Delta_G(v)$ . Note that  $\Delta_G(v)$  is the set of all vertices in  $V$  that can be reached by an even-length alternating path in  $G$  which starts with an edge not in  $M$ ; thus the sets  $\Delta_G(v)$  are easy to compute in polynomial time. The procedure TESTEXPANSION presented in Figure 1 uses the idea of tight sets to compute (approximately) the expansion ratio of the input graph  $G$ . If the expansion ratio is not too small, Corollary 2 can be applied to estimate  $N(G)$  directly. Otherwise, TESTEXPANSION returns a ‘constriction’ in  $G$  (a set  $A \subset V$  such that  $|\Gamma_G(A)|/|A|$  is close to 1) which permits an efficient decomposition into subproblems which can then be solved recursively. The specification of TESTEXPANSION is made explicit in Lemma 3.

**Lemma 3.** On input  $(G, \alpha)$ , where  $G$  is a bipartite graph and  $\alpha > 0$ , the algorithm TESTEXPANSION presented in Figure 1 *either*

- (i) correctly identifies  $G$  as an  $\alpha$ -expander, *or*

```

procedure TESTEXPANSION( $G, \alpha$ );
begin
  Let  $M$  be any perfect matching in  $G$ ;
  Reorder the vertices in  $V'$  so that
     $M$  matches each vertex  $v$  in  $V$  with the corresponding vertex  $v'$  in  $V'$ ;
  for each subset  $X \subset V$  with  $|X| < \frac{1}{2}\alpha n$  do begin
    Let  $\widehat{G}$  be the graph obtained from  $G$ 
      by removing vertices  $X, X'$ , and all incident edges;
  (1)   if there exists  $v \in V$  with  $\frac{1}{4}n < |\Delta_{\widehat{G}}(v)| \leq \frac{1}{2}n$  then  $A := \Delta_{\widehat{G}}(v)$ 
    else begin
      Let  $v_0, v_1, \dots, v_{n-|X|-1}$  be an enumeration of vertices  $v \in V - X$ 
        satisfying  $|\Delta_{\widehat{G}}(v_0)| \leq |\Delta_{\widehat{G}}(v_1)| \leq \dots \leq |\Delta_{\widehat{G}}(v_{n-|X|-1})|$ ;
       $i := 0$ ;  $A := \emptyset$ ;
  (2)   while  $|A \cup \Delta_{\widehat{G}}(v_i)| \leq \frac{1}{2}n$  do begin
         $A := A \cup \Delta_{\widehat{G}}(v_i)$ ;  $i := i + 1$ 
      end
    end;
  (3)   if  $|X|/|A| < 2\alpha$  then output  $A$  and halt
    end;
  (4)   output " $G$  is an  $\alpha$ -expander"
end

```

**Figure 1. Algorithm for testing the expansion factor of a graph**

(ii) produces a set  $A \subseteq V$  such that  $|A| \leq \frac{1}{2}n$  and  $|\Gamma_G(A)| < (1 + 2\alpha)|A|$ .

The execution time of TESTEXPANSION is  $\exp(O(\alpha n \log n))$ .

**Proof.** It is clear that the algorithm terminates, and either outputs a set  $A \subseteq V$ , or an assertion that  $G$  is an  $\alpha$ -expander. In the first instance we must assure ourselves that the set  $A$  satisfies the conditions  $|A| \leq \frac{1}{2}n$  and  $|\Gamma_G(A)| < (1 + 2\alpha)|A|$ , and in the second that the assertion is correct.

Suppose first that line (3) is reached and the condition  $|X|/|A| < 2\alpha$  is true. The set  $A$  is a union of tight sets, and hence is itself tight (with respect to the graph  $\widehat{G}$ ); equivalently,  $\Gamma_{\widehat{G}}(A) = A'$ . Thus  $|\Gamma_G(A)| \leq |A' \cup X'| = |A \cup X| < (1 + 2\alpha)|A|$ , as required.

Now suppose that the for-loop runs to completion, and line (4) is reached. We must demonstrate that the claim that  $G$  is an  $\alpha$ -expander is correct. Suppose to the contrary that there exists a set  $B \subset V$  such that  $|B| \leq \frac{1}{2}n$  and  $|\Gamma_G(B)| < (1 + \alpha)|B|$ . Note that  $\Gamma_G(B)$  can be written as a disjoint union  $B' + X'$  with  $|X'| < \alpha|B|$ . Since



$|X| < \frac{1}{2}\alpha n$ , the for-loop will eventually consider the set  $X$ ; observe that  $B$  is tight with respect to the graph  $\widehat{G}$  obtained by removing vertices  $X$ ,  $X'$ , and all incident edges. We distinguish two cases and obtain a contradiction in both.

CASE 1. There exists a vertex  $v \in V$  such that  $\frac{1}{4}n < |\Delta_{\widehat{G}}(v)| \leq \frac{1}{2}n$ . This situation is caught in line (1). Since  $|A| > \frac{1}{4}n$  and  $|X| < \frac{1}{2}\alpha n$ , the for-loop terminates prematurely at line (3), contrary to our assumption.

CASE 2. There does not exist a vertex  $v \in V$  such that  $\frac{1}{4}n < |\Delta_{\widehat{G}}(v)| \leq \frac{1}{2}n$ . After execution of the while-loop in line (2) we are left with a set  $A \subset V$  satisfying  $|A| \leq \frac{1}{2}n$  and  $|A \cup \Delta_{\widehat{G}}(v_i)| > \frac{1}{2}n$ . Now either  $|\Delta_{\widehat{G}}(v_i)| \leq \frac{1}{4}n$  or  $|\Delta_{\widehat{G}}(v_i)| > \frac{1}{2}n$ . If the first inequality holds,  $|A| > \frac{1}{4}n$  and we obtain a contradiction as in Case 1. If the second inequality holds then no  $v_j$  with  $j \geq i$  can be a member of  $B$ , and hence  $A \supseteq B$ . Thus  $|X|/|A| \leq |X|/|B| < \alpha$  and again the for-loop terminates prematurely at line (3).  $\square$

We now have all the ingredients for the proposed approximation scheme, which is presented in Figure 2 as the function COUNTPM. The analysis of COUNTPM is the subject of Theorem 4.

**Theorem 4.** With appropriate choices for the parameters  $t$  and  $\hat{\epsilon}$ , the algorithm COUNTPM of Figure 2 is a randomised approximation scheme for the number of perfect matchings in  $G$ . The execution time of COUNTPM is  $\epsilon^{-2} \exp(O(n^{1/2} \log^2 n))$ .

**Proof.** Let us first deal with the correctness of COUNTPM, deciding on the parameters  $t$  and  $\hat{\epsilon}$  as we do so. First imagine an idealised situation in which the procedure MARKOVSIM *always* returns the *exact* number of matchings in  $G$ . Then it is not too difficult to see that COUNTPM also produces the correct result. The crucial step to check is the division into sub-problems effected by the for-loop in line (5). Observe that any perfect matching  $M$  in  $G$  matches  $A \subset V$  with some subset  $B' \subseteq \Gamma_G(A)$  of size  $|A|$ , and that the set of all matchings in  $G$  can be partitioned according to the set  $B'$  so defined. The for-loop ranges over the blocks of this partition, accumulating the number of matchings in each block as it goes. To see that the body of the loop is correct, one merely needs to observe that the each perfect matching in the block defined by  $B'$  can be decomposed into a disjoint union of a perfect matching in  $G_0$  and a perfect matching in  $G_1$ .

Next set  $\hat{\epsilon} = \epsilon/2n_0$ , where  $n_0$  is the value of  $n$  at the top level of recursion, and suppose that MARKOVSIM always produces a result that is within ratio  $1 + \hat{\epsilon}$  of the correct value. (This is still an idealised situation, since the specification of MARKOVSIM allows large deviations with probability  $\frac{1}{4}$ .) Then a simple induction on  $n$  establishes that COUNTPM returns a result that is within ratio  $(1 + \hat{\epsilon})^n$  of  $N(G)$  when applied to a bipartite graph  $G$  with  $n + n$  vertices. In particular, the result returned from

```

function COUNTPM( $G$ );
  [[ Returns an approximation to the number of perfect matchings in  $G$ . ]]
  begin
    if  $n = 1$  then return 0 or 1 as appropriate;
    if  $G$  has no perfect matchings then return 0;
     $\alpha := n^{-1/2}$ ;
    (1) Execute TESTEXPANSION( $G, \alpha$ ) and TESTEXPANSION( $G^*, \alpha$ );
    (2) if  $G$  and  $G^*$  are both confirmed as  $\alpha$ -expanders then begin
    (3)   Call MARKOVSIM  $2t + 1$  times with parameters  $G, \alpha, \hat{\epsilon}$ ;
        [[ The parameters  $t$  and  $\hat{\epsilon}$  are appropriately initialised
            before entry to COUNTPM at the top level. ]]
    (4)   return the median of the  $2t + 1$  trials
    end else begin
      Wlog, let  $A \subset V$  satisfy  $|A| \leq \frac{1}{2}n$  and  $|\Gamma_G(A)| < (1 + 2\alpha)|A|$ ;
       $sum := 0$ ;
      (5)   for  $B' \subseteq \Gamma_G(A)$  with  $|B'| = |A|$  do begin
          Let  $G_0$  be the subgraph of  $G$  induced by  $A$  and  $B'$ ;
          Let  $G_1$  be the subgraph of  $G$  induced by  $V - A$  and  $V' - B'$ ;
      (6)    $sum := sum + \text{COUNTPM}(G_0) \times \text{COUNTPM}(G_1)$ 
      end;
      return  $sum$ 
    end
  end

```

**Figure 2. Recursive algorithm for counting perfect matchings**

the top-level call to COUNTPM is within ratio  $(1 + \hat{\epsilon})^{n_0} = (1 + \epsilon/2n_0)^{n_0} < 1 + \epsilon$ , as required.

Finally, suppose that MARKOVSIM is as specified, and produces a result that is within ratio  $1 + \hat{\epsilon}$  with probability at least  $\frac{3}{4}$ . Let us call the sequence of trials carried out in line (3) an *experiment*, and say that the experiment is a *success* if the median of the  $2t + 1$  trials approximates  $N(G)$  within ratio  $1 + \hat{\epsilon}$ . We shall choose  $t$  so that the probability that *every* experiment is a success is at least  $\frac{3}{4}$ . An easy induction on  $n$  establishes that the number of experiments is bounded by  $8^n$ . (This is a gross overestimate, as will be apparent once we have completed the analysis of the time complexity of COUNTPM.) By [5, Lemma 6.1], the failure probability of a single experiment may be reduced to  $8^{-(n_0+1)}$  by setting  $t = 18(n_0 + 1) = O(n_0)$ . (The point here is that  $t$  need not be very large.) With this choice of  $t$ , the probability that *every* experiment is a success is at least  $\frac{3}{4}$ . This completes the validation of COUNTPM.

Let  $T(n, \epsilon)$  be the (worst-case) time complexity of COUNTPM as a function of the size of  $G$  and the accuracy parameter  $\epsilon$ . (Technically, the time complexity depends on  $\hat{\epsilon}$  and  $t$ , and hence should be a function also of  $n_0$ ; however, the dependence on  $n_0$  is cubic, and can safely be swept under the carpet in the current context.) We show, by induction on  $n$ , that  $T(n, \epsilon) \leq \epsilon^{-2} \exp(cn^{1/2} \ln^2 n)$ , where  $c$  is a suitable constant. From Figure 2,

$$T(n, \epsilon) \leq \max \{T_1(n, \epsilon), T_2(n, \epsilon)\}, \quad (1)$$

where  $T_1(n, \epsilon)$  and  $T_2(n, \epsilon)$  are, respectively, the time complexities conditional on whether the ‘then’ or ‘else’ part of the if-statement is selected at line (2). The time complexity  $T_1$  is determined by the calls to TESTEXPANSION in line (1), and the sequence of  $2t + 1$  trials using MARKOVSIM in line (3). From Corollary 2 and Lemma 3, setting  $\alpha = n^{-1/2}$ , we have

$$\begin{aligned} T_1(n, \epsilon) &\leq \exp(O(n^{1/2} \log n)) + \epsilon^{-2} \exp(O(n^{1/2} \log^2 n)) \\ &\leq \epsilon^{-2} \exp(O(n^{1/2} \log^2 n)). \end{aligned} \quad (2)$$

The time complexity  $T_2$  is determined by the call to TESTEXPANSION in line (1) and the recursive calls to COUNTPM in line (6). Let  $S(k)$  be an upper bound on the number of subproblems generated (i.e., the number of times the for-loop is executed) when the set  $A$  returned by TESTEXPANSION has size  $k$ . By Lemma 3,

$$T_2(n, \epsilon) \leq \exp(O(n^{1/2} \log n)) + \max \{S(k)[T(k) + T(n - k)] : 1 \leq k \leq \frac{1}{2}n\}. \quad (3)$$

In preparation for the inductive step, we proceed to estimate each of the quantities appearing on the right hand side of (3). Suppose the set  $A$  returned by TESTEXPANSION is of size  $k$ ; then  $|\Gamma_G(A)| = k + d$ , where  $d < 2\alpha k = 2kn^{-1/2}$ , and

$$S(k) = \binom{k + d}{k} \leq n^d \leq \exp(2kn^{-1/2} \ln n) \quad (4)$$

$$\leq \exp(n^{1/2} \ln n). \quad (5)$$

Applying the induction hypothesis we obtain (since  $k \leq \frac{1}{2}n$ )

$$T(k, \epsilon) \leq \epsilon^{-2} \exp(ck^{1/2} \ln^2 k) \leq \epsilon^{-2} \exp(\frac{3}{4}cn^{1/2} \ln^2 n), \quad (6)$$

and, again,

$$\begin{aligned} T(n - k, \epsilon) &\leq \epsilon^{-2} \exp(c(n - k)^{1/2} \ln^2(n - k)) \\ &\leq \epsilon^{-2} \exp(c(1 - \frac{1}{2}kn^{-1})n^{1/2} \ln^2 n) \\ &\leq \epsilon^{-2} \exp(cn^{1/2} \ln^2 n - \frac{1}{2}ckn^{-1/2} \ln^2 n). \end{aligned} \quad (7)$$

Now suppose  $c \geq 8$ . By combining inequalities (5) and (6), we have

$$S(k)T(k, \epsilon) \leq \epsilon^{-2} \exp\left(\frac{7}{8}cn^{1/2} \ln^2 n\right), \quad (8)$$

and by combining (4) and (7),

$$\begin{aligned} S(k)T(n-k, \epsilon) &\leq \epsilon^{-2} \exp\left(cn^{1/2} \ln^2 n - \frac{1}{4}ckn^{-1/2} \ln^2 n\right) \\ &\leq \epsilon^{-2} \exp\left(cn^{1/2} \ln^2 n - 2n^{-1/2}\right) \\ &\leq (1 - n^{-1/2})\epsilon^{-2} \exp\left(cn^{1/2} \ln^2 n\right). \end{aligned} \quad (9)$$

It is immediate from (2) that  $T_1(n, \epsilon) \leq \epsilon^{-2} \exp(cn^{1/2} \ln^2 n)$  for  $c$  sufficiently large, and it follows easily from (3), (8), and (9) that  $T_2(n, \epsilon) \leq \epsilon^{-2} \exp(cn^{1/2} \ln^2 n)$ . Finally, from (1),  $T(n, \epsilon) \leq \epsilon^{-2} \exp(cn^{1/2} \ln^2 n)$ , which completes the inductive step.  $\square$

### 3. Extensions and open problems

The algorithm presented here rests on a tradeoff: a graph with a high expansion factor is amenable to direct solution using an existing approximation algorithm, while a graph with low expansion factor admits an efficient recursive decomposition into subproblems. The exponent  $\frac{1}{2}$  appearing in the expression for the time complexity arises quite naturally from a consideration of the crossover point for this tradeoff, and it seems unlikely that a further improvement in asymptotic time-complexity is possible without the introduction of some new technique. The ultimate goal must surely be to construct a fully polynomial approximation scheme (or to provide convincing complexity-theoretic evidence that no such scheme exists). However, a reasonable intermediate goal might be to design an approximation algorithm with ‘quasi-polynomial’ time complexity, i.e., with time complexity  $O(\exp(\text{poly}(\log n)))$ .

Of possible generalisations of the problem considered in this note, the two most natural appear to be (1) estimating the permanent of a general non-negative integer matrix, and (2) estimating the number of perfect matchings in a general (non-bipartite) graph. Provided the entries in the matrix are small, the first of these problems can be handled by reduction to the 0,1 case. Suppose  $A$  is an  $n \times n$  matrix of non-negative integers whose binary expansions are uniformly bounded in length by  $k$ . It is not difficult to show that there exists an  $m \times m$  matrix  $A'$  with 0,1-entries for which  $\text{per}(A') = \text{per}(A)$ ; here  $m = O(nk)$ , and the matrix  $A'$  is easily constructed in polynomial time. However, if the matrix entries are very large integers (of roughly  $n$  bits) it will be seen that this reduction provides no gain in efficiency over exact computation using Ryser’s algorithm. The question of whether Ryser’s method can be beaten for general non-negative matrices remains open.

It seems plausible that the second generalisation (to non-bipartite graphs) can be handled by methods similar to those employed in this note. The stumbling block is

Lemma 1: even if the lemma itself can be extended to non-bipartite graphs  $G$ , the proof presented here cannot.

## References

- [1] Andrei Z. BRODER, How hard is it to marry at random? (On the approximation of the permanent), *Proceedings of the 18th ACM Symposium on Theory of Computing*, 1986, pp. 50–58. Erratum in *Proceedings of the 20th ACM Symposium on Theory of Computing*, 1988, p. 551.
- [2] 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, editors), Colloquia Mathematica Societatis János Bolyai **25**, North-Holland, 1981.
- [3] Mark JERRUM, *An analysis of a Monte-Carlo algorithm for Estimating the Permanent*, Report ECS-LFCS-91-164, Department of Computer Science, University of Edinburgh, June 1991.
- [4] Mark JERRUM and Alistair SINCLAIR, Approximating the permanent, *SIAM Journal on Computing* **18** (1989), pp. 1149–1178.
- [5] Mark R. JERRUM, Leslie G. VALIANT, and Vijay V. VAZIRANI, Random generation of combinatorial structures from a uniform distribution, *Theoretical Computer Science* **43** (1986), pp. 169–188.
- [6] N. KARMAKAR, R. KARP, R. LIPTON, L. LOVÁSZ, and M. LUBY, *A Monte-Carlo Algorithm for Estimating the Permanent*, preprint 1988.
- [7] R. M. KARP and M. LUBY, Monte-Carlo algorithms for enumeration and reliability problems, *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, 1983, pp. 56–64.
- [8] L. LOVÁSZ and M. D. PLUMMER, *Matching Theory*, North-Holland, Amsterdam, 1986.
- [9] Henryk MINC, *Permanents*, Addison Wesley, 1978.
- [10] Alistair SINCLAIR, *Randomised Algorithms for Counting and Generating Combinatorial Structures*, PhD Thesis, University of Edinburgh, June 1988.
- [11] L. G. VALIANT, The complexity of computing the permanent, *Theoretical Computer Science* **8** (1979), pp. 189–201.