Tailoring the Permanent Formula to Problem Instances

Eric Bax

Computer Science Department
California Institute of Technology

Caltech-CS-TR-96-17

Tailoring the Permanent Formula to Problem Instances

Eric Bax*

July 10, 1996

Abstract

A parameterized set of finite-difference formulas have been developed for the permanent. One parameter setting produces Ryser's [2] inclusion and exclusion formula. Other parameter settings yield formulas that can be computed more efficiently.

One group of settings introduces symmetry, so only half the terms need to be computed. Some of these settings produce formulas that have many zero-valued terms when applied to matrices drawn from random distributions. Gathering the zero-valued terms and removing them from the computation substantially reduces the time required to compute the permanent [1].

This paper explores methods to tailor the parameter settings to specific matrices, choosing the formula based on the problem instance to increase the number of zero-valued terms.

Key words algorithms, combinatorial problems, permanent.
AMS subject classifications 05,68

*Supported by an NSF fellowship. Computer Science Department, California Institute of Technology 256-80, Pasadena, California, 91125 (eric@cs.caltech.edu).

1 Introduction

The permanent of an $n \times n$ matrix $A = [a_{ij}]$ is defined as follows.

$$per\ A = \sum_{j_1 \cdots j_n} a_{1j_1} \cdots a_{nj_n} \tag{1}$$

where $j_1 \ldots j_n$ is a permutation of $1 \ldots n$. Computing the permanent is a #P-complete problem [3].

Ryser [2] developed the following inclusion and exclusion formula to compute the permanent.

$$per\ A = \sum_{S \subseteq \{1,\ldots,n\}} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j \notin S} a_{ij} \tag{2}$$

The formula is a sum over 2^n terms, each of which can be computed in polynomial time. Each term is a product of row sums, with the columns that correspond to the elements of S zeroed.

Finite-difference formulas [1] are a generalization of Ryser's formula. Each term is a product of row sums, with the columns of A multiplied by a choice of values. Define the term $P(x_1, \ldots, x_n)$ to be the product of row sums of matrix A with each column j multiplied by variable x_j.

$$P(x_1, \ldots, x_n) \equiv \prod_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_j \tag{3}$$

The finite-difference formulas for the permanent have the following form.

$$\frac{1}{(u_1 - v_1) \cdots (u_n - v_n)} \sum_{\{x_1,\ldots,x_n\} \in \{u_1,v_1\} \times \cdots \times \{u_n,v_n\}} (-1)^{s(x_1,\ldots,x_n)} P(x_1, \ldots, x_n) \tag{4}$$

where $s(x_1, \ldots, x_n)$ is the number of variables x_j set to v_j, and for all j, $u_j \neq v_j$. The choice of column multipliers $\mathbf{u} = [u_j]$ and $\mathbf{v} = [v_j]$ determines a specific finite-difference formula.

Setting $(\mathbf{u}, \mathbf{v}) = (\mathbf{1}, \mathbf{0})$ produces Ryser's formula. The formula with column multipliers $(\mathbf{u}, \mathbf{v}) = (\mathbf{1}, -\mathbf{1})$ often has many zero-valued terms $P(\mathbf{x})$ when applied to matrices drawn from a random distribution. Methods have been developed to reduce computation by collecting and eliminating these terms [1].

In this paper, the goal is to tailor the column multipliers $(\mathbf{u}, \mathbf{v})$ to specific matrices A to increase the number of zero terms in the resulting formulas. We consider the $(\mathbf{1}, -\mathbf{1})$ formula to be a standard for comparison. First we develop a method to choose between ± 1 and ± 2 multipliers for each column. Then we develop methods to choose column multipliers from a larger domain.

2 Even and Odd Rows, ± 1 and ± 2

If every row in matrix A has an odd number of 1's, then the $(\mathbf{1}, -\mathbf{1})$ formula does not produce any zero terms at all. Consider the following problem instance matrix and the corresponding matrix with columns multiplied by variables.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ x_1 & x_2 & x_3 & 0 & 0 \\ 0 & x_2 & x_3 & x_4 & 0 \\ 0 & 0 & x_3 & x_4 & x_5 \\ x_1 & x_2 & 0 & 0 & x_5 \end{bmatrix} \tag{5}$$

Each row of B has an odd number of variables, so it is impossible to assign an equal number of $+1$'s and -1's to the variables in any row.

Zero terms can be introduced by setting $u_j = 2$ and $v_j = -2$ for some variable x_j and setting $u_j = 1$ and $v_j = -1$ for the other variables. For example, setting $u_1 = 2$ and $v_1 = -2$ introduces zero terms in every row containing x_1. A row is zeroed when the sum of the variables in its last four columns equals the value assigned to x_1. For the top row of B, there are $C(4,3)$ ways to assign $+1$'s and -1's to the variables x_2 through x_5 to get a sum of $+2$. By symmetry, there are also $C(4,3)$ assignments that sum to -2. So each assignment to x_1 has $C(4,3)$ zero terms, and the top row now has $2C(4,3) = 8$ zero terms. By the same reasoning, the second and fifth rows now have $2C(2,2) = 2$ zero terms each. On the other hand, the third and fourth rows do not contain x_1, so their column sums are still odd for all assignments. Hence, they have no zero terms.

2.1 ± 1 and ± 2 Procedures

One approach to choosing column multipliers $\mathbf{u}$ and $\mathbf{v}$ is to set all column multipliers to ± 1 initially, then set some multipliers to ± 2 to introduce zero terms in odd rows. Define $\sum_{j=1}^{n} a_{ij} u_j$ to be the row total for row i. For a row to have zero terms in a $(\mathbf{u}, -\mathbf{u})$ formula, the row must have an even row total – otherwise exact cancellation of variables is impossible. Thus, toggling a column multiplier between ± 1 and ± 2 toggles the possibility of zero terms for each row of B with a variable in the column.

One goal in allocating ± 1's and ± 2's as column multipliers is to maximize the number of rows with even row totals. Another goal is to limit the number of ± 2's. In the $(+\mathbf{1}, -\mathbf{1})$ formula, each row's zero terms come from the center of the binomial distribution. As more ± 2's are introduced, the zero terms recede away from the center of the distribution. For example, suppose a row has 10 variables. In the $(+\mathbf{1}, -\mathbf{1})$ formula, the row has $C(10,5) = 252$ zero terms. Now suppose two of the variables have column multipliers ± 2. In 2 cases, the ± 2 variables cancel, and there are $C(8,4)$ zero terms. In the other two cases, the ± 2 variables sum to 4 or -4, and there are $C(8,2)$ zero terms. This is a total of $2C(8,4) + 2C(8,2) = 196$ zero terms.

Suppose a row has k variables, d with ± 2 column multipliers and $k-d$ with ± 1 column multipliers. Also, suppose the total is even. Then the row's zero terms can be counted as a sum of cases. In each case, let m be the number of ± 2 variables assigned $+2$. The sum of the ± 2 variables is $+2(m)-2(d-m) = 4m-2d$. If there are not enough ± 1 variables to cancel the ± 2 variable sum, i.e. if $k-d < |4m-2d|$, then the case has no zero terms. Otherwise, in each zero term, $|4m-2d|$ of the $k-d$ ± 1 variables cancel the ± 2 variable sum. Also, half of the remaining $(k-d)-|4m-2d|$ ± 1 variables cancel the other half. So this formula counts the row's zero terms:

$$\sum_{\substack{m=1 \\ k-d \geq |4m-2d|}}^{d} C(d,m)C(k-d, |4m-2d| + \frac{1}{2}[(k-d)-|4m-2d|]) \quad (6)$$

$$\sum_{\substack{m=1 \\ k-d \geq |4m-2d|}}^{d} C(d,m)C(k-d, \frac{1}{2}[(k-d)+|4m-2d|]) \quad (7)$$

The following procedure assigns ± 1 and ± 2 column multipliers. Initially, all column multipliers are assigned ± 1. Each step executes one of the column multiplier toggles which most increases the number of rows with even totals. Since each step increases the number of rows with even totals, the procedure terminates after at most n steps. The procedure returns $\mathbf{u}$ and $\mathbf{v} = -\mathbf{u}$.

choose_multipliers_±1_±2
{
define $\text{even}(A, \mathbf{u}) \equiv |\{i | \sum_{j=1}^{n} a_{ij}u_j \text{ is even }\}|$
define $\text{toggle}(\mathbf{u}, j) \equiv \begin{cases} (u_1, \ldots, u_{j-1}, 1, u_{j+1}, \ldots, u_n) & \text{if } u_j = 2 \\ (u_1, \ldots, u_{j-1}, 2, u_{j+1}, \ldots, u_n) & \text{if } u_j = 1 \end{cases}$

initially $\mathbf{u} = \mathbf{1}$

while $\exists\ j \in \{1, \ldots, n\}$ such that even(A,toggle($\mathbf{u}, j$)) $>$ even($A, \mathbf{u}$)
  {
  $T = \{j |$ even(A,toggle($\mathbf{u}, j$)) $= \max_{\hat{j} \in \{1,\ldots,n\}}$ even(A,toggle($\mathbf{u}, \hat{j}$))$\}$
  choose j from T
  $\mathbf{u} :=$ toggle($\mathbf{u}, j$)
  }
return($\mathbf{u}, -\mathbf{u}$)
}

2.2 ± 1 and ± 2 Tests

Figures 1 and 2 compare the number of zero terms in the permanent formula with $(\mathbf{u}, \mathbf{v})$ selected by choose_multipliers_± 1_± 2 to the number of zero terms in the formula with $(\mathbf{u}, \mathbf{v}) = (\mathbf{1}, -\mathbf{1})$. The results displayed in the figures are averages over several tests performed on random 0-1 matrices drawn from the distribution $A(n,p)$. The figures show that the procedure choose_multipliers_± 1_± 2 produces a modest increase in the number of zero terms.

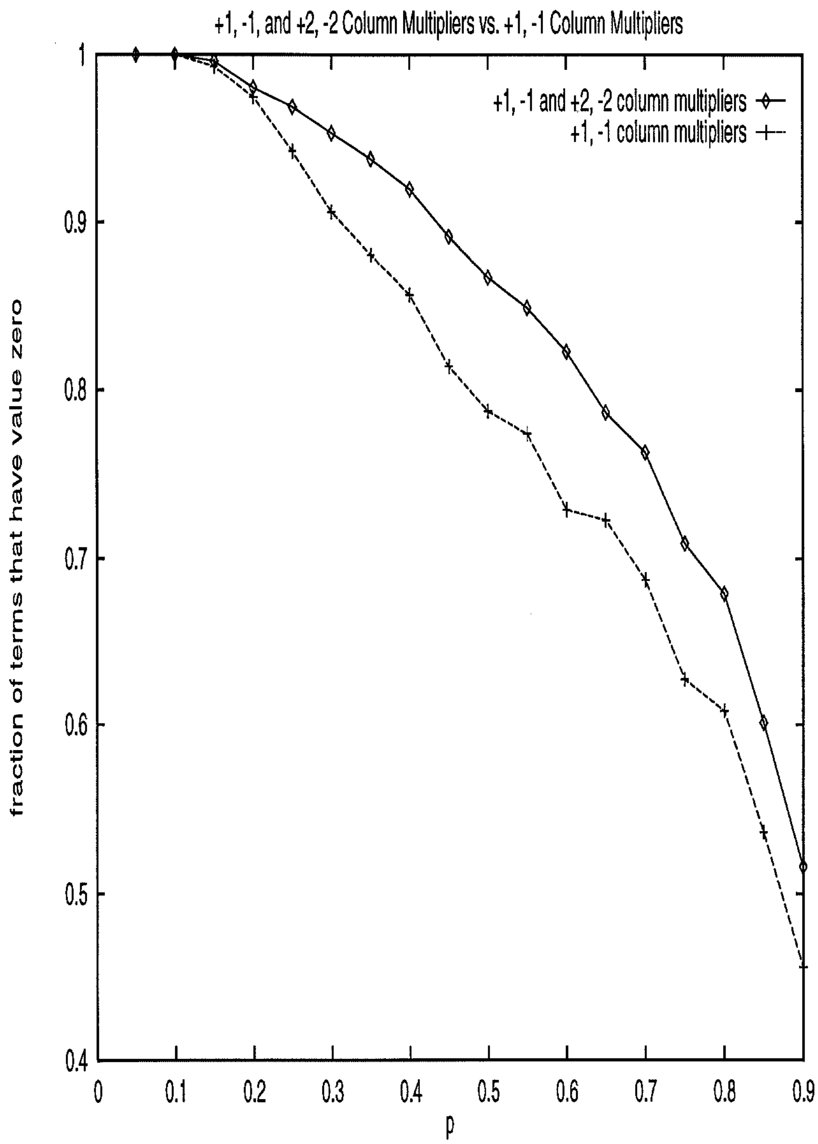

Figure 1: Results averaged over 100 10×10 matrices.

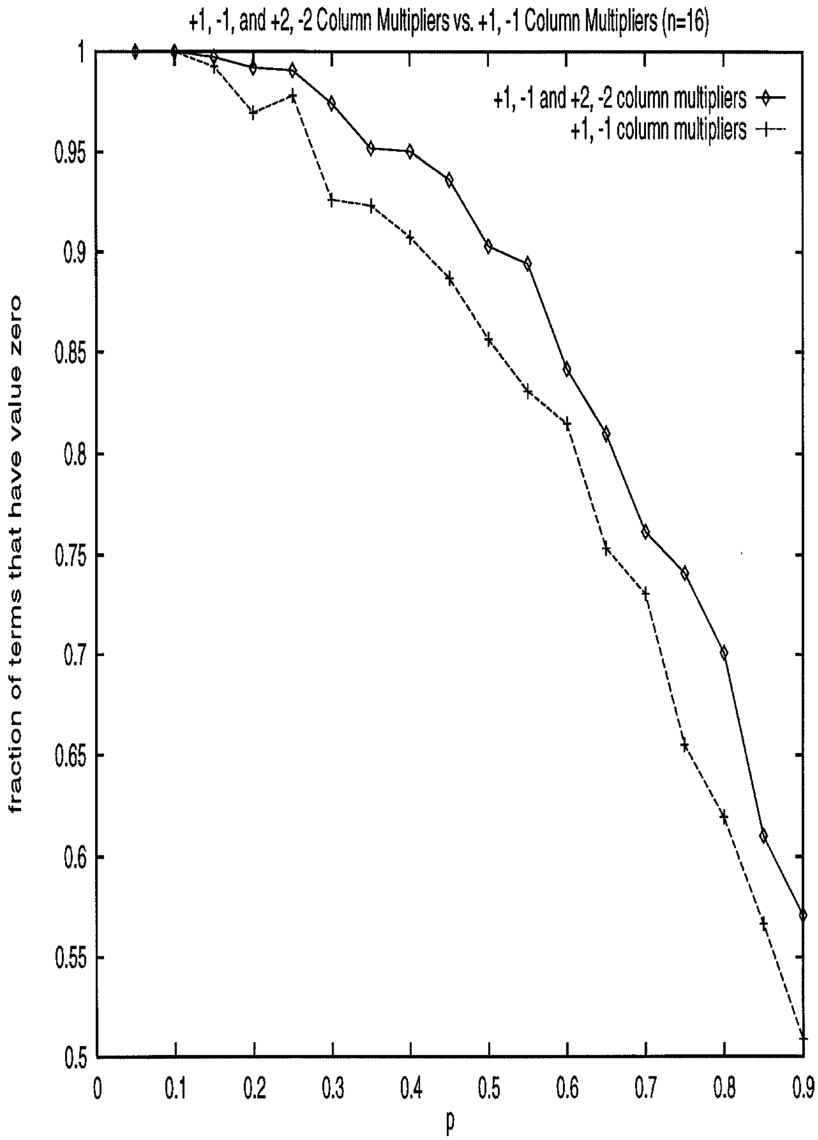

Figure 2: Results averaged over 10 16×16 matrices.

3 Choosing (u, v) by Iterative Ascent

In this part of the paper, we consider methods to increase the number of zero terms by adjusting the column multipliers iteratively. We begin with some column multipliers $(\mathbf{u}, \mathbf{v})$. We estimate the number of zero terms that would occur if the corresponding formula was applied to the problem instance at hand. We consider adjustments to some of the column multiplier values, estimating the number of zero terms that would be produced by each adjustment. We perform the adjustment that would produce the most zero terms, according to our estimates. Then, we perform the same process on our new column multipliers.

3.1 Iterative Ascent Procedures

We restrict our search to integer-valued column multipliers. Since column multipliers $(\mathbf{u}, \mathbf{v})$ with $\mathbf{u} = -\mathbf{v}$ have computational advantages due to symmetry [1], we restrict our search to these multipliers. Thus, we search for a vector $\mathbf{u}$ such that the $(\mathbf{u}, -\mathbf{u})$ formula has many zero terms when applied to problem instance A. Since the finite-difference formulas require $u_j \neq v_j \ \forall j$, we restrict the search to values of $\mathbf{u}$ with $u_j \neq 0 \ \forall j$. Note that if we exchange column multipliers u_j and v_j, then the resulting formula has the same number of zero terms as the original formula. Thus, any formula can be converted to a formula with $\mathbf{u} > \mathbf{0}$ and with the same number of zero terms by exchanging u_j and v_j for each negative u_j. Thus, we restrict our search to $\mathbf{u} > \mathbf{0}$.

Our iterative procedures perform local ascent steps in the lattice of positive integer-valued vectors. These procedures are subject to all of the problems associated with gradient ascent methods. For example, our iterative procedures may settle on local maxima that are not optimal.

Let $f(\mathbf{u})$ be an estimate of the number of zero terms in the $(\mathbf{u}, -\mathbf{u})$ formula for the permanent of matrix A. Let $\mathbf{u}^0$ be an initial value for $\mathbf{u}$. Let $N(\mathbf{u})$ be a set of column multiplier candidates that includes $\mathbf{u}$. Our iterative ascent procedures have the following form.

iterative_ascent
{
$\mathbf{u} = \mathbf{u}^0$

repeat
    $\mathbf{u} = \min_{\mathbf{w} \in N(\mathbf{u})} f(\mathbf{w})$
until the termination condition is satisfied

return $(\mathbf{u}, -\mathbf{u})$
}

The intial value can be set randomly, set to **1**, or set by the procedure choose_multipliers_±1_±2. If the values are drawn at random, it is probably best to favor small values in the distribution.

The termination condition can consist of a test for convergence coupled with a limit on the number of iterations. The iterative methods presented here are not guaranteed to converge, so a test for convergence alone is not sufficient.

Larger sets of candidates $N(\mathbf{u})$ produce more robust search steps, but they also have higher computational costs. It is tractable to conduct line searches, in which a single element of $\mathbf{u}$ is varied over some range. It is also feasible to vary a few elements at once. However, varying all elements independently is intractable, since it produces an exponential (in n) number of candidates.

Random sampling can provide an estimate $f(\mathbf{u})$ of the number of zero terms. Choose some assignments (x) at random from $\{u_1, -u_1\} \times \cdots \times \{u_n, -u_n\}$. For each assignment, compute the term $P(\mathbf{x})$. Multiply the fraction of the sampled terms with value zero by 2^n to estimate the number of zero terms. The accuracy of the estimate grows with the sample size. On the other hand, the higher variance resulting from smaller sample sizes may encourage escape from local maxima. So the best strategy may be to begin with a small sample size and increase it over the course of the iterations – a simulated annealing technique.

An estimate of the number of zero terms can be calculated by assuming independence among the terms zeroed by individual rows. Define U to be the set of assignments to $\mathbf{x}$ in the permanent formula.

$$U \equiv \{u_1, -u_1\} \times \cdots \times \{u_n, -u_n\} \tag{8}$$

Define Z_i to be the set of assignments for which row i has sum zero after the columns of A are multiplied by the elements of $\mathbf{x}$, i.e. let Z_i be the set of assignments $\mathbf{x}$ for which the terms $P(\mathbf{x})$ are zero because row i has sum zero.

$$Z_i \equiv \{\mathbf{x} | \sum_{i=1}^{n} a_{ij} x_j = 0\} \tag{9}$$

Define $q_i \equiv \frac{|Z_i|}{|U|}$, i.e. q_i is the fraction of assignments that zero row i.

If the row zeroings were independent, then the fraction of assignments that zero no row would be:

$$\prod_{i=1}^{n}(1 - q_i) \tag{10}$$

The fraction of zero terms would be the difference between this number and one. So we have the following estimate of the number of zero terms.

$$f(\mathbf{u}) = 2^n[1 - \prod_{i=1}^{n}(1 - q_i)] \tag{11}$$

This estimate can be computed efficiently by using the following dynamic programming procedure to calculate each $|Z_i|$. The procedure's operation is based on the principle that each assignment to the first $k-1$ variables $x_1, \ldots, x_{k-1}$ that generates sum s in the first $k-1$ elements of row i after column multiplication ($s = \sum_{j=1}^{k-1} a_{ij} x_j$) is the prefix of two assignments to the first k variables $x_1, \ldots, x_k$. The assignment with $x_k = u_k$ has sum $s + u_k$ in the first k elements, and the assignment with $x_k = v_k$ has sum $s + v_k$ in the first k elements. In the procedure, b^k is the minimum sum of the first k elements of row i after column multipliers have been applied to A.

$$b^k \equiv \min_{\mathbf{x} \in U} \sum_{j=1}^{k} a_{ij} x_j \tag{12}$$

The variable t^k is the corresponding maximum value.

$$t^k \equiv \max_{\mathbf{x} \in U} \sum_{j=1}^{k} a_{ij} x_j \tag{13}$$

On termination, each variable c_s^k contains the number of column multiplier assignments $\mathbf{x}$ that produce a row sum of s over the first k elements of row i.

$$c_s^k = |\{\mathbf{x} \in U | \sum_{j=1}^{k} a_{ij} x_j = s\}| \tag{14}$$

Hence, c_0^n is the number of assignments that zero row i. (The procedure is written for the general $(\mathbf{u}, \mathbf{v})$. To compute the estimate $f(\mathbf{u})$, call the procedure with $\mathbf{v} = -\mathbf{u}$.)

compute_$|Z_i|$(A,$\mathbf{u}$,$\mathbf{v}$,i)
{
define $b^k \equiv \sum_{j=1}^{k} a_{ij} \min(u_j, v_j)\ \forall\ k \in \{1, \ldots, n\}$, $b^0 \equiv 0$
define $t^k \equiv \sum_{j=1}^{k} a_{ij} \max(u_j, v_j)\ \forall\ k \in \{1, \ldots, n\}$, $t^0 \equiv 0$
initially $c_s^k = 0\ \forall\ k \in \{1, \ldots, n\}$, $s \in \{b^k, \ldots, t^k\}$, $c_0^0 = 1$

for $k = 1$ to n
    {
    $\forall\ s \in \{b^{k-1}, \ldots, t^{k-1}\}$
        {
        $c_{s+a_{ik}u_k}^k := c_{s+a_{ik}u_k}^k + c_s^{k-1}$

        $c_{s+a_{ik}v_k}^k := c_{s+a_{ik}v_k}^k + c_s^{k-1}$
        }

```
    }
return c_0^n
}
```

In reality, row zeroings can be highly correlated. For example, suppose $(\mathbf{u},\mathbf{v}) = (\mathbf{1},-\mathbf{1})$ and A has no zero rows. The assignment $\mathbf{x} = \mathbf{1}$ makes every row sum positive, so each row has probability zero of having a zero sum. On the other hand, among assignments with about half of the elements of $\mathbf{x}$ assigned $+1$ and about half assigned -1, each row has a relatively high probability of being zeroed. Thus, intersections of the sets of terms that zero each row are larger than they would be if row zeroings were independent. Consequently, the union of row zero sets is smaller than it would be if row zeroings were independent. So $f(\mathbf{u},\mathbf{v})$ overestimates the number of zero terms $|Z_1 \cup \ldots \cup Z_n|$.

Recall that when $(\mathbf{u},\mathbf{v}) = (\mathbf{1},-\mathbf{1})$, each row's zero terms are drawn from the center of the binomial distribution. Hence, q_i $=\mathrm{O}(\frac{1}{\sqrt{n}})$ for each row i. Substituting $q_i = \frac{1}{\sqrt{n}}$ into (10) gives a rough estimate of the expected fraction of nonzero terms if the row zeroings were independent:

$$\prod_{i=1}^{n}(1-\frac{1}{\sqrt{n}}) = (1-\frac{1}{\sqrt{n}})^n \asymp e^{-\sqrt{n}-\frac{1}{2}} \tag{15}$$

Earlier in this paper it was shown that the expected fraction of nonzero terms is actually much higher, e.g. it is $\mathrm{O}(\frac{1}{\sqrt{n\log n}})$ when $p = \frac{1}{2}$.

The intersections of row zero sets can be decreased by varying $(\mathbf{u},-\mathbf{u})$ from $(\mathbf{1},-\mathbf{1})$. But this process generally shrinks the row zero sets themselves. So there is a tradeoff. If the elements of $\mathbf{u}$ are too small, then the row zeroings are highly correlated, so the union of row zero sets is small. If the elements are too large, then there are very few row zeroings, so once again the union of row zero sets is small.

Our approximation function $f(\mathbf{u})$ can be extended so that it depends on some of the row zero set intersections. Define $q_S \equiv \frac{|\bigcup_{i\in S} Z_i|}{|U|}$, and $q_\emptyset \equiv 1$. By inclusion and exclusion:

$$\frac{|U-(Z_1\cup\ldots\cup Z_n)|}{|U|} = \sum_{S\subseteq\{1,\ldots,n\}} (-1)^{n-|S|} q_S \tag{16}$$

Assuming independence ($q_S = \prod_{i\in S} q_i$) gave the estimate:

$$\frac{|U-(Z_1\cup\ldots\cup Z_n)|}{|U|} \approx \sum_{S\subseteq\{1,\ldots,n\}} (-1)^{n-|S|} \prod_{i\in S} q_i = \prod_{i=1}^{n}(1-q_i) \tag{17}$$

Expand the product, group the factors by pairs, and expand each pair:

$$\prod_{i=1}^{n}(1-q_i) = (1-q_1)\dots(1-q_n) = (1-q_1-q_2+q_1q_2)\dots(1-q_{n-1}-q_n+q_{n-1}q_n) \tag{18}$$

To introduce some intersections, replace $q_1q_2, \dots, q_{n-1}q_n$ with $q_{\{1,2\}}, \dots, q_{\{n-1,n\}}$, i.e. remove the independence assumption on these pairs of row zero sets.

$$\frac{|U-(Z_1 \cup \dots \cup Z_n)|}{|U|} \approx (1-q_1-q_2+q_{\{1,2\}})\dots(1-q_{n-1}-q_n+q_{\{n-1,n\}}) \tag{19}$$

So we have the following estimate of the number of zero terms $|Z_1 \cup \dots \cup Z_n|$.

$$f(\mathbf{u}) = 2^n[1-(1-q_1-q_2+q_{\{1,2\}})\dots(1-q_{n-1}-q_n+q_{\{n-1,n\}})] \tag{20}$$

The dynamic programming procedure to compute $|Z_i|$ can be extended to compute $|Z_{i_1} \cap Z_{i_2}|$. On termination of the following procedure, $c^k_{s_1,s_2}$ contains the number of assignments $\mathbf{x}$ that produce partial row sum s_1 over the first k elements of row i_1 and partial row sum s_2 over the first k elements of row i_2 after column multiplication.

$$c^k_{s_1,s_2} = |\{\mathbf{x} \in U | \sum_{j=1}^{k} a_{i_1j}x_j = s_1 \text{ and } \sum_{j=1}^{k} a_{i_2j}x_j = s_2\}| \tag{21}$$

Hence, $c^k_{0,0}$ is the number of assignments that zero both row i_1 and row i_2.

compute_$|Z_{i_1} \cap Z_{i_2}|(A,\mathbf{u},\mathbf{v},i_1,i_2)$

{

define $b_1^k \equiv \sum_{j=1}^k a_{i_1j} \min(u_j, v_j)\ \forall\ k \in \{1,\dots,n\}$, $b_1^0 \equiv 0$

define $t_1^k \equiv \sum_{j=1}^k a_{i_1j} \max(u_j, v_j)\ \forall\ k \in \{1,\dots,n\}$, $t_1^0 \equiv 0$

define $b_2^k \equiv \sum_{j=1}^k a_{i_2j} \min(u_j, v_j)\ \forall\ k \in \{1,\dots,n\}$, $b_2^0 \equiv 0$

define $t_2^k \equiv \sum_{j=1}^k a_{i_2j} \max(u_j, v_j)\ \forall\ k \in \{1,\dots,n\}$, $t_2^0 \equiv 0$

initially $c^k_{s_1s_2} = 0\ \forall\ k \in \{1,\dots,n\}$, $(s_1,s_2) \in \{b_1^k,\dots,t_1^k\} \times \{b_2^k,\dots,t_2^k\}$, $c^0_{00} = 1$

for $k = 1$ to n

{

$\forall\ (s_1,s_2) \in \{b_1^{k-1},\dots,t_1^{k-1}\} \times \{b_2^{k-1},\dots,t_2^{k-1}\}$

{

$c^k_{s_1+a_{i_1k}u_k, s_2+a_{i_2k}u_k} := c^k_{s_1+a_{i_1k}u_k, s_2+a_{i_2k}u_k} + c^{k-1}_{s_1,s_2}$

$c^k_{s_1+a_{i_1k}v_k, s_2+a_{i_2k}v_k} := c^k_{s_1+a_{i_1k}v_k, s_2+a_{i_2k}v_k} + c^{k-1}_{s_1,s_2}$

```
        }
    }
return c^n_{0,0}
}
```

Similar estimates can be derived using intersections of three or more sets.

3.2 Iterative Ascent Tests

Figure 3 shows the results of tests to compare several iterative search methods against the $(\mathbf{1}, -\mathbf{1})$ formula. In each test, a random 0-1 matrix is drawn from $A(n,p)$, and each iterative method is applied to the matrix to produce some $(\mathbf{u}, -\mathbf{u})$. Then the fraction of zero terms is computed for each $(\mathbf{u}, -\mathbf{u})$ formula given by an iterative method, and for the standard $(\mathbf{1}, -\mathbf{1})$ formula as well. The results shown in the figure are averages over several tests.

In all of these tests, the initial value $\mathbf{u}^0$ is $\mathbf{1}$. The searches are terminated after 20 iterations. In each iteration, a pair (j_1, j_2) is selected at random from $\{1, \ldots, n\}^2$, and $N(\mathbf{u})$ is two-dimensional with a "radius" of two in the j_1 and j_2 elements:

$$N(\mathbf{u}) = \{\mathbf{w} | w_j = u_j \ \forall \ j \notin \{j_1, j_2\}, \tag{22}$$

$$w_{j_1} \in \{u_{j_1} - 2, \ldots, u_{j_1} + 2\}, w_{j_2} \in \{u_{j_2} - 2, \ldots, u_{j_2} + 2\}, \text{ and } \mathbf{w} > \mathbf{0}\} \tag{23}$$

The sampling estimator draws 50 random terms to estimate the nonzero term fraction for each $\mathbf{w}$ in the search. Figure 3 shows that iterative search with the sampling estimator generally failed to produce a formula with more zero terms than the standard $(\mathbf{1}, -\mathbf{1})$ formula. The accuracy may be improved by drawing more samples, but, given the size of the test matrices (10×10), using even 50 random terms is infeasible. Computing the permanent formula using $\mathbf{u} = -\mathbf{v}$ symmetry requires calculating at most $\frac{1}{2}2^{10} = 512$ terms. But the iterative formula to find multipliers computes $20 \times 5 \times 5 \times 50 = 25,000$ terms!

The estimators based on computing the cardinality of row zero term sets fared better. The estimator that computes the cardinality of some intersections generally outperformed the estimator that assumes independence among all row zero term sets.

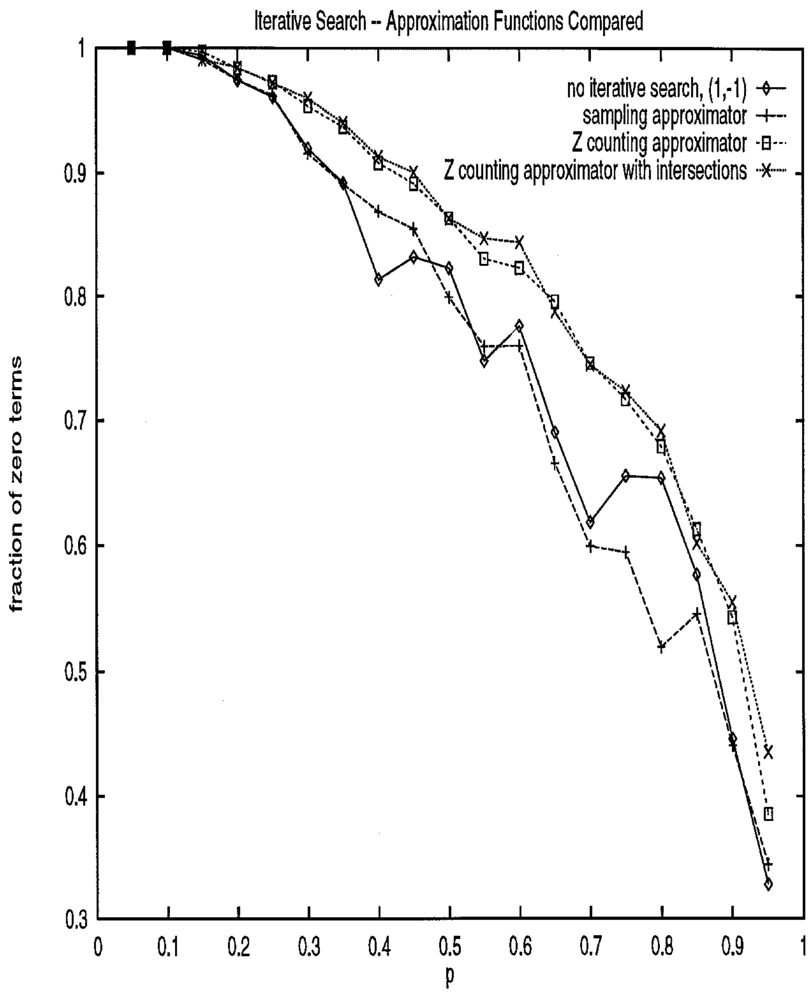

Figure 3: Results averaged over 10 10×10 matrices.

4 Conclusion

Our tests indicate that, on average, either of two simple strategies increases the number of zero terms in the permanent formula by adapting the column multipliers to the problem instance at hand. One succesful strategy is to assign ± 2 column multipliers to create zero terms in rows with odd numbers of entries. Another succesful strategy is to choose the column multipliers by iteratively ascending on an estimate of the number of zero terms in the formula. Both of these strategies are efficient. They operate in polynomial time to reduce the permanent computation, which requires exponential time in the worst case.

The strategies developed here are simple and ad hoc. Their success demonstrates the utility of adapting the permanent formula to the problem instance at hand. There is no reason to believe that the methods developed in this paper are optimal in any way; there are probably much better methods yet to be discovered.

References

[1] E. Bax and J. Franklin, A finite-difference sieve to compute the permanent, CalTech-CS-TR-96-04.

[2] H. J. Ryser, *Combinatorial Mathematics*, The Mathematical Association of America 1963, Ch. 2.

[3] L. G. Valiant, The complexity of computing the permanent, *Theoretical Computer Science*, 8 (1979) 189-201.