Approximate Counting by Dynamic Programming

Martin Dyer*

November 4th, 2002

Abstract

We give efficient algorithms to sample uniformly, and count approximately, solutions to the zero-one knapsack problem. The algorithm is based on using dynamic programming to provide a *deterministic* relative approximation. Then "dart throwing" techniques are used to give arbitrary approximation ratios. We extend this approach to several related problems: the m-constraint zero-one knapsack, the general integer knapsack (including its m-constraint version) and contingency tables with constantly many rows. We also indicate how further improvements can be obtained using randomized rounding.

1 Introduction

In this paper we describe efficient algorithms to uniformly sample and approximately count solutions to the zero-one knapsack problem and some related problems. Specifically we address the multiple constraint version of zero-one knapsack, the general integer knapsack with arbitrary upper bounds (both single and multiple constraint), and contingency tables with a constant number of rows. In each case the algorithms are based on a dynamic programming computation which provides a *deterministic* approximation ratio of polynomial size. Then simple "dart throwing" techniques give arbitrary approximation ratios.

Previous approaches to the problems we discuss here have been based almost exclusively on the Markov chain Monte Carlo (MCMC) approach. (See, for example, the survey of Jerrum and Sinclair [12].) One exception is the algorithm of Cryan and Dyer [2] for contingency tables with constantly many rows. This combines dynamic programming with volume approximation, but the approximate volume computation does itself involve MCMC methods. In fact, few results are known in approximate counting which do not rely on MCMC. Of course, the foundational paper of Karp and Luby [14] did not use MCMC, and the technique developed there for estimating a union of sets has been employed elsewhere. The papers of Karger [13] and Gore *et al* [9] use this idea, combined with application-specific methods.

*Supported by the EPSRC grant "Sharper Analysis of Randomized Algorithms: a Computational Approach" and by the EC IST network RAND-APX.

The zero-one knapsack problem has been approached by MCMC. The best result known, due to Morris and Sinclair [17, 18], gives sampling in time $O(n^{9/2+\epsilon})$, for any $\epsilon > 0$, for a problem with n variables. In section 2.1 we give an $O(n^3)$ time sampling algorithm and a fully polynomial randomized approximation scheme (*fpras*), with relative error ε, running in time $O(n^3 + \varepsilon^{-2}n^2)$, i.e. essentially the same time bound. In section 2.2 we show how this can be improved further to $O(n^{5/2}\sqrt{\log(\varepsilon^{-1})} + n^2\varepsilon^{-2})$ using randomized rounding.

The multiple constraint versions of the knapsack problem have also been considered previously [6, 16, 18]. Here the best result known, again due to Morris and Sinclair [16, 18], gives sampling with time bound $n^{2^{O(m)}}$. We improve this substantially in section 2.1 to give sampling, and an *fpras*, with a time bound of $O(n^{2m+1})$.

The general integer knapsack problem has perhaps been less studied from the viewpoint of approximate counting. The analysis of [6] was extended to this case, but the time bound for the Markov chain given there is $2^{O^*(\sqrt{n})}$ when there are n variables. It is likely that the methods of [16, 18] apply to this problem but it seems that this has not yet been done. In section 2.3 we show that the single constraint problem has an $O(n^5)$ sampling algorithm and an *fpras* with similar running time. Again, we can easily generalise these results to give running times $O(n^{2m+3})$ for the m-constraint case.

The problems of sampling and counting *contingency tables* have been widely studied. (See, for example, [1, 2, 3, 5, 7, 8, 10, 16].) The practical relevance of this problem was discussed by Diaconis and Efron [4]. It is still not known how to sample general contingency tables uniformly in polynomial time, but algorithms are known for some special cases. In particular, the case where the number of rows is considered to be a constant has been examined in [2, 3, 7, 10]. The previous best result in this area is the algorithm of Cryan and Dyer, which gives sampling in time $n^{O(m^2)}$ for a problem with m rows and n columns. In section 3 we improve this to $O(n^{4m+1})$, and give an *fpras* with similar time bound.

Dynamic programming has been used to construct FPTAS's for *optimization* problems of types similar to those we study here. (See Woeginger [20] and its references.) It seems that the reason the technique also works for counting is rather different. Nevertheless, there is a possibility of extending our results to other problems where dynamic programming FPTAS's exist for optimization.

2 Approximately counting knapsack solutions

Throughout, $\mathbb{N}$ will denote the set of all *non-negative* integers. For integers $i \leq j$, we will use $[i, j]$ to denote the set of integers $\{i, \ldots, j\}$, and $[j]$ to denote the set $[1, j]$ for $1 \leq j$.

2.1 The zero-one case

Let S denote the solution set of

$$\sum_{j=1}^{n} a_j x_j \leq b, \quad \text{with } x \in B_n = \{0,1\}^n,$$

where $0 \le a_1 \le a_2 \le \cdots \le a_n \le b$ are integers.[1]

Let k be such that $a_j \le b/n$ for $j \le k$ and either $a_{k+1} > b/n$ or $k = n$. Let $C = \{0,1\}^k \times \{0\}^{n-k}$. If $x \in C$ then $\sum_{j=1}^n a_j x_j \le \sum_{j=1}^k a_j \le kb/n \le b$, so $x \in S$. Thus $C \subseteq S$.

Let $\alpha_j = \lfloor n^2 a_j / b \rfloor$ and $\delta_j = n^2 a_j/b - \alpha_j$, so $0 \le \delta_j < 1$. Let S' be the solution set of

$$\sum_{j=1}^{n} \alpha_j x_j \le n^2, \quad \text{with } x \in B_n,$$

Now $|S'|$ can be determined in $O(n^3)$ time, using dynamic programming. Write $F(r,s) = |\{x \in B_r : \sum_{j=1}^r \alpha_j x_j \le s\}|$. In $O(n^3)$ time, the dynamic programming tabulates $F(r,s)$ $(1 \le r \le n,\ 0 \le s \le n^2)$, using the recursion

$$F(r,s) = F(r-1,s) + F(r-1, s-\alpha_r) \quad (r \ge 2), \qquad F(1,s) = \begin{cases} 1 & \text{if } s < \alpha_1, \\ 2 & \text{otherwise.} \end{cases}$$

Then we have$|S'| = F(n, n^2)$.

If $x \in S$, $\sum_{j=1}^n \alpha_j x_j \le (n^2/b) \sum_{j=1}^n a_j x_j \le (n^2/b) b = n^2$, so $x \in S'$. Thus $S \subseteq S'$ and $|S| \le |S'|$. If $S' \ne S$, suppose $x \in S' \setminus S$. Then clearly there exists an integer $p(x)$ such that $x_p = 1$ and $p \notin [k]$. Otherwise $x \in C \subseteq S \subseteq S'$, a contradiction. If there is more than one such integer, take $p(x)$ to be the smallest. Note that we have $\alpha_p \ge n$.

Define a map $f : S' \to B_n$, as follows. If $x \in S$ then $f(x) = x$. Otherwise $x \in S' \setminus S$, and $p(x)$ is well defined. Define $f(x) = y$, where $y_j = x_j$ for $j \ne p(x)$, and $y_p = 0$. If $x \in S' \setminus S$ then, with $y = f(x)$,

$$\begin{aligned} \sum_{j=1}^{n} a_j y_j &= \frac{b}{n^2} \sum_{j=1}^{n} (\alpha_j + \delta_j) y_j = \frac{b}{n^2} \Big(\sum_{j=1}^{n} \alpha_j y_j + \sum_{j=1}^{n} \delta_j y_j \Big) \\ &= \frac{b}{n^2} \Big(\sum_{j=1}^{n} \alpha_j x_j - \alpha_p + \sum_{j=1}^{n} \delta_j y_j \Big) \\ &\le \frac{b}{n^2} \Big(n^2 - n + n \Big) = b, \end{aligned}$$

so $f(x) \in S$. Hence $f(S') = S$. But, for $y \in S$, we have $|f^{-1}(y)| \le (n+1)$, since any element of $f^{-1}(y)$ may change a single coordinate of y or none. Thus $|S'| = |f^{-1}(S)| \le (n+1)|S|$. Hence $1 \le |S'|/|S| \le (n+1)$ so $|S'|/\sqrt{n+1}$ approximates $|S|$ deterministically within a factor $\sqrt{n+1}$ and can be computed in $O(n^3)$ time. Since knapsack is obviously self-reducible, existence of an *fpras* for the problem now follows from a general result of Sinclair and Jerrum [12]. However, we will describe a simpler and more efficient "dart-throwing" method.

The $F(r,s)$ table can be used to determine a uniform sample from S' in $O(n)$ time, by tracing back probabilistically from $F(n,n^2)$, as follows. Set $x_n = 0$ with probability $F(n-1, n^2)/F(n,n^2)$, else set $x_n = 1$ with the remaining probability $F(n-1, n^2 - \alpha_n)/F(n,n^2)$.

[1] A single (rational) linear inequality in zero-one variables can always be put in this form.

If $x_n = 0$, recursively determine $x_{n-1}, x_{n-2}, \ldots, x_2, x_1$ by tracing back from $F(n-1, n^2)$ and, if $x_n = 1$, similarly trace back recursively from $F(n-1, n^2 - \alpha_n)$. The resulting point of S' has probability at least $1/(n+1)$ of lying in S. If so, it is uniformly distributed in S, and we accept it. Otherwise we repeat the whole process independently. After $n+1$ repetitions we have a sample with probability at least $1 - e^{-1}$. Hence a sample of ν uniform points can be determined in $O(n^3 + n^2\nu)$ time[2] with probability at least $1 - e^{-\Omega(n)}$.

To have an *fpras* for $|S|$, we need only estimate the probability $\rho = |S|/|S'| \geq 1/(n+1)$, since $|S'| = F(n, n^2)$. With ν points, the sampling error is $O(1/\sqrt{\nu n})$. We require this to be smaller than $\varepsilon\rho = \Omega(\varepsilon/n)$. Hence we need $\nu = O(\varepsilon^{-2}n)$. The complexity of the *fpras* is then $O(n^3 + \varepsilon^{-2}n^2)$.

The m-constraint (or *multidimensional*) knapsack problem,

$$S = \bigcap_{i=1}^{m} S_i, \quad \text{where} \quad S_i = \{x \in B_n : \sum_{j=1}^{n} a_{ij}x_j \leq b_i\},$$

with $a_{ij} \geq 0$ ($i \in [m]$, $j \in [n]$)[3], can be solved by the same technique if m is a constant. Let $S' = \bigcap_{i=1}^{m} S'_i$, where $S'_i = \{x \in B_n : \sum_{j=1}^{n} \alpha_{ij}x_j \leq n^2\}$ with $\alpha_{ij} = \lfloor n^2 a_{ij}/b_i \rfloor$. We can show $S \subseteq S'$ exactly as before.

Let $K_i = \{j : a_{ij} \leq b_i/n\}$. For $x \in S' \setminus S$, let $I(x) = \{i : x \in S'_i \setminus S_i\}$. As before, for every $i \in I(x)$, there exists $p_i(x) \notin K_i$ such that $x_{p_i} = 1$. Construct $f(x) = y$ by $y_{p_i(x)} = 0$ for $i \in I(x)$ and $y_j = x_j$ otherwise. Then it can be shown as before that $f(x) \in S$. The inverse mapping changes some set of coordinates P with $0 \leq |P| \leq m$, so

$$|f^{-1}(y)| \leq 1 + n + \binom{n}{2} + \cdots + \binom{n}{m} \leq n^m \qquad (m, n \geq 2).$$

The dynamic programming computation to determine $|S'|$ takes $O(n^{2m+1})$ time. Using the same ideas as before, we can obtain a uniform sample of size ν from S in time $O(n^{2m+1} + n^{m+1}\nu)$, and an *fpras* for approximate counting which takes $O(n^{2m+1} + \varepsilon^{-2}n^{m+1})$ time.

2.2 An improvement using randomized rounding

We will show how to reduce the running time of the *fpras* for the zero-one knapsack problem by (almost) a $\sqrt{n}$ factor, using randomized rounding. If ε is as above, and η is a (small) constant error probability, let $K = \sqrt{-\frac{1}{2}n\ln(\eta\varepsilon)}$, and $\hat{\alpha}_j = \lfloor 2nK\alpha_j/b \rfloor$, $\delta_j = 2nK\alpha_j/b - \hat{\alpha}_j$. Now let W_j be independent random variables such that

$$\Pr(W_j = 1) = 1 - \Pr(W_j = 0) = \delta_j \qquad (j \in [n]),$$

and let $\alpha_j = \hat{\alpha}_j + W_j$. Hence we have $2nKa_j/b = \alpha_j + \delta_j - W_j$, and $\mathbf{E}[\alpha_j] = 2nKa_j/b$. Let S be defined as before and let $S' = \{x \in B_n : \sum_{j=1}^{n} \alpha_j x_j \leq (2n+1)K\}$. Then $|S'|$ can be determined by dynamic programming as before. We first show that

[2] Here and elsewhere we count arithmetic operations, rather than operations on bits.

[3] The problem provably has no *fpras* without this assumption, even if $b_i > 0$ for all $i \in [m]$.

Lemma 1. $\Pr(|S' \cap S| < (1-\varepsilon)|S|) \leq \eta$.

Proof. If $x \in S$, we bound $\Pr(x \notin S')$ as follows.

$$\sum_{j=1}^{n} \alpha_j x_j = \sum_{j=1}^{n} (2nKa_j/b - \delta_j + W_j)x_j \leq 2nK + \sum_{j=1}^{n} (W_j - \delta_j)x_j \leq (2n+1)K$$

with probability $e^{-2K^2/n} = \eta\varepsilon$, using the Hoeffding bound [11]. Summing over $x \in S$, we have $\mathbf{E}[\,|S \setminus S'|\,] \leq \eta\varepsilon|S|$. Hence, by the Markov inequality,

$$\Pr(|S \setminus S'| > \varepsilon|S|) \leq \frac{\eta\varepsilon|S|}{\varepsilon|S|} = \eta,$$

which implies the Lemma. □

We define $f : S' \to B_n$ as follows. If $x \in S' \cap S$, we set $f(x) = x$ and, if $x \in S' \setminus S$, let k be such that $a_k x_k > b/n$. Now define $y = f(x)$ by setting $x_k = 0$ as before.

Lemma 2. $\Pr(|S' \cap f^{-1}(S)| < (1-\varepsilon)|S'|) \leq \eta$.

Proof. If $x \in S' \setminus f^{-1}(S)$, then $x \in S'$ but $f(x) = y \notin S$. We bound $\Pr(y \notin S)$ as follows.

$$\begin{aligned}
\sum_{j=1}^{n} a_j y_j &\leq \sum_{j=1}^{n} a_j x_j - \frac{b}{n} = \frac{b}{2nK}\Big(\sum_{j=1}^{n} (\alpha_j + \delta_j - W_j)y_j - 2K\Big) \\
&\leq \frac{b}{2nK}\Big((2n-1)K + \sum_{j=1}^{n} (\delta_j - W_j)y_j\Big) \leq \frac{b}{2nK}(2nK) = b,
\end{aligned}$$

with probability $e^{-2K^2/n} = \eta\varepsilon$, again using the Hoeffding bound [11]. Summing over $x \in S'$, we have $\mathbf{E}[\,|S' \setminus f^{-1}(S)|\,] \leq \eta\varepsilon|S'|$. Now, by the Markov inequality,

$$\Pr(|S' \setminus f^{-1}(S)| > \varepsilon|S'|) \leq \frac{\eta\varepsilon|S'|}{\varepsilon|S'|} = \eta,$$

giving the Lemma. □

Thus, with probability at least $(1 - 2\eta)$, we have $|S' \cap S| \geq (1-\varepsilon)|S|$ and $|S' \cap f^{-1}(S)| \geq (1-\varepsilon)|S'|$. From these we may deduce

$$n|S| \geq |f^{-1}(S)| \geq |S' \cap f^{-1}(S)| \geq (1-\varepsilon)|S'|,$$

and hence $\rho = |S \cap S'|/|S'| \geq (1-\varepsilon)|S|/|S'| \geq (1-\varepsilon)^2/n$. Hence, by sampling from S', we can determine and estimate $\hat{\rho}$ of ρ satisfying $\leq \hat{\rho} \leq \rho\sqrt{1-\varepsilon}$ in $O(n^2/\varepsilon^2)$ time. Then let $\Psi = \hat{\rho}|S'|/\sqrt{1-\varepsilon}$ be our estimate of $|S|$. We have

$$\Psi \leq \frac{\rho|S'|}{1-\varepsilon} = \frac{|S' \cap S|}{1-\varepsilon} \leq \frac{|S|}{1-\varepsilon}, \quad \text{and} \quad \Psi \geq \rho|S'| = |S' \cap S| \geq (1-\varepsilon)|S|.$$

So we have an *fpras*, with overall running time $O(n^{5/2}\sqrt{\log(\varepsilon^{-1})} + n^2\varepsilon^{-2})$.

There are analogous improvements for the other problems we study, but we will not consider them here.

2.3 The general case

Let $U_r = \{\ 0 \le x_j \le u_j,\ j \in [r]\}$, where the u_j are given integers. We want to determine $|S|$, where

$$S = \{x : \sum_{j=1}^{n} a_j x_j \le b,\ x \in U_n\},$$

with $a_1, \ldots, a_n, b > 0$ given integers. Note that we can assume $u_j \le \lfloor b/a_j \rfloor$.

Let $h_j(x_j) = \lfloor 2n^2 a_j x_j / b \rfloor$ $(0 \le x_j \le u_j,\ j \in [n])$, and

$$S' = \{x : \sum_{j=1}^{n} h_j(x_j) \le 2n^2,\ x \in U_n\}.$$

Now let $C = \{x : a_j x_j \le b/n,\ j \in [n]\}$. It follows easily that $C \subseteq S \subseteq S'$. Thus, if $x \in S' \backslash S$, there exists $p(x)$ such that $a_p x_p > b/n$. Note that $h_p(x_p) \ge 2n$. Define $f : S' \to U$, by $f(x) = x$ if $x \in S$ and $f(x) = y$ otherwise, where $y_j = x_j$ for $j \ne p(x)$ and $y_p = \lfloor x_p/2 \rfloor$. Now, if $y = f(x)$ and $p = p(x)$,

$$\begin{aligned}
\sum_{j=1}^{n} a_j y_j &= \frac{b}{2n^2} \sum_{j=1}^{n} 2n^2 a_j y_j / b \;=\; \frac{b}{2n^2}\Big(\sum_{j \ne p} 2n^2 a_j x_j / b + 2n^2 a_p \lfloor x_p/2 \rfloor / b\Big) \\
&\le \frac{b}{2n^2}\Big(\sum_{j \ne p} (h_j(x_j) + 1) + n^2 a_p x_p / b\Big) \\
&\le \frac{b}{2n^2}\Big(\sum_{j \ne p} h_j(x_j) + n - 1 + \tfrac{1}{2}(h_p(x_p) + 1)\Big) \\
&\le \frac{b}{2n^2}\Big(\sum_{j \ne p} h_j(x_j) + n - 1 + h_p(x_p) - n + \tfrac{1}{2}\Big) \\
&\le \frac{b}{2n^2}\Big(2n^2 - \tfrac{1}{2}\Big) < b.
\end{aligned}$$

Thus $f(S') = S$. But $|f^{-1}(y)| \le 2n + 1$, since $y \in f^{-1}(y)$ and, for any $1 \le p \le n$, there are at most two possible values of x_p.

We calculate $F(r, s) = |\{x \in U_r : \sum_{j=1}^{r} h_j(x_j) \le s\}|$ by dynamic programming, with $|S'| = F(n, 2n^2)$. Let $\kappa_j = \dfrac{b}{2n^2 a_j}$, $\tau_j = \lfloor u_j/\kappa_j \rfloor \le 2n^2$. Then

$$\Delta_j(t) = |\{x_j : h_j(x_j) = t\}| = \lceil (t+1)\kappa_j \rceil - \lceil t\kappa_j \rceil \quad (0 \le t < \tau_j),$$

and $\Delta_j(\tau_j) = u_j - \lceil t\kappa_j \rceil$. Now the recurrence is

$$F(r, s) = \sum_{t=0}^{\tau_r} \Delta_r(t) F(r-1, s-t), \quad F(1, s) = 1 + \min\big[\lfloor s\kappa_1 \rfloor, u_1\big].$$

The table $F(r, s)$ $(1 \le r \le n,\ 0 \le s \le 2n^2)$ can be determined in $O(n^5)$ time. The probabilistic traceback takes $O(n^3)$ time, so we can generate a sample of size ν from S in

$O(n^5+n^4\nu)$ time. Again we need $O(\varepsilon^{-2}n)$ samples from S' for an *fpras* giving $O(n^5+\varepsilon^{-2}n^4)$ time. The generalisation to the m-constraint version is similar to the zero-one case, and leads to $O(n^{2m+3}+\nu n^{m+3})$ time for a sample of size ν, and $O(n^{2m+3}+\varepsilon^{-2}n^{m+3})$ time for an *fpras*.

3 Contingency tables with few rows

We consider $m \times n$ tables, where m is constant. We will assume $m \le n$, otherwise we will transpose the table. The row totals are r_i, $i \in [m]$, and the column totals c_j, $j \in [n]$, and $N = \sum_{i=1}^m r_i = \sum_{j=1}^n c_j$. Assume, without loss of generality, that $c_n = \max_{j=1}^n c_j$. We will also assume that $c_n \ge m^5$. Otherwise, it is possible to count *exactly* by dynamic programming in $O(n^{m+1})$ time, for constant m.

We denote the i,jth element of a matrix $x \in \mathbb{N}^{m\times n}$ by x_{ij}, and its jth column by x_j ($i \in [m], j \in [n]$). Also $x^* \in \mathbb{N}^{m\times(n-1)}$ denotes x with its nth column deleted. Let

$$X_j = \big\{x_j \in \mathbb{N}^m \ : \ \sum_{i=1}^m x_{ij} = c_j\big\} \qquad (j \in [n-1]),$$

and

$$X = \big\{x^* \ : \ x_j \in X_j,\ j \in [n-1]\big\}.$$

Letting $r = (r_1,\dots,r_m)$, the set S of contingency tables with totals r_i, c_j can be written

$$S = \big\{x^* \in X \ : \ \sum_{j=1}^{n-1} x_j \le r\big\}.$$

Let $h_j : X_j \to \mathbb{N}^m$ be defined by $[h_j(x_j)]_i = \lfloor 2n^2 x_{ij}/r_i \rfloor$ ($i \in [m]$), and let

$$S' = \big\{x^* \in X \ : \ \sum_{j=1}^{n-1} h_j(x_j) \le 2n^2\mathbf{1}\big\},$$

where $\mathbf{1}$ is the m-vector of 1's. Clearly $S \subseteq S'$.

For $t \in T = [0, 2n^2]^m$, we can calculate $F(k,t) = |\{x^* \in X : \sum_{j=1}^k h_j(x_j) \le t\}|$ by dynamic programming, with $|S'| = F(n-1, 2n^2\mathbf{1})$.

Let $\xi_i(t_i) = \left\lceil \dfrac{r_i t_i}{2n^2} \right\rceil$ and, for any $j \in [n-1]$, define

$$\Delta_j(t) = |\{x_j \in X_j : h_j(x_j) = t\}| = |\{x_j \in X_j : \xi_i(t_i) \le x_{ij} < \xi_i(t_i+1),\ i \in [m]\}|.$$

Then, if $s \in T$, the recurrence is

$$F(k,s) = \sum_{t\in T} \Delta_k(t) F(k-1, s-t), \quad F(1,s) = \Delta_1(s).$$

The table $F(k,s)$ ($k \in [n-1],\ s \in T$) can be determined in $O(n^{4m+1}D)$ time, where D is the time needed to determine $\Delta_j(t)$.

Lemma 3. *$\Delta_j(t)$ can be determined in $O(m2^m)$ arithmetic operations.*

Proof. Note that each $\Delta_j(t)$ is of the form

$$M = |\{\zeta = (\zeta_1, \ldots, \zeta_m) \in \mathbb{Z}^m \ : \ 0 \le \zeta_i \le u_i \ (i \in [m]), \ \sum_{i=1}^{m} \zeta_i = \xi\}|. \qquad (1)$$

For $\sigma \in \{0,1\}^m$, let $e(\sigma) = \sum_{i=1}^m \sigma_i$, $z(\sigma) = \xi - \sum_{i=1}^m \sigma_i(u_i + 1)$, and

$$Z(\sigma) = \big\{\zeta : \zeta_i \ge 0 \ (i \in [m]), \ \sum_{i=1}^{m} \zeta_i = z(\sigma)\big\}.$$

Note that $|Z(\sigma)| = \binom{z(\sigma)+m-1}{m-1}$. Now, using the principle of inclusion-exclusion, we have

$$M = \sum_{\sigma \in \{0,1\}^m} (-1)^{e(\sigma)} |Z(\sigma)| = \sum_{\sigma \in \{0,1\}^m} (-1)^{e(\sigma)} \binom{z(\sigma) + m - 1}{m - 1}.$$

Each term in the sum can be calculated in $O(m)$ arithmetic operations, and there are 2^m terms. ☐

Hence, for constant m, all $F(k, s)$ can be calculated in $O(n^{4m+1})$ time. Hence we can determine $|S'|$. The probabilistic traceback takes $O(n^{2m+1})$ time, given that we can select uniformly from sets of the form (1) in constant time (for fixed m). We will consider this point later.

We now construct the mapping f from S' to S. This is not as straightforward as the construction of the mappings in section 2. If $x^* \in S$, then we set $f(x^*) = x^*$. Otherwise let $I(x^*) = \{i : \sum_{j=1}^{n-1} x^*_{ij} > r_i\}$. For $i \in I(x^*)$, let $p(i)$ be such that $x^*_{ip} = \max_{j=1}^{n-1} x^*_{ij} > r_i/n$. Now, for any $x^* \in S'$,

$$\sum_{j=1}^{n-1} x^*_{ij} \le \frac{r_i}{2n^2} \sum_{j=1}^{n-1} (h_{ij}(x^*_{ij}) + 1) < \frac{r_i}{2n^2}(2n^2 + n) = \Big(1 + \frac{1}{2n}\Big) r_i.$$

Now let $J = \{j : j = p(i)$ for some $i \in I(x^*)\}$ and $\ell = |J| + 1$. Note that $\ell \le m$, since $x_j \in X_j$, $j \in [n-1]$, implies that $[m] \setminus I(x^*) \ne \emptyset$. Let $\hat{y}_{ij} = 0$ $(i \in [m], j \in J)$, $\hat{y}_{ij} = x^*_{ij}$ otherwise. Then $\rho_i = r_i - \sum_{j=1}^{n-1} \hat{y}_{ij} \ge 0$.

If $i \in I(x^*)$, $\rho_i \ge x^*_{ip} - \frac{1}{2n} r_i \ge \frac{1}{2n} r_i$. Thus, if $i \in I(x^*)$, $j \in J$, $x^*_{ij} \le x^*_{ip} \le \rho_i + \frac{1}{2n} r_i \le 2\rho_i$. If $i \notin I(x^*)$, $j \in J$, then clearly $x^*_{ij} \le \rho_i$. Also, $x^*_j \in X_j$ implies $x^*_{ij} \le c_j$. Thus we have $x^*_{ij} \le 2\min(\rho_i, c_j)$ for any $i \in [m]$, $j \in J$.

Let $J' = J \cup \{n\}$. We use the columns in J' to "complete" $\hat{y}$ to a contingency table y. We can do this by setting these columns to any $m \times \ell$ contingency table with row sums ρ_i, $i \in [m]$, and column sums c_j, $j \in J'$. The method we use is similar to that in [3]. Let $N' = \sum_{i=1}^m \rho_i = \sum_{j \in J'} c_j$, and let us assume without loss that the rows are re-numbered if necessary so that $\rho_m = \max_{i=1}^m \rho_i$. Note that $\rho_m \ge N'/m$ and $c_n \ge N'/\ell$.

Let $a_{ij} = \lfloor \min(\rho_i, c_j)/m^3 \rfloor$, $Q_{ij} = \lfloor x^*_{ij}/(a_{ij}+1) \rfloor \leq 2m^3$, $R_{ij} = x^*_{ij} \bmod (a_{ij}+1)$, so that $x^*_{ij} = (a_{ij}+1)Q_{ij} + R_{ij}$, for $i \in [m-1], j \in J$. Let

$$Q'_{ij} = \left\lfloor \frac{\rho_i c_j}{N'(a_{ij}+1)} \right\rfloor, \quad z_{ij} = (a_{ij}+1)Q'_{ij} + R_{ij} \qquad (i \in [m-1], j \in J).$$

We must show that the z_{ij} can be completed to a contingency table. This will be so if, and only if,

$$\sum_{j \in J} z_{ij} \;\leq\; \rho_i \quad (i \in [m-1]), \tag{2}$$

$$\sum_{i \in [m-1]} z_{ij} \;\leq\; c_j \quad (j \in J), \tag{3}$$

$$\sum_{i \in [m-1]} \sum_{j \in J} z_{ij} \;\geq\; N' - \rho_m - c_n. \tag{4}$$

For (2),

$$\sum_{j \in J} z_{ij} \;\leq \sum_{j \in J} \left(\frac{\rho_i c_j}{N'} + a_{ij} \right) \leq \sum_{j \in J} \left(\frac{\rho_i c_j}{N'} + \frac{\rho_i}{m^3} \right) < \rho_i \left(1 - \frac{1}{\ell} + \frac{1}{m^2} \right) < \rho_i.$$

Similarly, for (3),

$$\sum_{i=1}^{m-1} z_{ij} \;\leq\; \sum_{i=1}^{m-1} \left(\frac{\rho_i c_j}{N'} + a_{ij} \right) \leq \sum_{i=1}^{m-1} \left(\frac{\rho_i c_j}{N'} + \frac{c_j}{m^3} \right) < c_j \left(1 - \frac{1}{m} + \frac{1}{m^2} \right) < c_j.$$

Finally, for (4),

$$\begin{aligned}
\sum_{i=1}^{m-1} \sum_{j \in J} z_{ij} \;&>\; \sum_{i=1}^{m-1} \sum_{j \in J} \left(\frac{\rho_i c_j}{N'} - (a_{ij}+1) \right) \;\geq\; \sum_{i=1}^{m-1} \sum_{j \in J} \left(\frac{\rho_i c_j}{N'} - \frac{c_j}{m^3} - 1 \right) \\
&\geq\; \frac{(N' - \rho_m)(N' - c_n)}{N'} - \frac{N' - c_n}{m^2} - m^2 \\
&\geq\; N' - \rho_m - c_n + \frac{N'}{m^2} - \frac{N'}{m^2} + \frac{N'}{m^3} - m^2 \;\geq\; N' - \rho_m - c_n,
\end{aligned}$$

provided $N' \geq m^5$. But this is implied by our assumption that $c_n \geq m^5$.

We can now define the mapping f. Let $f(x^*) = y^*$, where $y^*_j = x^*_j$, $j \notin J$, and $y^*_j = z_j$, $j \in J$. For the inverse mapping, the set J contains at most $(m-1)$ of the $(n-1)$ columns, and there are at most n^{m-1} ways of selecting it. Given J, the ρ_i, $i \in [m]$, can be calculated, and we determine the largest ρ_i, which we assume to be ρ_m. Now the a_{ij} can be determined, and hence the R_{ij} from $R_{ij} = y^*_{ij} \bmod (a_{ij}+1)$, for $i \in [m-1], j \in J$. Finally we must select the Q_{ij}. Since $Q_{ij} \in [0, 2m^3]$, $i \in [m-1], j \in J$, there are at most $(2m^3+1)^{(m-1)^2}$ ways of selecting them all. For fixed m this is constant. Thus $|f^{-1}(y^*)| = O(n^{m-1})$.

We can generate as before by tracing back. However, at each stage we now need to generate a random point in a set M of the form (1). Let us defer this issue temporarily, and suppose we can do this in constant time for fixed m. It then follows, using the same ideas as before, that a sample of ν tables can be computed in time $O(n^{4m+1} + \nu\, n^{3m})$, and that there is an *fpras* with running time $O(n^{4m+1} + \varepsilon^{-2} n^{3m})$.

Let us now return to the question of generation in the trace back. Fortunately, we can use the method above to bootstrap itself. Note that any set of the form (1) is (essentially) the set of solutions to a $2 \times m$ contingency table, with row sums ξ, $\sum_{i=1}^{m} u_i - \xi$ and column sums $u_1, u_2, \ldots, u_m$. Thus the method above will generate a point in $O(m^9)$ time, provided that we can trace back when there are only two rows. Thus we need to generate a uniform point in

$$\zeta_1 + \zeta_2 = \xi, \quad 0 \le \zeta_1 \le u_1,\ 0 \le \zeta_2 \le u_2.$$

But this is straightforward. We choose $\zeta_1 \in [\max(0, \xi - u_2), \min(u_1, \xi)]$ uniformly at random, and then $\zeta_2 = \xi - \zeta_1$.

Finally, observe that this method for generation could in fact be used to count approximately in the dynamic programming phase. Thus we use Lemma 1 only for tables with two rows. It is then possible to make the implied constants in the time bounds for the sampling and approximation algorithms have polynomial dependence on m. We omit the details, since the appearance of m in the exponent of n makes this an issue of lesser importance.

References

[1] F. Chung, R. Graham and S.-T. Yau, On sampling with Markov chains. *Random Structures & Algorithms*, **9**, 1996, pp 55–77.

[2] M. Cryan and M. Dyer, A polynomial-time algorithm to approximately count contingency tables when the number of rows is constant. In *Proceedings of the 34th Annual Symposium on Theory of Computing*, 2002, pp 240–249.

[3] M. Cryan, M. Dyer, L. Goldberg, M. Jerrum and R. Martin, Rapidly mixing Markov chains for sampling contingency tables with a constant number of rows. To appear in *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, 2002. Full version available at `http://www.dcs.warwick.ac.uk/~leslie/papers/contingency.ps`.

[4] P. Diaconis and B. Efron, Testing for independence in a two-way table: new interpretations of the chi-square statistic (with discussion). *Annals of Statistics*, **13**, 1995, pp 845–913.

[5] P. Diaconis and L. Saloff-Coste, Random walk on contingency tables with fixed row and column sums. Technical Report, Department of Mathematics, Harvard University, 1995.

[6] M. Dyer, A. Frieze, R. Kannan, A. Kapoor, L. Perkovic and U. Vazirani, A mildly exponential algorithm for estimating the number of knapsack solutions. *Combinatorics, Probability and Computing*, **2**, 1993, pp 271–284.

[7] M. Dyer and C. Greenhill, Polynomial-time counting and sampling of two-rowed contingency tables. *Theoretical Computer Science*, **246**, 2000, pp 265–278.

[8] M. Dyer, R. Kannan and J. Mount, Sampling contingency tables. *Random Structures & Algorithms*, **10**, 1997, pp 487–506.

[9] V. Gore, M. Jerrum, S. Kannan, Z. Sweedyk and S. Mahaney, A quasi-polynomial time algorithm for sampling words from a context-free Language. *Information and Computation*, **134**, 1997, pp 59–74.

[10] D. Hernek, Random generation of $2 \times n$ contingency tables. *Random Structures & Algorithms*, **13**, 1998, pp 71–79.

[11] W. Hoeffding, Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, **58**, 1963, pp 13–30.

[12] M. Jerrum and A. Sinclair, Markov chain Monte Carlo method: an approach to approximate counting and integration. D.S. Hochbaum (Ed), *Approximation Algorithms for NP-Hard Problems*, PWS, Boston, 1997, pp 482–520.

[13] D. Karger, A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM J. Comput.*, bf 29, 1999, pp 492–514.

[14] R. Karp and M. Luby, Monte Carlo algorithms for enumeration and reliability problems. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, 1983, pp 56-64.

[15] B. Morris, Improved bounds for sampling contingency tables. *3rd International Workshop on Randomization and Approximation Techniques in Computer Science*, volume 1671 of *Lecture Notes in Computer Science*, 1999, pp 121-129.

[16] B. Morris, *Random Walks in Convex Sets*. PhD thesis, Department of Statistics, University of California, Berkeley, 2000.

[17] B. Morris and A.J. Sinclair, Random walks on truncated cubes and sampling 0-1 knapsack solutions. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, 1999, pp 230–240.

[18] B. Morris and A.J. Sinclair, Random walks on truncated cubes and sampling 0-1 knapsack solutions, July 2002, submitted. (An extended and improved version of [17].)

[19] A. Sinclair and M. Jerrum, Approximate counting, uniform generation, and rapidly mixing Markov chains. *Information and Computation*, **82**, 1989, pp 93–133.

[20] G. Woeginger, When does a dynamic programming formulation guarantee the existence of an FPTAS ?,*Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, 1999, pp 820–829.