COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **22** (3) (2002) 417–433

IMPROVED PSEUDORANDOM GENERATORS FOR COMBINATORIAL RECTANGLES

CHI-JEN LU

Received July 29, 1998

We construct a pseudorandom generator which uses $O(\log m + \log d + \log^{3/2} 1/\varepsilon)$ bits and approximates the volume of any combinatorial rectangle in $\{1,\ldots,m\}^d$ to within ε error. This improves on the previous construction using $O(\log m + \log d + \log^2 1/\varepsilon)$ bits by Armoni, Saks, Wigderson, and Zhou [4]. For a subclass of rectangles with at most $t \geq \log 1/\varepsilon$ nontrivial dimensions and each dimension being an interval, we also give a pseudorandom generator using $O(\log\log d + \log 1/\varepsilon \log^{1/2} \frac{t}{\log 1/\varepsilon})$ bits. This again improves the previous upper bound $O(\log\log d + \log 1/\varepsilon \log \frac{t}{\log 1/\varepsilon})$ by Chari, Rohatgi, and Srinivasan [5].

1. Introduction

Pseudorandom generators for combinatorial rectangles have been actively studied recently, because they are closely related to some fundamental problems in theoretical computer science, such as derandomizing RL, DNF approximate counting, and approximating the distributions of independent multivalued random variables.

Let V be a finite set with uniform distribution. The *volume* of a set $A \subseteq V$ is defined as

$$vol(A) = P_{x \in V}[x \in A] = \frac{|A|}{|V|}.$$

Let $\mathcal{A}$ be a family of subsets from V. We want to sample from a much smaller space, instead of from V, and still be able to approximate the volume of any

Mathematics Subject Classification (2000): 05B40, 68Q25, 68R05

subset $A \in \mathcal{A}$. We call a deterministic function $g : \{0,1\}^\ell \to V$ an *ε-generator* using ℓ bits for $\mathcal{A}$, if for all $A \in \mathcal{A}$,

$$|P_{y \in \{0,1\}^\ell}[g(y) \in A] - vol(A)| \leq \varepsilon.$$

For a positive integer k, let $[k]$ denote the set $\{1, \ldots, k\}$. For positive integers m and d, a *combinatorial rectangle* of type (m,d) is a subset of $[m]^d$ of the form $R_1 \times \cdots \times R_d$, where $R_i \subseteq [m]$ for all $i \in [d]$. Let $\mathcal{R}(m,d)$ denote the family of all such rectangles. The volume of a rectangle $R \in \mathcal{R}(m,d)$ is now $\prod_{i \in [d]} \frac{|R_i|}{m}$. Our goal is to find explicit ε-generators with small ℓ for $\mathcal{R}(m,d)$ and its subclasses.

As observed by Even, Goldreich, Luby, Nisan, and Veličković [6], this is a special case of constructing pseudorandom generators for RL. Nisan's generator for RL [14] is currently the best, using $O(\log^2 n)$ bits. Because it has many important applications and no improvement has been made for several years, one might hope that solving this special case could shed some light on the general problem.

It's easy to show that a random function mapping from $O(\log m + \log d + \log 1/\varepsilon)$ bits to $[m]^d$ is very likely to be an ε-generator for $\mathcal{R}(m,d)$. However, the efficient construction of an explicit one still remains open. Even *et al.* [6] gave two ε-generators. One uses $O((\log m + \log d + \log 1/\varepsilon) \log 1/\varepsilon)$ bits based on k-wise independence, and the other uses $O((\log m + \log d + \log 1/\varepsilon) \log d)$ bits based on Nisan's generator for RL. Armoni *et al.* [4] observed that the generator of Impagliazzo, Nisan, and Wigderson [9] for communication networks also gives an ε-generator for $\mathcal{R}(m,d)$ using $O(\log m + (\log d + \log 1/\varepsilon) \log d)$ bits, which is good when d is small. They then reduced the original problem to the case when d is small (a formal definition of reductions will be given in the next section), and used the INW-generator to get an ε-generator for $\mathcal{R}(m,d)$ using $O(\log m + \log d + \log^2 1/\varepsilon)$ bits.

When m, d, and $1/\varepsilon$ are polynomially related, say all $n^{\Theta(1)}$, all previous generators still use $\Theta(\log^2 n)$ bits, which is the current barrier for its generalized problem — constructing generators for RL. We break this barrier for the first time, and give an ε-generator for $\mathcal{R}(m,d)$ using $O(\log m + \log d + \log^{3/2} 1/\varepsilon)$ bits. Our construction is based on that of Armoni *et al.* [4], and uses two more reductions to further reduce the dimension before applying the INW-generator. The overall construction can be seen as a composition of several generators for rectangles. Independently, Radhakrishnan and Ta-Shma [18] have a generator using $O(\log m + (\log \log m + \log d + \log 1/\varepsilon) \log^{1/2} 1/\varepsilon)$ random bits, based on a very similar idea.

We also observe that further improvements can be made if one can do better for a special case. Let $\mathcal{R}(m,d,k)$ be the set of rectangles from $\mathcal{R}(m,d)$

with at most k nontrivial dimensions (those not equal to $[m]$). We show that if an explicit ε-generator using $O(k+\log m+\log d+\log 1/\varepsilon)$ bits for $\mathcal{R}(m,d,k)$ exists, we can construct an explicit ε-generator using $O(\log m+\log d+\log 1/\varepsilon \log\log 1/\varepsilon)$ bits for $\mathcal{R}(m,d)$. Unfortunately we still don't know how to construct such a generator for $\mathcal{R}(m,d,k)$.

Another interesting special case is for rectangles where each dimension is an interval. Let $\mathcal{B}(m,d,k)$ be the set of rectangles from $\mathcal{R}(m,d,k)$ with each dimension being an interval. Even *et al.* [6] observed that the problem of approximating the distribution of independent multivalued random variables can be reduced to this case. They gave a generator using $O(k+\log d+\log 1/\varepsilon)$ bits. This is good when $k=O(\log 1/\varepsilon)$. For the case $k \geq \log 1/\varepsilon$, Chari *et al.* [5] gave a generator using $O(\log\log d+\log 1/\varepsilon \log \frac{k}{\log 1/\varepsilon})$ bits. Here, we improve this again to $O(\log\log d+\log 1/\varepsilon \log^{1/2} \frac{k}{\log 1/\varepsilon})$.

We will not emphasize the efficiency of our generators, but one can easily check that all the generators can be computed in simultaneous $(md/\varepsilon)^{O(1)}$ time and $O(\log m+\log d+\log 1/\varepsilon)$ space.

It's worth mentioning that the hitting version of our problem has already been settled. Linial, Luby, Saks, and Zuckerman [11] gave an explicit generator using $O(\log m+\log\log d+\log 1/\varepsilon)$ bits that can hit any rectangle in $\mathcal{R}(m,d)$ of volume at least ε. In fact the work of Armoni *et al.* [4] followed closely this result, and so does ours.

2. Preliminaries

2.1. Notations

For a set V, we let 2^V denote the family of subsets of V. For a rectangle $R \in \mathcal{R}(m,d)$ and a set of indices $I \subseteq [d]$, we let R_I denote the subrectangle of R restricted to those dimensions in I. Similarly, for a vector $v \in [m]^d$ and $I \subseteq [d]$, let v_I be the subvector of v restricted to those dimensions in I.

A function from A to B can be seen as a vector in $B^{|A|}$, and vice versa. We say that a class $\mathcal{F}$ of functions can be identified with $[|\mathcal{F}|]$ if there is a one-to-one mapping from $[|\mathcal{F}|]$ onto $\mathcal{F}$ that can be efficiently computed. Whenever we can identify a class $\mathcal{F}$ of functions with $[|\mathcal{F}|]$, we can use numbers in $[|\mathcal{F}|]$ to represent functions in $\mathcal{F}$. For a function $f: A \to [m]^d$, an element $x \in A$, and an index $y \in [d]$, we will use $f(x)(y)$ to denote the yth dimension in the vector $f(x) \in [m]^d$.

When we sample from a finite set, the default distribution is the uniform one over that set. All the logarithms throughout this paper will have base 2.

2.2. k-wise Independent Hash Function Family

Let n_1, n_2 be integers. Recall the correspondence between functions and vectors. A family H of functions from $[n_1]$ to $[n_2]$ is called a (k,ε) independent hash function family if for any $I \subseteq [n_1]$ with $|I| \leq k$ and for any $v \in [n_2]^{|I|}$,

$$\left| P_{h \in H}[h_I = v] - \left(\frac{1}{n_2}\right)^{|I|} \right| \leq \varepsilon.$$

A $(k,0)$ independent hash function family is called a k-wise independent hash function family.

A standard construction of k-wise independent hash function family [12, 1], denoted as $\mathcal{H}_k(n_1,n_2)$, has size $(max\{n_1,n_2\})^k$ and can be identified with $[|\mathcal{H}_k(n_1,n_2)|]$.

A (k,ε) independent hash function family can be derived from the constructions of Alon, Goldreich, Hastad, and Peralta [2], or Naor and Naor [13], by using their $k \log n_2$-wise ε-biased sample space over $\{0,1\}^{n_1 \log n_2}$. Let $\mathcal{H}_{k,\varepsilon}(n_1,n_2)$ denote such a family, which has size $(\frac{k \log n_2}{\varepsilon})^{O(1)}$ and can be identified with $[|\mathcal{H}_{k,\varepsilon}(n_1,n_2)|]$.

2.3. Extractors

For a distribution D over a set S, let $D(i)$, for $i \in S$, denote the probability of i according to D, let $D(A) = \sum_{a \in A} D(a)$ for $A \subseteq S$, and define the min-entropy of D as $\min_{a \in S: D(a)>0} \log 1/D(a)$. The distance between two distributions D_1 and D_2 over S is defined as $\max_{A \subseteq S} |D_1(A) - D_2(A)|$. For a positive integer n, let U_n denote the uniform distribution over $\{0,1\}^n$.

Extractors were first defined by Nisan and Zuckerman [16]. They are used to extract randomness from weakly random sources, and turn out to have many other applications [15].

Definition 2.1. [16] A function $E : \{0,1\}^s \times \{0,1\}^t \to \{0,1\}^\ell$ is called an (s,r,t,ℓ,δ)-extractor if for x chosen from any distribution over $\{0,1\}^s$ with min-entropy at least r, and y chosen from U_t, the distribution of $E(x,y)$ has distance at most δ to U_ℓ.

An extractor takes s bits with min-entropy at least r, invests t truly random bits, and extracts ℓ quasi-random bits. One would like to have extractors with small t and large ℓ. A recent breakthrough by Trevisan [20] achieves $t = O(\log s/\delta)$ and $\ell = r^{O(1)}$ for any $r = s^{\Omega(1)}$, and leads to a series of further improvements (e.g. [17, 10, 19]).

2.4. Reductions

We adopt the notion of reduction introduced by Armoni *et al.* [4]. It enables us to reduce a harder problem to an easier one, and then focus our attention to solving the easier problem. A class $\mathcal{F}$ of functions from a set V_2 to a set V_1 defines a reduction from V_1 to V_2. Let $\mathcal{A}_1 \subseteq 2^{V_1}$ and $\mathcal{A}_2 \subseteq 2^{V_2}$. $\mathcal{F}$ is said to be $(\mathcal{A}_1, \mathcal{A}_2, \varepsilon)$-good [4], if for each $R \in \mathcal{A}_1$ the following hold:

1. $\forall f \in \mathcal{F}, f^{-1}(R) \in \mathcal{A}_2$, and
2. $|E_{f\in\mathcal{F}}[vol(f^{-1}(R))] - vol(R)| \leq \varepsilon$.

Suppose now that $\mathcal{F}$ is $(\mathcal{A}_1, \mathcal{A}_2, \varepsilon_1)$-good and $g : \{0,1\}^s \to V_2$ is an ε_2-generator for $\mathcal{A}_2$. Armoni *et al.* [4] showed that the function $g' : \{0,1\}^s \times \mathcal{F} \to V_1$, defined as $g'(y, f) = (f \circ g)(y)$, is an $(\varepsilon_1 + \varepsilon_2)$-generator for $\mathcal{A}_1$. The *reduction cost* of $\mathcal{F}$ is $\log|\mathcal{F}|$, which is the number of extra bits needed for the new generator. The following lemma follows immediately.

Lemma 2.1. [4] *For each i, $0 \leq i \leq \ell$, let V_i be a set and $\mathcal{A}_i \subseteq 2^{V_i}$. Suppose that $\mathcal{F}_i$ is $(\mathcal{A}_{i-1}, \mathcal{A}_i, \varepsilon_{i-1})$-good for $1 \leq i \leq \ell$, and $g : \{0,1\}^s \to V_\ell$ is an ε_ℓ-generator for $\mathcal{A}_\ell$. Then the function $g' : \{0,1\}^s \times \mathcal{F}_1 \times \cdots \times \mathcal{F}_\ell \to V_0$ defined as $g'(x, f_1, \ldots, f_\ell) = (f_1 \circ \cdots \circ f_\ell \circ g)(x)$, is a $\left(\sum_{i=0}^{\ell} \varepsilon_i\right)$-generator for $\mathcal{A}_0$.*

So to construct a generator for $\mathcal{A}_0$, it suffices to find a series of reductions from $\mathcal{A}_0$ to $\mathcal{A}_\ell$, and then find a generator for $\mathcal{A}_\ell$.

Notice that an $(\mathcal{A}_1, \mathcal{A}_2, \varepsilon)$-good reduction $\mathcal{F}$ actually corresponds to a special kind of ε-generator for $\mathcal{A}_1$. Let $h : V_2 \times \mathcal{F} \to V_1$ be defined as $h(y, f) = f(y)$. Then for all $R \in \mathcal{A}_1$,

$$\begin{aligned} |P_{(y,f)\in V_2\times\mathcal{F}}[h(y,f) \in R] - vol(R)| &= |E_{f\in\mathcal{F}}[P_{y\in V_2}[f(y) \in R]] - vol(R)| \\ &= |E_{f\in\mathcal{F}}[vol(f^{-1}(R))] - vol(R)|. \end{aligned}$$

So the second condition holds if and only if h is an ε-generator for $\mathcal{A}_1$. The first condition guarantees that one part of h's input, V_2, can come from the output of a generator for $\mathcal{A}_2$, and makes the composition of generators possible. So one way of finding a reduction is to use some generator that might use many bits but can be composed with other generators.

3. The Pseudorandom Generator for $\mathcal{R}(m, d)$

3.1. An Overview of the Construction

The INW-generator uses $O(\log m + (\log d + \log 1/\varepsilon)\log d)$ bits, which is good when d is small. The idea of Armoni *et al.* [4] is to reduce the dimension of rectangles first to $d' = (1/\varepsilon)^{O(1)}$ before applying the INW-generator.

In addition to that, we also reduce m to $m' = (1/\varepsilon)^{O(1)}$. The INW-generator for $\mathcal{R}(m', d')$ needs $O(\log m' + (\log d' + \log 1/\varepsilon)\log d') = O(\log 1/\varepsilon + \log^2 1/\varepsilon)$ bits. Observe that we do not lose by letting m' increase a little. By allowing m' to grow to $m'' = (1/\varepsilon)^{O(\sqrt{\log 1/\varepsilon})}$, we are able to reduce d' to $d'' = 2^{O(\sqrt{\log 1/\varepsilon})}$. The INW-generator now uses $O(\log^{3/2} 1/\varepsilon)$ bits for $R(m'', d'')$. The total reduction cost is $O(\log m + \log d + \log^{3/2} 1/\varepsilon)$, and we have the desired generator for $\mathcal{R}(m, d)$.

More precisely, we will use the following three reductions.

- $\mathcal{F}_1$ is called the first dimension reduction family, and is used to reduce d to $d_1 = (1/\varepsilon)^{O(1)}$. It is $(\mathcal{R}(m,d), \mathcal{R}(m_1, d_1), \varepsilon/4)$-good, where $m_1 = (md)^{O(1)}$. The reduction cost is $O(\log d)$.
- $\mathcal{F}_2$ is called the range reduction family, and is used to reduce m_1 to $m_2 = (1/\varepsilon)^{O(1)}$. It is $(\mathcal{R}(m_1, d_1), \mathcal{R}(m_2, d_2), \varepsilon/4)$-good, where $d_2 = d_1$. The reduction cost is $O(\log m + \log d + \log 1/\varepsilon)$.
- $\mathcal{F}_3$ is called the second dimension reduction family, and is used to reduce d_2 to $d_3 = 2^{(3\log \frac{12}{\varepsilon})/(k-1)}$, with k a parameter to be chosen to optimize our construction. It is $(\mathcal{R}(m_2, d_2), \mathcal{R}(m_3, d_3), \varepsilon/4)$-good, where $m_3 = |\mathcal{H}_k(d_2, m_2)| = (1/\varepsilon)^{O(k)}$. The reduction cost is $\log |\mathcal{H}_k(d_2, d_3)| = O(k \log 1/\varepsilon)$.

Together with the $\varepsilon/4$-generator for $\mathcal{R}(m_3, d_3)$ from the INW-generator, we have an ε-generator for $\mathcal{R}(m, d)$. The number of bits used depends on k, and choosing $k = \log^{1/2} 1/\varepsilon$ results in the minimum $O(\log m + \log d + \log^{3/2} 1/\varepsilon)$.

3.2. The First Dimension Reduction Function Family

Let $\mathcal{F}_1$ be the reduction family used by Armoni *et al.* [4], which is the composition of three reduction families. $\mathcal{F}_1$ is $(\mathcal{R}(m,d), \mathcal{R}(m_1, d_1), \varepsilon/4)$-good, where $m_1 = (dm)^{O(1)}$, $d_1 = (1/\varepsilon)^{O(1)}$, and each is a power of 2. $|\mathcal{F}_1| = d^{O(1)}$. Let $V_1 = [m_1]^{d_1}$.

3.3. The Range Reduction Function Family

Recall the definition of extractors from Section 2.3. The idea of using extractors for range reduction was inspired by that of Radhakrishnan and Ta-Shma [18]. We will borrow an extractor-based generator of Nisan and Zuckerman [16], which was originally used to fool randomized space-bounded computation. The extractor we need here is one that works well for sources with very high min-entropy, and the best one for our purpose is an extractor of Goldreich and Wigderson [8].

Lemma 3.1. [8] *There are constants c_1 and c_2 such that for any ℓ, s, γ, and δ, with $s>\gamma$, $s-\gamma>\ell$, and $\delta>2^{-(s-\ell-c_1\gamma)/c_2}$, an explicit $(s,s-\gamma,O(\gamma+\log\frac{1}{\delta}),\ell,\delta)$-extractor exists.*

Choose $\delta=\varepsilon/(4d_1)$, $\gamma=\lceil\log 1/\delta\rceil$, and $\ell=\log m_1$, where d_1 and m_1 are from the first reduction. Choose $s=\ell+c\gamma=O(\log m+\log d+\log 1/\varepsilon)$ for some constant c, such that $2^{-(s-\ell-c_1\gamma)/c_2}=2^{-((c-c_1)/c_2)\log 1/\delta}<2^{\log\delta}=\delta$. For this setting, let A denote the explicit $(s,s-\gamma,t,\ell,\delta)$-extractor guaranteed by Lemma 3.1, for some $t=O(\gamma+\log 1/\delta)=O(\log 1/\varepsilon)$.

The building block of Nisan and Zuckerman's generator [16], when using the extractor A, has the form $G:\{0,1\}^s\times\{0,1\}^{td_1}\to[m_1]^{d_1}$, where

$$G(x,y_1,\ldots,y_{d_1})=(A(x,y_1),\ldots,A(x,y_{d_1})).$$

For $R=R_1\times\cdots\times R_{d_1}\in\mathcal{R}(m_1,d_1)$, we first show that

$$|P_{x,y_1,\ldots,y_{d_1}}[(A(x,y_1),\ldots,A(x,y_{d_1}))\in R]-vol(R)|\le d_1\delta=\frac{\varepsilon}{4}.$$

For $i\in[d_1]$, let $D_i(x,y_1,\ldots,y_i)$ denote the event that $(A(x,y_1),\ldots,A(x,y_i))$ is in $R_1\times\cdots\times R_i$. Also let $p_i=P_{x,y_1,\ldots,y_i}[D_i(x,y_1,\ldots,y_i)]$, $q_i=\prod_{j=1}^{i}\frac{|R_j|}{m_1}$, and $r_{i+1}=P_{x,y_1,\ldots,y_{i+1}}[A(x,y_{i+1})\in R_{i+1}|D_i(x,y_1,\ldots,y_i)]$. The following is a simplified version of a lemma due to Nisan and Zuckerman [16], based on our choice of parameters.

Lemma 3.2. *For any $i\in[d_1]$, $|p_i-q_i|\le i\delta$.*

Proof. Use induction on i. It's true for $i=1$ from the definition of extractors. Assuming $|p_i-q_i|\le i\delta$, we will show that $|p_{i+1}-q_{i+1}|\le(i+1)\delta$.

$$\begin{aligned}|p_{i+1}-q_{i+1}|&=\left|p_ir_{i+1}-p_i\frac{|R_{i+1}|}{m_1}+p_i\frac{|R_{i+1}|}{m_1}-q_i\frac{|R_{i+1}|}{m_1}\right|\\&\le p_i\left|r_{i+1}-\frac{|R_{i+1}|}{m_1}\right|+|p_i-q_i|.\end{aligned}$$

There are two cases.

- $p_i\le\delta$: $|p_{i+1}-q_{i+1}|\le\delta+i\delta=(i+1)\delta$.
- $p_i>\delta$:

The distribution of x conditioned on $D_i(x,y_1,\ldots,y_i)$ has min-entropy at least $s-\log 1/\delta\ge s-\gamma$. From Lemma 3.1, $|P_{x,y_1,\ldots,y_i}[A(x,y_{i+1})\in R_{i+1}|D_i(x,y_1,\ldots,y_i)]-\frac{|R_{i+1}|}{m_1}|\le\delta$, and $|p_{i+1}-q_{i+1}|\le\delta+i\delta=(i+1)\delta$. ∎

Now let $m_2 = 2^t = (1/\varepsilon)^{O(1)}$, $d_2 = d_1$, and $V_2 = [m_2]^{d_2}$. Consider the reduction family $\mathcal{F}_2 = \{f_x \mid x \in \{0,1\}^s\}$, where $f_x : V_2 \to V_1$ is defined as follows

$$f_x(y_1, \ldots, y_{d_2}) = G(x, y_1, \ldots, y_{d_1}).$$

Then $f_x^{-1}(R) = R'_1 \times \cdots \times R'_{d_2} \in \mathcal{R}(m_2, d_2)$, where $R'_i = \{y_i \mid A(x, y_i) \in R_i\}$. Also,

$$\begin{aligned} |E_x[vol(f_x^{-1}(R))] - vol(R)| &= |P_{x,y_1,\ldots,y_{d_1}}[G(x, y_1, \ldots, y_{d_1}) \in R] - vol(R)| \\ &\leq \varepsilon/4. \end{aligned}$$

So we have the following lemma.

Lemma 3.3. *$\mathcal{F}_2$ is $(\mathcal{R}(m_1, d_1), \mathcal{R}(m_2, d_2), \varepsilon/4)$-good.*

3.4. The Second Dimension Reduction Function Family

Let $R = R_1 \times \cdots \times R_{d_2} \in \mathcal{R}(m_2, d_2)$. We want to partition the d_2 dimensions of R into d_3 parts using some function $h : [d_2] \to [d_3]$ in the natural way. For $q \in [d_3]$, those dimensions of R that are mapped to q form a subrectangle $R_{h^{-1}(q)} = \prod_{i \in h^{-1}(q)} R_i$, where $h^{-1}(q)$ is the set $\{i \in [d_2] : h(i) = q\}$. Recall the correspondence between fnctions and vectors. Based on the idea of Even *et al.* [6], the volume of a subrectangle can be approximated by sampling from the k-wise independent space $G = \mathcal{H}_k(d_2, m_2)$. Then $vol(R)$ can be approximated by sampling from d_3 independent copies of G, one for each subrectangle. So the rectangle $R^{(h)} = R'_1 \times \cdots \times R'_{d_3}$, where $R'_q = \{p \in G : p_{h^{-1}(q)} \in R_{h^{-1}(q)}\}$, should have a volume close to that of $R = R_{h^{-1}(1)} \times \cdots \times R_{h^{-1}(d_3)}$. The dimension is reduced to d_3. The error depends on the choice of k and h. We will show that for $k = O(\log^{1/2} 1/\varepsilon)$ and h chosen uniformly from $H = \mathcal{H}_k(d_2, d_3)$, the expected error is at most $\varepsilon/4$.

More formally, let $d_3 = 2^{\left(3 \log \frac{12}{\varepsilon}\right)/(k-1)}$, $m_3 = |G|$, $V_3 = [m_3]^{d_3}$, and $\mathcal{F}_3 = \{f_h : h \in H\}$, where $f_h : V_3 \to V_2$ is defined as follows

$$f_h(p_1, \ldots, p_{d_3}) = (p_{h(1)}(1), \ldots, p_{h(d_2)}(d_2)).$$

Then for $R \in \mathcal{R}(m_2, d_2)$ and $f_h \in \mathcal{F}_3$, $f_h^{-1}(R) = R^{(h)} \in \mathcal{R}(m_3, d_3)$.

We also need the following notation for the proofs below. For $R = R_1 \times \cdots \times R_{d_2} \in \mathcal{R}(m_2, d_2)$, let $\tilde{R}$ denote the rectangle $\overline{R}_1 \times \cdots \times \overline{R}_{d_2} \in \mathcal{R}(m_2, d_2)$,

where $\overline{R}_i = [m_2] \setminus R_i$. For $i, j \in [d_2]$ and $I \subseteq [d_2]$, denote

$$\begin{aligned}
\delta_i &= \frac{|\overline{R}_i|}{m_2}, \\
\pi(I) &= vol(\tilde{R}_I) = \prod_{i \in I} \delta_i, \text{ with } \pi(\emptyset) = 1 \\
\gamma(I) &= P_{p \in G}[p_I \in \tilde{R}_I], \text{ and} \\
\tau_j(I) &= \sum_{J \subseteq I, |J| = j} \pi(J), \text{ with } \tau_0(\emptyset) = 1.
\end{aligned}$$

The approximation error of each subrectangle can be bounded in the following way, as shown by Even *et al.* [6]. We include the proof for completeness.

Proposition 3.1. [6] $\forall I \subseteq [d_2], |P_{p \in G}[p_I \in R_I] - vol(R_I)| \leq \tau_k(I)$

Proof. G is a k-wise independent space, so for any $J \subseteq I$ with $|J| \leq k$, $\pi(J) = \gamma(J)$. From the principle of inclusion and exclusion, we have the following.

$$vol(R_I) = \prod_{i \in I}(1 - \delta_i) = \sum_{J \subseteq I}(-1)^{|J|}\pi(J) = \sum_{j=0}^{k}(-1)^j \tau_j(I) + \sum_{j=k+1}^{|I|}(-1)^j \tau_j(I).$$

$$P_{p \in G}[p_I \in R_I] = \sum_{J \subseteq I}(-1)^{|J|}\gamma(J) = \sum_{j=0}^{k}(-1)^j \tau_j(I) + \sum_{j=k+1}^{|I|}(-1)^j \sum_{J \subseteq I, |J| = j} \gamma(J).$$

Now the proposition follows because both $vol(R_I)$ and $P_{p \in G}[p_I \in R_I]$ fall between $\sum_{j=0}^{k-1}(-1)^j \tau_j(I)$ and $\sum_{j=0}^{k}(-1)^j \tau_j(I)$. ∎

The approximation error of any partition can be bounded as follows.

Lemma 3.4. $\forall h : [d_2] \to [d_3], |vol(R^{(h)}) - vol(R)| \leq \sum_{q \in [d_3]} \tau_k(h^{-1}(q))$

Proof.

$$\begin{aligned}
vol(R) &= \prod_{q \in [d_3]} vol(R_{h^{-1}(q)}). \\
vol(R^{(h)}) &= \prod_{q \in [d_3]} P_{p \in G}[p_{h^{-1}(q)} \in R_{h^{-1}(q)}].
\end{aligned}$$

This lemma follows from the previous proposition and the known fact that $|\prod_{i=1}^{\ell} x_i - \prod_{i=1}^{\ell} y_i| \leq \sum_{i=1}^{\ell} |x_i - y_i|$ when $0 \leq x_i, y_i \leq 1$ for all $i \in [\ell]$. ∎

Finally, we can bound the expected approximation error.

Lemma 3.5. *For $R \in \mathcal{R}(m_2, d_2)$, $|E_{h\in H}[vol(R^{(h)})] - vol(R)| \leq \frac{\varepsilon}{4}$.*

Proof.

$$\begin{aligned}
|E_{h\in H}[vol(R^{(h)})] - vol(R)| &\leq E_{h\in H}[|vol(R^{(h)}) - vol(R)|] \\
&\leq E_{h\in H}\left[\sum_{q\in[d_3]} \tau_k(h^{-1}(q))\right] \\
&= \sum_{q\in[d_3]} E_{h\in H}\left[\sum_{I\subseteq h^{-1}(q),|I|=k} \pi(I)\right] \\
&= \sum_{q\in[d_3]} \sum_{I\subseteq[d_2],|I|=k} P_{h\in H}[\forall i \in I\ h(i) = q]\pi(I) \\
&= \sum_{q\in[d_3]} \sum_{I\subseteq[d_2],|I|=k} (1/d_3)^k \pi(I) \\
&= (1/d_3)^{k-1}\tau_k([d_2]).
\end{aligned}$$

Let $\alpha = \sum_{i\in[d_2]} \delta_i$. There are two cases depending on the value of α.

- $\alpha \leq \log\frac{12}{\varepsilon}$:

$\tau_k([d_2])$ gets its maximum value when $\delta_i = \frac{\alpha}{d_2}$ for all $i \in [d_2]$. So $\tau_k([d_2]) \leq (\frac{ed_2}{k})^k(\frac{\alpha}{d_2})^k \leq \left(\frac{e\log\frac{12}{\varepsilon}}{k}\right)^k$, which is again maximized when $k = \log\frac{12}{\varepsilon}$. So for $d_3 = 2^{(3\log\frac{12}{\varepsilon})/(k-1)}$, we have

$$\begin{aligned}
|E_{h\in H}[vol(R^{(h)})] - vol(R)| &\leq 2^{-3\log\frac{12}{\varepsilon}} e^{\log\frac{12}{\varepsilon}} \\
&= 2^{-(3-\log e)\log\frac{12}{\varepsilon}} \\
&= \left(\frac{\varepsilon}{12}\right)^{3-\log e} \\
&\leq \frac{\varepsilon}{12}.
\end{aligned}$$

- $\alpha > \log\frac{12}{\varepsilon}$:

In this case, both $E_{h\in H}[vol(R^{(h)})]$ and $vol(R)$ are small, so their difference is small. First, $vol(R) = \prod_{i\in[d_2]}(1-\delta_i) \leq 2^{-\sum_{i\in[d_2]}\delta_i} < \frac{\varepsilon}{12}$.

Next, we show that $E_{h\in H}[vol(R^{(h)})] \leq \frac{\varepsilon}{3}$. Let d' be the smallest integer such that $\log\frac{12}{\varepsilon} - 1 < \sum_{i\in[d']} \delta_i \leq \log\frac{12}{\varepsilon}$. Let $R' = R_{[d']} \times [m_2]^{d_2-d'}$. From the

previous case $E_{h\in H}[vol(R'^{(h)})] \le vol(R') + \frac{\varepsilon}{12}$. So

$$\begin{aligned} E_{h\in H}[vol(R^{(h)})] &\le E_{h\in H}[vol(R'^{(h)})] \\ &\le vol(R') + \frac{\varepsilon}{12} \\ &\le 2^{-\sum_{i\in[d']}\delta_i} + \frac{\varepsilon}{12} \\ &\le 2^{-\log\frac{12}{\varepsilon}+1} + \frac{\varepsilon}{12} \\ &= \frac{\varepsilon}{4}. \end{aligned}$$

$0 \le E_{h\in H}[vol(R^{(h)})], vol(R) \le \frac{\varepsilon}{4}$. So $|E_{h\in H}[vol(R^{(h)})] - vol(R)| \le \frac{\varepsilon}{4}$. ∎

So we have the following lemma.

Lemma 3.6. *$\mathcal{F}_3$ is $(\mathcal{R}(m_2,d_2), \mathcal{R}(m_3,d_3), \frac{\varepsilon}{4})$-good.*

3.5. The Main Theorem

We have shown that $\mathcal{F}_i$ is $(\mathcal{R}(m_{i-1},d_{i-1}), \mathcal{R}(m_i,d_i), \frac{\varepsilon}{4})$-good, for $1 \le i \le 3$. The INW-generator gives us an $\frac{\varepsilon}{4}$-generator for $\mathcal{R}(m_3,d_3)$. From Lemma 2.1, we have an ε-generator for $\mathcal{R}(m,d)$. We summarize the key parameters used in the reductions:

- The first dimension reduction:
  - Cost: $\log|\mathcal{F}_1| = O(\log d)$.
  - New dimension: $d_1 = (1/\varepsilon)^{O(1)}$.
  - New range: $m_1 = (md)^{O(1)}$.
- The range reduction:
  - Cost: $\log|\mathcal{F}_2| = O(\log m + \log d + \log 1/\varepsilon)$.
  - New dimension: $d_2 = d_1$.
  - New range: $m_2 = (1/\varepsilon)^{O(1)}$.
- The second dimension reduction:
  - Cost: $\log|\mathcal{F}_3| = \log|\mathcal{H}_k(d_2,d_3)| \le k\log d_2 = O(k\log 1/\varepsilon)$.
  - New dimension: $d_3 = 2^{(3\log\frac{12}{\varepsilon})/(k-1)}$.
  - New range: $m_3 = |\mathcal{H}_k(d_2,m_2)| = (1/\varepsilon)^{O(k)}$.
- The $\frac{\varepsilon}{4}$-generator for $\mathcal{R}(m_3,d_3)$:

– Number of random bits: $O(\log m_3 + (\log d_3 + \log 1/\varepsilon)\log d_3) = O(k\log 1/\varepsilon + 1/k\log^2 1/\varepsilon)$.

The total number of random bits used by our generator is $O(\log d + \log m + k\log 1/\varepsilon + 1/k\log^2 1/\varepsilon)$, which gets its minimum value $O(\log m + \log d + \log^{3/2} 1/\varepsilon)$ at $k = O(\log^{1/2} 1/\varepsilon)$. So we have our main theorem.

Theorem 3.1. *There is an explicit ε-generator for $\mathcal{R}(m,d)$, using $O(\log m + \log d + \log^{3/2} 1/\varepsilon)$ bits.*

4. A Potential Improvement

The key component of our construction is the second dimension reduction family $\mathcal{F}_3$. In this section, we will study the possibility of improving this reduction and how this would lead to a more efficient generator construction.

First, we can replace the k-wise independent hash function family H, used for partitioning dimensions, by the (k,ε) independent hash family $H' = \mathcal{H}_{k,\varepsilon/d_3}(d_2,d_3)$, described in Section 2.2. It guarantees that for any $I \subseteq [d_2]$ with $|I| \le k$ and for any $y \in [d_3]^{d_2}$,

$$\left| P_{x\in H'}[x_I = y_I] - \left(\frac{1}{d_3}\right)^{|I|} \right| \le \frac{\varepsilon}{d_3}.$$

H' has size $(\frac{k\log d_2}{\varepsilon})^{O(1)} = (\frac{1}{\varepsilon})^{O(1)}$, and can be identified with $|H'|$. One can easily verify that only an additional $O(\varepsilon)$ error is introduced in Lemma 3.5, and now the reduction cost for $\mathcal{F}_3$ is $O(\log 1/\varepsilon)$. From now on we will use H' instead of H in $\mathcal{F}_3$.

Next, recall from the previous section that $m_3 = |\mathcal{H}_k(d_2,m_2)| = (1/\varepsilon)^{O(k)}$ and $d_3 = 2^{(3\log 1/\varepsilon)/(k-1)}$. Larger k implies smaller d_3 but larger m_3. The optimum is attained at $k = \Theta(\log^{1/2} 1/\varepsilon)$. If we can replace $\mathcal{H}_k(d_2,m_2)$ by a smaller space, we might be able to choose a larger k and get a smaller d_3. Remember that d_3 copies of $\mathcal{H}_k(d_2,m_2)$ are used to approximate the volumes of the d_3 subrectangles of R partitioned by a function $h:[d_2]\to[d_3]$. The approximation is guaranteed by the fact that for $R \in \mathcal{R}(m_2,d_2)$ and for $J \subseteq [d_2]$ with $|J| \le k$,

$$|\gamma(J) - \pi(J)| = |P_{p\in G}[p_J \in \tilde{R}_J] - vol(\tilde{R}_J)| = 0.$$

We want to use a smaller space by allowing a small error ε' instead of 0 above. The approximate k-wise independent space does not help here, because it needs $\Omega(k\log m_2 + \log 1/\varepsilon')$ bits to achieve an error ε' here, no better than G. However, observe that what we need here is to approximate the volume of a

rectangle with at most k nontrivial dimensions. This turns out to be a special case of our original problem — constructing a pseudorandom generator for $\mathcal{R}(m,d,k)$.

Suppose that $g:\{0,1\}^s \to [m_2]^{d_2}$ is an ε'-generator for $\mathcal{R}(m,d,k)$. In $\mathcal{F}_3$, we replace $\mathcal{H}_k(d_2,m_2)$ by the space generated by g. Let $m_3 = 2^s$. For $h \in H'$, let $f_h:[m_3]^{d_3} \to [m_2]^{d_2}$ be defined as follows

$$f_h(x_1,\ldots,x_{d_3}) = (g(x_{h(1)})(1),\ldots,g(x_{h(d_2)})(d_2)).$$

For $R = R_1 \times \cdots \times R_{d_2} \in \mathcal{R}(m_2,d_2)$, $f_h^{-1}(R) = R^{(h)} = R'_1 \times \cdots \times R'_{d_3} \in \mathcal{R}(m_3,d_3)$, where $R'_q = \{x \in [m_3] : g(x)_{h^{-1}(q)} \in R_{h^{-1}(q)}\}$. Then for $J \subseteq h^{-1}(q)$ with $|J| \le k$, $|P_{x\in[m_3]}[g(x)_J \in \tilde{R}_J] - vol(\tilde{R}_J)| \le \varepsilon'$. So, for any $h:[d_2]\to[d_3]$,

$$\begin{aligned} |vol(R^{(h)}) - vol(R)| &\le \sum_{q\in[d_3]} \left(\left(\sum_{J\subseteq h^{-1}(q), |J|\le k} \varepsilon' \right) + \tau_k(h^{-1}(q)) \right) \\ &\le d_2^{k+2}\varepsilon' + \sum_{q\in[d_3]} \tau_k(h^{-1}(q)), \end{aligned}$$

and $|E_{h\in H'}[vol(R^{(h)})] - vol(R)| \le d_2^{k+2}\varepsilon' + \frac{\varepsilon}{4} = \varepsilon/2$, for $\varepsilon' = \frac{\varepsilon}{4d_2^{k+2}}$.

So if we have a better ε'-generator for $\mathcal{R}(m_2,d_2,k)$, we can choose a larger k and thus a smaller d_3.

Theorem 4.1. *If there exists an explicit ε-generator for $\mathcal{R}(m,d,k)$ using $O(k+\log d+\log m+\log\frac{1}{\varepsilon})$ bits, then there exists an explicit ε-generator for $\mathcal{R}(m,d)$ using $O(\log d+\log m+\log\frac{1}{\varepsilon}\log\log\frac{1}{\varepsilon})$ bits.*

Proof. Using the ε'-generator for $\mathcal{R}(m_2,d_2,k)$ in $\mathcal{F}_3$ gives

$$m_3 = \left(\frac{2^k d_2 m_2}{\varepsilon/d_2^k}\right)^{O(1)} = \left(\frac{d_2^k}{\varepsilon}\right)^{O(1)}.$$

We want to repeatedly reduce the dimensions of rectangles. Notice that each time the dimension is reduced, we can choose a larger next k.

For $3 \le i \le \ell = \log\log\frac{12}{\varepsilon} - \log\log\log\frac{12}{\varepsilon}$, let

$$\begin{aligned} k_i &= 2^i, \\ d_i &= 2^{O((\log\frac{1}{\varepsilon})/k_i)} = 2^{O((\log\frac{1}{\varepsilon})/2^i)}, \\ \varepsilon_i &= \frac{\varepsilon}{4d_{i-1}^{k_i+2}} = \varepsilon^{O(1)}, \text{ and} \\ m_i &= \left(\frac{2^{k_i} m_{i-1} d_{i-1}}{\varepsilon_i}\right)^{O(1)} = \left(\frac{1}{\varepsilon}\right)^{O(i)} \end{aligned}$$

For $3 \le i \le \ell$, let $\mathcal{F}_i$ be the modified dimension reduction discussed previously in this section, using the assumed ε_i-generator for $\mathcal{R}(m_{i-1}, d_{i-1}, k_i)$. One can check that each $\mathcal{F}_i$ is $(\mathcal{R}(m_{i-1}, d_{i-1}), \mathcal{R}(m_i, d_i), \varepsilon)$-good. Using the INW-generator as an ε-generator for $\mathcal{R}(m_\ell, d_\ell)$, we have an $O(\varepsilon \log\log \frac{1}{\varepsilon})$-generator for $\mathcal{R}(m,d)$. The total number of bits used is

$$\begin{aligned}
&\sum_{i=1}^{\ell} \log |\mathcal{F}_i| + O(\log m_\ell + (\log d_\ell + \log \tfrac{1}{\varepsilon}) \log d_\ell) \\
&= O\left(\log m + \log d + \log \tfrac{1}{\varepsilon} + \log \tfrac{1}{\varepsilon} \log\log \tfrac{1}{\varepsilon} \right. \\
&\qquad \left. + \log m_\ell + \left(\log d_\ell + \log \tfrac{1}{\varepsilon}\right) \log d_\ell \right) \\
&= O(\log m + \log d + \log \tfrac{1}{\varepsilon} \log\log \tfrac{1}{\varepsilon}).
\end{aligned}$$

Replacing $\varepsilon \log\log \frac{1}{\varepsilon}$ by ε, we have an ε-generator for $\mathcal{R}(m,d)$ using $O(\log m + \log d + \log \frac{1}{\varepsilon} \log\log \frac{1}{\varepsilon})$ bits. ∎

We don't know yet how to construct such an explicit ε-generator for $\mathcal{R}(m,d,k)$ using $O(k + \log d + \log m + \log 1/\varepsilon)$ bits. Using an idea of Auer, Long, and Srinivasan [3], we can derive one using $O(\log k + \log m + \log^{3/2} 1/\varepsilon)$ bits, which improves their upper bound, but does not serve our purpose here.

5. The Pseudorandom Generator for $\mathcal{B}(m,d,t)$

Recall that $\mathcal{B}(m,d,t)$ denote the class of rectangles from $\mathcal{R}(m,d)$ with at most t nontrivial dimensions and each dimension being an interval. For $\mathcal{B}(m,d,t)$, Even *et al.* [6,7] have an ε-generator using $O(t + \log\log d + \log 1/\varepsilon)$ bits. Unfortunately, we cannot apply the iterative procedure in the previous section to $\mathcal{B}(m,d,d)$ because after applying the dimension reduction once, each dimension is no longer an interval.

For $t \ge \log 1/\varepsilon$, Chari *et al.* [5] had an ε-generator for $\mathcal{B}(m,d,t)$ using $O(\log\log d + \log \frac{1}{\varepsilon} \log \frac{t}{\log 1/\varepsilon})$ bits, a significant improvement in the dependence on t. This is improved again by the following theorem.

Theorem 5.1. *For $t \ge \log 1/\varepsilon$, there is an explicit ε-generator for $\mathcal{B}(m,d,t)$, using $O\left(\log\log d + \log \frac{1}{\varepsilon} \log^{1/2} \frac{t}{\log 1/\varepsilon}\right)$ bits.*

Proof. Here we use only one reduction, a modified dimension reduction similar to that in Section 4, while the first two reductions of Section 3 are not needed. Let $k \le t$ be a parameter to be chosen later. For $\varepsilon' = \varepsilon(\frac{k}{t})^{O(k)}$, let $g : \{0,1\}^s \to [m]^d$ be the ε'-generator of Even *et al.* for $\mathcal{B}(m,d,k)$. Let

$m' = 2^s$ and $d' = 2^{(3\log\frac{12}{\varepsilon})/(k-1)}$. Given $R = R_1 \times \cdots \times R_d \in \mathcal{B}(m,d,t)$, assume w.l.o.g. that the first t dimensions are nontrivial. Similar to Section 4, for $h \in H' = \mathcal{H}_{k,\varepsilon}(d,d')$, let the reduction function $f_h : [m']^{d'} \to [m]^d$ be defined as

$$f_h(x_1,\ldots,x_{d'}) = (g(x_{h(1)})(1),\ldots,g(x_{h(d)})(d)).$$

Then

$$R^{(h)} = f_h^{-1}(R) = R'_1 \times \cdots \times R'_d \in \mathcal{R}(m',d'),$$

where $R'_q = \{x \in [m'] : g(x)_{h^{-1}(q)} \in R_{h^{-1}(q)}\}$.

For $J \not\subseteq [t]$,

$$|P_{x\in[m']}[g(x)_J \in \tilde{R}_J] - vol(\tilde{R}_J)| = 0,$$

because $\overline{R}_j = \emptyset$ for $j \notin [t]$. For $J \subseteq [t]$ with $|J| \le k$,

$$|P_{x\in[m']}[g(x)_J \in \tilde{R}_J] - vol(\tilde{R}_J)| \le 2^{|J|}\varepsilon',$$

as each $\tilde{R}_J$ is the union of at most $2^{|J|}$ rectangles from $\mathcal{B}(m,|J|,|J|)$. Then

$$\begin{aligned}|E_{h\in H'}[vol(R^{(h)})] - vol(R)| &\le \sum_{j=0}^{k}\sum_{J\subseteq[t],|J|=j} 2^j\varepsilon' + E_{h\in H'}\left[\sum_{q\in[d']}\tau_k(h^{-1}(q))\right]\\ &\le \left(\frac{t}{k}\right)^{O(k)}\varepsilon' + O(\varepsilon)\\ &\le O(\varepsilon).\end{aligned}$$

Combined with the INW-generator, we get an ε-generator. The number of random bits used is

$$\begin{aligned}&|H'| + O\left(\log m' + \left(\log d' + \log\frac{1}{\varepsilon}\right)\log d'\right)\\ &= O\left(\log\frac{k\log d}{\varepsilon}\right) + O\left(k + \log\log d + \log\frac{1}{\varepsilon'} + \left(\frac{\log 1/\varepsilon}{k} + \log\frac{1}{\varepsilon}\right)\frac{\log 1/\varepsilon}{k}\right)\\ &= O\left(\log k + \log\log d + \log\frac{1}{\varepsilon} + k\log\frac{t}{k} + \frac{1}{k}\log^2\frac{1}{\varepsilon}\right),\end{aligned}$$

which is

$$O\left(\log\log d + \log\frac{1}{\varepsilon}\log^{1/2}\frac{t}{\log 1/\varepsilon}\right)$$

when $k = \frac{\log 1/\varepsilon}{\log^{1/2}\frac{t}{\log 1/\varepsilon}}$. ∎

Acknowledgements. We would like to thank David Barrington for correcting some mistakes and making useful suggestions. We would like to thank Shiyu Zhou for telling us the result in [18] and for some helpful comments. We would also like to thank Amnon Ta-Shma, Jaikumar Radhakrishnan, and Avi Wigderson for reading a preliminary version of this paper.

References

[1] N. Alon, L. Babai, and A. Itai: A fast and simple randomized parallel algorithm for the maximal independent set problem, *Journal of Algorithms*, **7** (1986), 567–583.

[2] N. Alon, O. Goldreich, J. Hastad, and R. Peralta: Simple constructions of almost k-wise independent random variables, *Random Structures and Algorithms*, **3(3)** (1992), 289–303.

[3] P. Auer, P. Long, and A. Srinivasan: Approximating hyper-rectangles: learning and pseudo-random sets, *Journal of Computer and System Sciences*, **57** (1998), 376–388.

[4] R. Armoni, M. Saks, A. Wigderson, and S. Zhou: Discrepancy sets and pseudorandom generators for combinatorial rectangles, In: *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, 412–421, 1996.

[5] S. Chari, P. Rohatgi, and A. Srinivasan: Improved algorithms via approximations of probability distributions, *Journal of Computer and System Sciences*, **61** (2000), 81–107.

[6] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Velicković: Approximations of general independent distributions, In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, 10–16, 1992. Journal version appeared as the following.

[7] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Velicković: Efficient approximation of product distributions, *Random Structures and Algorithms*, **13(1)** (1998), 1–16.

[8] O. Goldreich and A. Wigderson: Tiny families of functions with random properties: a quality-size trade-off for hashing, *Random Structures and Algorithms*, **11(4)** (1997), 315–343.

[9] R. Impagliazzo, N. Nisan, and A. Wigderson: Pseudorandomness for network algorithms, In: *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, 356–364, 1994.

[10] R. Impagliazzo, R. Shaltiel, and A. Wigderson: Extractors and pseudo-random generators with optimal seed length, In: *Proceedings of 32nd Symposium on Theory of Computing*, 1–10, 2000.

[11] N. Linial, M. Luby, M. Saks, and D. Zuckerman: Efficient construction of a small hitting set for combinatorial rectangles in high dimension, *Combinatorica*, **17** (1997), 215–234.

[12] M. Luby: A simple parallel algorithm for the maximal independent set problem, *SIAM Journal on Computing*, **15(4)** (1986), 1036–1053,

[13] J. Naor and M. Naor: Small-bias probability spaces: efficient constructions and applications, *SIAM Journal on Computing*, **22(4)** (1990), 838–856.

[14] N. Nisan: Pseudorandom generators for space-bounded computation, *Combinatorica*, **12** (1992), 449–461.

[15] N. Nisan: Extracting randomness: how and why—a survey, In: *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, 44–58, 1996.

[16] N. Nisan and D. Zuckerman: Randomness is linear in space, *Journal of Computer and System Sciences*, **52(1)** (1996), 43–52.

[17] R. Raz, O. Reingold, and S. Vadhan: Extracting all the randomness and reducing the error in Trevisan's extractors, In: *Proceedings of 31st Symposium on Theory of Computing*, 149–158, 1999.

[18] J. Radhakrishnan and A. Ta-Shma: Private communication.
[19] A. Ta-Shma, C. Umans, and D. Zuckerman: Loss-less condensers, unbalanced expanders, and extractors, to appear in *Proceedings of 33rd Symposium on Theory of Computing*, 2001.
[20] L. Trevisan: Construction of extractors using pseudorandom generators, In: *Proceedings of 31st Symposium on Theory of Computing*, 141–148, 1999.

Chi-Jen Lu

Institute of Information Science
Academia Sinica
Taipei, Taiwan, ROC
cjlu@iis.sinica.edu.tw