Almost Optimal Dispersers

Amnon Ta-Shma *
ICSI

Abstract

A (K, ϵ) disperser graph $G = (V_1, V_2, E)$ is a bipartite graph with the property that for any subset $A \subseteq V_1$ of cardinality K, the neighbors of A cover at least $1 - \epsilon$ fraction of the vertices of V_2. Such graphs have many applications in derandomization. Saks, Srinivasan and Zhou presented an explicit construction of $(K = 2^k, \epsilon)$ disperser graphs $G = (V = [2^n], W, E)$ with an almost optimal degree $D = poly(n, \epsilon^{-1})$, for every $k \geq n^{\Omega(1)}$. We extend their result for any parameter $k \leq n$.

1 Introduction

A disperser is a sparse graph with strong random-like properties. As such, explicit dispersers have numerous applications in derandomization (many of them appearing in the excellent survey paper by Nisan [Nis96]). The question whether explicit constructions of such graphs do exist attracted much research in the last decade [Sip88, Zuc90, Zuc91, NZ93, SZ94, SSZ95, Zuc96]. Saks, Srinivasan and Zhou [SSZ95] showed an almost optimal disperser construction for certain parameters. In this paper we show how to extend their work to give explicit constructions for the whole range of parameters.

Definition 1.1 (disperser) A bipartite graph $G = (V = [N = 2^n], W = [M], E)$ with a left degree D is a (K, ϵ) disperser, if any subset $A \subseteq V$ of cardinality K has more than $(1 - \epsilon)M$ distinct neighbors in W.

Given k and ϵ (as functions of the growing parameter n) we would like to have the degree as small as possible, while maintaining M as large as possible. It is known that any non-trivial disperser must have degree D that is at least $D \geq \Omega(\log(N)\frac{1}{\epsilon})$ and the size of the right hand side M is at most $M \leq O(\frac{KD}{\log(\frac{1}{\epsilon})})$ [RTS97]. Indeed, if we build such a graph by choosing D independent neighbors for each $v \in V$, we get an optimal disperser with tight degree D (up to a multiplicative factor) and maximal number of right hand side vertices (again, up to a multiplicative factor).

Saks, Srinivasan and Zhou [SSZ95] showed a disperser construction with degree $D = poly(n, \epsilon^{-1})$ for sets of size $K = 2^k$ with $k \geq n^{\Omega(1)}$. Their work was based on earlier results and a new combinatorial construction of a small family of segmentations (that will be described later). We simplify the SSZ combinatorial construction, giving a simpler analysis with better bounds. We use the new family of segmentations to build the disperser graph in much the same way as is done in [SSZ95], but we prove the correctness of the construction using the terminology of [Ta-96], which results in a simpler proof with tighter bounds.

Theorem 1 For every constant $\gamma < 1$, $\epsilon \geq 2^{-n^\gamma}$ and any $k \leq n$, there is an explicit $(K = 2^k, \epsilon)$ disperser $G = (V = [N = 2^n], W = [M = 2^{k - poly(\log n, \log(\frac{1}{\epsilon}))}], E)$ with degree $D \leq poly(n, \frac{1}{\epsilon})$.

Although the degree of our disperser is almost optimal, the entropy-loss is not. In an ideal graph we can hope that any set of K vertices of degree D covers $\Omega(KD)$ vertices. In the disperser we presented, however, the right hand size has size $M = K/2^{polylog(n)}$. For convenience we measure the log of this loss, i.e., the entropy loss of a disperser is $\log(K) + \log(D) - \log(M)$. The [SSZ95] disperser has $n^{\Omega(1)}$ entropy loss, while our disperser has $poly(\log(n), \log(\frac{1}{\epsilon}))$ entropy loss. The reason we achieve much smaller entropy loss is connected to the fact that we have good dispersers for any parameter k, and uses the existence of a good extractor presented in [Ta-96]. Yet, even the entropy loss

*Current address: International Computer Science Institute, 1947 Center Street, Berkeley California 94704, USA. E-mail: amnon@icsi.berkeley.edu . This work was partially done while the author was at the Department of Computer Science, Weizmann Institute of Science, Rehovot, Israel. This work was supported by a Phil Zacharia postdoctoral fellowship.

STOC '98 Dallas Texas USA

we achieve is still away from the optimal one which is only $\log\log(\frac{1}{\epsilon}) + O(1)$, and reducing the min-entropy loss to the optimal is an important open problem with many applications (e.g. for the construction of explicit a-expanding graphs and depth 2 super-concentrators, see the survey paper of [Nis96]).

The paper is constructed as follows: in Section 2 we give definitions and a few preliminaries. In Section 3 we present the new small family of segmentations, in Section 4 we show how to build from it a structure that in particular implies the existence of good dispersers. Finally, in Section 5 we discuss some open problems.

2 Preliminaries

In the following we give formal definitions for the different objects we are going to handle later. However, to get a wider picture and a better understanding of the relationships between these different objects we recommend the reader to look at the survey paper of [Nis96].

We are going to work with distributions. A probability distribution X over Λ, is a function $X : \Lambda \mapsto [0,1]$ s.t. $\Sigma_{a\in\Lambda} X(a) = 1$. We measure the amount of randomness in the distribution by considering the min-entropy.

Definition 2.1 (min-entropy) The min-entropy of a distribution X is $H_\infty(X) = min_a(-\log_2 X(a))$,

In general we would like to build an extractor which is a function that extracts randomness from a random source X having high minentropy. It is easy to see that no such deterministic function exists, so we are forced to use some (hopefully few) truly random bits. We say $E(x,y)$ extracts randomness from X if the distribution of $E(X,U)$, obtained by picking x according to the distribution X (we denote this by $x \in X$) and y uniformly at random (we denote this as $y \in U$) and computing $E(x,y)$, is ϵ close to the uniform distribution U.

Definition 2.2 (statistical difference) Two distributions X and Y over the same space Λ have statistical distance $d(X,Y) = \frac{1}{2}|X-Y|_1 = \frac{1}{2}\Sigma_{a\in\Lambda}|X(a)-Y(a)|$.

Definition 2.3 (extractor for a class C) Let C be a set of distributions X over $\{0,1\}^n$. A function $E : \{0,1\}^n \times \{0,1\}^t \mapsto \{0,1\}^m$ is an ϵ extractor for C, if for any distribution $X \in C$ the distribution of $E(X,U_t)$ (i.e. the distribution over $\{0,1\}^m$ obtained by choosing $x \in X$, and $y \in_U \{0,1\}^t$ and computing $E(x,y)$) is ϵ-close to the uniform distribution U_m over $\{0,1\}^m$.

The ultimate goal is finding an optimal extractor for all distributions having k min-entropy, which is still open for most cases [Nis96]. However, it turns out that the problem is much simpler when we restrict ourselves to random sources having more structure:

Definition 2.4 [CG88] (block-wise source) Suppose X is a distribution over $\{0,1\}^n$, and π is a partition of $[1..n]$ into l consecutive blocks. Define the induced random variable X_i^π which is the result of the random variable X when restricted to the i'th block of π. Thus, $X = X_1^\pi \circ \ldots \circ X_l^\pi$ where the X_i^π are possibly correlated.

We say X is a $(\pi, z_1, \ldots, z_l)$ block-wise source, if for any $x = x_1 \circ \ldots x_l \in X$ and any $1 \le i \le l$ we have

$$H_\infty(X_i^\pi \mid X_{i-1}^\pi = x_{i-1}, \ldots, X_1^\pi = x_1) \ge z_i$$

Many times we omit the partition π and simply say that X is a $(z_1, \ldots, z_l)$ block-wise source.

Nisan and Zuckerman [NZ93] and later Srinivasan and Zuckerman [SZ94] showed how to extract randomness from block-wise sources:

Definition 2.5 (a block-wise extractor) A $((z_1, \ldots, z_l), \epsilon)$ block-wise extractor is a function $E : \{0,1\}^n \times \{0,1\}^t \mapsto \{0,1\}^m$ that is an extractor for the set of all $(\pi; z_1, \ldots, z_l)$ block-wise sources. E may depend on π, and when we want to emphasize this we write E_π.

Definition 2.6 (Explicit Extractor): We say an extractor $E : \{0,1\}^n \times \{0,1\}^t \mapsto \{0,1\}^m$ is explicit, if $E(x,y)$ can be computed in time polynomial in the input length $|x| + |y| = n + t$. We say a block-wise extractor is explicit, if given π, x, y, $E_\pi(x,y)$ can be computed in time polynomial in $n + t$.

Nisan and Zuckerman [NZ93] showed a nice block-wise extractor that was later improved by Srinivasan and Zuckerman [SZ94], yielding:

Fact 2.1 [SZ94] There is a constant $C_{sz} > 1$ s.t. for any $b, \epsilon > 0, l \ge 1$, $z_i' \ge C_{sz}^{l-i} b$ there is an explicit $(\pi; z_1', \ldots, z_l'; \epsilon)$ block-wise extractor $E_\pi : \{0,1\}^n \times \{0,1\}^t \mapsto \{0,1\}^m$ with $m = \Omega(C_{sz}^l \cdot b)$ output bits and $t = O(\log(n) + b + \log(\frac{1}{\epsilon}))$ truly random bits.

Extractors are much stronger than dispersers. In fact the relationship between extractors and dispersers is much the same as between one-sided error and two-sided error computations. Next, we are going to weaken the notion of an extractor to that of a somewhere random extractor. We start with a definition of a somewhere random source, taken with some changes from [Ta-96]:

Definition 2.7 [Ta-96] [1] (somewhere random source) $B = B_1 \circ \ldots \circ B_b$ is a b–block (m, ϵ, η) somewhere random source, if each B_i is a random variable over $\{0,1\}^m$, and there is a random variable Y over $[0..b]$ s.t.:

[1] The definition here is a bit different to the one appearing in [Ta-96]. There the conditions are $Prob(Y = 0) \le \eta$ and for any i, $d((B_i|Y = i), U_m) \le \epsilon$. The two definitions are actually equivalent. To see that our definition implies the one in [Ta-96], define $Y'(x)$ to be $Y(x)$ if $Y(x) = i$ and $Prob(Y = i) \ge \frac{\eta}{b}$, and 0 otherwise.

- For any $i \in [1..b]$: $Prob(Y = i) \geq \frac{\eta}{b} \implies d((B_i|Y = i), U_m) \leq \epsilon$.
- $Prob(Y = 0) \leq \eta$.

We call Y an (ϵ, η) selector for B.

We then define a somewhere random extractor to be a function whose output is a somewhere random source.

Definition 2.8 (somewhere random extractor) Let $E : \{0,1\}^n \times \{0,1\}^t \mapsto (\{0,1\}^m)^b$ be a function. Given a distribution X on $\{0,1\}^n$ we can define the distribution $E(X, U_t) = B_1 \circ \ldots \circ B_b$ obtained by picking $x \in X, y \in U_t$ and computing $E(x, y)$.

We say E is a (k, ϵ, η) somewhere-random extractor if for any distribution X with $H_\infty(X) \geq k$, $E(X, U_t) = B_1 \circ \ldots \circ B_b$ is a b-block (m, ϵ, η) somewhere random source. When $\epsilon = \eta$ we say E is an (m, ϵ) somewhere-random extractor.

Given a random source with k min-entropy an extractor is forced to output a single distribution that is close to uniform. A somewhere random extractor can instead output many distributions, with the guarantee that one of which (but may be only one) is uniform. Thus, a somewhere random extractor is much closer in spirit to a disperser. However, it is still stronger than a disperser because the requirement that one of the distributions is close to uniform is stronger than the requirement that we cover many vertices.

Lemma 2.1 Let $\eta < \frac{1}{2}$. Let $D : \{0,1\}^n \times \{0,1\}^t \mapsto (\{0,1\}^m)^b$ be a (k, ϵ, η) somewhere random extractor. Suppose $D(x, y) = D_1(x, y) \circ \ldots \circ D_b(x, y)$. Define the bipartite graph $G = (V = \{0,1\}^n, W = \{0,1\}^m, E)$ where $(v, w) \in E$ iff there is some $1 \leq i \leq b$ and some $z \in \{0,1\}^t$ s.t. $w = D_i(v, z)$. Then G is a $(K = 2^k, \epsilon)$ disperser.

Proof: Let $X \subseteq V$ be of cardinality 2^k. let $Y = Y(x, z)$ be the selector function for D. Since $\eta < \frac{1}{2}$ there must be some $i > 0$ s.t. $Prob(Y = i) \geq \frac{\eta}{b}$. Hence, $d(\ (D_i(X, U_t) \mid Y = i)\ ,\ U_m) \leq \epsilon$. Thus, even when we restrict ourselves only to $x \in X$ and $z \in \{0,1\}^t$ s.t. $Y(x, z) = i$, these edges induce a distribution which is ϵ close to the uniform distribution over W. In particular, these edges miss at most an ϵ fraction of W. Therefore, in particular, the set of all neighbors of X, $\Gamma(X)$ miss at most an ϵ fraction of W. □

In Section 4 we construct good somewhere random extractors, and combining it with the above lemma we get Theorem 1.

3 A Small Family of Segmentations

In this section we strengthen a combinatorial lemma appearing in [SSZ95]. Loosely speaking, the lemma claims there is a small family of segmentations of $[1..n]$ into few blocks l, s.t. for any possible way of dividing weight among these n points, there is at least one segmentation in the family s.t. any block in the segmentation has high weight.

Definition 3.1 (segmentations) We say π partitions $[1..n]$ into l blocks if it produces l segments $B_1 = [1..b_1], B_2 = [b_1 + 1..b_2], \ldots, B_l = [b_{l-1} + 1..n]$. We call π a segmentation.

A family F of segmentations of $[1..n]$ into l blocks is $(k, [z_1 = z, \ldots, z_l = z])$–good for any weight function $p : [1..n] \mapsto [0, w]$ if for any such p with $\Sigma_i p(i) \geq k$ there is at least one segmentation $\pi \in F$ s.t. π partitions $[1..n]$ into blocks $B_1, \ldots, B_l$ s.t. for every $1 \leq i \leq l$ the total weight of vertices in B_i is at least z_i.

The next lemma is a generalization of the lemma appearing in [SSZ95]. The proof presents a somewhat simpler algorithm, with a simpler analysis and better parameters. The idea behind the construction is, though, essentially the same as in [SSZ95].

Lemma 3.1 Suppose $\frac{k}{2^l} - z \log(n) \geq w$ for some positive values k, l, n, z, w. Then there is a family F of segmentations of $[1..n]$ into l blocks that is $(k, [z_1 = z, \ldots, z_l = z])$–good for any $p : [1..n] \mapsto [0, w]$, and such that the size of F is at most n^2.

Proof: W.l.o.g we can assume n is a power of two, for otherwise we can just add dummy zero weights. We take a balanced binary tree over n leaves. The leaves of the subtree whose root is v cover a consequent subset of $[1..n]$ which we denote by $dom(v)$.

Construction of F :

- Take all possible paths from the root to a leave.
- For each path $v_1, \ldots, v_{\log(n)}$ take all subsets of size $l - 1$ of $\{v_1, \ldots, v_{\log(n)}\}$.
- Each such subset $\{v_{i_1}, \ldots, v_{i_{l-1}}\}$ of $l - 1$ vertices determines a partitioning of $[1..n]$ into l blocks as follows: each v_{i_k} puts a partition point on the middle point of $dom(v_{i_k})$.

Clearly there are n possible paths, and each path can have at most $2^{\log(n)} = n$ subsets. Thus, the number of partitions in F is at most n^2.

F is good :

Let $p : [1..n] \mapsto [0..w]$ be s.t. $\Sigma p_i \geq k$. Under p the weight of a vertex v, $weight(v)$, is the sum of the weights of vertices in $dom(v)$. Let us concentrate on a distinguished path p_{hcavy}: the path that starts at the root, and each time goes to the heavier son. Let us call a vertex v on p_{hcavy} good if both the weights of his left and right sons are at least z.

Claim 3.1 There are at least $l-1$ good vertices in p_{heavy}.

Proof: The weight of the root is k. When we encounter a good vertex we choose the heavy son and therefore we decrease the weight by at most half. When we encounter a bad vertex we lose at most z weight. If there are t good vertices then the leaf must have weight at least $\frac{k}{2^t} - \log(n)z$. However, any leaf has weight at most w. Thus, $\frac{k}{2^t} - \log(n)z \leq w$. However, we started with a big enough k, namely, $w \leq \frac{k}{2^t} - \log(n)z$ and hence $t \geq l$. □

Thus, these $l-1$ good vertices define a partition into l blocks s.t. each of them has weight at least z.

□

A similar analysis for the case where the different blocks have different weights give:

Lemma 3.2 Suppose $\frac{k}{2^t} - \Sigma_{i=1}^{l} z_i \geq w$ for some positive values k, l, n, z_i, w. Then there is a family F of segmentations of $[1..n]$ into l blocks that is $(k, [z_1, \ldots, z_l])$-good for any $p : [1..n] \mapsto [0, w]$, and such that the size of F is at most n^2.

The proof is similar to the proof of lemma 3.1 and appears in the appendix.

4 Efficient Somewhere random extractors

4.1 The basic lemma

Imagine being an extractor. You are given a string $x \in \{0,1\}^n$ that has large min-entropy. The first question you might ask yourself is "which of the bits of X are "more" random?". It turns out that instead of measuring the surprise of the i'th bit in the random source X, an even better idea is to consider the surprise of the i'th bit in the given string, i.e., to consider $q_i = Prob(X_i = x_i \mid x_{i-1}, \ldots, x_1)$ as our measure of surprise. This idea originates in the work of [NZ93].

When taking this as our measure of surprise we can see that if X has high min-entropy, almost all strings $x \in X$ have many surprising bits. This can be viewed as giving weights to the n bits, the weight of the i'th bit corresponds to its amount of "surprise", that add up together to something large. At the bit level we do not know which bit has high weight. However, at the block level we can use the small family of segmentations, and almost by definition, one of the segmentations in our small family must be good in the sense that it partitions $[1..n]$ into surprising blocks. This is the same as saying that the resulting blocks form a somewhere random source, which we already know to handle. Thus trying all the possible segmentations (and there aren't too many of them) we know one of them will work and give us an almost uniform distribution. This idea, of trying all possible segmentations from a small fixed family, comes directly from the [SSZ95] paper, but the implementation here is more direct than in [NZ93, SSZ95], resulting in a simpler and stronger analysis.

Lemma 4.1 Suppose

- $\frac{k}{2^t} - \Sigma_{i=1}^{l} z_i \geq w = \log(\frac{n}{\epsilon})$, and
- There is an explicit $((z'_1 = z_1 - \log(\frac{|F|}{\epsilon}), \ldots, z'_l = z_l - \log(\frac{|F|}{\epsilon}), \epsilon)$ block-wise extractor $E_\pi : \{0,1\}^n \times \{0,1\}^t \mapsto \{0,1\}^m$.

Then there is an explicit (k, ϵ) somewhere-random extractor $E : \{0,1\}^n \times \{0,1\}^t \mapsto (\{0,1\}^m)^{|F|}$.

Proof:

From Lemma 3.2 we know there exists an explicit family F of segmentations of $[1..n]$ into l blocks that is $(k, [z_1, \ldots, z_l])$-good for any $p : [1..n] \mapsto [0, w]$, and the size of the family F is at most n^2.

Let X be any random source over $\{0,1\}^n$ with $H_\infty(X) \geq k$ and let $x \in X$.

Algorithm :

Choose y uniformly from $\{0,1\}^t$.

Any $\pi \in F$ partitions $[1..n]$ into l segments. For any such π let $B_\pi = E_\pi(x, y)$. The output is the $|F|$ blocks B_π for any $\pi \in F$.

Correctness :

We call an $x \in X$ "rare" if there is some i s.t. $Prob(X_i = x_i \mid x_{i-1}, \ldots, x_1) \leq \frac{\epsilon}{n}$. It is easy to verify that $Prob(x \text{ is rare}) \leq n\frac{\epsilon}{n} = \epsilon$. If x is rare we let $Y = 0$ (i.e., we failed).

For a non-rare $x \in X$ denote $q_i \stackrel{\text{def}}{=} Prob(X_i = x_i \mid X_{i-1} = x_{i-1}, \ldots, X_1 = x_1)$. Clearly $\prod q_i = Prob(X = x) \leq 2^{-k}$. Define $p = p(x) : [1..n] \mapsto [0, w]$ by $p_i = \log(\frac{1}{q_i})$. It can be easily checked that $\Sigma p_i \geq k$, and $0 \leq p_i$. Furthermore, since x is not rare $p_i(x) \leq \log(\frac{n}{\epsilon}) = w$. Therefore, there is at least one partition in F that is good for p, and let us fix such one π. Let $Y = Y(x) = \pi$. Note that the weight function $p = p(x)$ and the segmentation $Y = Y(x)$ depend on x.

Denote by X_i^π the distribution of X when restricted to B_i^π, the i'th segment of π. Similarly for a string $b \in \{0,1\}^n$, b_i^π denotes the i'th block of b under the segmentation π. We now show that:

Claim 4.1 For any $\delta > 0$, if $Prob(Y = \pi) \geq \delta$ then $((X_1^\pi \circ \ldots \circ X_l^\pi) \mid Y = \pi)$ is a $(z_1 - \log(\frac{1}{\delta}), \ldots, z_l - \log(\frac{1}{\delta}))$ block-wise source.

Proof: For $1 \leq i \leq l$ let $b_1^\pi, \ldots, b_i^\pi$ be such that they can be extended to some b with $Y(b) = \pi$.

Since $Y(b) = \pi$ we have that under the weight function $p = p(b)$, the weight of B_i^π is at least z_i. Consequently:

$$Prob(X_i^\pi = b_i^\pi \mid X_1{}^\pi = b_1^\pi, \ldots, X_{i-1}^\pi = b_{i-1}^\pi) =$$

$$\prod_{j \in B_i^\pi} Prob(X_j = b_j | X_1 = b_1, \ldots, X_{j-1} = b_{j-1}) =$$

$$\prod_{j \in B_i^\pi} q_j = 2^{-\Sigma_{j \in B_i^\pi} p_j} \leq 2^{-z_i}$$

Therefore,

$$Prob(X_i^\pi = b_i^\pi | X_1^\pi = b_1^\pi, \ldots, X_{i-1}^\pi = b_{i-1}^\pi, Y = \pi)$$

$$\leq \frac{Prob(X_i^\pi = b_i^\pi \mid X_1^\pi = b_1^\pi, \ldots, X_{i-1}^\pi = b_{i-1}^\pi)}{Prob(Y = \pi)} \leq \frac{2^{-z_i}}{\delta}$$

And the claim follows.

Therefore, if we pick $\delta = \frac{\epsilon}{|F|}$, we get that whenever $Prob(Y = \pi) \geq \frac{\epsilon}{|F|}$ we have that $(X_1^\pi \circ \ldots \circ X_l^\pi \mid Y = \pi)$ is a $(z_1', \ldots, z_l')$ block-wise source $(z_i' = z_i - \log(\frac{1}{\delta}))$, and therefore $(B_\pi \mid Y = \pi)$ is ϵ close to uniform. And the correctness follows.

□

4.2 Plugging specific parameters

For the basic lemma to work we need good block-wise extractors, and as mentioned in Section 2 good block-wise extractors do exist. Combining the combinatorial construction of Lemma 3.2 (or even Lemma 3.1) and the block-wise extractor of [SZ94] (Fact 2.1) with Lemma 4.1, and plugging the parameters $l = O(\log\log(n))$, $t = O(\log(n) + \log(\frac{1}{\epsilon}))$ and $k = polylog(n)\log(\frac{1}{\epsilon})$ we get:

Corollary 4.2 There is a $(k = poly(\log n, \log(\frac{1}{\epsilon})), O(\epsilon))$ somewhere random extractor $D : \{0,1\}^n \times \{0,1\}^t \mapsto (\{0,1\}^{\log^2 n})^{n^2}$ with $t = O(\log(n) + \log(\frac{1}{\epsilon}))$.

In general if we treat l as a parameter (which might be smaller than $\log\log(n)$ or bigger) we get:

Theorem 2 There is a $(k = O(2^{O(l)}(\log n + \log(\frac{1}{\epsilon}))), O(\epsilon))$ somewhere random extractor $D : \{0,1\}^n \times \{0,1\}^t \mapsto (\{0,1\}^{2^l \log(n)})^{n^2}$ with $t = O(\log(n) + \log(\frac{1}{\epsilon}))$.

If we take $\epsilon \leq 1/poly(n)$, we get a useful somewhere random extractor for any minentropy, even those that are smaller than $\log^2 n$! Given a source with $2^{O(l)}\log(n)$ entropy, we can invest the optimal number of $O(\log(n) + \log(\frac{1}{\epsilon}))$ truly random bits, and get in return blocks with $2^l \log n$ somewhere quasi-random bits. As we saw, this in particular implies good dispersers for any min-entropy. The entropy loss in the system is, however, not optimal.

4.3 Composing somewhere random extractors

Now we show that if we can obtain $\log^2 n$ quasi-random bits then we can use them to further extract all of the remaining min-entropy in the source using the extractor in [Ta-96].

Algorithm 4.1 Suppose

- $E_1 : \{0,1\}^n \times \{0,1\}^{d_1} \mapsto (\{0,1\}^{d_2})^l$ is a (k_1, ζ_1, η_1)- somewhere-random extractor,
- $E_2 : \{0,1\}^n \times \{0,1\}^{d_2} \mapsto \{0,1\}^{m_2}$ is a (k_2, ζ_2)- extractor

Define $E_2 \ominus E_1 : \{0,1\}^n \times \{0,1\}^{d_1} \mapsto (\{0,1\}^{m_2})^{nl}$ as follows:

Input : $a \in \{0,1\}^n$, $r_1 \in \{0,1\}^{d_1}$.

Output :

For $1 \leq i \leq n$ and $1 \leq j \leq l$,

- Let q_i^j be the j'th block in the output of $E_1(a_{[i,n]}, r_1)$. And,
- Let $z_i^j = E_2(a_{[1,i-1]}, q_i^j)$.

The output contains nl blocks, where the (i,j) output block is z_i^j.

Theorem 3 (Implicit in [Ta-96]) Suppose E_1, E_2 are as above. Then for every safety parameter $s > 0$, $E_1 \ominus E_2$ is an $(k_1 + k_2 + s, \zeta_1 + \zeta_2, O(n(\eta_1 + 2^{-s/3}))$- somewhere random extractor.

In Theorem 2 we obtained a somewhere random extractor outputting blocks of length $polylog(n)$. In [Ta-96] it was shown that with that amount of truly random bits we can extract all of the remaining min-entropy:

Fact 4.1 [Ta-96] For every constant $\gamma < 1$, $\epsilon \geq 2^{-n^\gamma}$, and every $k = k(n)$ there is an explicit (k, ϵ) extractor $E : \{0,1\}^n \times \{0,1\}^{polylog(n)\cdot\log(\frac{1}{\epsilon})} \mapsto \{0,1\}^k$.

Putting it together we get:

Theorem 4 For every constant $\gamma < 1$, $\epsilon \geq 2^{-n^\gamma}$ and any $k \leq n$, there is an explicit $(k, O(\epsilon))$ somewhere random extractor $D : \{0,1\}^n \times \{0,1\}^t \mapsto (\{0,1\}^{k - poly(\log n, \log(\frac{1}{\epsilon}))})^{n^3}$ with $t = O(\log(n) + \log(\frac{1}{\epsilon}))$.

5 Further Work

In this paper we constructed efficient somewhere random extractors for sources having any min-entropy. In particular this reduces the problem of finding explicit general extractors, to that of finding explicit extractors for somewhere random sources, for if we can do the later than by our result we can also do the former. Extractors for somewhere random sources are called "mergers"

in [Ta-96], where some explicit constructions with non-optimal degree are presented. It will be very interesting (and useful) to have a direct (and hopefully efficient) construction for mergers.

Another open problem is reducing the entropy loss to $O(\log(n))$. As stated earlier this has some beautiful consequences (e.g. for the construction of explicit depth 2 superconcentrators, see [Nis96]). It seems that Theorem 2 gives a lead for making progress on this problem, as it supplies an optimal somewhere-random extractor working for very low min-entropies and retrieving a (relatively) large amount of randomness. It seems that once we have good somewhere random mergers we might use Theorem 2 to further reduce our min-entropy losses.

References

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM Journal on Computing, 17(2):230–261, 1988.

[Nis96] N. Nisan. Refining randomness: Why and how. In Annual Conference on Structure in Complexity Theory, 1996.

[NZ93] N. Nisan and D. Zuckerman. More deterministic simulation in logspace. In Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, ACM, pages 235–244, 1993.

[RTS97] J. Radhakrishnan and A. Ta-Shma. Tight bounds for depth-two superconcentrators. In IEEE Symposium on Foundations of Computer Science (FOCS), 1997.

[Sip88] Sipser. Expanders, randomness, or time versus space. Journal of Computer and System Sciences, 36, 1988.

[SSZ95] M. Saks, A. Srinivasan, and S. Zhou. Explicit dispersers with polylog degree. In Proceedings of the 26th Annual ACM Symposium on the Theory of Computing, ACM, 1995.

[SZ94] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. In Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science, 1994.

[Ta-96] A. Ta-Shma. On extracting randomness from weak random sources. In STOC, 1996.

[Zuc90] D. Zuckerman. General weak random sources. In Proceedings of the 31st Annual IEEE Symposium on the Foundations of Computer Science, pages 534–543, 1990.

[Zuc91] D. Zuckerman. Simulating BPP using a general weak random source. In Proceedings of the 32nd Annual IEEE Symposium on the Foundations of Computer Science, pages 79–89, 1991.

[Zuc96] David Zuckerman. Randomness-optimal sampling, extractors, and constructive leader election. In ACM Symposium on Theory of Computing (STOC), pages 286–295, Philadelphia, Pennsylvania, May 1996.

A A Small Family of Segmentations

We prove Lemma 3.2.

Proof:

The construction of F is exactly as it is in Lemma 3.1. Hence, clearly, $|F| \leq n^2$. Next we prove that F is good.

F is good :

Let $p : [1..n] \mapsto [0..w]$ be s.t. $\Sigma p_i \geq k$. As before we concentrate on the distinguished path p_{heavy}: the path that starts at the root and each time goes to the heavier son. We go along the path p_{heavy} from the root to the leaf and we label the vertices as being good or bad. As before, good vertices will be chosen as partition points, and the partition point is laid in the middle of the domain they cover. Our main claim will be that if k is big enough there are always enough good points on the path p_{heavy}.

Suppose we labeled the first i vertices $v_1, \ldots, v_{i-1}$ of the path p_{heavy} as being good or bad and declared t of them as being good yielding t partition points $\{p_1, \ldots, p_t\}$. Denote by q the middle point of $dom(v_i)$, and by a and b ($a < q < b$) the partition points in $\{p_1, \ldots, p_t\}$ that are closest to q (if no partition point smaller (or bigger) than q exists we take the end point). Let us denote by t_r the number of partition points greater than q ($t_r \geq 0$), and similarly t_l is the number of partition points smaller than q. We call v_i good if:

- v_{i+1} (the heavy son of v_i) is the left son of v_i and $weight([\lceil q \rceil..b)) \geq z_{l-t_r}$, or
- v_{i+1} (the heavy son of v_i) is the right son of v_i and $weight([a..\lfloor q \rfloor)) \geq z_{t_l+1}$.

The t partition points $\{p_1, \ldots, p_t\}$ partition $[1..n]$ into $t+1$ blocks. Furthermore, t of these blocks $B_1, \ldots, B_t$ will never be partitioned again. We call these blocks inactive. At each time there is only one active block. Also notice that the inactive blocks cover a prefix and a suffix of $[1..n]$, and therefore we know exactly what their final index should be. Let us denote by k_j the weight of the j'th block from the left. We know that $k_j \geq z_j$. Next we show that at any stage the remaining weight $k - \Sigma_{j=1}^{t} weight(B_j)$ is big.

Lemma A.1 After finding t good vertices $k - \Sigma_{j=1}^{t} weight(B_j)$ is at least $\frac{k}{2^t} - \Sigma_{j=1}^{t_l} z_j - \Sigma_{j=l-t_r+1}^{l} z_j$.

Proof: By induction on t.

The case $t = 0$ ($t_l = t_r = 0$) is trivial.

For $t = 1$ we look at the first good vertex $v = v_i$. Since $v = v_i$ is the first good vertex, we get that the left and right flanks (the prefix and suffix outside $dom(v)$) have weight less than z_l and z_1 respectively. W.l.o.g let us assume v_{i+1} is the left son of v_i. Then $B_1 = [q..b]$, and since v is good it has weight at least z_1. However, its weight is also bounded by above by $\frac{w(dom(v))}{2} + z_1$, since it is composed of the lighter half $[q..c]$ of $dom(v)$ that can contribute at most $w(dom(v))/2$, and the remaining region $[c..n]$ that weights less than z_1. Hence, the remaining weight is at least as required.

Now, assume for t and let us prove for $t+1$. Again, let us consider the t'th good vertex $v = v_i$. Say q is the middle point of $dom(v)$ and that a and b ($a < q < b$) are the partition points in $\{p_1, \ldots, p_t\}$ that are closest to q (if no partition point smaller (or bigger) than q exists we take the end point). W.l.o.g we assume v_{i+1} is the right son of v_i.

By induction we know that even ignoring $B_1, \ldots, B_t$ we still have at least $\frac{k}{2^t}$ minus the sum of the corresponding z_j. Let us now find the weight of the new added block B_{t+1}. By an argument similar to the case $t = 1$, we see that its weight is at most half of what was left, plus a new z_j corresponding the index of the new block. Thus, what is left when we remove the weight of B_{t+1} is at least as stated. □

Now suppose we were able to find only $t < l - 1$ good partition points. This means that we have t inactive blocks $B_1, \ldots, B_t$, t_l covering a prefix $[1..a]$, and t_r covering a postfix $[b..n]$. The remaining block $B = [a + 1..b - 1]$ is unable to support a new block neither in the left, nor in the right. So, $w(B) < z_{t_l+1} + z_{l-t_r} + w$. However, by Lemma A.1, $w(B)$ is at least $\frac{k}{2^t} - \Sigma_{j=1}^{t_l} z_j - \Sigma_{j=l-t_r+1}^{l} z_j$. Thus, $\Sigma_i z_i + w > \frac{k}{2^t} \geq \frac{k}{2^t}$. A contradiction.

□