Simple analysis of graph tests for linearity and PCP

Johan Håstad* Avi Wigderson†

October 5, 2006

Abstract

We give a simple analysis of the PCP with low amortized query complexity of Samorodnitsky and Trevisan [16]. The analysis also applies to the linearity testing over finite fields, giving a better estimate of the acceptance probability in terms of the distance of the tested function to the closest linear function.

Keywords: Linearity testing, PCP, graph test, iterated test, pseudorandomness.

1 Introduction

In the celebrated PCP-theorem [3, 2] it is proved that any arbitrary statement in NP can be checked by a probabilistic verifier which uses $O(\log n)$ random coins and reads only a constant number of bits. Such a proof that is checked by a probabilistic verifier is called a Probabilistically Checkable Proof or simply a PCP.

Apart from being a striking theorem on its own this fact has far reaching consequences for the approximability of NP-hard optimization problems. This connection, which was first established in [11], has produced a large number of results. To obtain sharp inapproximability results it is necessary to have very efficient PCPs.

One aspect that is important is the tradeoff between the number of bits read by the verifier in the proof and the probability that the verifier accepts the proof. Recently, Samorodnitsky and Trevisan [16] constructed a PCP in which the verifier uses logarithmic randomness, reads q bits in the proof and accepts a proof of an incorrect statement with probability $2^{-q+O(\sqrt{q})}$. The main purpose of this paper is to give a simpler proof of this result.

A related problem is Linearity Testing: given oracle access to a Boolean function f on n bits, determine whether it is close to a linear function over $GF[2]^n$. This too was analyzed in [16], who

*Royal Institute of Technology, Stockholm, work done while visiting Institute for Advanced Study, supported by NSF grant CCR-9987077.

†Hebrew University and Institute for Advanced Study, partially supported by NSF grants CCR-9987007 and CCR-9987845

showed that the error in their test has the same dependence on the number of queries as above. We show that our simple analysis carries over to this problem as well. Moreover, we obtain much better error bounds in terms of the distance from f to the closest affine function. In [16] this distance was a lower bound on the error of their test, independently of the number of queries, whereas we can decrease it exponentially. Indeed, as a function of the number of queries our error bounds are near optimal. Since the proof of the linearity test avoids some technicalities of the PCP construction we present this analysis first in Section 3. It turns out to extend naturally from the known case of linear functions over Z_2, to any Z_p for prime p.

Both linearity testing and the PCP proof in [16] use the notion of a "graph test" - each edge in the graph specifies a "basic test", and this set of (dependent!) basic tests is performed simultaneously. We view our analysis of this result as simple since it gives a transparent and intuitive reduction from analyzing the graph test to analyzing a small variant of a single basic test. While [16] also give such a reduction, it is not as direct, and has an intermediate step which seems to miss an intuitive explanation.

A more direct advantage of our analysis can be seen as follows. It is easy to see that the performance of the graph test increases (i.e. the error of the test decreases) with the density of the graph. Therefore [16] use a complete graph. Our analysis reveals another parameter which improves the performance - the largest induced matching in a "typical" subgraph of our graph. While high density and high induced matching seem contradictory, a remarkable construction of [17] gives graphs of nearly quadratic density that are disjoint union of nearly linear size induced matchings. The existence of such graphs is essential to our exponentially improved bounds on linearity testing. For completeness we sketch the construction of [17] in the appendix.

For a more thorough discussion of PCPs and their properties we refer to the papers [7], [12] and for a discussion of the history of the current problem we refer to [16].

2 Preliminaries

Here we recall the Fourier transform over a field of two elements, which will be needed both for linearity testing over this field, as well as for PCPs.

All our Boolean functions map into ± 1 where we let -1 correspond to true. The most commonly used operation is exclusive-or which in our notation is multiplication. For $x, y \in \{-1, 1\}^n$, let $(x_i)_{i=1}^n$ denote the individual coordinates and let xy denote coordinate-wise multiplication. The boolean operator $\wedge$ is defined in the natural way and note that it is *not* multiplication.

Our essential tool is the discrete Fourier transform given by

$$\hat{f}_\alpha = 2^{-n} \sum_{x \in \{-1,1\}^n} f(x)\chi_\alpha(x)$$

where $\alpha \subseteq [n]$ and χ_α are the character functions defined by $\chi_\alpha(x) = \prod_{i \in \alpha} x_i$. We have the

inversion formula

$$f(x) = \sum_{\alpha} \hat{f}_\alpha \chi_\alpha(x)$$

and Parseval's identity tells us that

$$\sum_{\alpha} \hat{f}_\alpha^2 = 2^{-n} \sum_{x} f^2(x) = 1$$

where the last equality comes from the fact that f takes values ± 1.

3 Linearity testing

In the first subsection we define the graph test and informally state our results. In the second we give our simple proof of the bound of [16]. In the third we show that our analysis leads to a much better bound, and demonstrate its near-optimality. All this is done over Z_2. In the last subsection we show that all the results extend naturally to Z_p for every prime p.

3.1 Graph tests - old and new bounds

An n-variate Boolean function is called *linear* if it is the exclusive-or of some fixed subset of its variables. A function is called *affine* if it is either linear or the complement of a linear function.

We are given oracle access to a function f and we are interested to test whether f is close to a linear function. Note that in the current formalism the linear functions are given by the χ_α. Moreover, $\hat{f}_\alpha$ gives the correlation of f with χ_α, and so the largest fraction of inputs in which f agrees with any affine function is given by

$$\frac{1 + \max_\alpha |\hat{f}_\alpha|}{2}$$

and thus it is natural to analyze the performance of a linearity test in terms of $d(f) = \max_\alpha |\hat{f}_\alpha|$. For the remainder of this section we state our results in terms of this distance $d(f)$.

A natural test, first suggested and analyzed by [8] (and thus usually called the BLR test), is to pick two independent random inputs x, y, and test if $f(xy) = f(x)f(y)$. Clearly, if f is linear, it passes this test with probability 1. The main problem is analyzing the acceptance probability if f is "far" from any linear function. As mentioned, this is called the error (or soundness) of the test. It was analyzed in [8], and then in [5], bounding it by $1/2 + d(f)/2$.

Clearly, repeating the BLR test independently many times reduces this error exponentially. However, motivated from issues of saving randomness and reducing the number of queries it was natural to try and analyze *dependent* tests. Such a family of tests, called *graph tests*, was suggested by Samorodnitsky and Trevisan [16].

Graph Test We are given a graph G with k vertices and edge set E and the test proceeds as follows.

1. Pick points $x^{(i)} \in \{-1,1\}^n$ for $i = 1, 2, \ldots k$ independently with the uniform distribution.

2. For each $(i,j) \in E$, test if
$$f(x^{(i)}x^{(j)}) = f(x^{(i)})f(x^{(j)}).$$
and accept if this is true in all cases.

Note that the test makes $k + |E|$ queries and that it performs the BLR linearity test for each edge in the graph reusing old answers. The remarkable property of this test is that despite the fact that these $|E|$ tests are very dependent (being generated only from k points), their joint outcome behaves almost as if they were $|E|$ independent BLR tests. More precisely, denote the acceptance probability of this test by $e(G,f)$. The task is to get the best upper bound on $e(G,f)$ as a function G and $d(f)$. The main result of [16] (stated as Theorem 3.2 below) is

$$e(G,f) \leq 2^{-|E|} + d(f).$$

We first present (in subsection 3.2) our simple analysis of this result, and then proceed (in subsection 3.3) to give some improvements. To explain them, observe that in terms of dependence of the number of queries, the best choice of G is a complete graph. So let $e(k,f) = e(K_k,f)$.

The [16] bound in this case is

$$e(k,f) \leq 2^{-\binom{k}{2}} + d(f),$$

achieved with $k + \binom{k}{2}$ queries. Surprisingly, they show that no hypergraph test (extended in a natural way) can do better on the first component of this bound, when f is the inner product function.

Still, there seems plenty of room for improvement in the second component, but it is not clear how to use their analysis to get it. Using our analysis (and the special graphs of [17] mentioned in the introduction) we proceed to show that

$$e(k,f) \leq 2^{-k^{2-o(1)}} + d(f)^{k^{1-o(1)}}.$$

At the end of this section we note that up to the $o(1)$ terms this bound is best possible in both parameters, giving (for every $d \leq k$) a function f with $d(f) = 2^{-d}$ and $e(k,f) \geq 2^{-dk} = d(f)^k$.

3.2 Simple analysis of the graph test

To analyze the graph test note that the verifier accepts iff

$$\prod_{(i,j)\in E} \frac{1 + f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)})}{2}$$

equals 1. Since this expression takes only 0/1 values, the acceptance probability is its expectation. Expanding the product we arrive at

$$2^{-|E|} \sum_{S \subseteq E} \prod_{(i,j)\in S} f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)}) \tag{1}$$

and we are interested in calculating the expected value of each term. The following lemma is sufficient to establish old results.

Lemma 3.1 *For any $S \neq \emptyset$ we have*

$$E\left[\prod_{(i,j)\in S} f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)})\right] \leq d(f).$$

Proof: Suppose, without loss of generality, that $(1,2) \in S$. We focus on this edge, leaving the variables $x^{(1)}, x^{(2)}$ alone, and fix all other variables to constants. This reduces the analysis of the graph test to (almost) that of one BLR edge test.

Fix $x^{(3)}, \ldots x^{(k)}$ to values $\bar{x}^{(3)} \ldots \bar{x}^{(k)}$ such that

$$E_{x^{(1)},x^{(2)},x_3^{(3)}=\bar{x}^{(3)},\ldots x^{(k)}=\bar{x}^{(k)}}\left[\prod_{(i,j)\in S} f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)})\right] \geq$$

$$E_{x^{(1)},x^{(2)},x^{(3)}\ldots x^{(k)}}\left[\prod_{(i,j)\in S} f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)})\right].$$

With all the values except $x^{(1)}$ and $x^{(2)}$ given constant values we have that

$$\prod_{(i,j)\in S} f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)}) = f(x^{(1)}x^{(2)})g(x^{(1)})h(x^{(2)}),$$

where g and h are two Boolean functions. To be more specific

$$g(x^{(1)}) = f(x^{(1)}) \prod_{j\geq 3 \wedge (1,j)\in S} f(x^{(1)}\bar{x}^{(j)})f(x^{(1)})$$

and a similar formula is true for h. Terms that do not depend on either $x^{(1)}$ or $x^{(2)}$ only contribute a Boolean constant which can be incorporated into h.

Although we started out with one single function we are now in a situation where we are checking a "linear consistency" property of three different, only somewhat related functions. This situation, for three completely independent functions, was already analyzed by Aumann et al. [4] (extending the analysis of [5]) and we use their analysis. The key is to replace each function by its Fourier-expansion.

$$\begin{aligned} E_{x^{(1)},x^{(2)}}[f(x^{(1)}x^{(2)})g(x^{(1)})h(x^{(2)})] &= E_{x^{(1)},x^{(2)}}\left[\sum_{\alpha,\beta,\gamma} \hat{f}_\alpha\chi_\alpha(x^{(1)}x^{(2)})\hat{g}_\beta\chi_\beta(x^{(1)})\hat{h}_\gamma\chi_\gamma(x^{(2)})\right] = \\ &= \sum_{\alpha,\beta,\gamma} \hat{f}_\alpha\hat{g}_\beta\hat{h}_\gamma E_{x^{(1)},x^{(2)}}\left[\chi_\alpha(x^{(1)}x^{(2)})\chi_\beta(x^{(1)})\chi_\gamma(x^{(2)})\right]. \quad (2) \end{aligned}$$

It is not difficult to see that the inner expected value equals 0 unless $\alpha = \beta = \gamma$ in which case it equals 1 and hence (2) equals $\sum_\alpha \hat{f}_\alpha \hat{g}_\alpha \hat{h}_\alpha$. Using Cauchy-Schwartz, we can bound it by

$$\sum_\alpha \hat{f}_\alpha \hat{g}_\alpha \hat{h}_\alpha \leq \max_\alpha |\hat{f}_\alpha| \sum_\alpha |\hat{g}_\alpha \hat{h}_\alpha| \leq \max_\alpha |\hat{f}_\alpha| (\sum_\alpha \hat{g}_\alpha^2)^{1/2} (\sum_\alpha \hat{h}_\alpha^2)^{1/2} \leq \max_\alpha |\hat{f}_\alpha| = d(f). \tag{3}$$

■

Using (1), estimating the term when $S = \emptyset$ by 1, and applying Lemma 3.1 when S is not empty we get.

Theorem 3.2 *[16] The probability that the linearity test accepts is bounded by* $2^{-|E|} + d(f)$.

3.2.1 A (slightly) different proof for the basic test.

In this section we outline a different way to estimate the probability that a graph test accepts. It is in the obvious senses worse than the analysis in the previous section. It is slightly more complicated and gives worse bounds. It is, however, different and still rather simple and since one of the main motivations for the current paper is to present alternative proof-techniques to be used in future papers, we feel that it is useful to present it.

We analyze the graph test by induction. Order the $|E|$ tests in any order. We want to prove that the probability that the l first tests accept is bounded by

$$2^{-l}(1 + 2^{k^2} d(f))^l + l 2^{-2k^2}.$$

This is worse than the previous bound, but since in general k is a constant and $d(f)$ is arbitrarily small the difference is not as great as it might look at a first glance. We prove this by induction over l and the base case $l = 0$ is clearly true.

Suppose for notational convenience that the l'th test corresponds to the edge $(1, 2)$. Now consider a fixed a value of $\bar{x} = x^{(3)}, x^{(4)} \ldots x^{(k)}$. Let us assume that none of the tests involving only pairs of these fixed inputs reject. Then the event that the first $l - 1$ tests accept can be written as $Q_1(x^{(1)}) \wedge Q_2(x^{(2)})$ for two predicates Q_1 and Q_2. We say that a value of $\bar{x}$ is *low* if

$$Pr_{x^{(1)}, x^{(2)}}[Q_1(x^{(1)}) \wedge Q_2(x^{(2)})] \leq 2^{-2k^2}$$

and otherwise it is called *high*. Let us look at all executions of the first l tests of the protocol. Those corresponding to low $\bar{x}$ contributes at most 2^{-2k^2} to the acceptance probability and thus it is sufficient to prove that executions corresponding to high $\bar{x}$ contributes at most

$$2^{-l}(1 + 2^{k^2} d(f))^l + (l - 1) 2^{-2k^2}. \tag{4}$$

To establish this first note that, by induction, the probability that the first $l - 1$ tests accept and $\bar{x}$ is high is bounded by

$$2^{1-l}(1 + 2^{k^2} d(f))^{(l-1)} + (l - 1) 2^{-2k^2}. \tag{5}$$

This follows since this estimate is true without the condition that $\bar{x}$ is high. Now let us look at the conditional probability that the l'th test accepts. This probability is easily seen to be

$$\frac{1 + E[f(x^{(1)}x^{(2)})f(x^{(1)})f(x^{(2)}) \mid Q_1(x^{(1)}) \wedge Q_2(x^{(2)})]}{2}. \tag{6}$$

Now let f_1 be a function that agrees with f when Q_1 is true and is 0 otherwise and define f_2 similarly by agreement with Q_2. Let q_1 the probability that Q_1 is true on a random input and define q_2 similarly. Then the expected value in (6) equals

$$\frac{E[f(x^{(1)}x^{(2)})f_1(x^{(1)})f_2(x^{(2)})]}{q_1 q_2}$$

This expected value in the numerator can be analyzed using the Fourier transform along the same lines as equations (2) and (3) obtaining the bound

$$\max_\alpha |\hat{f}_\alpha| (\sum_\alpha \hat{f}^2_{1,\alpha})^{1/2} (\sum_\alpha \hat{f}^2_{2,\alpha})^{1/2} \leq (q_1)^{1/2}(q_2)^{1/2} \max_\alpha |\hat{f}_\alpha|,$$

where the last inequality follows from the fact that the L_2-norm of f_i is $(q_i)^{1/2}$.

This implies (using the definition of the "high" case to bound $q_1 q_2$) that

$$(1 + 2^{k^2} d(f))/2$$

is an upper bound of the probability that the l'th test accepts given that the previous tests accepted and that $\bar{x}$ was high. Multiplying this by the probability the the first $l-1$ tests accepts as given by (5) we obtain the desired bound (4) for the contributions of the high $\bar{x}$. The proof of the claimed bound is complete.

3.3 Improved analysis of the graph test

The above bound is clearly optimal as a function of $|E|$ since a random function passes the linearity test defined by G with probability $2^{-|E|}$. We can hope to get a sharper bound as a function of $d(f)$. This is the aim of the present subsection. Towards this end we first give a definition

Definition 3.3 *A graph G has an induced matching of size m if there are $2m$ vertices such that there are exactly m edges supported on these vertices and these form a matching.*

We have

Lemma 3.4 *If the set S has an induced matching of size m then*

$$E\left[\prod_{(i,j)\in S} f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)})\right] \leq d(f)^m.$$

Proof: In the previous proof we reduced the analysis of the graph test to that of one BLR test. Here we reduce it to that of m *independent* BLR tests, in essentially the same way – fixing the values of all sample points except the endpoints of an induced matching of size m. Suppose without loss of generality that $(2i-1, 2i) \in S$ for $i = 1, 2, \ldots m$ and that there are no other edges in S between any pair of these $2m$ vertices. Fix the values of $x_{2m+1}, \ldots x_k$ to constants without decreasing the expected value. The induced function can be written as

$$\prod_{i=1}^{m} f(x^{(2i-1)}x^{(2i)})g_i(x^{(2i-1)})h_i(x^{(2i)}).$$

The different factors are independent and the expected value of each term can be estimated as in Lemma 3.1. ∎

To use Lemma 3.4 we want to find a graph G so that most subgraphs of G have large induced matchings. Note that for this purpose the complete graph K_k is quite bad, since a typical subgraph will only have an induced matching of size about $O(\log k)$. We instead use a remarkable construction from [17]. Let us first state formally what we need.

Definition 3.5 *A bipartite graph G is a union of t matchings of size r if $E = \cup_{i=1}^{t} M_i$ where M_i is an induced matching of size r in G and $M_i \bigcap M_j = \emptyset$ for $i \neq 0$.*

We have the following lemma

Lemma 3.6 *If G is the union of t matchings of size r then the probability that linearity test defined by G accepts is bounded above by*

$$e^{-tr/8} + d(f)^{r/4}$$

Proof: We use the expansion (1). If S contains an induced matching of size $r/4$ then, by Lemma 3.4, the corresponding term is bounded by $d(f)^{r/4}$ and we need to count the number of S for which there is no such matching. For each M_i this means that the S contains at most $r/4 - 1$ edges from M_i. From Theorem A.1.1 of [1] it follows that the probability of this happening, for a fixed i, is bounded by $e^{-r/8}$. The event of this happening is independent for different i and hence the lemma follows. ∎

We have the following elegant result of Rusza and Szemeredi (which for completeness we prove in the appendix)

Theorem 3.7 *[17] There exist a bipartite graph on $2k$ vertices which is the union of $k/3$ matchings each of size $k^{1-o(1)}$.*

Combining Lemma 3.6 and Theorem 3.7 we get the below theorem.

Theorem 3.8 *The probability to accept in the linearity test of the complete graph on k vertices is bounded by*

$$\min(\, 2^{-\binom{k}{2}} + d(f),\ 2^{-k^{2-o(1)}} + d(f)^{k^{1-o(1)}}).$$

We conclude by demonstrating the near-optimality of the last bound.

Theorem 3.9 *The graph test accepts a random function with probability $2^{-|E|}$. Furthermore, for any $d \leq n/2$ there is a function with $d(f) = 2^{-d}$ such that the acceptance probability of the complete graph test is at least*

$$2^{-dk} = d(f)^k.$$

Proof: To verify the first claim fix any choice of $(x^{(i)})_{i=1}^k$. The condition that the test accepts f can be written as $|E|$ homogeneous linear equations in the values of f. The probability that a random function satisfies these equations is at least $2^{-|E|}$.

For the second claim define

$$f(x) = \prod_{i=1}^{d} (x_i \wedge x_{i+d}).$$

It is not difficult to see that for any $\alpha \subseteq [2d]$ we have $|\hat{f}_\alpha| = 2^{-d}$ while for other α we have $\hat{f}_\alpha = 0$. Now if $x_i^{(j)} = 1$ for $1 \leq j \leq k$ and $1 \leq i \leq d$ then f is equal to 1 for all queried points and hence the test accepts. Thus the test accepts with probability at least 2^{-dk}. ∎

3.4 Larger finite fields

In this subsection we extend the results of this section to the groups Z_p for prime $p > 2$. This extension, for a test similar to the basic BLR test has been analyzed earlier [13] and we obtain similar results for this basic case. The extension to graph tests appears to be new but is straightforward.

As before with Z_2, we write Z_p multiplicatively, namely as the group of p'th roots of unity, which we call G. The linear functions on n variables are identified in this multiplicative notation with the characters $x^\alpha = \prod_i x_i^{\alpha_i}$, with $x \in G^n$ and $\alpha \in [p]^n$.

Given access to an oracle for a function $f : G^n \to G$, we want a test whose acceptance probability is related to the distance of f from the closest linear function. As before, we plan to analyze it using the Fourier transform of f, given by the unique expansion of f as a linear combination of characters

$$f(x) = \sum_\alpha \hat{f}_\alpha x^\alpha.$$

The main difference from the case $p = 2$ is that now the coefficients $\hat{f}_\alpha$ may be complex, and the agreement between f and x_α is not as immediate and we need some notation.

Let ζ be a p'th root of unity. All our complex number will be of the form $\sum_{i=0}^{p-1} r_i \zeta^i$ for rational numbers r_i and this is the field extension $Q[\zeta]$. For $1 \leq a \leq p-1$ let σ_a be a homomorphism of $Q[\zeta]$

that sends ζ to ζ^a. Note that if x is a p'th root of unity $\sigma_a(x) = x^a$ and the mapping is extended by linearity. The main reason this is useful for us is the following standard lemma which we state without a proof.

Lemma 3.10 *If x is a p'th root of unity then $\sum_{i=0}^{p-1} x^i = 0$ unless $x = 1$ in which case the sum equals p.*

We next establish.

Lemma 3.11 *Let f be a function mapping into the p'th roots of unity, then the fraction of inputs on which f agrees with the linear function x^α is*

$$\frac{1}{p}(1 + \sum_{a=1}^{p-1} \sigma_a(\hat{f}_\alpha)).$$

Proof: By the previous lemma the probability in question is

$$p^{-n} \sum_x p^{-1} \sum_{a=0}^{p-1} (f(x)x^{-\alpha})^a.$$

The terms corresponding to $a = 0$ contributes $\frac{1}{p}$. For the other terms we switch the order of summation and replace f by its Fourier-expansion giving for a fixed a

$$\sum_x (f(x)x^{-\alpha})^a = \sum_x \sigma_a(f(x)x^{-\alpha}) = \sum_x \sigma_a(\sum_\beta \hat{f}_\beta x^\beta x^{-\alpha}) = \sum_\beta \sigma_a(\hat{f}_\beta) \sum_x \sigma_a(x^{\beta-\alpha}).$$

The inner sum is 0 unless $\beta = \alpha$ in which case it is p^n. The lemma now follows. ∎

It is straightforward to extend the analysis of the case $p = 2$ to obtain the bound $\max_\alpha |\hat{f}_\alpha|$ but this does not immediately imply agreement with a linear function. There are two ways to ways to get this direct correspondence for the basic BLR test. The first is to change the basic BLR test to testing

$$f(x)^a f(y)^b = f(x^a y^b)$$

for random x, y and $a, b \in \{1, 2 \ldots p-1\}$. The second possibility is to stay with the original test and access f in a way to make the random exponents unnecessary. Since the latter alternative works nicely for the graph test this is the path we take.

Definition 3.12 f respects exponentiation *if for any x and any a, $1 \leq a \leq p-1$ we have $f(x)^a = f(x^a)$.*

Any linear function respects exponentiation and we can make sure that an unknown function given to us by a table has this property by the following access rule. From every class of $p-1$ inputs of the type $\{x^a : 1 \leq a \leq p-1\}$, pick (arbitrarily) a unique representative, and access it whenever the value of f on any of these inputs is needed (answering in a way that respects exponentiation). We now have the following lemma.

Lemma 3.13 *Assume that f respects exponentiation, then $\hat{f}_\alpha$ is real for any α.*

Proof: We have for any $a \neq 0$

$$p^n \sigma_a(\hat{f}_\alpha) = \sigma_a(\sum_x f(x)x^\alpha) = \sum_x f(x)^a x^{a\alpha} =$$

$$= \sum_x f(x^a)x^{a\alpha} = \sum_x f(x)x^\alpha = p^n \hat{f}_\alpha$$

and hence this number is real. ∎

It is now natural to define, as before $d(f) = max_\alpha \hat{f}_\alpha$. Now the basic test is identical to the old test. Pick random $x, y \in G^n$ and test that $f(x)f(y) = f(xy)$.

Lemma 3.14 *The acceptance probability of the BLR test extended to Z_p and applied to a function that respects exponentiation is*

$$\frac{1}{p}\left(1 + (p-1)\sum_\alpha \hat{f}_\alpha^3\right) \leq \frac{1}{p}\left(1 + (p-1)max_\alpha|\hat{f}_\alpha|\right).$$

Proof: The probability that the basic test accepts is the expectation of

$$\frac{1}{p}\left(\sum_{a=0}^{p-1}(f(x)f(y)f(xy)^{-1})^a\right).$$

The term corresponding to $a = 0$ is 1 and to estimate any other term we use the fact the f respects exponentiation and replace each term by the Fourier-expansion to obtain

$$f(x^a)f(y^a)f(x^{-a}y^{-a}) = \sum_{\alpha,\beta,\gamma} \hat{f}_\alpha x^{a\alpha}\hat{f}_\beta y^{a\beta}\hat{f}_\gamma x^{-a\gamma}y^{-a\gamma}.$$

The expectation, over a random x and y of a term is 0 unless $\alpha = \beta = \gamma$ and the lemma follows. ∎

Also the graph test is identical to the one described earlier - we pick the k points corresponding to the vertices at random, and on every edge perform the basic test. The graph test accepts iff all basic tests succeed. The analog of the main theorem of the previous subsection is

Theorem 3.15 *The probability to accept in the linearity test over Z_p of the complete graph on k vertices on function f that respects exponentiation is bounded by*

$$\min(\, p^{-\binom{k}{2}} + d(f),\ p^{-k^{2-o(1)}} + d(f)^{k^{1-o(1)}}).$$

Proof: The proof is completely analogous to the case $p = 2$ and let us only give the highlights. The test accepts iff

$$\prod_{(i,j)\in E}\left(\frac{1}{p}\sum_{a=0}^{p-1}(f(x^{(i)}x^{(j)})f(x^{(i)})f(x^{(j)}))^a\right) \tag{7}$$

equals 1 and otherwise this expression equals 0. Expanding the product and manipulating as before this leads that we need to estimate expressions of the form

$$f(x^{(i)}x^{(j)})^a g(x^{(i)})h(x^{(j)})$$

where g and h take values that a p'th roots of unity and a is nonzero. Replacing each term by the Fourier transform and using Plancherel's equality this can be estimated by $\max_\alpha |\hat{f}_\alpha|$ and the first bound follows.

The extension to get the second bound is exactly the same as in the $p = 2$ case.

■

4 Analyzing PCPs

In this section we show that the same idea employed for the analysis of the graph test for linearity testing extends to provide a simple analysis of the graph test used by [16] for PCPs. This is done in subsection 4.2. We then try to obtain an improved bound in the same sense we did in the previous section. We point why it seems impossible, and content ourselves with a minor improvement in the same spirit (given in subsection 4.3). But first we define the PCP and its graph test.

4.1 The PCP and its graph test

Many efficient PCPs, such as the one given in [16] are conveniently analyzed using the formalism of an outer and inner verifier. This could also be done here, but to help the reader not familiar with this formalism we give a more explicit analysis. Using the results of [2] (as explicitly done in [10]) one can prove that there is a constant $c < 1$ such that it is NP-hard to distinguish satisfiable 3-SAT formulas from those where only a fraction c of the clauses can be satisfied by any assignment. This formula furthermore has the property than any clause is of length exactly 3 and any variable appears in exactly 5 clauses.

Given a 3-SAT formula $\varphi = C_1 \wedge C_2 \dots C_m$ which is either satisfiable or where one can only satisfy a fraction c of the clauses one can design a two-prover interactive with verifier V as follows.

The two-prover protocol

1. V chooses a clause C_k uniformly at random and a variable x_j, again uniformly at random, appearing in C_k. V sends k to prover P_1 and j to prover P_2.
2. V receives a value for x_j from P_2 and values for all variables appearing in C_k from P_1. V accepts if the two values for x_j agree and the clause C_k is satisfied.

It is not difficult to see that if a fraction c of the clauses can be satisfied simultaneously then the optimal strategy of P_1 and P_2 convinces V with probability $(2 + c)/3$. Thus it is NP-hard to distinguish the case when this probability is 1 and when it is some constant strictly smaller than 1.

To make the gap larger one runs this protocol u times in parallel and in this protocol u random clauses are sent to P_1, u variables (one from each clause) are sent to P_2. The verifier accepts in this protocol if the assignments returned by the provers satisfy all the picked clauses and are consistent. By the fundamental result by Raz [15], the probability that the verifier accepts when only a constant fraction $c < 1$ of the clauses are satisfied is bounded by d_c^u for some absolute constant $d_c < 1$.

This two-prover protocol is now turned into a PCP by, for each question to either P_1 or P_2 writing down the answer in coded form. As many other papers we use the marvelous *long code* introduced by Bellare et al [7].

Definition 4.1 *The* long code *of an assignment $x \in \{-1,1\}^t$ is obtained by for each function $f : \{-1,1\}^t \mapsto \{-1,1\}$ writing down the value $f(x)$.*

Thus the long code of a string of length t is a string of length 2^{2^t}. Note that even though a prover is supposed to write down a long code for an assignment we have no way to guarantee that a cheating prover does not write down a string which is not the correct long code of anything. We analyze such arbitrary tables by the Fourier-expansion and in the current situation this is given by

$$\sum_{\alpha \subseteq \{-1,1\}^t} \hat{A}_\alpha \chi_\alpha(f),$$

where

$$\chi_\alpha(f) = \prod_{x \in \alpha} f(x).$$

If A is indeed a correct long code of a string $x^{(0)}$ then $\hat{A}_{\{x^{(0)}\}} = 1$ while all the other Fourier coefficients are 0.

We can, to a limited extent, put some restrictions on the tables produced by the prover.

Definition 4.2 *A table A is folded over true if $A(f) = -A(-f)$ for any f.*

Definition 4.3 *A table A is conditioned upon h if $A(f) = A(f \wedge h)$ for any f.*

To make sure that an arbitrary long code is folded we access the table as follows. For each pair $(f, -f)$ we choose (in some arbitrary but fixed way) one representative. If f is chosen, then if the value of the table is required at f it is accessed the normal way by reading $A(f)$. If the value at $-f$ is required then also in this case $A(f)$ is read but the result is negated. If $-f$ is chosen from the pair the procedures are reversed.

Similarly we can make sure that a given table is properly conditioned by always reading $A(f \wedge h)$ when the value for f is needed. Folding over true and conditioning can be done at the same time.

Let us now give the consequences of folding and conditioning for the Fourier coefficients. The proofs are easy and left to the reader but they can also be found in [12].

Lemma 4.4 *If A is folded over true and $\hat{A}_\alpha \neq 0$ then $|\alpha|$ is odd and in particular α is non-empty.*

Lemma 4.5 *If A is conditioned upon h and $\hat{A}_\alpha \neq 0$ then for every $x \in \alpha$, $h(x)$ is true.*

Concluding, the written proof used in our PCP is the following. For every subset U of size u we have the Boolean string of length 2^{2^u}. Also, for every subset W of size $w \leq 3u$ we have a Boolean string of length 2^{2^w}. In a correct proof for a satisfiable formula all these strings are long codes of the restriction of the same satisfying assignment to the relevant subsets.

The test of this written proof is now performed as follows.

The PCP graph test

1. The verifier V chooses u variables, each picked uniformly and independently from the others. Let the chosen set be U.

2. V chooses k random functions f_i, $i = 1, 2 \ldots k$ on U. These are chosen randomly and independently. Let A be the string (hopefully long code) corresponding to the set U in the written proof.

3. Repeat the following steps independently for $j = 1, 2, \ldots k$. For each variable in U choose a random clause containing it. Let h_j be the conjunction of the chosen clauses and let W_j be the set of variables appearing in the chosen clauses. Choose g_j to be a random function with uniform probability on W_j. Let B_j be the string (hopefully long code) corresponding to the set W_j in the written proof, folded over true and conditioned upon h_j. Note that U is a subset of W_j for all j.

4. For $1 \leq i, j \leq k$ choose a function μ_{ij} on W_j which, independently at each point takes the value 1 with probability $1 - \epsilon$ and the value -1 with probability ϵ. Set $g_{ij} = g_j f_i \mu_{ij}$, i.e. for each $y \in \{-1, 1\}^{W_j}$ set $g_{ij}(y) = g_j(y) f_i(\pi(y)) \mu_{ij}(y)$ where π is the projection from W_j to U. Test whether

$$B_j(g_{ij}) = B_j(g_j) A(f_i).$$

5. If all tests accept, V accepts and otherwise it rejects.

The test above is performed for all possible pairs (i, j). Note however that unlike the linearity test we have questions of two different types (as the f_i and g_j live on different domains) and thus G must in this case be a bipartite graph.

4.2 Simple analysis of the PCP graph test

It is easy to see that the completeness of the test is at least $(1 - \epsilon)^{|E|}$ and we need to analyze the soundness.

Similarly to the linearity test the verifier accepts if

$$\prod_{(i,j)\in E} \frac{1 + A(f_i)B_j(g_j)B_j(g_{ij})}{2}$$

equals one. We expand this product getting

$$2^{-|E|} \sum_{S\subseteq E} \prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j(g_{ij}). \tag{8}$$

The main lemma of this section shows that for any S, a positive expectation of the above expression yields a strategy for the two prover game of related success probability. As we know the later must be small, we are able to upper bound the soundness.

Lemma 4.6 *Suppose S is nonempty and*

$$E\left[\prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j(g_{ij})\right] = \delta$$

where the expectation is taken over all coin tosses of the PCP verifier. Then there is a strategy for the two provers in the two-prover game that convinces the its verifier with probability at least $4\epsilon\delta^2$.

Proof: Suppose without loss of generality that $(1,1) \in S$. Now for a fixed U fix values of $(W_j, g_j, f_i, \mu_{ij})$, $i,j \geq 2$ and $\mu_{1j}, j > 1$ $\mu_{i1}, i > 1$ which does not decrease the expectation (taken over f_1, W_1, g_1, μ_{11}) of the considered expression. This product can now be written as

$$A'(f_1)B_1(g_{11})C(g_1) \tag{9}$$

where A' and C are Boolean functions and B_1 is the original long code on W_1. The function A' is a function that depends on the constants chosen above. However note that these constants only depend on the value of U and hence A' is a fixed function on U. In particular A' does not depend on W_1 (or g_1 or μ_{11}). The function C is a Boolean function on W_1 which is defined by a product that contains terms of the form $B_1(g_{i1})$. It is difficult to control but we only need that it is a Boolean function. For reasons of typography let us for the remainder of this proof rename B_1 to B and W_1 to W.

Let us fix U and W for the moment, and substitute the Fourier expansion of each function in (9), taking the expected values over f_1, g_1 and μ_{11}. We get

$$E_{f_1,g_1,\mu_{11}}\left[\sum_{\alpha,\beta_1,\beta_2} \hat{A}'_\alpha \hat{B}_{\beta_1}\hat{C}_{\beta_2}\chi_\alpha(f_1)\chi_{\beta_1}(f_1 g_1 \mu_{11})\chi_{\beta_2}(g_1)\right] =$$
$$\sum_{\alpha,\beta_1,\beta_2} \hat{A}'_\alpha \hat{B}_{\beta_1}\hat{C}_{\beta_2} E_{f_1,g_1,\mu_{11}}\left[\chi_\alpha(f_1)\chi_{\beta_1}(f_1 g_1 \mu_{11})\chi_{\beta_2}(g_1)\right].$$

Now, unless $\beta_1 = \beta_2 = \beta$ the inner expected value is 0. Taking the expected value over f_1 we see that unless $\pi_2(\beta) = \alpha$ the value is also 0. Here π_2 is the mod 2 projection i.e. $\pi_2(\beta)$ contains x iff there is an odd number of $y \in \beta$ such that $\pi(y) = x$. Finally $E[\chi_\beta(\mu_{11})] = (1-2\epsilon)^{|\beta|}$ and we obtain the overall result

$$\sum_\beta \hat{A}'_{\pi_2(\beta)} \hat{B}_\beta \hat{C}_\beta (1-2\epsilon)^{|\beta|}.$$

By Cauchy-Schwartz inequality this is bounded by

$$\left(\sum_\beta \hat{C}_\beta^2\right)^{1/2} \left(\sum_\beta (\hat{A}'_{\pi_2(\beta)} \hat{B}_\beta (1-2\epsilon)^{|\beta|})^2\right)^{1/2} = \left(\sum_\beta \hat{A}'^2_{\pi_2(\beta)} \hat{B}_\beta^2 (1-2\epsilon)^{2|\beta|}\right)^{1/2}.$$

So, under the hypothesis of the lemma, taking expectations over U and W we get

$$E_{U,W}\left[\left(\sum_\beta \hat{A}'^2_{\pi_2(\beta)} \hat{B}_\beta^2 (1-2\epsilon)^{2|\beta|}\right)^{1/2}\right] \geq \delta.$$

Since $E[X^2] \geq E[X]^2$ for any random variable X we obtain

$$E_{U,W}\left[\sum_\beta \hat{A}'^2_{\pi_2(\beta)} \hat{B}_\beta^2 (1-2\epsilon)^{2|\beta|}\right] \geq \delta^2. \tag{10}$$

Now consider the following strategy for the provers in the two-prover game. Prover P_1, on receiving W, picks a random β with probability $\hat{B}_\beta^2$ and then a random $y \in \beta$. Prover P_2, on receiving U, picks a random α with probability $\hat{A}'^2_\alpha$ and then returns a random x in α. By Lemma 4.5, the answer returned by P_1 always satisfies the chosen clauses. Also note that by Lemma 4.4, β is of odd size and hence neither it nor $\pi_2(\beta)$ is empty. Since A' is not folded over true α might be empty and in such a case P_2 sends some default string. The probability of convincing V in the two prover game is now exactly the probability that $\alpha = \pi(\beta)$, which is at least

$$\sum_\beta \hat{B}_\beta^2 \hat{A}'^2_{\pi_2(\beta)} |\beta|^{-1}. \tag{11}$$

We have the inequality $x^{-1} \geq e^{-x}$ valid for any $x > 0$ and applying this we see that

$$(4\epsilon|\beta|)^{-1} \geq e^{-4\epsilon|\beta|} \geq (1-2\epsilon)^{-2|\beta|}$$

and thus we see that (11) is at least 4ϵ times the value of (10) and hence has expected value at least $4\epsilon\delta^2$. ∎

Since the soundness of the two prover protocol is d_c^u, Lemma 4.6 is sufficient to get the following result (which is already a bit stronger than what is stated by Samorodnitsky and Trevisan [16]).

Theorem 4.7 *The soundness of the above described PCP with G the complete bipartite graph is at most*

$$2^{-k^2} + \left(\frac{d_c^u}{4\epsilon}\right)^{1/2}$$

4.3 Improved analysis

In the linearity testing we succeeded in improving the obtained bound by raising the second term of the upper bound to a high power. We explain where and why this idea fails here, and give the best bound it implies, sightly improving the theorem above (essentially squaring the second term).

We first note that as long as U remains fixed, the same improvement obtained in the case of linearity testing *is* possible.

Lemma 4.8 *Fix the value U, suppose S has and induced matching of size m and*

$$E\left[\prod_{(i,j)\in S} B_j(g_{ij})B_j(g_j)A(f_i)\right] = \delta_U,$$

where the expected value is taken over all other random choices of the verifier. Then there is a strategy for the two provers in the two-prover game that given that U is chosen, convinces the verifier with probability at least $4\epsilon\delta_U^{2/m}$.

Proof: We proceed as in the proof of Lemma 3.4. Suppose $(i,i) \in S$ for $1 \le i \le m$ and that there are no other edges on these vertices. We fix values of $(W_j, g_j, f_i, \mu_{ij}), i, j > m$ and μ_{ij} with $i \le m$ and $j > m$ or $i > m$ and $j \le m$ to values that do not decrease the expected value. Reasoning as in the proof of Lemma 4.6 we get, under the hypothesis of the lemma that there are Boolean functions A_i' and C_i such that

$$E\left[\prod_{i=1}^{m} A_i'(f_i)B_i(g_{ii})C_i(g_i)\right] \ge \delta_U$$

where this expectations is over the surviving random variables excluding U. Since the factors are independent there is one i such that

$$E\left[A_i'(f_i)B_i(g_{ii})C_i(g_i)\right] \ge \delta_U^{1/m}.$$

The rest of the proof is now essentially identical to the corresponding part of Lemma 4.6. ∎

Unfortunately the corresponding strengthening does not carry over to the full analysis of the PCP. The problem being that from $E_U[\delta_U] = \delta$ the best lower bound for $m \ge 2$ that can be obtained for $E_U[\delta_U^{2/m}]$ is δ. Thus it is only useful to have $m = 2$ giving a moderate improvement over the results of [16].

Since we know that the soundness of the two-prover game is d_c^u we get that terms corresponding to an S which contains an induced matching of size m for $m = 1$ and $m = 2$ can be at most

$$\left(\frac{d_c^u}{4\epsilon}\right)^{m/2}$$

in absolute value. The empty graph is the only graph that does not contain a matching of size 1 and we need to estimate the number of graphs that do not contain a matching of size 2. We have the following lemma.

Lemma 4.9 *The number of bipartite graphs with k vertices in each part that do not contain a matching of size 2 is bounded by $(k!)^2\binom{2k-1}{k}$.*

Proof: Suppose the two parts of the vertices are V_1 and V_2. For $i = 1, 2, \ldots k$ let S_i be the subset of V_2 connected to i'th vertex of V_1. If there is no matching of size 2 then for any pair (i, j) we either have $S_i \subseteq S_j$ or $S_j \subseteq S_i$. We conclude that if there is no matching of size 2 then there is a permutation π such that

$$S_{\pi(1)} \subseteq S_{\pi(2)} \subseteq \ldots \subseteq S_{\pi(k)}.$$

Such a chain is uniquely described by the order in which elements are added and how many elements are added at each point in time. The order is given by a permutation σ and the number of ways to partition k elements into k pieces is, by a standard argument at most $\binom{2k-1}{k}$. Since there are at most $k!$ choices for each of the permutations π and σ, the lemma follows. ∎

Note that we do not get a 1-1 correspondence since often neither π nor σ is uniquely determined. The overestimate is not too bad since when $|S_{\pi(i)}| = i$, both π and σ are uniquely determined and hence the number of such graphs is at least $(k!)^2$ and thus the lemma is not too far from the truth.

Using the expansion (8), the bound 1 when S is empty, the bound $\left(\frac{d_c^u}{4\epsilon}\right)^{1/2}$ when the maximal size of an induced matching is 1 and $\frac{d_c^u}{4\epsilon}$ in the remaining cases, we get a final estimate for the acceptance probability. The result is only moderately stronger than the corresponding theorem of Samorodnitsky and Trevisan [16], and the main contribution is that our proof is simpler.

Theorem 4.10 *The soundness of the above described PCP with G the complete graph is at most*

$$2^{-k^2} + 2^{-k^2}(k!)^2 2^{2k}\left(\frac{d_c^u}{4\epsilon}\right)^{1/2} + \left(\frac{d_c^u}{4\epsilon}\right).$$

4.4 PCPs in larger fields

The results by Samorodnitsky and Trevisan have been extended to the case where each symbol is in Z_p by Engebretsen [9]. The current analysis also applies to that case. Let us briefly recall the setup and state the result.

In this case each symbol in the proof is an element from Z_p which we again write multiplicatively as the p'th roots of unity. We have the same underlying 2-prover protocol but in the PCP we change the ordinary long code to long-p-code in which a table is indexed by all functions f mapping into Z_p. In a correct long-p-code for x the value at f should be $f(x)$. For Boolean h define $f \wedge h(x)$ as $f(x)$ if $h(x)$ is true and as 1 if $h(x)$ is false. Long-p-codes can be folded and conditioned.

Definition 4.11 *A table A is p-folded if $A(zf) = zA(f)$ for any f and any $z \in Z_p$.*

Definition 4.12 *A table A is conditioned upon h if $A(f) = A(f \wedge h)$ for any f.*

The extension of Fourier transforms to the case of Z_p has already been described in Section 3.4 and we only state the consequences of folding and conditioning. Note that in the present case a linear function is written f^α where α is a function mapping into $0, 1, 2 \ldots p-1$ and

$$f^\alpha = \prod_x f(x)^{\alpha(x)}.$$

Proofs of the below two lemmas can be found in [12].

Lemma 4.13 *If A is a p-folded and $\hat{A}_\alpha \neq 0$ then $\sum_x \alpha(x) \equiv 1 \bmod p$ and in particular α is non-zero.*

Lemma 4.14 *If A is conditioned upon h and $\hat{A}_\alpha \neq 0$ then for every x with $\alpha(x) \neq 0$, $h(x)$ is true.*

We could have, as in the linearity test, asked for the tables to respect exponentiation, but this is not needed and hence we do not.

The definition of the graph test is verbally the same except that the error functions μ_{ij} which takes the value 1 with probability $1 - \epsilon$ and with probability ϵ a random value in Z_p.

The analysis is adding the same small modifications that we needed in the linearity testing in larger fields to the analysis of the PCP for $p = 2$. We start with an expansion similar to (7) and expand the product. We analyze each individual term using the Fourier expansion (as is done in the simple case of one test in [12]). The successful strategies of the provers are given by the Fourier coefficients of the tables and we simply state the theorem (which was originally proved in [9]).

Theorem 4.15 *[9] For any k and any $\epsilon > 0$ any language in NP admits a polynomial size PCP that reads $2k + k^2$ symbols from Z_p, has completeness $1 - \epsilon$ and soundness $p^{-k^2}(1 + \epsilon)$.*

5 Conclusion

We have given a very simple analysis of the test given by Samorodnitsky and Trevisan for linearity testing and for PCPs with optimal query complexity. Our hope is that this will help in analyzing more complicated tests that might be useful to obtain stronger results.

The second author also wishes to convey the following intuition (not fully shared by the first author), relating our analysis of the graph test to the analysis of pseudorandom generators. Indeed, the graph test generates from a small random sample many (dependent) tests, which behave as though they were independent, and therefore can be viewed as some kind of pseudorandom generator.

Those familiar with the set-up of the NW-generator [14] will recognize in Section 3 a more detailed correspondence.

- The seed of the generator is the k query points of the graph test.
- The output of the generator are the results of individual linearity test on one edge (pair of seed points) of the graph. Moreover, the intersection of any two such sets is "small", namely one test point (this is a trivial "design").
- The output of the generator has to "fool" (i.e. look uniform to) all linear tests (as expressed by equation (2)).
- This is proved by fixing all but two of the seed points, and reducing the "pseudo-randomness" of the output to the "hardness" of one edge test, conveniently provided by the [4] linearity test.

Needless to say, some of the complications that arise in following precisely the NW analysis in this context are confusing and unnecessary in this simple context, and indeed the resulting analysis we described here need not refer to it at all. But perhaps there are other problems where this analogy and viewpoint may help, as it was here.

Acknowledgment We are grateful to Madhu Sudan for very fruitful discussions. We also thank Roy Meshulam and Benny Sudakov for helpful discussions. We are most grateful to Subhash Khot for pointing out a flaw in an earlier version of the paper.

References

[1] N. Alon and J. Spencer, *The probabilistic Method*, 2nd edition, 2000, Wiley, New York.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[3] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[4] Y. Aumann, J. Håstad, M. Rabin, and M. Sudan Linear consistency testing. *Journal of Computer and System Sciences,* Vol 62, 2001, pp 589-607.

[5] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42 (6):1781–1796, November 1996.

[6] F. Behrend, On sequences of integers containing no arithmetic progression. *Časopis Pěst. Mat.*, 67: 235–239, 1938.

[7] M. Bellare, O. Goldreich and M. Sudan. Free bits, PCP's and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804-915, 1998.

[8] M. Blum, M. Luby and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47: 549–595, 1993.

[9] L. Engebretsen Lower bounds for non-Boolean constrain satisfaction, *ECCC* TR00-042.

[10] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45: 634–652, 1998.

[11] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[12] J. Håstad. Some optimal inapproximability results. *Journal of ACM*, Vol 48, 2001, pp 798-859.

[13] M. Kiwi, Probabilistically Checkable Proofs and the testing of Hadamard-like codes. Ph. D. Thesis, MIT.

[14] N. Nisan, A. Wigderson, Hardness vs. Randomness. *Journal of Computer Systems and Sciences*, 49(2): 149–167, 1994.

[15] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[16] A. Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. *Proc. of 32nd STOC*, 191–199, 2000.

[17] Ruzsa and E. Szemeredi, Triple systems with no six points carrying three triangles. Combinatorics (Proc. Fifth Hungarian Colloq.), Keszthely, 1976.

A The construction of the graphs

We now explain the construction, due to Rusza and Szemeredi [17], of dense graphs whose edge set can be partitioned to a linear number of *induced* matchings of nearly linear size.

Let $[n]$ denote the set of the first n integers. We will construct bipartite graphs on two sets of vertices labeled by $[3n]$ as follows. Fix a subset $A \subseteq [n]$. For any element $i \in [n]$, we let M_i to be the matching consisting of all edges $\{(a+i, a+2i) : a \in A\}$ (all these integers are in $[3n]$. Now define $G(A)$ to be the union of these M_i over all $i \in [n]$.

Theorem A.1 *Assume that A has no three-term arithmetic progression. Then all M_i are induced in $G(A)$.*

Proof: Assume to the contrary that one of the matching is not induced. This means that for some $i, j \in [n]$ and $a, b \in A$ we have $(a+i, a+2i), (b+i, b+2i) \in M_i$ but also $(a+i, b+2i) \in M_j$. This means that for some $c \in A$ we have the system of equations

$$a + i = c + j \qquad b + 2i = c + 2j$$

from which we conclude that $2a = b + c$, a contradiction. ∎

It remains to give a large set $A \subset [n]$ without a three-term arithmetic progression. The best known construction is by Behrend [6] which we describe below. The proof is not difficult given the construction, and we only provide a sketch.

Pick integers d, s, and let t be the smallest integer so that $(2d+1)^t \geq n$. Let $A_{d,s}$ be the set of all integers of the form $\sum_{i=0}^{t} a_i (2d+1)^i$ with the integers a_i satisfying

1. For all i, $0 \leq a_i \leq d$
2. $\sum_{i=0}^{t} a_i^2 = s$

Theorem A.2 *1. For every d, s the set $A_{d,s}$ has no three-term arithmetic progression.*

2. For $d = \lceil 2^{\sqrt{\log n}} \rceil$ and some choice of s, $|A_{d,s}| \geq n/2^{O(\sqrt{\log n})} = n^{1-o(1)}$

Proof: (Sketch)
For (1), note that the condition $0 \leq a_i \leq d$ implies that any three-term arithmetic progression must give three colinear vectors $(a_i)_{i=0}^{t}$ and there can be no three colinear vectors of the same L_2 norm. For (2), take the value of s which maximizes the size of $A_{d,s}$. The right hand side is simply the average size. ∎