A PCP Characterization of NP with Optimal Amortized Query Complexity

Alex Samorodnitsky* Luca Trevisan†

October 20, 1999

Abstract

For every $\varepsilon > 0$ and integers k and q (with $k \leq q \leq k + k^2/4$), we present a PCP characterization of NP where the verifier queries q bits (of which only k are free bits), accepts a correct proof with probability $\geq 1 - \varepsilon$ and accepts a "proof" of a wrong statement with probability $\leq 2^{-(q-k)}$.

In particular, for every $\delta > 0$ we have a PCP characterization of NP where the verifier has, simultaneously, $1 + \delta$ amortized query complexity and δ amortized free bit complexity. Both results are tight, unless $P = NP$.

The optimal amortized query complexity of our verifier implies essentially tight non-approximability results for constraint satisfaction problems. Specifically, we can show that k-CSP, the problem of finding an assignment satisfying the maximum number of given constraints (where each constraint involves at most k variables) is NP-hard to approximate to within a factor $2^{-k+O(\sqrt{k})}$. The problem can be approximated to within a factor 2^{-k+1}, and was known to be NP-hard to approximate to within a factor about $2^{-2k/3}$. We can also prove some new separation results between different PCP model.

A PCP characterization of NP with optimal amortized free bit complexity implies that for every $\delta > 0$ it is hard to approximate the maximum clique problem to within a $n^{1-\delta}$ factor. Such a characterization had already been proved by Håstad [Hås96], in a celebrated recent breakthrough. Our construction gives an alternative, simpler, proof of this result.

Our techniques also give a tight analysis of *linearity testing* algorithms with low amortized query complexity. As in the case of PCP, we show that it is possible to have a linearity testing algorithm that makes q queries and has error bounded from above by $2^{-q+O(\sqrt{q})}$. We also prove a lower bound showing that, for a certain, fairly general, class of testing algorithms, our analysis is tight even in the lower order term. That is, we show that the error of a q-query testing algorithm in this class has to be at least $2^{-q+\Omega(\sqrt{q})}$.

*`asamor@ias.edu`. Institute for Advanced Study and DIMACS

†`luca@cs.columbia.edu`. Columbia University. Work partly done at DIMACS.

1 Introduction

The PCP Theorem [AS98, ALM$^+$98] gives a characterization of NP that is both useful and fascinating. It is fascinating for the surprisingly strong point that it makes about the power of randomness in computations, and it is useful for its applications to the study of the approximability of optimization problems, by a connection that was initiated by [FGL$^+$91] and then stretched to an amazing extent.

In the last seven years, research on probabilistically checkable proofs has focused on achieving *quantitative* strengthening of the PCP Theorem, involving increasingly efficient verifiers [BGLR93, FK94, BS94, BGS98, Hås96, Hås97, Tre98, ST98] (and also, in a somewhat different direction, [RS97, AS97, DFK$^+$99]). Such improvements of the PCP Theorem have been mostly driven by the search for improved non-approximability results, and some of the efficiency measures used for verifiers have been tailored to the goal of proving non-approximability results for specific problems (see [Bel96] for a survey on efficiency parameters for PCP and their relation to non-approximability results). In addition, the PCP model defines complexity classes of independent interest, and, at least for the most natural efficiency parameters, it is a natural question to ask what is the strongest version of the PCP Theorem, whether or not it yields improved non-approximability results.

The original version of the PCP Theorem states that proofs for any NP language can be encoded in such a way that their validity can be verified by only reading a constant number of bits, and with an error probability[1] that is upper bounded by a constant. In particular, the verifier of [ALM$^+$98] had an error probability at most $1/2$, and the number of queries was at most $10,000$ (a folklore estimation not explicitly made in the paper). The general goal is to construct verifiers having small error and small query complexity: while it is easy to trade off one parameter for the other, it is hard to optimize them simultaneously (and there are inherent limitations). In particular, already from [ALM$^+$98] one can show that there is a PCP characterization of NP where the verifier makes 3 queries, and the error probability is bounded away from 1 (this is implicit in the [ALM$^+$98] reduction of PCP computations to 3SAT); on the other hand, it is possible to get verifiers having an arbitrarily small error $s > 0$, and having query complexity $O(\log(1/s))$. Furthermore, one can show that, in a PCP characterization of NP, a verifier having error s must have query complexity at least $\log(1/s)$ (unless P=NP) [BGS98, Tre96]. So the best we can hope for (in terms of trade-off between error probability and number of queries) is to construct verifiers having error s and query complexity $\bar{q}\log 1/s$ where $\bar{q} > 1$ is some constant that we would like to be as small as possible. The parameter $\bar{q}$ (the ratio between number of queries and logarithm of inverse error probability) is called the *amortized query complexity* of the verifier.

Another important PCP parameter is the *free bit complexity* of the verifier. We say that a verifier uses f free bits if there is a subset of f queries such that for any possible outcome to these queries there is only one possible answer to the other queries that would make the verifier accept.[2] The amortized free bit complexity of a verifier that uses f free bits and that has error s is $f/\log(1/s)$. The amortized free bit parameter is important for its application to the approximability of the Max Clique problem. In particular, if there is a PCP characterization of NP where the verifier has amortized free bit complexity $\bar{f}$, then Clique is hard to approximate to within a factor roughly $n^{1/(1-\bar{f})}$.

[1] A PCP verifier can make an error in two possible ways: it can reject a valid proof, or it can accept a "proof" of an invalid statement. In this paper we will only consider PCP verifiers that accept a valid proof with probability at least $1-\varepsilon$, where $\varepsilon > 0$ is a constant that can be made arbitrarily small independently of the other parameters of interest. Therefore we will use the term "error probability" as a synonym of "soundness," i.e. (an upper bound to) the probability of accepting a proof of an incorrect statement.

[2] A more general definition is given in [BGS98].

1.1 Previous Work

Previous results showed that there is a PCP characterization of NP with amortized query complexity $1.5+\delta$ for any $\delta > 0$ [ST98]. In terms of free bit complexity, Håstad showed that for every $\delta > 0$ there is a construction with δ amortized free bits. The construction of [Hås96] uses an unbounded number of amortized query bits. On the other hand, it was known that a PCP verifier that makes q queries and has an error probability less than 2^{1-q} can only recognize languages in P [BGS98, Tre96]. So, unless $P = NP$, PCP characterizations of NP must employ verifiers having amortized query complexity bigger than 1.

The main open question left from the work of [ST98] is a construction with $1+\varepsilon$ amortized query bits. As will be discussed later, this has implications for constraint satisfaction problems and for the relation between different PCP models. A somewhat different question was to get a simpler proof of the result of Håstad [Hås96], whose analysis was very involved.

1.2 Linearity Testing

In [Tre98, ST98], PCP constructions are achieved by first analysing the related and somewhat simpler *linearity testing* problem, and then adapting linearity testing algorithms, and their analysis, to the PCP setting. In this paper we follow a similar route.

Recall that a function $f : \{0,1\}^n \to \{0,1\}$ is *linear* if for every $x, y \in \{0,1\}^n$ it holds $f(x) \oplus f(y) = f(x \oplus y)$. Equivalently, f is linear if there exists a subset $\alpha \subseteq \{1,\ldots,n\}$ such that $f(x) = \bigoplus_{i\in\alpha} x_i$. In the linearity testing problem we are given oracle access to a function $f : \{0,1\}^n \to \{0,1\}$ and we want to determine whether f is linear, or if f is very far from every linear function, where the distance $\mathbf{Dist}(f,g)$ between two functions is the fraction of points where they disagree (for a function f and a family $\mathcal{F}$ we also use the notation $\mathbf{Dist}(f,\mathcal{F}) = \min_{g\in\mathcal{F}} \mathbf{Dist}(f,g)$.) Let us call LIN the set of linear functions. We want a randomized test that always accepts linear functions and that accepts with very small probability functions that are far from the set of linear functions. More precisely, we are interested in having a small acceptance probability for the functions that have a low correlation with being linear, that is functions f such that $\mathbf{Dist}(f,\mathsf{LIN}) \approx 1/2$. We will say that a testing algorithm has "error probability at most s," if for every function f such that $1/2-\varepsilon \le \mathbf{Dist}(f,\mathsf{LIN}) \le 1/2+\varepsilon$ we have that the test accepts f with probability at most $s+\varepsilon$. The amortized query complexity of a test having error s that makes q queries is defined as usual as $q/\log(1/s)$. This definition may look somewhat contrived, but testing algorithm having a certain error probability under this definition have a strong analogy to PCP constructions with the same soundness.

Trevisan [Tre98] described a family of linearity testing algorithms and was able to analyse some of them. The best ones had amortized query complexity about 1.5. Sudan and Trevisan [ST98] extended this analysis to the PCP setting.

1.3 Our Results

We denote by $\mathrm{naPCP}_{c,s}[\log, q]$ the class of problems that admit a proof system where the verifier runs in polynomial time, uses a logarithmic number of random bits, reads non-adaptively at most q bits of the proof and satisfies the following properties: a valid proof of a correct statement is accepted with probability at least c and any "proof" of an incorrect statement is accepted with probability at most s. In this paper we show the following.

Theorem 1 (Main) *For every $\varepsilon > 0$ and integers k, q, with $q \le k + k^2/4$, $NP = naPCP_{1-\varepsilon,s}[\log, q]$ where $s = 2^{k-q}$. In addition, the free bit complexity of the verifier is k.*

In other terms, we give a construction that makes q queries, uses about $2\sqrt{q}$ free bits, and has soundness about $2^{2\sqrt{q}-q}$. The amortized query complexity of our verifier is thus about $1+2/\sqrt{q}$, and its amortized free bit complexity is $2/\sqrt{q}$. While this does not perfectly match known impossibility results (that only rule out a combination of q queries and soundness 2^{1-q}) we do prove a tight $1+\delta$ amortized query complexity, and we derive an alternative proof of the δ amortized free bit complexity result by Håstad. Our proof seems to be considerably simpler (at a both a technical and a conceptual level) than Håstad's original one, although it uses related machinery.

We can also prove a very strong result in terms of linearity testing with low amortized query complexity.

Theorem 2 *For every k, q with $q \leq k + \binom{k}{2}$ there is a testing algorithm A that makes q accesses to a function f given as an oracle, and if f is linear, then A accepts with probability* 1. *If $1/2 - \varepsilon \leq$ $\mathbf{Dist}(f, \mathsf{LIN}) \leq 1/2 + \varepsilon$, then A accepts with probability at most $2^{k-q} + \varepsilon$.*

Since our testing algorithm makes q queries and has a probability at most $2^{\sqrt{2q}-q}$ of incorrectly accepting a function f that is from linear, its amortized query complexity tends to 1 for large q. Our result is proved using testing algorithms described in [Tre98], whose analysis was not possible using the techniques of [Tre98].

The testing algorithms introduced in [Tre98] are associated to undirected graphs. To each graph $G = (V, E)$ there is an associated testing algorithm that makes $q = |V| + |E|$ queries and invokes $|E|$ instances of an atomic linearity test due to [BLR90]. Each atomic test has error probability 1/2, and if they were invoked independently, the error probability of the composed test would be $2^{-|E|}$. Indeed, the invocations of the atomic tests are not independent, however we prove that when the tests are invoked on functions very far from being linear, the atomic tests have low correlation, and the error probability of the composed test is $2^{-|E|}$. For a complete graph, the error probabilty is $2^{\Theta(\sqrt{q})-q}$. There is a natural way of generalizing the tests of [Tre98] so that each test is associated to a hypergraph $H = (V, E)$, and the test associated to H makes $q = |V| + |E|$ queries and makes $|E|$ invocations to the [BLR90] atomic test. In analogy to what we prove in this paper, one would be tempted to conjecture that such a test has error probability $2^{-|E|}$ that, as a function of the number of queries, is as low as $2^{O(q^{1/d})-q}$ for a d-uniform hypergraph, and even $2^{O(\log q)-q}$ for the complete hypergraph.

Instead, we are able to show that any hypergraph-based testing algorithm that makes q queries must have an error probability at least $2^{\Omega(\sqrt{q})-q}$. It remains an open question whether there are linearity testing algorithms (of different kind) having error probability $2^{o(\sqrt{q})-q}$ where q is the number of queries. The only general lower bound is that a testing algorithm that makes q queries must have error probability at least $2^{\Omega(\log q)-q}$.

1.4 Applications

We refer to [Tre98] and [ST98] for discussions on the application of PCP with low amortized query complexity.

For an integer $k \geq 2$, the Max kCSP problem is the variation of Max kSAT where we are given a set of boolean expressions, each one involving at most k variables, and we want to find an assignment of values to the variables such that the maximum number of expressions is satisfied. This family of problems, as well as some special cases, is very well studied [KMSV99, Cre95, Tre96, Tre97, TSSW96, KSW97, Zwi98a, Zwi98b]. Max kCSP is NP-hard to approximate to within a factor of roughly $2^{-2k/3}$ [ST98]. The best known algorithm has an approximation ratio $2^{-(k-1)}$ [Tre96] (note that a random solution is 2^{-k}-approximate.) In general, if $NP = \text{naPCP}_{c,s}[\log, q]$, then Max qCSP is NP-hard to approximate to within a factor larger than s/c (see e.g. [Tre96]). The following, almost tight, non-approximability result is then an immediate consequence of Theorem 1.

Corollary 1 *For every k, the Max kCSP problem is NP-hard to approximate to within $2^{-k+O(\sqrt{k})}$.*

To illustrate another consequence of our main result, consider the PCP model where each entry of the proof is not a bit, but rather an element of the alphabet $\{0,1\}^l$. If a verifier makes k queries in this model, it can be simulated by an ordinary PCP verifier that makes kl queries. Intuitively, this reduction should not be tight, since making kl unrestricted queries seems to give more power than being constrained to query k blocks of l-bits data. A negative result on the power of this PCP classes was (in a slightly different language) given in [STX98]. Specifically, it was shown that a verifier that reads k entries in a proof having entries of size l cannot have an error less than $2^{-l(k-1)}$. This means that if we define the query complexity as lk, and the amortized query complexity similarly, a verifier that reads k block must have amortized query complexity at least $k/(k-1)$. So, another consequence of Theorem 1 is that for every fixed k, and for sufficiently large l, there is a separation between the power of a verifier that reads k entries of size l is less than the power of a verifier that reads lk bits without restrictions. (Only a separation result for the case $k=2$ was known before [ST98].)

1.5 Techniques

In this paper we analyse linearity testing algorithms and PCP verifiers that were considered in [Tre98, ST98], but whose analysis was beyond the techniques of those papers. The main contribution of [Tre98] was the definition of a family of tests/verifiers. In [ST98] a new *composition theorem* was proved, showing how "inner verifiers" developed in analogy to the linearity testing algorithms of [Tre98] could yield PCP constructions. Both [Tre98] and [ST98] make use of ideas appeared in previous papers, most notably [Hås96, Hås97]. The reader can find additional references and proper credits in [Tre98, ST98].

We first consider the point where the analysis of the linearity testing algorithms in [Tre98] broke down. A similar difficulty arises in PCP constructions. In the remainder of this discussion, we will assume that the boolean values $\{0,1\}$ are represented as $\{1,-1\}$. In particular, $\oplus$ becomes multiplication, and a function $f:\{1,-1\}^n \to \{1,-1\}$ is linear if and only if there is a subset $\alpha \subseteq \{1,\ldots,n\}$ such that $f(x)=\prod_{i\in\alpha} x_i$. We also associate to every function $f:\{1,-1\}^n \to \mathbf{R}$ (as a special case, to every function $f:\{1,-1\}^n \to \{1,-1\}$), a sequence of 2^n real values, a real value denoted $\hat{f}_\alpha$ for any $\alpha \subseteq \{1,\ldots,n\}$. Such values are called the Fourier coefficients of f, and their definition and properties are not important in the following discussion. In [Tre98] it was proved that a key technical lemma in the analysis of linearity testing algorithms was to show that for every graph of the form $([k],S)$ and for every function $f:\{1,-1\}^n \to \{1,-1\}$,

$$\mathop{\mathbf{E}}_{x_1,\ldots,x_k\in\{1,-1\}^n}\left[\prod_{(i,j)\in S} f(x_i)f(x_j)f(x_i\cdot x_j)\right] \le \max_\alpha |\hat{f}_\alpha| \tag{1}$$

In [Tre98], Expression (1) was proved only for the special case where each connected component of $([k],S)$ is either a path or a bipartite graph with the smallest component of the bipartition having one or two vertices. The techniques of [Tre98], in the cases where they worked, also proved Expression (1) for all functions $f:\{1,-1\}^n \to \mathbf{R}$ having $\mathop{\mathbf{E}}_x f^2(x) \le 1$. However, for general graphs, there are functions for which $\mathop{\mathbf{E}}_x f^2(x) \le 1$, and for which Expression (1) does not hold. So the fact that the techniques of [Tre98] generalized to arbitrary real functions with $\mathbf{E}_x f^2(x) \leq 1$ was a fundamental limitation.

In this paper we show a technical lemma that reduces the general case of Expression (1) to the special case studied in [Tre98]. The reduction is a technically simple application of Cauchy-Swartchz, and only works for functions $f : \{1, -1\}^n \to \{1, -1\}$.

Similarly, we are able to show that the analysis of PCP "inner" verifiers that eluded the techniques of [ST98] can be reduced to the analysis of inner verifiers that were analysed in [ST98]. Once a tight analysis of inner verifiers is obtained, a "composition theorem" proved in [ST98] gives a PCP characterization of NP.

1.6 Organization of the Paper

We prove a simple probabilistic lemma in Section 2. The lemma gives the reduction of the general case of the analysis of linearity test and "inner verifiers" to the special cases studied in [Tre98, ST98]. Our results on linearity testing are presented in Section 3. Preliminary definitions on PCP are given in Section 4, and our results on PCP are in Section 5. We present in Section 6 a lower bound on the error probability of a generalized class of linearity testing algorithms.

2 A Probabilistic Lemma

Lemma 3 below is the main inequality we use in this paper. It is a consequence of the Cauchy-Schwarz inequality.

Lemma 3 *For every two integers k, l for every two functions $F : \{1, -1\}^k \times \{1, -1\}^l \to \mathbf{R}$ and $G : \{1, -1\}^l \to \mathbf{R}$, we have*

$$\mathop{\mathbf{E}}_{x,y}[F(x,y)G(y)] \leq \sqrt{\mathop{\mathbf{E}}_{x_1,x_2,y}[F(x_1,y)F(x_2,y)]}\sqrt{\mathop{\mathbf{E}}_{y}[G^2(y)]} \tag{2}$$

PROOF:

$$\begin{aligned}
\mathop{\mathbf{E}}_{x,y}[F(x,y)G(y)] &= \mathop{\mathbf{E}}_{y}[(\mathop{\mathbf{E}}_{x}[F(x,y)])G(y)] \\
&\leq \sqrt{\mathop{\mathbf{E}}_{y}[(\mathop{\mathbf{E}}_{x}[F(x,y)])^2]}\sqrt{\mathop{\mathbf{E}}_{y}[(G(y))^2]} \\
&= \sqrt{\mathop{\mathbf{E}}_{y}[\mathop{\mathbf{E}}_{x_1}[F(x_1,y)] \cdot \mathop{\mathbf{E}}_{x_2}[F(x_2,y)]]}\sqrt{\mathop{\mathbf{E}}_{y}[(G(y)^2]} \\
&= \sqrt{\mathop{\mathbf{E}}_{x_1,x_2,y}[F(x_1,y)F(x_2,y)]}\sqrt{\mathop{\mathbf{E}}_{y}[G^2(y)]}
\end{aligned}$$

The second inequality is a consequence of the following (average) version of the Cauchy-Schwarz inequality: for any two random variables A, B over the same sample space, $\mathbf{E}[AB] \leq \sqrt{\mathbf{E}[A^2]}\sqrt{\mathbf{E}[B^2]}$. $\square$

In particular, if the range of G is contained in the interval $[-1, 1]$, then $G^2(\cdot)$ is always at most one, and the expression can be furtherly simplified.

Corollary 2 *For every two integers k, l for every two functions $F : \{1, -1\}^k \times \{1, -1\}^l \to \mathbf{R}$ and $G : \{1, -1\}^l \to [-1, 1]$, we have*

$$\mathop{\mathbf{E}}_{x,y}[F(x,y)G(y)] \leq \sqrt{\mathop{\mathbf{E}}_{x_1,x_2,y}[F(x_1,y)F(x_2,y)]} \tag{3}$$

3 Linearity Test with Amortized Query Complexity $1+\varepsilon$

Let $([k], E)$ be an undirected graph on k vertices. Consider the test

> $\mathsf{LinTestGraph}(G; f)$
> Choose uniformly at random $x_1, \ldots, x_k \in \{1,-1\}^n$
> **if** $f(x_i)f(x_j)f(x_ix_j) = 1$ for all $(i,j) \in E$
> **then accept**
> **else reject**

It has been shown in [Tre98] (and it is an easy calculation) that

$$\mathbf{Pr}[\mathsf{LinTestGraph}(G; f) \text{ accepts }] = \frac{1}{2^{|E|}} \sum_{S \subseteq E} \mathop{\mathbf{E}}_{x_1,\ldots,x_k} \left[\prod_{(i,j)\in S} f(x_i)f(x_j)f(x_ix_j) \right] . \tag{4}$$

where we use the convention that a product ranging over an empty set is 1.

It has also been shown in [Tre98] that for every k

$$\mathop{\mathbf{E}}_{z_1,z_2,x_1,\ldots,x_k} \prod_{i\in[2],\ j\in[k]} f(z_i)f(x_j)f(z_ix_j) \le \max_\alpha \hat{f}_\alpha^2$$

The main result of this section is the following lemma.

Lemma 4 *Let $f : \{1,-1\}^n \to \{1,-1\}$, let $k \ge 2$ be an arbitrary integer, and let $([k], S)$ be an arbitrary graph. Then $\mathop{\mathbf{E}}_{x_1,\ldots,x_k} \prod_{(i,j)\in S} f(x_i)f(x_j)f(x_ix_j) \le \max_\alpha |\hat{f}_\alpha|$.*

Proof: Let us assume without loss of generality that node (1,2) is an edge. We can define two functions $L : \{1,-1\}^{k\times n} \to \{1,-1\}$ and $R : \{1,-1\}^{(k-1)\times n} \to \{1,-1\}$ as follows:

$$L(x_1, x_2, \ldots, x_k) = \prod_{(1,j)\in S} f(x_1)f(x_1x_j)f(x_j)$$

and

$$R(x_2, \ldots, x_k) = \prod_{(i,j)\in S, i,j\neq 1} f(x_i)f(x_ix_j)f(x_j)$$

That is, the expression that we want to bound is the expectation of $L(x_1, \ldots, x_k)R(x_2, \ldots, x_k)$. We can invoke Corollary 2 and see that

$$\begin{aligned} \mathop{\mathbf{E}}_{x_1,\ldots,x_k} \prod_{(i,j)\in S} f(x_i)f(x_j)f(x_ix_j) &= \mathop{\mathbf{E}}_{x_1,\ldots,x_k} L(x_1, \ldots, x_k)R(x_2, \ldots, x_k) \\ &\le \sqrt{\mathop{\mathbf{E}}_{z_1,z_2,x_2,\ldots,x_k} L(z_1, x_2, \ldots, x_k)L(z_2, x_2, \ldots, xk)} \end{aligned}$$

where

$$\mathop{\mathbf{E}}_{z_1,z_2,x_2,\ldots,x_k} L(z_1, x_2, \ldots, x_k)L(z_2, x_2, \ldots, xk) = \mathop{\mathbf{E}}_{z_1,z_2,x_2,\ldots,x_k} \prod_{(1,j)\in S} (f(z_1)f(z_1x_j)f(x_j))(f(z_2)f(z_2x_j)f(x_j))$$

Now, if we call d the degree of vertex 1 in S, and we call $y_1, \ldots, y_d$ the variables x_j corresponding to a j adjacent to 1, the above expression becomes

$$= \mathop{\mathbf{E}}_{z_1, z_2, y_1, \ldots, y_d} \prod_{i \in [2], j \in [d]} f(z_i) f(z_i y_j) f(y_j) \leq \max_\alpha \hat{f}_\alpha^2$$

□

A tight analysis of graph-tests is now an immediate consequence of Lemma 4 and Expression 4

Theorem 5 *Let $f : \{1, -1\}^n \to \{1, -1\}$, let $k \geq 2$ be an arbitrary integer, and let $G = ([k], E)$ be an arbitrary graph. Then $\mathbf{Pr}[\mathsf{LinTestGraph}(G; f)$ accepts $] \leq \frac{1}{2^{|E|}} + \frac{2^{|E|-1}}{2^{|E|}} \max_\alpha |\hat{f}_\alpha|$.*

4 Background on PCP

In this section we give the definition of "inner" verifier and we state a result of [ST98] that reduces the task of proving a PCP characterization of NP to the task of constructing an inner verifier with appropriate parameter.

We begin by introducing some additional notation. For an integer n, we denote by $\mathcal{F}_n$ the set of functions $f : [n] \to \{1, -1\}$. The operator $\circ$ denotes composition of functions, i.e. if $f \in \mathcal{F}_n$ and $\pi : [m] \to [n]$ then the function $f \circ \pi \in \mathcal{F}_m$ is defined as $(f \circ \pi)(b) = f(\pi(b))$ for any $b \in [m]$.

The Long code is the set of linear functions whose support is a singleton, i.e. $\mathsf{LONG}_n = \{l_{\{a\}} : a \in [n]\}$. We say that $l_{\{a\}}$ is the Long code of a. Thus, the Long code is formed by n codewords of length 2^n.

Finally, we need a notion analogous to that of *folding* from [BGS98]. Observe that if $A = l_{\{a\}}$ is a codeword of the Long code, then $A(f) = f(a) = -(-f(a)) = -A(-f)$ for any f; for any function $A : \mathcal{F}_n \to \{1, -1\}$ we will define a new function A' that satisfies such a property. The definition of A' is as follows:

$$A'(f) = \begin{cases} A(f) & \text{If } f(1) = 1 \\ -A(-f) & \text{If } f(1) = -1. \end{cases}$$

We stress that, for any f, $A'(f)$ can be evaluated with one query to A, moreover A' is equal to A if A is a codeword of the Long code.

We can now give the formal definition of inner verifier, and the "composition theorem" that shows that the existence of an inner verifier with appropriate parameters implies a PCP characterization of NP.

Definition 6 (k-Inner Verifier [ST98]) *A k-*inner verifier *is a randomized oracle algorithm V that is given a sequence of functions $\pi_1, \ldots, \pi_k$ where $\pi_j : \{1, -1\}^m \to \{1, -1\}^n$, and has oracle access to a function $A : \mathcal{F}_n \to \{1, -1\}$ and to a sequence of functions $B_1, \ldots, B_k$ where $B_j : \mathcal{F}_m \to \{1, -1\}$.*

Definition 7 (Decoding Procedure) *A* decoding procedure *is a randomized algorithm D such that on input an integer parameter n and a function $A : \mathcal{F}_n \to \{1, -1\}$ returns an element of $[n]$.*

Definition 8 (Good Inner Verifier) *A k-inner verifier V is (c, s, q)-good with respect to a decoding procedure D if for any $\pi_1, \ldots, \pi_k : [m] \to [n]$, any $A : \mathcal{F}_n \to \{1, -1\}$, and any $B_1, \ldots, B_k : \mathcal{F}_m \to \{1, -1\}$, the following properties hold.*

- [Number of Queries] *V makes at total number of at most q non-adaptive oracle queries.*

- [Completeness] *if A is the Long code of a, and B_i is the long code of b_i, and $\pi_i(b_i) = a$, then* $\mathbf{Pr}[V(A', B'_1, \ldots, B'_k, \pi_1, \ldots, \pi_k)\text{accepts}] \geq c$.

- [Soundness] *For any constant $\delta > 0$, there is a positive constant $\delta' > 0$ independent of m, n, (but possibly dependent on δ) such that:*
If $\mathbf{Pr}[V(A', B'_1, \ldots, B'_k, \pi_1, \ldots, \pi_k)\text{accepts}] \geq s + \delta$.
Then $\mathbf{Pr}[$ *at least two values out of* $D(n, A'), \pi_1(D(m, B'_1)), \ldots, \pi_k(D(m, B'_k))$ *are equal*$] \geq \delta'$.

The Composition Theorem from [ST98] is as follows.[3]

Theorem 9 ([ST98]) *If there exists a (c, s, q)-good k-inner verifier V with respect to a decoding procedure D then for any $\varepsilon > 0$* $\mathrm{NP} = \mathrm{naPCP}_{c,s+\varepsilon}[\log, q]$.

5 PCP with Amortized Query Complexity $1 + \varepsilon$

5.1 Construction

We will denote by $\mathcal{UF}_n$ the uniform distribution over functions in $\mathcal{F}_n$. For any $\varepsilon > 0$, we define the distribution $\mathcal{SF}_{\varepsilon,n}$ over $\mathcal{F}_n$ as follows: in order to sample a function e according to $\mathcal{SF}_{\varepsilon,n}$, for every $a \in \{1, -1\}^n$ we fix $e(a) = 1$ with probability $1 - \varepsilon$, and we fix $e(a) = -1$ with probability ε.

5.1.1 Verifier

For any $\varepsilon > 0$, integers h, k, and bipartite graph $G = ([h], [k], E)$ our inner verifier $\mathsf{Inner}_{G,\varepsilon}$ is described in Figure 1. $\mathsf{Inner}_{G,\varepsilon}$ is obtained by iterating a basic 3-query inner verifier by Håstad [Hås97]. The basic protocol would access two tables A and B, would pick a function f uniformly from the domain of A, a function g uniformly from the domain of B, and a function e from the domain of B but with the non-uniform distribution $\mathcal{SF}_{\varepsilon,m}$; the verifier would accept iff $A(f)B(g) = B((f \circ \pi)ge)$. By recycling queries, we manage to execute $|E|$ iterations of the basic protocol while using only $h + k + |E|$ queries instead of $3|E|$ queries.

When $k = h$, and $|E| = [k] \times [k]$, our verifier makes k^2 iterations of the basic protocol by making only $2k + k^2$ queries (instead of $3k^2$).

5.1.2 Decoding Procedure

The decoding procedure D is based on the fact that, by Parseval's identity, the squares of the Fourier coefficients $\hat{A}_\alpha$'s and $\hat{B}_\beta$'s sum to 1 and can hence be thought of as a probability distribution.

For a table $A : \{1, -1\}^n \to \{1, -1\}$, the decoding procedure is defined as follows:

- Pick a set $\alpha \subseteq [n]$ with probability $\hat{A}^2_\alpha$; pick a random element $a \in \alpha$, return a. (Notice that this is well defined only when $\hat{A}_\emptyset = 0$, which is true for a folded A.)

[3] Theorem 9 was stated in this form in [ST98] and was a generalization of previous work by several people. We do not have space to give proper references here, but a history of related results can be found in [ST98].

$\mathsf{Inner}_{G,\varepsilon}(A, B_1, \ldots, B_k, \pi_1, \ldots, \pi_k)$
  Sample independently $f_1, \ldots, f_h$ from $\mathcal{UF}_n$,
    $g_1, \ldots, g_k$ from $\mathcal{UF}_m$
    and, for every $(i, j) \in E$, $e_{i,j}$ from $\mathcal{SF}_{\varepsilon,m}$
  if for all $(i, j) \in E$
    $A'(f_i)B'_j(g_j) = B'_j((f_i \circ \pi_j)g_j e_{i,j})$
      then accept
      else reject

Figure 1: The inner verifier.

5.2 Analysis

Let us fix parameters h, k, graph $G = ([h], [k], E)$, parameter $\varepsilon > 0$, and consider a possible input $(A, B_1, \ldots, B_k, \pi_1, \ldots, \pi_k)$ for $\mathsf{Inner}_{G,\varepsilon}$. To simplify notation, we assume that $A, B_1, \ldots, B_k$ are already the virtual folded tables accessed by the verifier.
The acceptance probability of the inner verifier is given by the following proposition.

Proposition 10

$$\mathbf{Pr}[\mathsf{Inner}_{G,\varepsilon}(A, B_1, \ldots, B_k, \pi_1, \ldots, \pi_k) \text{ accepts }] =$$

$$\frac{1}{2^{|E|}} \sum_{S \subseteq E} \left(\mathop{\mathbf{E}}_{\{f_i\}_{i=1}^h, \{g_j\}_{j=1}^k, \{e_{i,j}\}_{(i,j)\in E}} \left[\prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j((f_i \circ \pi_j)g_j e_{i,j}) \right] \right)$$

Where f_i are sampled according to $\mathcal{UF}_n$, g_j according to $\mathcal{UF}_m$, and $e_{i,j}$ according to $\mathcal{SF}_{\varepsilon,m}$. We will introduce some auxiliary notation to simplify the expression. In particular, we define $t(f, g, j) = \mathop{\mathbf{E}}_e[A(f)B_j(g)B_j((f \circ \pi_j)ge)]$, where e is sampled according to the distribution $\mathcal{SF}_{\varepsilon,m}$.

We also define, for any set $S \subseteq E$,

$$T_S = \mathop{\mathbf{E}}_{f_1,\ldots,f_h,g_1,\ldots,g_k} \left[\prod_{(i,j)\in S} \mathop{\mathbf{E}}_{e_{i,j}} [A(f_i)B_j(g_j)B_j((f_i \circ \pi_j)g_j e_{i,j})] \right] = \mathop{\mathbf{E}}_{f_1,\ldots,f_h,g_1,\ldots,g_k} \left[\prod_{(i,j)\in S} t(f_i, g_j, j) \right]$$

Then we have

$$\mathbf{Pr}[\mathsf{Inner}_{G,\varepsilon}(A, B_1, \ldots, B_k, \pi_1, \ldots, \pi_k) \text{ accepts }] = \frac{1}{2^{|E|}} \sum_{S \subseteq E} T_S$$

Our goal is to show that whenever, for a non-empty S, T_S is noticeably large, then the decoding procedure has a noticeable probability of success. A result from [ST98] gives such an analysis for the case where S is a subset of $[2] \times [d]$ for some d. In our analysis we will need the special case of the analysis of [ST98] restricted to $S = [2] \times [d]$.

Lemma 11 ([ST98]) *Let $d \geq 1$ be fixed. For every δ, there is a $\delta' = \mathrm{poly}(\delta)$ such that if $T_{[2]\times[d]} \geq \delta$ then the decoding procedure succeeds with probability at least δ'.*

Lemma 12 *Let $S \subseteq [h] \times [k]$ be non-empty. Then for every δ, there is a $\delta' = \mathrm{poly}(\delta)$ such that if $T_S > \delta$ then the decoding procedure succeeds with probability at least δ'.*

Proof: Up to renaming of the variables, we can assume that S contains a pair of the form $(1, j)$, and furthermore that all the pairs of the form $(1, j)$ are precisely $(1, 1), \ldots, (1, d)$ where $d \geq 1$.

Define

$$F(f_1, \ldots, f_k, g_1, \ldots, g_k) = \prod_{(1,j)\in S} t(f_1, g_j, j) = \prod_{j=1}^{d} t(f_1, g_j, j)$$

and

$$G(f_2, \ldots, f_k, g_1, \ldots, g_k) = \prod_{(i,j)\in S, i\neq 1} t(f_i, g_j, j)$$

so that we have

$$T_S = \mathop{\mathbf{E}}_{f_1,\ldots,f_h,g_1,\ldots,g_k} F(f_1, \ldots, f_h, g_1, \ldots, g_k) G(f_2, \ldots, f_h, g_1, \ldots, g_k)$$

using Corollary 2 we get that

$$\begin{aligned} T_S &\leq \sqrt{\mathop{\mathbf{E}}_{f_1',f_2',f_2,\ldots,f_k,g_1,\ldots,g_k} F(f_1', f_2, g_1, \ldots, g_k) F(f_2', f_2, g_1, \ldots, g_k)} \\ &= \sqrt{\mathop{\mathbf{E}}_{f_1',f_2',g_1,\ldots,g_d} \prod_{i\in[2],j\in[d]} t(f_i', g_j, j)} \end{aligned}$$

It follows that

$$\delta < T_S \leq \sqrt{T_{[2]\times[d]}}$$

and using Lemma 11 we can conclude that there exists a $\delta' = \mathrm{poly}(\delta^2) = \mathrm{poly}(\delta)$ such that the decoding procedure succeeds with probability at least δ'.

□

We can now prove the main result of this section. Notice that Theorem 1 is a consequence of Theorem 9 and of the following theorem (with an appropriate setting of parameters).

Theorem 13 $\mathsf{Inner}_{G,\varepsilon}$ *is a* $((1-\varepsilon)^{|E|}, 2^{-|E|}, h + k + |E|)$*-good* k*-inner verifier.*

Proof: [Of Theorem 13] Clearly $\mathsf{Inner}_{G,\varepsilon}$ makes $h + k + |E|$ queries and accepts valid proofs with probability is $(1-\varepsilon)^{|E|}$. Furthermore if its acceptance probability is at least $2^{-|E|} + \delta$, then there is a non-empty S such that $T_S > \delta$, so that there is a δ' (depending only on δ) such that the decoding procedure succeeds with probability at least δ'. □

6 Hypergraph Tests

It is natural to extend the family of tests, described in section 3, by associating a linearity test $\mathsf{LinTestHypergraph}(H)$ with every hypergraph $H = ([k], E)$ on k vertices, in the following way:

```
LinTestHypergraph(H; f)
  Choose uniformly at random x_1, ..., x_k ∈ {1, −1}^n
  if ∏_{i∈T} f(x_i) · f(∏_{i∈S} x_i) = 1 for all T ∈ E
    then accept
    else reject
```

Our intent in this section is to show, that using hypergraph tests would not improve the lower bound of $1 + \Omega\left(1/\sqrt{q}\right)$ amortized complexity for q queries.

We will do so, by describing an explicit function $f_n : \{1,-1\}^n \to \{1,-1\}$, which has small Fourier coefficients, and for any hypergraph H the acceptance probability of the test $\mathsf{LinTestHypergraph}(H)$ on f_n is large. In boolean notation, the function $f_n(x) = f_n(x_1...x_n)$ will be just $(x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus ....$

Definition 14 *Set $g : \{1,-1\}^2 \to \{1,-1\}$ by $g = 1$ on all inputs but $(-1,-1)$, where $g = -1$. (Note that g is just a transcription of a boolean boolean AND) to the $1,-1$ notation.*

The function $f_n : \{1,-1\}^n \to \mathbf{R}$ is set to be $f_n(x_1,\ldots,x_n) = g(x_1,x_2)g(x_3,x_4)\cdots g(x_{n-1},x_n)$ if n is even and $f_n(x_1...x_n) = f_{n-1}(x_1,\ldots,x_{n-1})$ if n is odd.

We point out, that $\max|\hat{f}| = \max \hat{f} = 2^{-\lfloor n/2 \rfloor}$. since $\max \hat{g} = \frac{1}{2}$. In the rest of this section we will prove the following result.

Proposition 15 *For any hypergraph H and linearity test $\mathsf{LinTestHypergraph}(H)$, with free bit complexity k and query complexity q, holds:*

$$\mathbf{Pr}[\mathsf{LinTestHypergraph}(H; f_n) \text{ accepts }] \geq \max\{2^{k-q}, 2^{-\binom{k}{2}}\} \ .$$

First, note that, similarly to (4),

$$\mathbf{Pr}[\mathsf{LinTestHypergraph}(H; f) \text{ accepts }] = \frac{1}{2^{|E|}} \sum_{\mathcal{S} \subseteq E} \mathop{\mathbf{E}}_{x_1,\ldots,x_k} \left[\prod_{T \in \mathcal{S}} \prod_{i \in T} f(x_i) \cdot f\left(\prod_{i \in T} x_i \right) \right] \qquad (5)$$

In order to simplify this expression, we need to introduce some notation. Let $\mathcal{F} = \{\{1\},...,\{k\}\} \cup E := \{F_1...F_q\}$, be a family of all the vertices and the edges of H, viewed as subsets of $\{1,\ldots,k\}$. Let $\mathbf{A}$ be a $k \times q$ zero-one matrix whose q columns are given by $F_1...F_q$, which we view as $0,1$ vectors of length k (in particular, the first k columns of $\mathbf{A}$ form the $k \times k$ identity matrix). Let u_T, for $T \in E$ be a zero-one vector of length q which is 1 if F_i is either T or a singleton, corresponding to a vertex than T passes through; and 0 otherwise.

For $\mathcal{R} \subseteq 2^{[k]}$, let $\mathbf{E}(f,\mathcal{R}) := \mathop{\mathbf{E}}_{x_1,\ldots,x_k} \left[\prod_{R \in \mathcal{R}} f\left(\prod_{l \in R} x_l \right) \right]$.

Let $U = \text{ Span } (u_T)_{T \in E}$ be a d-dimensional subspace of $\mathbf{Z}_2^q$, then for a boolean f, (5) is just

$$\sum_{\mathcal{S} \subseteq E} \mathop{\mathbf{E}}_{x_1,\ldots,x_k} \left[\prod_{i \,:\, \bigoplus_{T \in \mathcal{S}} u_T(i) = 1} f\left(\prod_{l \in F_i} x_l \right) \right] = \frac{1}{2^d} \sum_{u = (u(1),\ldots u(q)) \in U} \mathop{\mathbf{E}}_{x_1,\ldots,x_k} \left[\prod_{i \,:\, u(i)=1} f\left(\prod_{l \in F_i} x_l \right) \right] =$$

$$\frac{1}{2^d} \sum_{u = (u(1),\ldots u(q)) \in U} \mathbf{E}(f, \{F_i : u(i) = 1\}). \qquad (6)$$

Our goal is to show that many of the terms $\mathbf{E}(f, \{F_i : u(i) = 1\})$ are 1.

Definition 16 *A family $\mathcal{R} \subseteq 2^{[k]}$ is an "even cover", iff any element $i \in \{1,\ldots,k\}$ and any pair of elements $i \neq j \in \{1,\ldots,k\}$ are covered an even number of times by the sets $R \in \mathcal{R}$.*

Lemma 17 *Let $\mathcal{R} \subseteq 2^{[k]}$. If $\mathcal{R}$ is an even cover, than $\mathbf{E}(f_n, \mathcal{R}) = 1$. For any $\mathcal{R}$, $\mathbf{E}(f_n, \mathcal{R}) \geq 0$.*

PROOF: Let $\mathcal{R} \subseteq 2^{[k]}$. Observe, that for any integer m and function $f : \{1,-1\}^m \to \mathbf{R}$, the average $\mathbf{E}(f,\mathcal{R}) := \mathop{\mathbf{E}}_{x_1,\ldots,x_k \in \{1,-1\}^m} \left[\prod_{R \in \mathcal{R}} f\left(\prod_{l \in R} x_l \right) \right]$ is well-defined. The definition of f_n as a product of $\lfloor n/2 \rfloor$ disjoint copies of g implies $\mathbf{E}(f_n, \mathcal{R}) = (\mathbf{E}(g, \mathcal{R}))^{\lfloor n/2 \rfloor}$. Therefore, it suffices to prove the lemma for g.

We start with the first item, and claim that for *any* choice of $x_1, \ldots, x_k \in \{-1,1\}^2$ holds $\prod_{R \in \mathcal{R}} g\left(\prod_{l \in R} x_l\right) = 1$.

For this purpose, instead of working with functions from $\{-1,1\}^n$ to $\{-1,1\}$, it will be convenient to revert to boolean notation, and to deal with functions from $\{0,1\}^n$ to $\{0,1\}$.

Let $x_i = (a_i, b_i) \in \mathbf{Z}^2$, for $i = 1...k$. Recalling the definition of g as the boolean product, we want to show, that $\sum_{R \in \mathcal{R}} \left(\sum_{j \in R} a_j\right) \cdot \left(\sum_{l \in R} b_l\right) = 0$ (in $\mathbf{Z}_2$), for any choice of $a_1...a_k$, $b_1...b_k \in \mathbf{Z}$.

For $i = 1...k$, let N_i be the number of times that i is covered by the sets in $\mathcal{R}$, and for $i \neq j$ let M_{ij} be the number of times the pair i, j is covered by the sets in $\mathcal{R}$. Opening up the brackets, we see that

$$\sum_{R \in \mathcal{R}} \left(\sum_{j \in R} a_j\right) \cdot \left(\sum_{l \in R} b_l\right) = \sum_{i=1}^{t} N_i \; a_i \cdot b_i \quad + \sum_{1 \le i \neq j \le t} M_{ij} \; a_i \cdot b_j = 0,$$

since in our case all the numbers N_i and M_{ij} are even.

We pass to the second item of the lemma. We have to show, that the boolean function

$$h(a_1...a_k, b_1...b_k) := \sum_{R \in \mathcal{R}} \left(\sum_{j \in R} a_j\right) \cdot \left(\sum_{l \in R} b_l\right)$$

has at least as many zeroes as it has ones on $\mathbf{Z}_2^k \times \mathbf{Z}_2^k$. Note, that for any fixed choice of $a_1...a_k$, the function h becomes a *linear functional* on Z_2^k. The claim follows, since a linear functional is zero with probability either 1 (if it is identically zero) or $\frac{1}{2}$. $\square$

Proof: First, we show the set W of vectors $u \in U$, such that the family $\mathcal{R} = \{F_i \;:\; u(i) = 1\}$ is an even cover, to be a linear subspace of U. For this purpose, consider the following transformation Φ from $\mathbf{Z}_2^k$ to $\mathbf{Z}_2^{k(k+1)/2}$: For $v \in \mathbf{Z}_2^k$, the vector $\Phi(v)$ will be indexed by pairs (i,i) and (i,j), for $1 \le i < j \le k$, where $\Phi(v)_{ii} = v_i \cdot v_i = v_i$, and $\Phi(v)_{ij} = v_i \cdot v_j$. The multiplication is in $\mathbf{Z}_2$.

The point is in the following simple fact: A family $\mathcal{R} \subseteq 2^{[k]}$ is an even cover of {1,...,k} iff $\sum_{R \in \mathcal{R}} \Phi(R) = 0$. The summation here is in $Z_2^{k(k+1)/2}$.

Let $\Phi(\mathbf{A})$ be the $(k(k+1)/2) \times q$ zero-one matrix with columns $\Phi(F_1)...\Phi(F_q)$, and let $V \subseteq \mathbf{Z}_2^q$ be the row space of $\Phi(\mathbf{A})$. We have:

$$W = \left\{ u \in U \;:\; \sum_i u(i)\Phi(F_i) = 0 \right\} = U \cap V^{\perp}, \tag{7}$$

where $V^{\perp} \subseteq \mathbf{Z}_2^q$ denotes the linear subspace of vectors orthogonal to V.

Therefore, we have to solve the question of finding a lower bound for the size of the intersection $U \cap V^{\perp}$ of two subspaces.

Let $X \subseteq \mathbf{Z}_2^q$ be the subspace spanned by the first k rows of $\Phi(\mathbf{A})$. Note, that these are, by the definition of Φ, precisely the rows of $\mathbf{A}$ itself, and, in particular, $\dim X = k$. We claim that U and $V^{\perp}$ are subspaces of $X^{\perp}$. This is immediate for $V^{\perp}$, since $X \subseteq V$. As for U, recall $U = \text{ Span } \{u_T\}_{T \in E}$. Therefore, it is enough to show that for any $T \in E$ holds $u_T \in X^{\perp}$, which just restates the obvious fact that the sets $\{\{i\} \;:\; i \in T\}$ and T itself, cover every vertex in $\{1...k\}$ an even number of times.

We recall two simple facts from linear algebra: (1) For any subspace X of $\mathbf{Z}_2^q$ holds $\dim(X) + \dim(X^{\perp}) = q$.
(2) For any two subspaces Y and Z of a space S holds $\dim(Y \cap Z) \ge \dim(Y) + \dim(Z) - \dim(S)$.

Therefore we have two lower bounds on $\dim(U \cap V^{\perp})$: Trivially,

$$\dim(U \cap V^{\perp}) \geq 0,$$

on the other hand,

$$\dim(U \cap V^{\perp}) \geq d + (q - k(k+1)/2) - (q-k) = d - k(k-1)/2.$$

□

Lemma 18 *The number of vectors $u \in U$, such that the family $\mathcal{R} = \{F_i \ : \ u(i) = 1\}$ is an even cover, is at least* $\max\{1, 2^{d-\binom{k}{2}}\}$.

Lemma 19

$$d \leq q - k.$$

PROOF: In the notation of the proof of lemma 18, we have shown that U is a subspace of $X^{\perp}$. Therefore:

$$q - k = \dim(X^{\perp}) \geq \dim(U) = d.$$

□

Proposition 15 now follows from lemma 17, lemma 18, lemma 19 and (6).

We are now ready to state and prove the main result of this section.

Theorem 20 *For any hypergraph H the amortized query complexity of* LinTestHypergraph(H) *is at least* $1 + \Omega(1/\sqrt{q})$.

PROOF: The theorem follows from proposition 15, by checking the two cases: $q \leq k(k+1)/2$, or $q > k(k+1)/2$. □

Acknowledgments

Thanks to Madhu Sudan for suggestions that significantly simplified our earlier proofs.

References

[ALM+98] 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. Preliminary version in *Proc. of FOCS'92.*

[AS97] S. Arora and M. Sudan. Improved low degree testing and its applications. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 485–495, 1997.

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in *Proc. of FOCS'92.*

[Bel96] M. Bellare. Proof checking and approximation: Towards tight results. *Sigact News*, 27(1), 1996.

[BGLR93] M Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 294–304, 1993. See also the errata sheet in *Proc of STOC'94*.

[BGS98] 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. Preliminary version in *Proc. of FOCS'95*.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the 22nd ACM Symposium on Theory of Computing*, pages 73–83, 1990.

[BS94] M. Bellare and M. Sudan. Improved non-approximability results. In *Proceedings of the 26th ACM Symposium on Theory of Computing*, pages 184–193, 1994.

[Cre95] N. Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences*, 51(3):511–522, 1995.

[DFK+99] I. Dinur, E. Fischer, G. Kindler, R. Raz, and S. Safra. PCP characterizations of NP: Towards a polynomially-small error-probability. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, pages 29–40, 1999.

[FGL+91] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pages 2–12, 1991.

[FK94] U. Feige and J. Kilian. Two prover protocols - low error at affordable rates. In *Proceedings of the 26th ACM Symposium on Theory of Computing*, pages 172–183, 1994.

[Hås96] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, pages 627–636, 1996.

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 1–10, 1997.

[KMSV99] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28(1):164–191, 1999. Preliminary version in *Proc. of FOCS'94*.

[KSW97] S. Khanna, M. Sudan, and D.P. Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 11–20, 1997.

[RS97] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 475–484, 1997.

[ST98] M. Sudan and L. Trevisan. Probabilistically checkable proofs with low amortized query complexity. In *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science*, 1998.

[STX98] M. Serna, L. Trevisan, and F. Xhafa. The parallel approximability of non-boolean constraint satisfaction and restricted integer linear programming. In *Proceedings of the 15th Symposium on Theoretical Aspects of Computer Science*, pages 488–498. LNCS 1373, Springer-Verlag, 1998.

[Tre96] L. Trevisan. Positive linear programming, parallel approximation, and PCP's. In *Proceedings of the 4th European Symposium on Algorithms*, pages 62–75. LNCS 1136, Springer-Verlag, 1996.

[Tre97] L. Trevisan. Approximating satisfiable satisfiability problems. In *Proceedings of the 5th European Symposium on Algorithms*, pages 472–485. LNCS 1284, Springer-Verlag, 1997.

[Tre98] L. Trevisan. Recycling queries in PCPs and in linearity tests. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998.

[TSSW96] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, pages 617–626, 1996.

[Zwi98a] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[Zwi98b] U. Zwick. Finding almost satisfying assignment. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998.