Small PCPs with Low Query Complexity

Prahladh Harsha and Madhu Sudan *

Laboratory for Computer Science, Massachusetts Institute of Technology,
545 Technology Square, Cambridge, MA 02139.
{prahladh,madhu}@mit.edu

Abstract. Most known constructions of probabilistically checkable proofs (PCPs) either blow up the proof size by a large polynomial, or have a high (though constant) query complexity. In this paper we give a transformation with slightly-super-cubic blowup in proof size, with a low query complexity. Specifically, the verifier probes the proof in 16 bits and rejects every proof of a false assertion with probability arbitrarily close to $\frac{1}{2}$, while accepting corrects proofs of theorems with probability one. The proof is obtained by revisiting known constructions and improving numerous components therein. In the process we abstract a number of new modules that may be of use in other PCP constructions.

1 Introduction

Probabilistically checkable proofs (PCP) have played a major role in proving the hardness of approximation of various combinatorial optimization problems. Constructions of PCPs have been the subject of active research in the last ten years. In the last decade, there have been several "efficient" construction of PCPs which in turn have resulted in tighter inapproximability results. Arora et al. [1] showed that it is possible to transform any proof into a probabilistically checkable one of polynomial size, such that it is verifiable with a constant number of queries. Valid proofs are accepted with probability one (this parameter is termed the completeness of the proof), while any purported proof of an invalid assertion is rejected with probability $1/2$ (this parameter is the soundness of the proof). Neither the proof size, nor the query complexity is explicitly described there; however the latter is estimated to be around 10^6.

Subsequently much success has been achieved in improving the parameters of PCPs, constructing highly efficient proof systems either in terms of their size or their query complexity. The best result in terms of the former is a result of Polishchuk and Spielman [12]. They show how any proof can be transformed into a probabilistically checkable proof with only a mild blowup in the proof size, of $n^{1+\epsilon}$ for arbitrarily small $\epsilon > 0$ and that is checkable with only a constant number of queries. This number of queries however is of the order of $O(1/\epsilon^2)$, with the constant hidden by the big-Oh being some multiple of the query complexity of [1]. On the other hand, Håstad [10] has constructed PCPs for arbitrary NP statements where the query complexity is a mere three bits (for completeness almost 1 and soundness $1/2$). However the blowup in the proof size of Håstad's PCPs has an exponent proportional to the query complexity of the PCP of [1].

* Supported in part by a Sloan Foundation Fellowship and NSF Career Award CCR-9875511.

Thus neither of these "nearly-optimal" results provides simultaneous optimality of the two parameters. It is reasonable to wonder if this inefficiency in the combination of the two parameters is inherent; and our paper is motivated by this question.

We examine the size and query complexity of PCPs jointly and obtain a construction with reasonable performance in both parameters. The only previous work that mentions the joint size vs. query complexity of PCPs is a work of Friedl and Sudan [8], who indicate that NP has PCPs with nearly quadratic size complexity and in which the verifier queries the proof for 165 bits. The main technical ingredient in their proof was an improved analysis of the "low-degree test". Subsequent to this work, the analysis of low-degree tests has been substantially improved. Raz and Safra [13] and Arora and Sudan [3] have given highly efficient analysis of different low-degree tests. Furthermore, techniques available for "proof composition" have improved, as also have the construction for terminal "inner verifiers". In particular, the work of Håstad [9, 10], has significantly strengthened the ability to analyze inner verifiers used at the final composition step of PCP constructions.

In view of these improvements, it is natural to expect the performance of PCP constructions to improve. Our work confirms this expectation. However, our work exposes an enormous number of complications in the natural path of improvement. We resolve most of these, with little loss in performance and thereby obtain the following result: Satisfiability has a PCP verifier that makes at most 16 oracle queries to a proof of size at most $n^{3+o(1)}$, where n is the size of the instance of satisfiability. Satisfiable instances have proofs that are accepted with probability one, while unsatisfiable instances are accepted with probability arbitrarily close to $1/2$. (See Main Theorem 1.)

We also raise several technical questions whose positive resolution may lead to a PCP of nearly quadratic size and query complexity of 6. Surprisingly, no non-trivial limitations are known on the joint size + query complexity of PCPs. In particular, it is open as to whether nearly linear sized PCPs with query complexity of 3 exist for NP statements.

2 Overview

We first recall the standard definition of the class $\mathrm{PCP}_{c,s}[r,q]$.

Definition 1. *For functions $r, q : \mathbb{Z}^+ \to \mathbb{Z}^+$, a probabilistic oracle machine (or verifier) V is (r,q)-restricted if on input x of length n, the verifier tosses at most $r(n)$ random coins and queries an oracle π for at most $q(n)$ bits. A language $L \in \mathrm{PCP}_{c,s}[r,q]$ if there exists an (r,q)-restricted verifier V that satisfies the following properties on input x.*

Completeness *If $x \in L$ then there exists π such that V on oracle access to π accepts with probability at least c.*

Soundness *If $x \notin L$ then for every oracle π, the verifier V accepts with probability strictly less than s.*

While our principal interest is in the size of a PCP and not in the randomness, it is well-known that the size of a probabilistically checkable proof (or more precisely, the number of distinct queries to the oracle π) is at most $2^{r(n)+q(n)}$. Thus the size is implicitly governed by the randomness and query complexity of a PCP. The main result of this paper is the following.

Main Theorem 1. *For every $\varepsilon, \mu > 0$,* $\mathrm{SAT} \in \mathrm{PCP}_{1,\frac{1}{2}+\mu}\left[(3+\varepsilon)\log n, 16\right]$.

Remark: Actually the constants ε and μ above can be replaced by some $o(1)$ functions; but we don't derive them explicitly.

It follows from the parameters that the associated proof is of size at most $O(n^{3+\varepsilon})$. Cook [6] showed that any language in NTIME$(t(n))$ could be reduced to SAT in $O(t(n)\log t(n))$ time such that instances of size n are mapped to Boolean formulae of size at most $O(t(n)\log t(n))$. Combining this with the Main Theorem 1, we have that every language in NP has a PCP with at most a slightly super-cubic blowup in proof size and a query complexity as low as 16 bits.

2.1 MIP and Recursive Proof Composition

As pointed out earlier, the parameters we seek are such that no existing proof system achieves them. Hence we work our way through the PCP construction of Arora et al. [1] and make every step as efficient as possible. The key ingredient in their construction (as well as most subsequent constructions) is the notion of recursive composition of proofs, a paradigm introduced by Arora and Safra [2]. The paradigm of recursive composition is best described in terms of multi-prover interactive proof systems (MIPs).

Definition 2. *For integer p, and functions $r, a : \mathbb{Z}^+ \to \mathbb{Z}^+$, an MIP verifier V is (p, r, a)-restricted if it interacts with p mutually-non-interacting provers $\pi_1, \ldots, \pi_p$ in the following restricted manner. On input x of length n, V picks a random $r(n)$-bit string R and generates p queries $q_1, \ldots, q_p$ and a circuit C of size at most $a(n)$. The verifier then issues query q_i to prover π_i. The provers respond with answers $a_1, \ldots, a_p$ each of length at most $a(n)$ and the verifier accepts x iff $C(a_1, \ldots, a_p) = \mathsf{true}$. A language L belongs to $\mathrm{MIP}_{c,s}[p, r, a]$ if there exists a (p, r, a)-restricted MIP verifier V such that on input x:*

Completeness *If $x \in L$ then there exist $\pi_1, \ldots, \pi_p$ such that V accepts with probability at least c.*

Soundness *If $x \notin L$ then for every $\pi_1, \ldots, \pi_p$, V accepts with probability less than s.*

It is easy to see that $\mathrm{MIP}_{c,s}[p, r, a]$ is a subclass of $\mathrm{PCP}_{c,s}[r, pa]$ and thus it is beneficial to show that SAT is contained in MIP with nice parameters. However, much stronger benefits are obtained if the containment has a small number of provers, even if the answer size complexity (a) is not very small. This is because the verifier's actions can usually be simulated by a much more efficient verification procedure, one with much smaller answer size complexity, at the cost of a few more provers. Results of this nature are termed proof composition lemmas; and the efficient simulators of the MIP verification procedure are usually called "inner verification procedures".

The next three lemmas divide the task of proving Main Theorem 1 into smaller subtasks. The first gives a starting MIP for satisfiability, with 3 provers, but polylogarithmic answer size. We next give the composition lemma that is used in the intermediate stages. The final lemma gives our terminal composition lemma – the one that reduces answer sizes from some slowly growing function to a constant.

Lemma 2. *For every* $\varepsilon, \mu > 0$, $\mathrm{SAT} \in \mathrm{MIP}_{1,\mu}[3, (3+\varepsilon)\log n, \mathrm{poly}\log n]$.

Lemma 2 is proven in Sect. 3. This lemma is critical to bounding the proof size. This lemma follows the proof of a similar one (the "parallelization" step) in [1]; however various aspects are improved. We show how to incorporate advances made by Polishchuk and Spielman [12], and how to take advantage of the low-degree test of Raz and Safra [13]. Most importantly, we show how to save a quadratic blowup in this phase that would be incurred by a direct use of the parallelization step in [1].

The first composition lemma we use is an off-the-shelf product due to [3]. Similar lemmas are implicit in the works of Bellare et al. [5] and Raz and Safra [13].

Lemma 3 ([3]). *For every* $\epsilon > 0$ *and* $p < \infty$*, there exist constants* c_1, c_2, c_3 *such that for every* $r, a : \mathbb{Z}^+ \to \mathbb{Z}^+$,

$$\mathrm{MIP}_{1,\epsilon}[p, r, a] \subseteq \mathrm{MIP}_{1,\epsilon^{1/(2p+2)}}[p+3, r + c_1 \log a, c_2 (\log a)^{c_3}] \ .$$

The next lemma shows how to truncate the recursion. This lemma is proved in Sect. 4 using a "Fourier-analysis" based proof, as in [9, 10]. This is the first time that this style of analysis has been applied to MIPs with more than 2 provers. All previous analyses seem to have focused on composition with canonical 2-prover proof systems at the outer level. Our analysis reveals surprising complications and forces us to use a large number (seven) of extra bits to effect the truncation.

Lemma 4. *For every* $\epsilon > 0$ *and* $p < \infty$*, there exists a* $\gamma > 0$ *such that for every* $r, a : \mathbb{Z}^+ \to \mathbb{Z}^+$,

$$\mathrm{MIP}_{1,\gamma}[p, r, a] \subseteq \mathrm{PCP}_{1,\frac{1}{2}+\epsilon}[r + O\left(2^{pa}\right), p + 7] \ .$$

Proof (of Main Theorem 1). The proof is straightforward given the above lemmas. We first apply Lemma 2 to get a 3-prover MIP for SAT, then apply Lemma 3 twice to get a 6- and then a 9-prover MIP for SAT. The answer size in the final stage is $\mathrm{poly}\log\log\log n$. Applying Lemma 4 at this stage we obtain a 16-query PCP for SAT; and the total randomness in all stages remains $(3+\varepsilon)\log n$. □

Organization of the paper: In Section 3, we prove Lemma 2. For this purpose, we present the Polynomial Constraint Satisfaction problem in Section 3.2 and discuss its hardness. We then discuss the Low degree Test in Section 3.3. Most aspects of the proofs in Section 3 are drawn from previous works of [1, 3, 12, 13]. Hence, we abstract the main results in this section and leave the detailed proofs to the full version of the paper[1] In Section 4, we present the proof of Lemma 4. In section 5 we suggest possible approaches for improvements in the joint size-query complexity of PCPs.

3 A Randomness Efficient MIP for SAT

In this section, we use the term "length-preserving reductions", to refer to reductions in which the length of the target instance of the reduction is nearly-linear ($O(n^{1+\epsilon})$ for arbitrarily small ϵ) in the length of the source instance.

[1] A full version of this paper can be found at `ftp://ftp.eccc.uni-trier.de/pub/eccc/reports/2000/TR00-061/`

To prove membership in SAT, we first transform SAT into an algebraic problem. This transformation comes in two phases. First we transform it to an algebraic problem (that we call AP for lack of a better name) in which the constraints can be enumerated compactly. Then we transform it to a promise problem on polynomials, called Polynomial Constraint Satisfaction (PCS), with a large associated gap. We then show how to provide an MIP verifier for the PCS problem.

Though most of these results are implicit in the literature, we find that abstracting them cleanly significantly improves the exposition of PCPs. The first problem, AP, could be proved to be NP-hard almost immediately, if one did not require length-preserving reductions. We show how the results of Polishchuk and Spielman [12] imply a length preserving reduction from SAT to this problem. We then reduce this problem to PCS. This step mimics the sum-check protocol of Lund et al. [11]. The technical importance of this intermediate step is the fact that it does *not* refer to "low-degree" tests in its analysis. Low-degree tests are primitives used to test if the function described by a given oracle is close to some (unknown) multivariate polynomial of low-degree. Low-degree tests have played a central role in the constructions of PCPs. Here we separate (to a large extent) their role from other algebraic manipulations used to obtain PCPs/MIPs for SAT .

In the final step, we show how to translate the use of state-of-the-art low-degree tests, in particular the test of Raz and Safra [13], in conjunction with the hardness of PCS to obtain a 3-prover MIP for SAT. This part follows a proof of Arora et al. [1] (their parallelization step); however a direct implementation would involve $6 \log n$ randomness, or an n^6 blow up in the size of the proof. Part of this is a cubic blow up due to the use of the low-degree test and we are unable to get around this part. Direct use of the parallelization also results in a quadratic blowup of the resulting proof. We save on this by creating a variant of the parallelization step of [1] that uses higher dimensional varieties instead of 1-dimensional ones.

3.1 A Compactly Described Algebraic NP-hard Problem

Definition 3. *For functions $m, h : \mathbb{Z}^+ \to \mathbb{Z}^+$, the problem $\mathrm{AP}_{m,h}$ has as its instances $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$ where: H is a field of size $h(n)$, $\psi : H^7 \to H$ is a constant degree polynomial, T is an arbitrary function from H^m to H and the ρ_i's are linear maps from H^m to H^m, for $m = m(n)$. (T is specified by a table of values, and ρ_i's by $m \times m$ matrices.) $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$ if there exists an assignment $A : H^m \to H$ such that for every $x \in H^m$, $\psi(T(x), A(\rho_1(x)), \ldots, A(\rho_6(x))) = 0$.*

The above problem is just a simple variant of standard constraint satisfaction problems, the only difference being that its variables and constraints are now indexed by elements of H^m. The only algebra in the above problem is in the fact that the functions ρ_i, which dictate which variables participate in which constraint, are linear functions. The following statement, abstracted from [12], gives the desired hardness of AP.

Lemma 5. *There exists a constant c such that for any pair of functions $m, h : \mathbb{Z}^+ \to \mathbb{Z}^+$ satisfying $h(n)^{m(n)-c} \geq n$ and $h(n)^{m(n)} = O(n^{1+o(1)})$, SAT reduces to $\mathrm{AP}_{m,h}$ under length preserving reductions.*

We note that Szegedy [16] has given an alternate abstraction of the result of [12] which focuses on some different aspects and does not suffice for our purposes.

3.2 Polynomial Constraint Satisfaction

We next present an instance of an algebraic constraint satisfaction problem. This differs from the previous one in that its constraints are "wider", the relationship between constraints and variables that appear in it is arbitrary (and not linear), and the hardness is not established for arbitrary assignment functions, but only for low-degree functions. All the above changes only make the problem harder, so we ought to gain something – and we gain in the gap of the hardness. The problem is shown to be hard even if the goal is only to separate satisfiable instances from instances in which only ϵ fraction of the constraints are satisfiable. We define this gap version of the problem first.

Definition 4. *For $\epsilon : \mathbb{Z}^+ \to \mathbb{R}^+$, and $m, b, q : \mathbb{Z}^+ \to \mathbb{Z}^+$ the promise problem $\mathrm{GapPCS}_{\epsilon,m,b,q}$ has as instances $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$, where $d, k, s \leq b(n)$ are integers and $\mathbb{F}$ is a field of size $q(n)$ and $C_j = (A_j; x_1^{(j)}, \ldots, x_k^{(j)})$ is an algebraic constraint, given by an algebraic circuit A_j of size s on k inputs and $x_1^{(j)}, \ldots, x_k^{(j)} \in \mathbb{F}^m$, for $m = m(n)$. $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ is a YES instance if there exists a polynomial $p : \mathbb{F}^m \to \mathbb{F}$ of degree at most d such that for every $j \in \{1, \ldots, t\}$, the constraint C_j is satisfied by p, i.e., $A_j(p(x_1^{(j)}), \ldots, p(x_k^{(j)})) = 0$. $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ is a NO instance if for every polynomial $p : \mathbb{F}^m \to \mathbb{F}$ of degree at most d it is the case that at most $\epsilon(n) \cdot t$ of the constraints C_j are satisfied.*

Lemma 6. *There exist constants c_1, c_2 such that for every choice of functions ϵ, m, b, q satisfying $(b(n)/m(n))^{m(n)-c_1} \geq n$, $q(n)^{m(n)} = O\left(n^{1+o(n)}\right)$, $q(n) \geq c_2 b(n)/\epsilon(n)$, SAT reduces to $\mathrm{GapPCS}_{\epsilon,m,b,q}$ under length preserving reductions.*

(The problem $\mathrm{AP}_{m,h}$ is used as an intermediate problem in the reduction. However we don't mention this in the lemma, since the choice of parameters m, h may confuse the statement further.) The proof of this lemma is inspired by the sum-check protocol of Lund et al. [11] while the specific steps in our proof follow the proof in Sudan [15].

3.3 Low-Degree Tests

Using GapPCS it is easy to produce a simple probabilistically checkable proof for SAT. Given an instance of SAT, reduce it to an instance $\mathcal{I}$ of GapPCS ; and provide as proof the polynomial $p : \mathbb{F}^m \to \mathbb{F}$ as a table of values. To verify correctness a verifier first "checks" that p is close to some polynomial and then verifies that a random constraint C_j is satisfied by p. Low-degree tests are procedures designed to address the first part of this verification step – i.e., to verify that an arbitrary function $f : \mathbb{F}^m \to \mathbb{F}$ is close to some (unknown) polynomial p of degree d.

Low-degree tests have been a subject of much research in the context of program checking and PCPs. For our purposes, we need tests that have very low probability of error. Two such tests with analyses are known, one due to Raz and Safra [13] and another due to Rubinfeld and Sudan [14] (with low-error analysis by Arora and Sudan [3])

For our purposes the test of Raz and Safra is more efficient. We describe their results first and then compare its utility with the result in [3].

A plane in $\mathbb{F}^m$ is a collection of points parametrized by two variables. Specifically, given $a, b, c \in \mathbb{F}^m$ the plane $\wp_{a,b,c} = \{\wp_{a,b,c}(t_1, t_2) = a + t_1 b + t_2 c | t_1, t_2 \in \mathbb{F}\}$. Several parameterizations are possible for a given plane. We assume some canonical one is fixed for every plane, and thus the plane is equivalent to the set of points it contains. The low-degree test uses the fact that for any polynomial $p : \mathbb{F}^m \to \mathbb{F}$ of degree d, the function $p_\wp : \mathbb{F}^2 \to \mathbb{F}$ given by $p_\wp(t_1, t_2) = p(\wp(t_1, t_2))$ is a bivariate polynomial of degree d. The verifier tests this property for a function f by picking a random plane through $\mathbb{F}^m$ and verifying that there *exists* a bivariate polynomial that has good agreement with f restricted to this plane. The verifier expects an auxiliary oracle f_{planes} that gives such a bivariate polynomial for every plane. This motivates the test below.

Low-Degree Test (Plane-Point Test)
Input: A function $f : \mathbb{F}^m \to \mathbb{F}$ and an oracle f_{planes}, which for each plane in $\mathbb{F}^m$ gives a bivariate degree d polynomial.

1. Choose a random point in the space $x \in_R \mathbb{F}^m$.
2. Choose a random plane $\wp$ passing through x in $\mathbb{F}^m$.
3. Query f_{planes} on $\wp$ to obtain the polynomial $h_\wp$. Query f on x.
4. Accept iff the value of the polynomial $h_\wp$ at x agrees with $f(x)$.

It is clear that if f is a degree d polynomial, then there exists an oracle f_{planes} such that the above test accepts with probability 1. It is non-trivial to prove any converse and Raz and Safra give a strikingly strong converse. (see Theorem 7)

First some more notation. Let $\text{LDT}^{f, f_{\text{planes}}}(x, \wp)$ denote the outcome of the above test on oracle access to f and f_{planes}. Let $f, g : \mathbb{F}^m \to \mathbb{F}$ have agreement δ if $\Pr_{x \in \mathbb{F}^m}[f(x) = g(x)] = \delta$.

Theorem 7. *There exist constants c_0, c_1 such that for every positive real δ, integers m, d and field $\mathbb{F}$ satisfying $|\mathbb{F}| \geq c_0 d(m/\delta)^{c_1}$, the following holds: Fix $f : \mathbb{F}^m \to \mathbb{F}$ and f_{planes}. Let $\{P_1, \ldots, P_l\}$ be the set of all m-variate polynomials of degree d that have agreement at least $\delta/2$ with the function $f : \mathbb{F}^m \to \mathbb{F}$. Then*

$$\Pr_{x,\wp}[f(x) \notin \{P_1(x), \ldots, P_l(x)\} \text{ and } \text{LDT}^{f, f_{\text{planes}}}(x, \wp) = \mathsf{accept}] \leq \delta.$$

Remarks:

1. The actual theorem statement of Raz and Safra differs in a few aspects. The main difference being that the exact bound on the agreement probability described is different; and the fact that the claim may only say that if the low-degree test passes with probability greater than δ, then there exists some polynomial that agrees with f in some fraction of the points. The full version of this paper will include a proof of the above theorem from the statement of Raz and Safra.

2. The cubic blowup in our proof size occurs from the oracle f_{planes} which has size cubic in the size of the oracle f. A possible way to make the proof shorter would be to use an oracle for f restricted only to lines. (i.e., an analogous line-point test to the above test) The analysis of [3] does apply to such a test. However they require the field size to be (at least) a fourth power of the degree; and this results in a blowup in the proof to (at least) an eighth power. Note that the above theorem only needs a linear relationship between the degree and the field size.

3.4 Putting Them Together

As pointed out earlier a simple PCP for GapPCS can be constructed based on the low-degree test above. A proof would be an oracle f representing the polynomial and the auxiliary oracle f_{planes}. The verifier performs a low-degree test on f and then picks a random constraint C_j and verifies that C_j is satisfied by the assignment f. But the naive implementation would make k queries to the oracle f and this is too many queries. The same problem was faced by Arora et al. [1] who solved it by running a curve through the k points and then asking a new oracle f_{curves} to return the value of f restricted to this curve. This solution cuts down the number of queries to 3, but the analysis of correctness works only if $|\mathbb{F}| \geq kd$. In our case, this would impose an additional quadratic blowup in the proof size and we would like to avoid this. We do so by picking r-dimensional varieties (algebraic surfaces) that pass through the given k points. This cuts down the degree to $rk^{1/r}$. However some additional complications arise: The variety needs to pass through many random points, but not at the expense of too much randomness. We deal with these issues below.

A variety $\mathcal{V} : \mathbb{F}^r \to \mathbb{F}^m$ is a collection of m functions, $\mathcal{V} = \langle \mathcal{V}_1, \ldots, \mathcal{V}_m \rangle$, $\mathcal{V}_i : \mathbb{F}^r \to \mathbb{F}$. A variety is of degree D if all the functions $\mathcal{V}_1, \ldots, \mathcal{V}_m$ are polynomials of degree D. For a variety $\mathcal{V}$ and function $f : \mathbb{F}^m \to \mathbb{F}$, the restriction of f to $\mathcal{V}$ is the function $f|_{\mathcal{V}} : \mathbb{F}^r \to \mathbb{F}$ given by $f|_{\mathcal{V}}(a_1, \ldots, a_r) = f(\mathcal{V}(a_1, \ldots, a_r))$. Note that the restriction of a degree d polynomial $p : \mathbb{F}^m \to \mathbb{F}$ to an r-dimensional variety $\mathcal{V}$ of degree D is an r-variate polynomial of degree Dd.

Let $S \subseteq \mathbb{F}$ be of cardinality $k^{1/r}$. Let $z_1, \ldots, z_k$ be some canonical ordering of the points in S^r. Let $\mathcal{V}^{(0)}_{S,x_1,\ldots,x_k} : \mathbb{F}^r \to \mathbb{F}^m$ denote a canonical variety of degree $r|S|$ that satisfies $\mathcal{V}^{(0)}_{S,x_1,\ldots,x_k}(z_i) = x_i$ for every $i \in \{1, \ldots, k\}$. Let $Z_S : \mathbb{F}^r \to \mathbb{F}$ be the function given by $Z_S(y_1, \ldots, y_r) = \prod_{i=1}^{r} \prod_{a \in S}(y_i - a)$; i.e. $Z_S(z_i) = 0$. Let $\alpha = \langle \alpha_1, \ldots, \alpha_m \rangle \in \mathbb{F}^m$. Let $\mathcal{V}^{(1)}_{S,\alpha}$ be the variety $\langle \alpha_1 Z_S, \ldots, \alpha_m Z_S \rangle$. We will let $\mathcal{V}_{S,\alpha,x_1,\ldots,x_k}$ be the variety $\mathcal{V}^{(0)}_{S,x_1,\ldots,x_k} + \mathcal{V}^{(1)}_{S,\alpha}$. Note that if α is chosen at random, $\mathcal{V}_{S,\alpha,x_1,\ldots,x_k}(z_i) = x_i$ for $z_i \in S^r$ and $\mathcal{V}_{S,\alpha,x_1,\ldots,x_k}(z)$ is distributed uniformly over $\mathbb{F}^m$ if $z \in (\mathbb{F} - S)^r$. These varieties will replace the role of the curves of [1]. We note that Dinur et al. also use higher dimensional varieties in the proof of PCP-related theorems [7]. Their use of varieties is for purposes quite different from ours.

We are now ready to describe the MIP verifier for $\mathrm{GapPCS}_{\epsilon,m,b,q}$. (Henceforth, we shall assume that t, the number of constraints in $\mathrm{GapPCS}_{\epsilon,m,b,q}$ instance is at most q^{2m}. In fact, for our reduction from SAT (Lemma 6), t is exactly equal to q^m.)

MIP Verifier$^{f,f_{\text{planes}},f_{\text{varieties}}}(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$.
Notation: r is a parameter to be specified. Let $S \subseteq \mathbb{F}$ be such that $|S| = k^{1/r}$.

1. Pick $a, b, c \in \mathbb{F}^m$ and $z \in (\mathbb{F} - S)^r$ at random.
2. Let $\wp = \wp_{a,b,c}$. Use b, c to compute $j \in \{1, \ldots, t\}$ at random (i.e., j is fixed given b, c, but is distributed uniformly when b and c are random.) Compute α such that $\mathcal{V}(z) = a$ for $\mathcal{V} = \mathcal{V}_{S,\alpha,x_1^{(j)},\ldots,x_k^{(j)}}$.

3. Query $f(a)$, $f_{\text{planes}}(\wp)$ and $f_{\text{varieties}}(\mathcal{V})$.
   Let $g = f_{\text{planes}}(\wp)$ and $h = f_{\text{varieties}}(\mathcal{V})$.
4. Accept if all the conditions below are true:
   (a) g and f agree at a.
   (b) h and f agree at a.
   (c) A_j accepts the inputs $h(z_1), \ldots, h(z_k)$.

Complexity: Clearly the verifier V makes exactly 3 queries. Also, exactly $3m \log q + r \log q$ random bits are used by the verifier. The answer sizes are at most $O((drk^{1/r} + r)^r \log q)$ bits.

Now to prove the correctness of the verifier. Clearly, if the input instance is a YES instance then there exists a polynomial P of degree d that satisfies all the constraints of the input instance. Choosing $f = P$ and constructing f_{planes} and $f_{\text{varieties}}$ to be restrictions of P to the respective planes and varieties, we notice that the MIP verifier accepts with probability one. We now bound the soundness of the verifier.

Claim 8. *Let δ be any constant that satisfies the conditions of Theorem 7 and $\delta \geq 4\sqrt{\frac{d}{q}}$ where $q = |\mathbb{F}|$. Then the soundness of the* MIP *Verifier is at most $\delta + 4\epsilon/\delta + 4rk^{\frac{1}{r}}d/\delta(q - k^{\frac{1}{r}})$.*

Proof. Let $P_1, \ldots, P_l$ be all the polynomials of degree d that have agreement at least $\delta/2$ with f. (Note $l \leq 4/\delta$ since $\delta/2 \geq 2\sqrt{d/q}$) Now suppose, the MIP Verifier had accepted a NO instance. Then one of the following events must have taken place.
Event 1: $f(a) \notin \{P_1(a), \ldots, P_l(a)\}$ and $\mathrm{LDT}^{f, f_{\text{planes}}}(a, \wp) = \texttt{accept}$.
We have from Theorem 7, that Event 1 could have happened with probability at most δ.
Event 2: There exists an $i \in \{1, \ldots, l\}$, such that constraint C_j is satisfiable with respect to polynomial P_i. (i.e., $A_j(P_i(x_1^{(j)}), \ldots, P_i(x_k^{(j)})) = 0$).
As the input instance is a NO instance of $\mathrm{GapPCS}_{\epsilon,m,b,q}$, this events happens with probability at most $l\epsilon \leq 4\epsilon/\delta$.
Event 3: For all $i \in \{1, \ldots, p\}$, $P_i|_{\mathcal{V}} \neq h$, but the value of h at a is contained in $\{P_1(a), \ldots, P_l(a)\}$.
To bound the probability of this event happening, we reinterpret the randomness of the MIP verifier. First pick $b, c, \alpha \in \mathbb{F}^m$. From this we generate the constraint C_j and this defines the variety $\mathcal{V} = \mathcal{V}_{S,\alpha,x_1^{(j)},\ldots,x_k^{(j)}}$. Now we pick $z \in (\mathbb{F} - S)^r$ at random and this defines $a = \mathcal{V}(z)$. We can bound the probability of the event in consideration after we have chosen $\mathcal{V}$, as purely a function of the random variable z as follows. Fix any i and $\mathcal{V}$ such that $P_i|_{\mathcal{V}} \neq h$. Note that the value of h at a equals $h(z)$ (by definition. of a, z and $\mathcal{V}$). Further $P_i(a) = P_i|_{\mathcal{V}}(z)$. But z is chosen at random from $(\mathbb{F} - S)^r$. By the Schwartz-Zippel lemma, the probability of agreement on this domain is at most $rk^{1/r}d/(|\mathbb{F}| - |S|)$. Using the union bound over the i's we get that this event happens with probability at most $lrk^{1/r}d/(|\mathbb{F}| - |S|) \leq 4rk^{\frac{1}{r}}d/\delta(q - k^{\frac{1}{r}})$.

We thus have that the probability of the verifier accepting a NO instance is at most $\delta + 4\epsilon/\delta + 4rk^{\frac{1}{r}}d/\delta(q - k^{\frac{1}{r}})$. □

We can now complete the construction of a 3-prover MIP for SAT and give the proof of Lemma 2.

Proof (of Lemma 2). Choose $\delta = \frac{\mu}{3}$. Let c_0, c_1 be the constants that appear in Theorem 7. Choose $\varepsilon' = \varepsilon/2$ where ε is the soundness of the MIP, we wish to prove. Choose $\epsilon = \min\{\delta\mu/12, \varepsilon'/3(9+c_1)\}$. Let n be the size of the SAT instance. Let $m = \epsilon \log n / \log\log n$, $b = (\log n)^{3+\frac{1}{\epsilon}}$ and $q = (\log n)^{9+c_1+\frac{1}{\epsilon}}$. Note that this choice of parameters satisfies the requirements of Lemma 6. Hence, SAT reduces to $\mathrm{GapPCS}_{\epsilon,m,b,q}$ under length preserving reductions. Combining this reduction with the MIP verifier for GapPCS, we have a MIP verifier for SAT. Also δ satisfies the requirements of Claim 8. Thus, this MIP verifier has soundness as given by Claim 8. Setting $r = \frac{1}{\epsilon}$, we have that for sufficiently large n, $4rk^{\frac{1}{r}}d/\delta(q - k^{\frac{1}{r}}) \leq 8rk^{\frac{1}{r}}d/q\delta \leq \mu/3$. Hence, the soundness of the MIP verifier is at most $\delta + 4\epsilon/\delta + \mu/3 \leq \mu$. The randomness used is exactly $3m \log q + r \log q$ which with the present choice of parameters is $(3+\varepsilon')\log n + \mathrm{poly}\log n \leq (3+\varepsilon)\log n$. The answer sizes are clearly $\mathrm{poly}\log n$. Thus, $\mathrm{SAT} \in \mathrm{MIP}_{1,\frac{1}{2}+\mu}[(3+\varepsilon)\log n, \mathrm{poly}\log n]$. □

4 Constant Query Inner Verifier for MIPs

In this section, we truncate the recursion by constructing a constant query "inner verifier" for a p-prover interactive proof system. An inner verifier is a subroutine designed to simplify the task of an MIP verifier. Say an MIP verifier V_{out}, on input x and random string R, generated queries $q_1, \ldots, q_p$ and a linear sized circuit C. In the standard protocol the verifier would send query q_i to prover Π_i and receive some answer a_i. The verifier accepts if $C(a_1, \ldots, a_p) = \mathtt{true}$. An inner verifier reduces the answer size complexity of this protocol by accessing oracles $A_1, \ldots, A_p$, which are supposedly encodings of the responses $a_1, \ldots, a_p$, and an auxiliary oracle B, and probabilistically verifying that the A_i's really correspond to some commitment to strings $a_1, \ldots, a_p$ that satisfy the circuit C. The hope is to get the inner verifier to do all this with very few queries to the oracles $A_1, \ldots, A_p$ and B and we do so with one (bit) query each to the A_i's and seven queries to B. For encoding the responses $a_1, \ldots, a_p$, we use the *long code* of Bellare et al. [4]. We then adapt the techniques of Håstad [9, 10] to develop and analyze a protocol for the inner verifier.

Let $\mathcal{A} = \{+1, -1\}^a$ and $\mathcal{B} = \{(a_1, \ldots, a_p) | C(a_1, \ldots, a_p) = -1\}$. Let π_i be the projection function $\pi_i : \mathcal{B} \to \mathcal{A}$ which maps $(a_1, \ldots, a_p)$ to a_i. By abuse of notation, for $\beta \subseteq \mathcal{B}$, let $\pi_i(\beta)$ denote $\{\pi_i(x) | x \in \beta\}$. Queries to the oracle A_i will be functions $f : \mathcal{A} \to \{+1, -1\}$. Queries to the oracle B will be functions $g : \mathcal{B} \to \{+1, -1\}$. The inner verifier expects the oracles to provide the long codes of the strings $a_1, \ldots, a_p$, i.e., $A_i(f) = f(a_i)$ and $B(g) = g(a_1, \ldots, a_p)$. Of course, we can not assume these properties; they need to be verified explicitly by the inner verifier. We will assume however that the tables are "folded", i.e., $A_i(f) = -A_i(-f)$ and $B(g) = -B(-g)$ for every i, f, g. (This is implemented by issuing only one of the queries f or $-f$ for every f and inferring the other value, if needed by complementing it.) We are now ready to specify the inner verifier.

$\mathbf{V}_{\mathrm{inner}}{}^{A_1,\ldots,A_p,B}(\mathcal{A}, \mathcal{B}, \pi_1, \ldots, \pi_p)$.

1. For each each $i \in \{1, \ldots, p\}$, choose $f_i : \mathcal{A} \to \{+1, -1\}$ at random.
2. Choose $f, g_1, g_2, h_1, h_2 : \mathcal{B} \to \{+1, -1\}$ at random and independently.

3. Let $g = f\,(g_1 \wedge g_2)\,(\Pi f_i \circ \pi_i))$ and $h = f\,(h_1 \wedge h_2)\,(\Pi f_i \circ \pi_i))$.
4. Read the following bits from the oracles $A_1, \ldots, A_p, B$
   $y_i = A_i(f_i)$, for each $i \in \{1, \ldots, p\}$.
   $w = B(f)$.
   $u_1 = B(g_1);\ u_2 = B(g_2);\ v_1 = B(h_1);\ v_2 = B(h_2)$
   $z_1 = B(g);\ z_2 = B(h)$
5. Accept iff $w \prod_{i=1}^{p} y_i = (u_1 \wedge u_2)z_1 = (v_1 \wedge v_2)z_2$

It is clear that if $a_1, \ldots, a_p$ are such that $C(a_1, \ldots, a_p) = -1$ and for every i and f, $A_i(f) = f(a_i)$ and for every g, $B(g) = g(a_1, \ldots, a_p)$, then the inner verifier accepts with probability one. The following lemma gives a soundness condition for the inner verifier, by showing that if the acceptance probability of the inner verifier is sufficiently high then the oracles $A_1, \ldots, A_p$ are non-trivially close to the encoding of strings $a_1, \ldots, a_p$ that satisfy $C(a_1, \ldots, a_p) = -1$. The proof uses, by now standard, Fourier analysis.

Note that the oracle A_i can be viewed as a function mapping the set of functions $\{\mathcal{A} \to \{+1, -1\}\}$ to the reals. Let the inner product of two oracles A and A' be defined as $\langle A, A' \rangle = 2^{-|\mathcal{A}|} \sum_f A(f)A'(f)$. For $\alpha \subseteq \mathcal{A}$, let $\chi_\alpha(f) = \prod_{a \in \alpha} f(a)$. Then the χ_α's give an orthonormal basis for the space of oracles A. This allows us to express $A(\cdot) = \sum_\alpha \hat{A}_\alpha \chi_\alpha(\cdot)$, where $\hat{A}_\alpha = \langle A, \chi_\alpha \rangle$ are the Fourier coefficients of A. In what follows, we let $\hat{A}_{i,\alpha}$ denote the α^{th} Fourier coefficient of the table A_i. Similarly one can define a basis for the space of oracles B and the Fourier coefficients of any one oracle.

Our next claim lays out the precise soundness condition in terms of the Fourier coefficients of the oracles $A_1, \ldots, A_p$.

Claim 9. *For every $\epsilon > 0$, there exists a $\delta > 0$ such that if ${V_{\mathrm{inner}}}^{A_1,\ldots,A_p,B}(\mathcal{A}, \mathcal{B}, \pi_1, \ldots, \pi_p)$ accepts with probability at least $\frac{1}{2} + \epsilon$, then there exist $a_1, \ldots, a_p \in \mathcal{A}$ such that $C(a_1, \ldots, a_p) = -1$ and $|\hat{A}_{i,\{a_i\}}| \geq \delta$ for every $i \in \{1, \ldots, p\}$.*

There is a natural way to compose a p-prover MIP verifier V_{out} with an inner verifier such as V_{inner} above so as to preserve perfect completeness. The number of queries issued by the composed verifier is exactly that of the inner verifier. The randomness is the sum of the randomness. The analysis of the soundness of such a verifier is also standard and in particular shows that if the composed verifier accepts with probability $\frac{1}{2} + 2\epsilon$, then there exist provers $\Pi_1, \ldots, \Pi_p$ such that V_{out} accepts them with probability at least $\epsilon \cdot \delta^{2p}$, where δ is from Claim 9 above. Thus we get a proof of Lemma 4.

5 Scope for Further Improvements

The following are a few approaches which would further reduce the size-query complexity in the construction of PCPs described in this paper.

1. An improved low-error analysis of the low-degree test of Rubinfeld and Sudan [14] in the case when the field size is linear in the degree of the polynomial. (It is to be noted that the current best analysis [3] requires the field size to be at least a fourth power of the degree.) Such an analysis would reduce the proof blowup to nearly quadratic.

2. It is known that for every $\epsilon, \delta > 0$, $\mathrm{MIP}_{1,\epsilon}[1, 0, n] \subseteq \mathrm{PCP}_{1-\delta, \frac{1}{2}}[c \log n, 3]$ from the results of Håstad [10]. Traditionally, results of this nature have led to the construction of inner verifiers for p-prover MIPs and thus showing that for every $\delta > 0$ and p there exists $\epsilon > 0$ and c such that

$$\mathrm{MIP}_{1,\epsilon}[p, r, a] \subseteq \mathrm{PCP}_{1-\delta, \frac{1}{2}}[r + c \log a, p + 3] \ .$$

Proving a result of this nature would reduce the query complexity of the small PCPs constructed in this paper to 6.

References

1. ARORA, S., LUND, C., MOTWANI, R., SUDAN, M., AND SZEGEDY, M. Proof verification and the hardness of approximation problems. *Journal of the ACM 45*, 3 (May 1998), 501–555.
2. ARORA, S., AND SAFRA, S. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM 45*, 1 (Jan. 1998), 70–122.
3. ARORA, S., AND SUDAN, M. Improved low degree testing and its applications. In *Proc. 29th ACM Symp. on Theory of Computing* (El Paso, Texas, 4–6 May 1997), pp. 485–495.
4. BELLARE, M., GOLDREICH, O., AND SUDAN, M. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal of Computing 27*, 3 (June 1998), 804–915.
5. BELLARE, M., GOLDWASSER, S., LUND, C., AND RUSSELL, A. Efficient probabilistically checkable proofs and applications to approximation. In *Proc. 25th ACM Symp. on Theory of Computing* (San Diego, California, 16–18 May 1993), pp. 294–304.
6. COOK, S. A. Short propositional formulas represent nondeterministic computations. *Information Processing Letters 26*, 5 (11 Jan. 1988), 269–270.
7. DINUR, I., FISCHER, E., KINDLER, G., RAZ, R., AND SAFRA, S. PCP characterizations of NP: Towards a polynomially-small error-probability. In *Proc. 31th ACM Symp. on Theory of Computing* (Atlanta, Georgia, 1–4 May 1999), pp. 29–40.
8. FRIEDL, K., AND SUDAN, M. Some improvements to total degree tests. In *Proc. 3rd Israel Symposium on Theoretical and Computing Systems* (1995).
9. HÅSTAD, J. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proc. 37nd IEEE Symp. on Foundations of Comp. Science* (Burlington, Vermont, 14–16 Oct. 1996), pp. 627–636.
10. HÅSTAD, J. Some optimal inapproximability results. In *Proc. 29th ACM Symp. on Theory of Computing* (El Paso, Texas, 4–6 May 1997), pp. 1–10.
11. LUND, C., FORTNOW, L., KARLOFF, H., AND NISAN, N. Algebraic methods for interactive proof systems. In *Proc. 31st IEEE Symp. on Foundations of Comp. Science* (St. Louis, Missouri, 22–24 Oct. 1990), pp. 2–10.
12. POLISHCHUK, A., AND SPIELMAN, D. A. Nearly-linear size holographic proofs. In *Proc. 26th ACM Symp. on Theory of Computing* (Montréal, Québec, Canada, 23–25 May 1994), pp. 194–203.
13. RAZ, R., AND SAFRA, S. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proc. 29th ACM Symp. on Theory of Computing* (El Paso, Texas, 4–6 May 1997), pp. 475–484.
14. RUBINFELD, R., AND SUDAN, M. Robust characterizations of polynomials with applications to program testing. *SIAM Journal of Computing 25*, 2 (Apr. 1996), 252–271.
15. SUDAN, M. *Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems*. PhD thesis, University of California, Berkeley, Oct. 1992.
16. SZEGEDY, M. Many-valued logics and holographic proofs. In *Automata, Languages and Programming, 26st International Colloquium* (Prague, Czech Republic, 11–15 July 1999), J. Wiedermann, P. van Emde Boas, and M. Nielsen, Eds., vol. 1644 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 676–686.