Electronic Colloquium on Computational Complexity, Report No. 48 (1998)

ECCC

Approximating-CVP to Within Almost-Polynomial Factors is NP-hard

I. Dinur * G. Kindler * S. Safra *

Abstract

This paper shows the closest vector in a lattice to be NP-hard to approximate to within any factor up to $2^{(\log n)^{1-\epsilon}}$ where $\epsilon = (\log \log n)^{-\alpha}$ for any constant $\alpha < \frac{1}{2}$.

Introduction

Background

A lattice $L = L(v_1, .., v_n)$, for vectors $v_1, .., v_n \in R^n$ is the set of all integer linear combinations of $v_1, .., v_n$, that is, $L = \{\sum a_i v_i \mid a_i \in Z\}$. Given a lattice L and an arbitrary vector y, the Closest Vector Problem (CVP) is to find a vector in L closest to y. The Shortest Vector Problem (SVP) is the homogeneous analog of CVP, i.e. finding the shortest non-zero vector in L.

These lattice problems have been introduced in the previous century, and have been studied since. Minkowsky and Dirichlet tried, with little success, to come up with lattice approximation algorithms. It was much later that the lattice reduction algorithm was presented by Lenstra, Lenstra and Lovász [L³82] , achieving a polynomial-time algorithm approximating the Shortest Lattice Vector to within an exponential factor $2^{\frac{dim}{2}}$. Babai [Bab86] applied L³'s methods to present an algorithm that approximates CVP to within a similar factor. Schnorr [Sch85] improved on L³'s technique, reducing the factor of approximation to $(1+\epsilon)^n$, for any constant $\epsilon > 0$, for both CVP and SVP. These positive approximation results are still quite weak, achieving only extremely large (exponential) factors. The question naturally arises: What are the factors of approximation to within which these problems can be approximated in polynomial time?

Interest in lattice problems has been recently renewed due to a result of Ajtai [Ajt96], showing a reduction, from the worst-case of a restricted version of SVP, to the average-case of the same problem. Finding a problem whose average case complexity is known to be as hard as the worst-case of some other problem is quite an achievement by itself from complexity theoretic perspective. Yet such a result has significant cryptographic applications, as shown in [AD96]. Showing

*Tel-Aviv University

ISSN 1433-8092

NP-hardness for that specific restriction of SVP – although unlikely as discussed below – would imply a cryptosystem whose breaking would imply P=NP.

CVP was shown to be NP-hard for any l_p norm in [vEB81], where it was also conjectured that SVP is NP-hard. Arora et al. [ABSS93] utilized the PCP characterization of NP to show that CVP is NP-hard to approximate to within any constant, and quasi-NP-hard to approximate to within $2^{(\log n)^{1-\epsilon}}$ for any constant $\epsilon > 0$.

As to SVP, only recently, [Ajt97] showed a randomized reduction from the NP-complete problem Subset-Sum to SVP. This has been improved [CN97], showing approximation hardness for some small factor $(1 + \frac{1}{dim^{-\epsilon}})$. Very recently [Mic98] has significantly strengthened Ajtai's result, showing SVP hard to approximate to within some constant factor. The proof in [Mic98] relies on the PCP characterization of NP and is carried out via a reduction from gap-CVP (shown NP-hard for any constant gap in [ABSS93]). Using gap-CVP allows, in addition to the significant improvement in the gap, a major simplification of the main technical lemma from [Ajt97]. Better hardness results for gap-CVP may result in hardness results for gap-SVP for larger gaps.

So far there is still a huge gap between the positive results, approximating these problems to within exponential factors, and the above hardness results. Nevertheless, some other results provide a discouraging indication for improving the hardness result beyond a certain factor. [LLS90] showed that approximating CVP to within $dim^{1.5}$ is in co-NP, and recently [GG] showed that approximating both SVP and CVP to within $\sqrt{dim}$ is in NP$\cap$ co-AM. Hence showing the unlikelihood of any of these problems to be NP-hard.

The strongest hardness result likely to be true for these problems hence, is that they are hard to approximate to within a constant power of the dimension. The proof of [ABSS93] utilizes amplification techniques that cause the size of the instance, hence the dimension, to grow faster than the factor for which hardness of approximation is obtained. It is therefore unlikely that using this technique, even if allowing a super-polynomial blow-up, one can obtain such strong results. It seems that it will always be the case that the factor for which hardness of approximation is proven never reaches beyond the barrier of $2^{(\log dim)^{1-\epsilon}}$ for any constant $\epsilon > 0$.

Our Results

This paper improves on [ABSS93] in two ways. First, it goes beyond the barrier of constant ϵ, proving CVP hard to approximate to within a factor of $2^{(\log dim)^{1-\epsilon}}$ where ϵ, rather than being an arbitrarily small constant, is $(\log\log dim)^{-\alpha}$ for any $\alpha < \frac{1}{2}$. This is the first hardness result for CVP reaching beyond the above mentioned barrier. Furthermore, our result shows approximating CVP is NP-hard for large factors, compared to the previously known *quasi* NP hardness .

The best known $\mathcal{PCP}$ characterization of NP (and even the conjectured one) seems inappropriate in order to show hardness of approximating CVP to within large factors. We introduce, for that purpose, a new characterization of NP,

$\mathcal{SSAT}$, and prove the hardness of gap-CVP using this new characterization. The $\mathcal{SSAT}$ characterization is different from the $\mathcal{PCP}$ characterization, despite relying on similar techniques in its proof.

Let SAT[F] be the following problem: An instance of SAT[F] is a set of tests (Boolean functions) over a common set of variables that range over a finite range F. An instance is accepted if each *test* can be assigned a satisfying value, such that the assignments to the tests are everywhere consistent, that is, each variable is given the same value by the assignments of all the tests depending on it.

The gap version of this problem, Super-SAT ($\mathcal{SSAT}$ for short), is as follows: $\mathcal{SSAT}$ is the same as SAT[F] except not all non-satisfiable instances must be rejected. We generalize the notion of assignment to that of super-assignment[1] – formal linear combinations of assignments with integer coefficients – and modify the acceptance condition accordingly. If there is a super-assignment to the tests, of norm smaller than g, which is everywhere consistent (in a sense similar to that described above), then that instance is not necessarily rejected (any outcome is acceptable).

We show (theorem 1) that solving this problem is NP-hard for $g \leq 2^{(\log n)^{1-\epsilon}}$ where $\epsilon = (\log\log n)^{-\alpha}$ for any positive constant $\alpha < \frac{1}{2}$. (n denotes, as usual, the size of the instance).

Improving the hardness of approximation factor to a constant power of n, namely where $g = n^{\epsilon}$ for some constant ϵ (conjecture 2), would imply CVP to be hard to approximate to within a constant power of the dimension.

We also show that our proof works for lattices over finite-fields (instead of Z). This in particular implies NP-hardness for approximating the nearest-codeword [ABSS93] within factor g.

Structure of the Paper

Section 1 presents the new characterization of NP, $\mathcal{SSAT}$. It starts by formally defining $\mathcal{SSAT}$ and then states theorem 1 which asserts that it is NP-hard for large factors of approximation. Section 2 gives a naive hardness proof (via a super-polynomial construction) which will be used as a basis for the complete proof. Section 3 covers the main part of the proof of theorem 1. The proof relies on a consistency lemma whose proof is shown in section 5. In section 4 we show the simple reduction from $\mathcal{SSAT}$ to CVP, and sketch the extension of the result for finite fields. Finally, in section 5 we return to the proof of the consistency lemma.

1 Super-SAT - $\mathcal{SSAT}$

In this section we define a new characterization of NP, named $\mathcal{SSAT}$. Let us begin by defining SAT[$\mathcal{F}$], which is actually SAT from a consistency point of view. An instance of SAT[$\mathcal{F}$] is a set of tests (Boolean functions) over a common

[1] A super-assignment can be thought of as a super-position of assignments.

set of variables that range over a field $\mathcal{F}$ of $\leq$ polynomial size. An assignment to a *test* maps to each test one of the test's satisfying values. The assignments to each test ψ have a different range, denoted $\mathcal{R}_\psi$. We denote $\mathcal{R} \stackrel{def}{=} \bigcup \mathcal{R}_\psi$. An instance is accepted iff there is an assignment to the tests that is everywhere consistent, that is, each variable is given the same value by the assignments to all tests that depend on it. It is easy to see that this problem is NP-complete.

$\mathcal{SSAT}$ is a gap variant of this problem, obtained by allowing certain non-satisfiable instances to be accepted. The gap of $\mathcal{SSAT}$ is no longer the fractional gap of the $\mathcal{PCP}$ (i.e. finding the maximal fraction of satisfiable tests) but of a different nature.

We will introduce a new notion of super-assignment to the tests, that is, a formal linear combination of assignments. We will allow acceptance of non-satisfiable instances that have 'short' and 'consistent' super-assignments.

Definition 1 (Super-Assignment to Tests) *A super-assignment is a function M mapping to each $\psi \in \Psi$ a value from $Z^{\mathcal{R}_\psi}$. $M(\psi)$ is a vector of integer coefficients, one for each possible value $r \in \mathcal{R}_\psi$. Denote by $M(\psi)[r]$ the r^{th} coordinate of $M(\psi)$.*

M is said to be *non-trivial* if $\forall \psi \in \Psi$, $||M(\psi)|| > 0$, where $||M(\psi)||$ denotes l_1 norm. Note that $||M(\psi)|| > 0$ means $||M(\psi)|| \geq 1$ since all the entries are integers. For a test ψ we think of all the values receiving non-zero coefficients in $M(\psi)$ as being simultaneously 'assigned' to ψ. The non-triviality requirement means that each test must be assigned at least one value.

A *natural assignment* (an assignment in the usual sense) is identified with a super-assignment where ψ is assigned a unit vector with a 1 in the corresponding coordinate.

Definition 2 (Norm of a Super-Assignment) *The norm of a super-assignment M is $||M|| = \frac{1}{|\Psi|}\sum_{\psi\in\Psi} ||M(\psi)||$.*

The norm of a natural super-assignment is 1. The gap of $\mathcal{SSAT}$ will be formulated in terms of the norm of the minimal super-assignment that maintains consistency. In the SAT[$\mathcal{F}$] problem a satisfying assignment is one that is everywhere consistent: For every pair of tests with a mutual variable, the assignments to the tests, restricted to the variable, are equal. We extend this notion to super-assignments by defining the projection of a super-assignment to a test onto each of its variables. Consistency between tests will amount to equality of projections on mutual variables.

Definition 3 (Projection) *A natural assignment r to a test induces an assignment to each variable x, denoted $r|_x$. Similarly, a super-assignment, induces a super-assignment on a variable by taking the formal linear combination of the assignments' restrictions.*

Let M be a super-assignment to the tests. We define the projection of $M(\psi)$ on a variable x of ψ, $\pi_x(M(\psi)) \in Z^{|\mathcal{F}|}$, in the natural way:

$$\forall f \in \mathcal{F}: \qquad \pi_x(M(\psi))[f] \stackrel{def}{=} \sum_{r\in\mathcal{R},\ r|_x = f} M(\psi)[r]$$

We shall now proceed to define the notion of consistency between tests. If the projections of two tests on a mutual variable x are equal (in other words, they both give x the same super-assignment), we say that the super-assignments of the tests are consistent (match).

Definition 4 (Consistency) *Let M be a super-assignment to the tests in Ψ. M is consistent if for every pair of tests ψ_i and ψ_j with a mutual variable x,*

$$\pi_x(M(\psi_i)) = \pi_x(M(\psi_j))$$

We can now define the $\mathcal{SSAT}$ problem.

$\mathcal{SSAT}$**:** An instance of $\mathcal{SSAT}$ with parameter g consists of a system of tests (Boolean functions) $\Psi = \{\psi_1, ..., \psi_m\}$ over common variables from $\mathcal{V} = \{x_1, .., x_n\}$ ranging over $\mathcal{F}$. Each test depends on exactly two variables. The problem is to determine if the instance falls into one of the following two cases,

Yes: There is a consistent natural assignment for Ψ.

No: No non-trivial consistent super-assignment is of norm $\leq g$.

Theorem 1 ($\mathcal{SSAT}$ **Theorem)** *$\mathcal{SSAT}$ is NP-hard for $g = 2^{(\log n)^{1-\epsilon}}$ where $\epsilon = (\log\log n)^{-c}$ for any $c < \frac{1}{2}$.*

We suggest a stronger conjecture. If true, it would imply that CVP is hard to approximate to within a *constant power* of the dimension.

Conjecture 2 *$\mathcal{SSAT}$ is hard for $g = n^c$ for some constant $c < \frac{1}{2}$.*

The $\mathcal{SSAT}$ theorem (theorem 1) can be viewed as an extension of Cook's theorem [Coo71] in the following way. An algorithm solving $\mathcal{SSAT}$ is required to accept if the test system is satisfiable. However, the algorithm is allowed to accept non-satisfiable instances that have a consistent super-assignment of norm $\leq g$. It must only reject when any consistent super-assignment for Ψ is of norm $> g$. We are, in fact, adding slackness between the acceptance and rejection cases.

The Depend Parameter. In the above formulation, the $\mathcal{SSAT}$ tests depend on exactly two variables. Consider the following modification. Let each test depend on a *polynomial* number of variables, as long as the number of satisfying values per each test is polynomially bounded. The reduction from this modification to the above formulation is simple:

- Add one new variable for every test. The variable for ψ will range over the satisfying values of ψ, $\mathcal{R}_\psi$.
- Replace the tests with a test for every pair of (test,variable) verifying that the values match.

The range of the new variables is still polynomial because of the restriction on the number of satisfying assignments to each test. The $\mathcal{SSAT}$ gap property is maintained, and every new test depends on exactly two variables. Note that this simple transformation in a $\mathcal{PCP}$ test system will severely increase the error probability.

Proving the NP-hardness of $\mathcal{SSAT}$, we construct a test system where the depend is much larger than 2. We show, in exchange, that the number of satisfying assignments for every test is polynomial. Such a test system can then be translated to a $\mathcal{SSAT}$ test system by the above transformation.

2 The Initial Construction

In this section we give a 'naive' hardness proof for $\mathcal{SSAT}$, via a super-polynomial construction. This is done via a reduction from $\mathcal{PCP}$ to $\mathcal{SSAT}$, that has super-polynomial variable range. This proof will be used as a basis for the final construction. NP-hardness (with a polynomial range for every test) will be proven in the following section (section 3).
Our starting point is the $\mathcal{PCP}$ characterization of NP. We can actually rely on any of the known $\mathcal{PCP}$ theorems ([AS92, ALM$^+$92, RS96, DFK$^+$98]) since the only property we need is a constant error probability.

Theorem 3 (PCP Theorem [DFK$^+$98]) *Given a system of tests $\Phi = \{\phi_1, ..., \phi_n\}$ over variables $\mathcal{V} = \{x_1, .., x_{n'}\}$ such that each test depends on $O(1)$ variables, and each variable ranges over a field $\mathcal{F}$ where $|\mathcal{F}| = O(2^{(\log n)^{1-\epsilon}})$ for any constant $\epsilon > 0$. It is NP-hard to distinguish between the following two cases:*

Yes: There is an assignment to the variables such that all $\phi_1, ..., \phi_n$ are satisfied.

No: No assignment can satisfy more than $\frac{2}{|\mathcal{F}|}$ fraction of the ϕ_i's.

We shall construct a new test system Ψ with an $\mathcal{SSAT}$ gap, based on Φ, and show a reduction from the $\mathcal{PCP}$ instance to the $\mathcal{SSAT}$ instance.

2.1 Cancellations

One may wonder if perhaps Φ already possesses the $\mathcal{SSAT}$ gap property. Suppose that for every 'no' instance of Φ, all of the consistent super-assignments are of norm $> g$. We could then take Φ for our final construction. Unfortunately, this is not necessarily the case. The reason is that we verify consistency of super-assignments by comparing projections of tests on mutual variables. There is a possibility that the super-assignment somehow locally cancels values on each variable and hence yields false consistency. (For example, let $(1,3), (3,3), (3,1)$ be the satisfying assignments of $\varphi(x, y)$; then the super-assignment $1 \cdot (1,3) - 1 \cdot (3,3) + 1 \cdot (3,1)$ projects to a natural assignment of 1 on both x and y although $(1,1)$ doesn't even satisfy φ).

Since we do not know the exact structure of the assignments to the tests, the cancellation problem cannot be ruled out. To solve this cancellation problem, we add auxiliary variables that serve as an error correcting code. We will show that for every test only a negligible fraction of its variables can be canceled in the above sense, and deduce that a consistent super-assignment must in fact be globally consistent.

2.2 Low Degree Functions

We begin with a few basic definitions relating to low degree functions. We will use these definitions shortly to construct the naive test system.

Definition 5 ($[r,d]$-LDF) *Let $\mathcal{F} = \mathcal{Z}_p$ for some prime p, and let $\mathcal{D} = \mathcal{F}^d$ be a domain. A low-degree function (LDF for short) with parameters $[r,d]$ is a polynomial function over $\mathcal{F}^d$ whose degree in each variable is no more than r.*

Denote by $LDF_{r,d}$ the set of all $[r,d]$-LDFs. We frequently use the following property of LDFs,

Proposition 1 *Let f and g be two distinct $[r,d]$-LDFs. The fraction of points $x \in \mathcal{D}$ on which $f(x) = g(x)$ is $\leq \frac{rd}{|\mathcal{F}|}$.*

2.3 The Low Degree Extension

Let $x_1, .., x_n$ be Φ's variables. We embed them in a larger domain (as done in numerous $\mathcal{PCP}$ papers): We view the variables as points of a set $\mathcal{H}^d$ (where $\mathcal{H}$ and d are chosen so that $|\mathcal{H}|^d = n$). We then extend the set $\mathcal{H}^d$ to a domain $\mathcal{F}^d \supset \mathcal{H}^d$ by taking $\mathcal{F} \supset \mathcal{H}$ to be a field of size $|\mathcal{F}| = |\mathcal{H}|^{O(1)}$. We have a variable for each point in the domain $\mathcal{F}^d$. The points of the extended domain $\mathcal{F}^d$ serve as the auxiliary variables that help eliminate the cancellation problem.

Satisfying assignments to the new variables would be *extensions* of satisfying assignments to the original variables in the following sense. Let $A : \mathcal{H}^d \to \mathcal{F}$ be an assignment to the original variables. There exists exactly one polynomial $\hat{A} : \mathcal{F}^d \to \mathcal{F}$ such that $\hat{A}$ *extends* A, and $\hat{A}$ has degree h in each of its variables. $\hat{A}$ is called the h degree extension of A in $\mathcal{F}$.

2.4 The Construction

We now proceed to describe the naive construction. This construction possesses the desired $\mathcal{SSAT}$ gap but it inflates the size of the generated instance.

Parameters. Denote the size of the original $\mathcal{PCP}$ instance by n. Let $c < \frac{1}{2}$ be arbitrary. We choose $\epsilon = (\log\log n)^{-c}$, $|\mathcal{H}| = 2^{O((\log n)^{1-2\epsilon})}$, $|\mathcal{F}| = |\mathcal{H}|^{O(1)}$ and $d \approx (\log n)^{2\epsilon}$. These parameters will be fixed throughout the rest of the paper.

Variables. We shall have one variable for every point $x \in \mathcal{D} \overset{def}{=} \mathcal{F}^d$. The original variables of Φ are identified with the subset $\mathcal{H}^d \subset \mathcal{F}^d$ of the new variables.

Tests. The tests of Ψ will correspond to affine subspaces (cubes) of $\mathcal{D}$. We define a *t-cube* of $\mathcal{D}$ to be an affine subspace of $\mathcal{D}$ of dimension t. Assume w.l.o.g. that all of the tests in Φ depend on exactly $D = O(1)$ variables. For a test $\varphi \in \Phi$ that depends on $x_{i_1}, .., x_{i_D}$, define the D-tuple $\tau_\varphi \overset{def}{=} (x_{i_1}, .., x_{i_D})$.

Denote by $\mathcal{S}_{\tau_\varphi}$ the set of all $D+3$-cubes that contain the points of τ_φ. Let $\mathcal{T} = \{\tau_\varphi\}_{\varphi \in \Phi}$, and define $\mathcal{S}_\mathcal{T} = \bigcup_{\tau \in \mathcal{T}} \mathcal{S}_\tau$.

For every cube $\mathcal{C} \in \mathcal{S}_{\tau_\varphi}$, Ψ has a test that depends on the variables corresponding to the points of $\mathcal{C}$. This test accepts only $[dh, D+3]$-LDFs whose restriction to the tuple points of τ_φ satisfy $\varphi \in \Phi$. We call $\mathcal{T}$ the tuple-set of the test system. Note that the tests of Ψ are determined by the tuples.

Super-Assignments. A super-assignment to a test (a cube) is, by definition, a formal linear combination of LDFs. We shall call such an object a super-polynomial. We include the explicit definition of a super-polynomial for clarity,

Definition 6 ($[r,t]$-Super-Polynomial) *An $[r,t]$-super-polynomial is a function $\mathcal{P} : LDF_{r,t} \to Z$ that assigns each $[r,t]$-LDF an integer coefficient. One may think of $\mathcal{P}$ as a vector with $|LDF_{r,t}|$ integer coordinates.*

Denote by $SLDF_{r,t}$ the set of all $[r,t]$-super-polynomials. The norm and projection of a super-polynomial are defined as in the general case for super-assignments. Consistency of super-polynomials amounts to equality of projections on each mutual point. The projection $\pi_\mathcal{C}(\mathcal{P})$ of a super-polynomial $\mathcal{P}$ on a cube $\mathcal{C}$ is naturally defined as the formal linear combination of the restrictions of the LDFs in $\mathcal{P}$ to the cube.

We shall now describe a property of super-polynomials that will help get over the cancellation problem: low-ambiguity. A point x_0 is called *ambiguous* for a super-polynomial $\mathcal{M}$, if there are two LDFs $P_1 \neq P_2$ that each have a non-zero coefficient in $\mathcal{M}$, and $P_1(x_0) = P_2(x_0)$. The ambiguous points are the only points that are candidates for cancellation. Only a negligible fraction of the points are ambiguous.

Proposition 2 (Low Ambiguity) *Let $\mathcal{P}$ be an $[r,t]$-super-polynomial of norm $\leq g$. The fraction of ambiguous points in $\mathcal{D}$ is $\leq amb(r,t,g) \overset{def}{=} \binom{g}{2}\frac{rt}{|\mathcal{F}|}$.*

We omit the simple proof of this proposition. We now know that no more than $amb(r,t,g)$ fraction of the variables of a test (points of a cube) can be canceled. Note the relation between the norm g of the super-polynomial and the bound on its ambiguity. Using the low-ambiguity property we'll be able to deduce global consistency from consistency of super-assignments, as seen in the following lemma.

This lemma is actually a special case of a more general consistency lemma (lemma 2) that will be proven in section 5.

Lemma 1 (The Naive Consistency Lemma) *Let $M : \mathcal{S}_{\mathcal{T}} \to SLDF_{r,t}$ be a super-assignment of norm $\leq g < |\mathcal{F}|^{\frac{1}{100}}$ and assume that $amb(r,t,g) \ll |\mathcal{F}|^{-\frac{1}{2}}$. If M is consistent (i.e. for every pair of cubes $\mathcal{C}_1, \mathcal{C}_2$ with a mutual point x – the projections of $M(\mathcal{C}_1)$ and $M(\mathcal{C}_2)$ on x are equal); then there exists a global super-polynomial $\mathcal{G}$ of degree h on $\mathcal{D}$ such that $||\mathcal{G}|| \leq 2g$ and*

$$\Pr_{\mathcal{C} \in \mathcal{S}_{\mathcal{T}}} [M(\mathcal{C}) = \pi_{\mathcal{C}}(\mathcal{G})] > 1 - \frac{g^{c_1}}{|\mathcal{F}|^{c_2}}$$

For an appropriate choice of $g = |\mathcal{F}|^{c_3}$, we obtain a global super-polynomial that agrees with all but a negligible fraction of the cubes in $\mathcal{S}_{\mathcal{T}}$. This is the aforementioned global consistency.

2.5 The Construction is Correct

Completeness. If Φ is totally satisfiable then there exists an LDF f over $\mathcal{F}^d$ that extends the satisfying assignment to the x_i's (the low degree extension of these values). f is of degree $|\mathcal{H}| - 1$ in each of its d variables. Its restrictions to the cubes of Ψ will supply the consistent natural assignment for Ψ.

Soundness. The naive consistency lemma (lemma 1) implies that if there is a non-trivial consistent super-assignment to the cubes whose norm is small enough, then there is a global super-polynomial $\mathcal{G}$ of low norm whose projections on most of the cubes equal their assigned super-polynomial. Consider any LDF P that appears in $\mathcal{G}$ with a non-zero coefficient. The low-ambiguity property implies that P's values appear in most of the points (i.e. aren't ambiguous or canceled). It follows easily that P's value appears in most of the cubes. Now, since for most cubes $\mathcal{C} \in \mathcal{S}_{\mathcal{T}}$, $M(\mathcal{C})$ equals the projection of $\mathcal{G}$ on $\mathcal{C}$ – we deduce that P appears in $M(\mathcal{C})$ for most $\mathcal{C} \in \mathcal{S}_{\mathcal{T}}$. Taking P's value on the points of $\mathcal{H}^d$ produces a satisfying assignment to over half of the tests in Φ. This shows that if the $\mathcal{PCP}$ instance was a 'no' instance then any consistent super-assignment for the $\mathcal{SSAT}$ instance must be of norm $\geq g$.

Size. The method of choosing parameters is as follows. For the construction to be correct we need $amb(h,t,g) = \binom{g}{2}\frac{ht}{|\mathcal{F}|}$ to be negligible. Note that g is the gap and we want it to be large. We therefore choose a large field $\mathcal{F}$ ($|\mathcal{F}| = 2^{(\log n)^{1-2\epsilon}}$) and the degree h is hence also large (recall that $h \overset{def}{=} |\mathcal{H}|$ is polynomial in $|\mathcal{F}|$).

The problem with this construction is the range of the assignments to the tests. The range of satisfying assignments for the tests must be polynomial in size. This is required for the reduction to CVP, shown in section 4 to work. However, the number of $[h,t]$-LDFs is at least $|\mathcal{F}|^h$, i.e. super-polynomial. We overcome this problem by an iterative substitution of the cubes, as will be seen in the next section.

3 The Final Construction

In the previous section we constructed a test system that possessed the $\mathcal{SSAT}$ property (a consistent super-assignment of small-norm for it, implies that the original test system was a 'yes' instance). We shall maintain this property while decreasing the range of the tests. Since every cube ranged over too many LDFs, we represent each cube by new variables that have a smaller range. This replacement procedure will be repeated several times until the final variables have a polynomial range.

3.1 Cube-Systems

An $[r,t]$-*cube-system* is a specific form of a test system. There is an underlying set of domains (copies of $\mathcal{F}^d$). The variables correspond to the points in these domains. Some points are mutual to several domains, that is the *same* variable represents these points in each of the domains. The tests in the cube-system correspond to cubes in these domains defined by a set of tuples. The satisfying assignments to the tests will be $[r,t]$-LDFs. The naive construction is an example of a $[dh, D+3]$-cube-system with one domain.

We shall show how one may transform an $[r,t]$-cube-system into an $[r',t']$-cube-system where $r' \ll r$, and $t' \approx t$. The main property of this transformation will be that a consistent assignment to the resulting cube-system induces an almost consistent assignment to the initial cube-system (preserving the norm). The exact meaning of "almost consistent" will become clear later, when we state the soundness theorem (theorem 4).

The initial cube-system Ψ. Let $\mathcal{D}_1, .., \mathcal{D}_k$ be domains that may have some mutual points. (It may be helpful to think, at first, of $k=1$ and of Ψ as being the test system from the naive construction). Let $\mathcal{T} = \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_k$, where $\mathcal{T}_i$ is a set of t-tuples in $\mathcal{D}_i$. Define, as before, $\mathcal{S}_{\mathcal{T}_i}$ to be the set of $t+3$-cubes of $\mathcal{D}_i$ that contain at least one tuple from $\mathcal{T}_i$. Let Ψ be a cube-system as follows - Ψ has a test for each cube in $\mathcal{S}_{\mathcal{T}} = \bigcup \mathcal{S}_{\mathcal{T}_i}$, and a variable for each point $x \in \bigcup_i \mathcal{D}_i$.

3.2 The b-transformation of Ψ to Ψ'.

In this section we show how to take a cube-system and 'break' the representation of each cube into many new cubes. The new cubes will range over super-polynomials of considerably lower degree (thereby decreasing their range).

Proposition 3 *Let $b > 1$. Let Ψ be an $[r,t]$-cube system. There exists (polynomially constructible) a $[bt\log_b r, t+4]$-cube-system Ψ' that "represents" Ψ.*

By "represents" we mean that consistent super-assignments to Ψ naturally translate to almost consistent super-assignments to Ψ' (with roughly the same norm) and vice versa. The exact meaning of this representation will become clear in the end of this section.

We repeat this b-transformation step iteratively, startingfrom the naive construction, until we reach super-polynomials with small enough degree and dimension (enough so that the range is polynomial). The b-transformation is thus the key step in the reduction. It is the tool that enables us to keep the entire construction polynomial in size, while attaining larger gap factors.

Embedding Extension

We will replace each cube by a new set of cubes such that the super-assignments to the new cubes have a smaller range. We first extend every cube $\mathcal{C}$ to a larger domain $\mathcal{D}_{\mathcal{C}}$. Sub-cubes of this domain will become the new tests of the new cube system Ψ'.

Let $\mathcal{C}$ be a t-cube, and let f be an $[r,t]$-LDF on $\mathcal{C}$. We map $\mathcal{C}$ to an extension domain $ext(\mathcal{C})$, and f to an extension LDF f_{ext} using an embedding technique from [DFK+98], as described below.

We map the points of $\mathcal{C}$ to a manifold in $\mathcal{D}_{\mathcal{C}} \overset{def}{=} ext(\mathcal{C})$ by $E : \mathcal{C} \to \mathcal{D}_{\mathcal{C}}$ as follows. E maps an arbitrary point $x = (\xi_1, .., \xi_t) \in \mathcal{C}$ to $y \in \mathcal{D}_{\mathcal{C}}$, $y = E(x) = (\eta_1, .., \eta_{kt})$ by replacing each axis ξ_i with k axises $\eta_{i,1}, .., \eta_{i,k}$ such that the following equations hold,

$$\forall i, m \;\; \eta_{i,m} = \xi_i^{b^m} \qquad (*)$$

Now let f be an $[r,t]$-LDF. We map f to a polynomial over $\mathcal{D}_{\mathcal{C}}$ by mapping each of the monomials,

$$\forall i, r \quad \xi_i^r \longrightarrow \eta_{i,0}^{b_0}\eta_{i,1}^{b_1}...\eta_{i,k}^{b_k} \qquad (**)$$

where $b_0 b_1 ... b_k$ is the base b representation of r.

f_{ext} is an LDF of degree b in each variable, and dimension $t \log_b r$. Taking the restriction of f_{ext} to the manifold defined by equations $(*)$, will give f. (This can be easily seen by substituting the manifold equations into each of the monomials). In other words, there is a natural 1-1 mapping of the original cube $\mathcal{C}$ to the manifold of $ext(\mathcal{C})$ defined by equations $(*)$. Computing f on a point $x \in \mathcal{C}$ is equivalent to computing $E(f) = f_{ext}$ on the point $E(x)$.

The new cube-system Ψ'.

For every cube $\mathcal{C}$, consider its extension domain $ext(\mathcal{C})$. Any $[r,t]$-LDF f over $\mathcal{C}$ is naturally mapped to a $[b, t\log_b r]$-LDF, f_{ext} over $ext(\mathcal{C})$. We shall replace every cube $\mathcal{C}$ by a set of cubes (defined by the tuples below) of its extension domain $ext(\mathcal{C})$. Assignments to these new cubes will range over $[b, t\log_b r]$-LDFs.

Domains. For every cube $\mathcal{C} \in \mathcal{S}_{\mathcal{T}}$ we have a new domain $\mathcal{D}_{\mathcal{C}} \overset{def}{=} ext(\mathcal{C})$.

Tuples. We now define a new collection of tuples (that will generate our new cubes). For every $\tau \in \mathcal{T}$ and $\mathcal{C} \in \mathcal{S}_\tau$, let

$$\mathcal{T}_\mathcal{C} = \{(x, \tau) \mid \ x \in \mathcal{C}\}$$

be a set of $(t+1)$-tuples. We consider the tuples in $\mathcal{T}_\mathcal{C}$ as belonging to the domain $\mathcal{D}_\mathcal{C}$. The tuple collection of Ψ' is $\mathcal{T}' \overset{def}{=} \bigcup_\mathcal{C} \mathcal{T}_\mathcal{C}$.

Variables. For every point in the the extension $\mathcal{D}_\mathcal{C}$ of $\mathcal{C}$, Ψ' will have a variable. The extensions of a point $x \in \mathcal{C}_1 \cap \mathcal{C}_2$, in $\mathcal{D}_{\mathcal{C}_1}$ and $\mathcal{D}_{\mathcal{C}_2}$ are considered mutual, and therefore will be represented by *the same* variable in Ψ'.

Tests. There will be a test for every $(t+4)$-cube in $\mathcal{S}' \overset{def}{=} \mathcal{S}_\mathcal{T}$. Before we describe the range of the tests, let us dwell for a minute on the meaning of this replacement transformation.

Let $\mathcal{C}$ be a cube in the previous cube-system, Ψ, ranging over $[r,t]$-LDFs. $\mathcal{C}$ was mapped to $\mathcal{D}_\mathcal{C}$, its extension domain, where f is naturally mapped to f_{ext}, a $[b, t\log_b r]$-LDF. We have new cubes that allegedly represent $(t+4)$-cubes of $\mathcal{D}_\mathcal{C}$. We would like their values to represent the value of the original variable $\mathcal{C}$, i.e. we would like them to represent restrictions of f_{ext} to the sub-cubes of $\mathcal{D}_\mathcal{C}$. The range of our new cube-variables, will therefore be, $[bt\log_b r, t+4]$-LDFs. This completes the description of the b-transformation.

3.3 The Whole Construction

Let us step back to examine the bigger picture. We begin with the naive cube-system.We transform it (by a b-transformation) into Ψ_1, and then transform Ψ_1 into Ψ_2 etc. In the end, the resulting cube-system will be shown to have the $\mathcal{SSAT}$ gap property. Also note, that the transformations only depend on our choice of the b parameter.

Tree Structure. The recursive structure of the construction is easily depicted as a tree of cubes. The root of the tree is the domain of Φ, the original $\mathcal{PCP}$ test system. Directly beneath the root are all of the cubes of Ψ_0, the naive construction. In the b-transformation, every cube was replaced by a set of cubes in its extension domain. These will be placed directly beneath the cube in the tree. Except for the root, every node in the tree can be dually viewed either as a cube, or as a domain that is the embedding extension of that cube.

Alternatively, the nodes of the tree each correspond to a tuple. The first level tuples (directly beneath the root) are the tuples defined by the tests of Φ plus three random points. the offspring of any node $\mathcal{C}$ in the tree, are the tuples that contain the node's tuple, plus a random point from the manifold $E(\mathcal{C}) \subset ext(\mathcal{C})$, plus three random points in the extension domain $ext(\mathcal{C})$.

We shall return to this tree structure for the correctness proof. Let us first complete the description of the construction by giving the exact sequence of

'b's used in the b-transformation sequence; the last cube-system in the sequence being the final construction.

Notation. It will be convenient to use the following parameters for a cube-system Ψ :

[t] $t = t(\Psi)$ The cube dimension of Ψ.

[D] $D = D(\Psi)$ The dimension of Ψ.

[r] $r = r(\Psi)$ The overall degree of the assignments for Ψ.

We can construct, (proposition 3), the b-transformation Ψ' of Ψ with the following parameters :

[t] $D(\Psi') = t \log_b r$

[D] $t(\Psi') = t + 4$

[r] $r(\Psi') = tb \log_b r$

The variable range of the new cube-system is, for the right choice of b, considerably smaller than that of the old system. However, one such transformation is not enough - we need to repeat the transformation several times $(O(\frac{1}{\epsilon}))$ to get our desired polynomial range.

The Initial Cube-System

Let Ψ_2 denote the naive $\mathcal{SSAT}$ system (we start with Ψ_2 for notational convenience). Let us recall some of its parameters:

[t] $D(\Psi_2) = \log^{2\epsilon} n$

[D] $t(\Psi_2) = const$

[r] $r(\Psi_2) = 2^{O(\log^{1-2\epsilon} n)}$

The Values Chosen for b.

The following are the values of b that are used in the series of b-transformations:

- *The b reduction phase.* The *first* 'b' used will be denoted b_3 for convenience, and is defined as $b_3 = 2^{\log^{1-3\epsilon} n}$. The second '$b$', b_4, is defined $b_4 = 2^{\log^{1-4\epsilon} n}$. We continue to take $b_k = 2^{\log^{1-k\epsilon} n}$, for $k \le K$, where $K \overset{def}{=} \lfloor \frac{1}{\epsilon} \rfloor$. After that we can no longer proceed, because the power of the logarithm would become negative.
- *The sub-logarithmic phase.* We use $b = 2$ for three additional iterations.

These values of b define the resulting cube-system. We need to show that the resulting cube-system indeed possesses the desired parameters.

The Parameters During the b Reduction Phase

Denote $\Psi_3, \Psi_4, \ldots$ the systems obtained from the $b_3, b_4, \ldots$ transformations respectively. Let us analyze their parameters.
The parameters of Ψ_3, calculated as in the definition of the b-transformation:

[t] $t(\Psi_3) = t(\Psi_2) + 4$

[D] $D(\Psi_3) = t(\Psi_2) \log_{b_3} r(\Psi_2)$

[r] $r(\Psi_3) = t(\Psi_2) b_3 \log_{b_3} r(\Psi_2)$

Actually this calculation is good for the general k iteration in the phase :

[t] $t(\Psi_k) = t(\Psi_{k-1}) + 4$

[D] $D(\Psi_k) = t(\Psi_{k-1}) \log_{b_k} r(\Psi_{k-1})$

[r] $r(\Psi_k) = t(\Psi_{k-1}) b_k \log_{b_k} r(\Psi_{k-1})$

Let us now analyze the explicit behavior of these parameters, as functions of n.

Proposition 4 *For every* $2 \leq k \leq K-1$,

$$r(\Psi_k) \leq 2^{2 \log^{(1-k\epsilon)} n}$$

Proof: We prove the proposition by induction on k. Let us check the base of the induction ($k = 2$):

$$r(\Psi_2) = 2^{\log^{1-2\epsilon} n} \leq 2^{2 \log^{1-2\epsilon} n}$$

Now for the inductive step. From the inductive hypotheses we have

$$r(\Psi_{k-1}) \leq 2^{2 \log^{(1-(k-1)\epsilon)} n}$$

Since we chose $b_k = 2^{\log^{1-k\epsilon} n}$

$$\log_{b_k} r(\Psi_{k-1}) \leq 2 \log^{\epsilon} n \tag{1}$$

Recall that $\epsilon = (\log \log n)^{-c}$ for $c < \frac{1}{2}$, and that $K = \lfloor \frac{1}{\epsilon} \rfloor$. Thus

$$\begin{aligned} qcnt = 2t(\Psi_k) \leq O(K) &\leq O((\log\log n)^c) \\ &\leq (\log\log n)^{1-c} \\ &= 2^{\epsilon \log\log n} \\ &= \log^{\epsilon} n \end{aligned} \tag{1}$$

Let us also note that

$$1 - k\epsilon \geq \epsilon \tag{3}$$

Therefore, by the recursive formula for r,

$$\begin{array}{rcl} r(\Psi_k) & = & t(\Psi_{k-1})b_k \log_{b_k} r(\Psi_{k-1}) \\ & \overset{(1)}{\leq} & O(t(\Psi_{k-1})2^{\log^{1-k\epsilon} n}) \log^{\epsilon} n \\ & \overset{(2)}{\leq} & O(\log^{2\epsilon} n 2^{\log^{1-k\epsilon} n}) \\ & \leq & 2^{\log^{\epsilon} n} 2^{\log^{1-k\epsilon} n} \\ & \overset{(3)}{\leq} & 2^{2\log^{1-k\epsilon} n} \end{array}$$

■

The last parameter in this phase, b_K, can be computed similarly, and we obtain

$$r(\Psi_K) = O(K)b_K \log_{b_K} r(\Psi_{K-1}) \leq 2^{2\log^{\epsilon} n}$$

The Parameters During the Sub-Logarithmic Phase

Ψ_K is the last system constructed using $b > 2$. The parameters of Ψ_K are

[t] $t(\Psi_K) = O(\frac{1}{\epsilon})$

[r] $r(\Psi_K) \leq 2^{2\log^{\epsilon} n}$

We now calculate the parameters of the systems generated from Ψ_K. The reader is reminded that we use $b = 2$ from now on, and since we only make a constant number of additional steps, it is enough to assume $t \leq O(K)$ for these systems. It is thus only left to calculate r. We use the general b transformation formula :

$$\begin{array}{rcl} r(\Psi_{K+1}) & = & t(\Psi_K)b_{K+1} \log_{b_{K+1}} r(\Psi_K) \\ & \leq & O(K) \log_2 2^{2\log^{\epsilon} n} \\ & \leq & O(K \log^{\epsilon} n) \\ & \leq & \log^{2\epsilon} n \end{array}$$

Following the same calculation for Ψ_{K+2} we have

$$r(\Psi_{K+2}) \leq O(K) \cdot 2\epsilon \log\log n = O(\log\log n)$$

Last but not least, the r parameter for Ψ_{K+3} is bounded by

$$r(\Psi_{K+3}) \leq O(K \log\log\log n) = O(K \log^{(3)} n) = O(K^2)$$

3.4 The Size of the Construction

The previous section completed the description of the construction. Ψ_{K+3} is our final $\mathcal{SSAT}$ test system. In this section we show that it is polynomial in size. Its correctness is proven in the next subsection.

The Variable Range

The variables of Ψ_{K+3} range over $[r,d]$-LDFs where r and d are the r and d parameters of Ψ_{K+3}. Let us compute the size of the range. The number of monomials of degree $r(\Psi_{K+3}) = O(K^2)$, and dimension $d(\Psi_{K+3}) = O(K)$ is bounded by

$$r^d = O(K^2)^{O(K)} = 2^{O(K \log K)} = 2^{(\log\log n)^c O(\log^{(3)} n)}$$

The number of polynomials is therefore bounded by

$$|\mathcal{F}|^{2^{(\log\log n)^c O(\log^{(3)} n)}} = 2^{2^{(\log\log n)^c O(\log^{(3)} n)} \cdot (\log n)^{(1-2\epsilon)}}$$

This is polynomial iff the following inequality holds,

$$(\log n)^{-2\epsilon} \cdot 2^{(\log\log n)^c O(\log^{(3)} n)} \leq 1$$

indeed substituting ϵ gives,

$$(\log n)^{-2\epsilon} = 2^{\frac{-2\log\log n}{(\log\log n)^c}} = 2^{-2(\log\log n)^{1-c}}$$

and,

$$\begin{aligned} (\log n)^{-2\epsilon} \cdot 2^{(\log\log n)^c O(\log^{(3)} n)} &= \\ = \quad & 2^{-2(\log\log n)^{1-c}} \cdot 2^{(\log\log n)^c O(\log^{(3)} n)} \\ = \quad & 2^{-2(\log\log n)^{1-c} + (\log\log n)^c O(\log^{(3)} n)} \end{aligned}$$

and therefore, since $1 - c > c$ ($c < \frac{1}{2}$), the range is polynomial in n.

The Number of Tests and Variables

It is only left to verify that the blowup is polynomial (i.e. the number of tests and variables is polynomial).

Let Ψ' be the outcome of the transformation procedure on Ψ. The blowup in the transformation step, i.e. the ratio between $v(\Psi')$ and $v(\Psi)$, is the number of $t(\Psi')$ dimensional cubes in the extension of each cube of Ψ. The formula for the extension's dimension $D(\Psi')$ is $t(\Psi)\log_{b_{\Psi'}} r(\Psi)$. By proposition 4 we have that $r(\Psi) \leq 2^{2\log^{(1-k\epsilon)} n}$ where k is the number of the transformation step. By the choice of $b_{\Psi'}$ we get that

$$\log_{b_{\Psi'}} r(\Psi) \leq \log_{2^{\log^{(1-(k+1)\epsilon)} n}} \left(2^{2(\log n)^{(1-k\epsilon)}}\right) \leq 2(\log n)^{\epsilon}$$

and therefore

$$D(\Psi') \leq 2t(\Psi)(\log n)^{\epsilon} \leq O(K)(\log n)^{\epsilon}$$

The blowup in a single step is bounded by

$$(|\mathcal{F}|^D)^t = |\mathcal{F}|^{t(\Psi')D(\Psi')} \leq |\mathcal{F}|^{O((\log\log n)^{2c}(\log n)^{\epsilon})}$$

and since we take $K = O((\log\log n)^c)$ steps, the overall blowup is bounded by the $O((\log\log n)^c)$ exponent of the single step blowup. The overall blowup is therefore bounded by

$$|\mathcal{F}|^{O((\log\log n)^{3c}\log^{\epsilon} n)} \leq 2^{O((\log n)^{1-2\epsilon})(\log\log n)^{3c}(\log n)^{\epsilon}}$$

$$\leq 2^{O((\log n)^{1-\epsilon})(\log\log n)^{3c}} \leq 2^{\log n}$$

and is therefore polynomial in n.

3.5 Correctness of the Construction

We have shown a sequence of polynomial-sized cube-systems $\Psi_2, \Psi_3, ..., \Psi_{K+3}$. In this section we will prove that each of them (and in particular, the final cube system) possess the $\mathcal{SSAT}$ gap property. For simplicity of notation we shall rename the cube-systems $\Psi_1, ..., \Psi_K$ (stretching K to be $K+2$) where Ψ_1 is the naive cube-system, and Ψ_K is the cube system taken to be the final construction. We need to show completeness and soundness of the construction.

Completeness. We need to show that 'yes' instances of $\mathcal{PCP}$ map to 'yes' instances of $\mathcal{SSAT}$; i.e. if the original test system Φ was satisfiable, then Ψ_K is satisfiable. Let I be the satisfying assignment for Φ. Taking the low-degree-extension of I, and then projecting it to the cubes, will obviously satisfy Ψ_1 (the naive construction). A satisfying assignment to Ψ_i translates into one for Ψ_{i+1} by computing (for each cube $\mathcal{C}$) the embedding extension of the LDF assigned to $\mathcal{C}$ and then computing its restrictions to cubes of the extension domain $ext(\mathcal{C})$. The resulting super-assignment for Ψ_K is obviously consistent.

Soundness. We need to show that 'no' instances of $\mathcal{PCP}$ map to 'no' instances of $\mathcal{SSAT}$. We assume that the constructed $\mathcal{SSAT}$ instance has a satisfying super-assignment of norm $\leq g$, and show that Φ – the PCP test system we started with – is satisfiable.

Theorem 4 (Soundness) *Let $g \overset{def}{=} |\mathcal{F}|^{\frac{1}{100}\cdot\frac{1}{a}^K}$ for some constant $a > 1$. If there exists a satisfying super-assignment of size $\leq g$ for Ψ_K, then Φ (the PCP system we began with) is satisfiable.*

Before proceeding to the soundness proof, let us verify that the above g gives the desired parameters in the g-$\mathcal{SSAT}$ theorem (theorem 1).

Proposition 5 $\forall c < \frac{1}{2}$, $\exists c < c' < \frac{1}{2}$ *such that*

$$\begin{aligned} |\mathcal{F}| &= 2^{\log^{1-\epsilon(c')} n} \\ g = |\mathcal{F}|^{\frac{1}{100}\cdot\frac{1}{a}^K} &= 2^{\log^{1-\epsilon(c)} n} \end{aligned}$$

where $\epsilon(c) \overset{def}{=} (\log\log n)^{-c}$.

Proof: The proposition follows from the following inequalities. In the use of *exp* here, we intend to the exponent in base 2.

$$\forall c < 0.5 : \qquad a^{\frac{1}{\epsilon(c)}} \leq a^{\sqrt{\log\log n}}$$

$$\begin{array}{rcl} g > |\mathcal{F}|^{\frac{1}{a}\sqrt{\log\log n}} & = & exp(\frac{\log n}{a^{\sqrt{\log\log n}}\log^{2\epsilon(c')} n}) \\ & > & exp(\frac{\log n}{\log^{\epsilon(c)} n}) \end{array}$$

the last inequality following from,

$$\begin{array}{rcl} a^{\sqrt{\log\log n}}\log^{\epsilon(c')} n & = & exp(O(\sqrt{\log\log n}) + (\log\log n)^{1-c'}) \\ & = & exp(O((\log\log n)^{1-c'})) \\ & < & 2^{(\log\log n)^{1-c}} \\ & = & \log^{\epsilon(c)} n \end{array}$$

■

This proposition implies that the soundness parameters indeed provide the desired hardness result for $\mathcal{SSAT}$.

We proceed to prove the soundness theorem.

Proof Structure

We begin with $\mathcal{M}_K$, a consistent super-assignment for Ψ_K, of size $\leq g$. it induces (by projection) a super-assignment m for the variables (points). Since $\mathcal{M}_K$ is consistent, m is well defined. m is used as the "underlying point super-assignment" for the rest of the proof.

Let us return to the tree view of the construction. We put every cube of the final construction as a leaf in the tree, and the internal nodes are cubes of the intermediate cube-systems (level i in the tree corresponds to the cubes of Ψ_i). We define a 'good' cube (node in the tree) by considering the leaves of its sub-tree:

Definition 7 *Let $\mathcal{C}$ be a cube in Ψ_i. $\mathcal{C}$ is said to be* good *if $avg(\mathcal{C}) \leq g^{a^i+1}$; where $avg(\mathcal{C})$ is the average norm of $\mathcal{M}_K$, over the cubes in Ψ_K that are derived from $\mathcal{C}$ (i.e. cubes of Ψ_K that are leaves in $\mathcal{C}$'s subtree). We denote by $Good_i$ the set of cubes of Ψ_i that are good.*

The proof consists of four propositions. We shall first show that most cubes are good (proposition 6), and that most of the children of a good cube are good (proposition 7). We then proceed to show that for every good cube there is a super-polynomial that agrees with m on most of the cube's points (proposition 8). Finally, we take these super-polynomials, and obtain from them an assignment that satisfies more than half of the $\mathcal{PCP}$ test system Φ (proposition 9).

Most cubes are good

Proposition 6 *Let $0 < i \leq K$, at least $1 - g^{-a^i}$ of the cubes in level i of the tree are good.*

Proof: For every cube in level i, consider the average norm of its subtree's leaves. The average over the subtree averages is simply the average norm of the leaves, g. By the Markov inequality, no more than $\frac{1}{l}$ of the subtrees have an average larger than lg. Taking $l \overset{def}{=} g^{a^i}$ concludes the argument. ■

Most Subcubes of a good cube are good

For any cube $\mathcal{C}$, denote by $child(\mathcal{C})$ the set of cubes directly beneath $\mathcal{C}$ in the tree. These are the cubes in the embedding extension $ext(\mathcal{C})$.

Proposition 7 *If $\mathcal{C} \in Good_i$, then*

$$\Pr_{\mathcal{C}' \in child(\mathcal{C})} (\mathcal{C}' \in Good_{i+1}) > 1 - g^{-(a-1)\cdot a^i}$$

Proof: $\mathcal{C}$ is good, hence $avg(\mathcal{C}) \leq g^{a^i+1}$. Had there been more than a $g^{-(a-1)\cdot a^i}$ fraction of bad subcubes, then the total average would be

$$> g^{-(a-1)\cdot a^i} \cdot g^{a^{i+1}+1} = g^{a\cdot a^i-(a-1)\cdot a^i+1} = g^{a^i+1}$$

■

A Super-Polynomial per Good Cube

Proposition 8 *Let $\mathcal{C}_0 \in Good_i$. There exists a super-polynomial of norm $\leq 2^{K-i} avg(\mathcal{C}_0)$ that agrees with m on a $1 - g^{-a^i}$ fraction of $\mathcal{C}_0$'s points.*

Proof: We prove this statement by induction on $K - i$.

The Base of the Induction ($i = K$). We know that $\mathcal{M}_K$ is a consistent super-assignment. Every $\mathcal{C} \in Good_K$ has a super-polynomial ($\mathcal{M}_K(\mathcal{C})$) of norm $\leq g^{a^K+1}$, by definition of $Good_K$. $\mathcal{M}_K(\mathcal{C})$ agrees (non-ambiguously) with m on at least $1 - amb(r, t, s)$ of the points of $\mathcal{C}$. Since

$$amb(r,t,s) < \frac{rts^2}{|\mathcal{F}|} \leq |\mathcal{F}|^{-\frac{1}{2}}$$

the claim follows, using the consistency lemma (see below).

The Inductive Step $(i < K)$. $avg(\mathcal{C}_0) \geq avg_{\mathcal{C}' \in child(\mathcal{C}_0) \cap Good_{i+1}}(avg(\mathcal{C}'))$, since taking only the good children can only decrease the average. By the inductive hypothesis, every $\mathcal{C}' \in child(\mathcal{C}_0) \cap Good_{i+1}$ has a super-polynomial with norm $\leq 2^{K-i-1} avg(\mathcal{C}')$. Define a super-assignment to the cubes in $\mathcal{C}' \in Good_{i+1} \cap child(\mathcal{C}_0)$ by setting $\mathcal{M}(\mathcal{C}')$ to be that super-polynomial. The average norm of these super-polynomials is $\leq 2^{K-i-1} avg(\mathcal{C}_0)$. For any cube $\mathcal{C}' \in child(\mathcal{C}_0) - Good_{i+1}$ assign the trivial super-polynomial. It then follows that $||\mathcal{M}|| \leq 2^{K-i-1} avg(\mathcal{C}_0)$.

We now state a consistency lemma that will imply the existence of the desired super-polynomial on $\mathcal{C}_0$.

Lemma 2 (Consistency Lemma) *Let $\mathcal{T}$ be a set of t-tuples. Let $\mathcal{D}$ be a domain, and let $\mathcal{S}_{\mathcal{T}}$ be the set of $(t+3)$-cubes of $\mathcal{D}$ that contain the points of at least one tuple in $\mathcal{T}$. Let $\mathcal{M} : \mathcal{S}_{\mathcal{T}} \to SLDF_{r,t+3}$ be a super-assignment; $||\mathcal{M}|| \leq s < |\mathcal{F}|^{\frac{1}{100}}$. Let m be the underlying super-assignment to the points. Denote by $G \subset \mathcal{S}_{\mathcal{T}}$, the set of all cubes $\mathcal{C}$ for which $\mathcal{M}(\mathcal{C})$ agrees with m on at least $1-\alpha$ of $\mathcal{C}$'s points, for some $0 < \alpha < \frac{1}{100}$.*

If at least $1-\alpha$ of the cubes in $\mathcal{S}_{\mathcal{T}}$ are in G then there exists a global super-polynomial $\mathcal{G}$ of degree r on $\mathcal{D}$, with $||\mathcal{G}|| \leq 2s$ and

$$\Pr_{\mathcal{C} \in_R G}(\mathcal{M}(\mathcal{C}) = \pi_{\mathcal{C}}(\mathcal{G})) > 1 - |\mathcal{F}|^{-\frac{1}{2}}(2s+1)$$

We defer the proof of this lemma to section 5. We would like to apply this lemma to the domain $\mathcal{D} = ext(\mathcal{C})$, and the super-assignment $\mathcal{M}$ defined above. Let us see that the super-assignment $\mathcal{M}$ obeys the requirements of the consistency lemma. We take $\alpha = g^{-(a-1)a^i}$, and $s = g^{a^i+1}$. For every cube $\mathcal{C}' \in Good_{i+1} \cap child(\mathcal{C}_0)$, $\mathcal{M}(\mathcal{C}')$ agrees with m on at least $1 - g^{-a^{1+1}} > 1 - \alpha$ (by the inductive hypothesis). The fraction of good cubes in $child(\mathcal{C}_0)$ is, by proposition 7, $\geq 1 - g^{-(a-1)a^i} = 1 - \alpha$. Finally, we can apply the consistency lemma (lemma 2).

We thus obtain a super-polynomial on $\mathcal{D}$ of norm $\leq 2 \cdot 2^{K-i-1} avg(\mathcal{C}_0) = 2^{K-i} avg(\mathcal{C}_0)$ that agrees with $1 - |\mathcal{F}|^{-\frac{1}{2}}(2^{K-i} avg(\mathcal{C}_0) + 1)$ of the super-polynomials on the cubes $\mathcal{C}' \in child(\mathcal{C}_0) \cap Good_{i+1}$. Recall that on every good cube $\mathcal{M}$ agreed with m on the tuple-points of the cube. This means that $\mathcal{G}$ agrees with m on 'almost all' of the tuples (because agreeing with one good cube on a tuple means agreeing with the tuple). Let us examine the meaning of 'almost all':
Since $2^{K-i} avg(\mathcal{C}_0) < |\mathcal{F}|^{\frac{2}{100}}$, we have

$$|\mathcal{F}|^{-\frac{1}{2}}(2^{K-i} avg(\mathcal{C}_0) + 1) \leq |\mathcal{F}|^{-\frac{1}{2}+\frac{2}{100}} < \frac{|\mathcal{F}|^{-\frac{1}{100}}}{2} \leq \frac{g^{-a^i}}{2}$$

hence the fraction of cubes in $child(\mathcal{C}_0)$ that are good, and agree with $\mathcal{G}$ is at least $1 - g^{-a^i}$ (the good cubes make up at least half of the cubes). This implies that $\mathcal{G}$ agrees with at least this fraction of tuples. Since we have exactly one tuple per point in the manifold of $\mathcal{C}_0$ (as defined in the construction), we have that $\mathcal{G}_{\mathcal{C}_0}$ (the restriction of $\mathcal{G}$ to the manifold) agrees with $> 1 - g^{-a^i}$ of the manifold points that correspond to $\mathcal{C}_0$, and with all of the tuple-points of $\mathcal{C}_0$ itself.

■

Constructing an assignment for Φ

Proposition 9 *There is an assignment that satisfies more than half of the $\mathcal{PCP}$ test system Φ.*

Proof: We now have an super-polynomial for every good cube. Define $\mathcal{M}_1$ to assign (as before) the cubes in $Good_1$ this super-polynomial, and the trivial super-polynomial to the rest of the cubes in level 1. By proposition 6 we have that $1 - g^{-(a-1)a}$ of the cubes are good, and their super-polynomial agrees with $1 - g^{-a}$ of their points *and* with their tuple points. We know that the average norm of $\mathcal{M}_1$ is $\leq 2^{K-1}avg(root) = 2^{K-1}g$. Using the consistency lemma one more time with $\alpha = g^{-a}$ yields a global super-polynomial that agrees with almost all of the good cubes, and in particular with half of the cubes in Ψ_1. This super-polynomial is not trivial, because m – the underlying point super-assignment – is non-trivial on most of its points (this follows from the low-ambiguity proposition, proposition 2).

Take one LDF P that appears in this super-polynomial. It appears in most of the good cubes (it may be canceled on a negligible fraction of the cubes). For every point $x \in \mathcal{H}^d$, assign $P(x)$. This will satisfy at least half of the tests in the $\mathcal{PCP}$ test-system, Φ. This follows since each $\varphi \in \Phi$ is represented by a tuple that is, in turn, contained by a cube that with probability $> \frac{1}{2}$ has P's restriction appearing in its super-polynomial. Note that the $\mathcal{PCP}$ property that we used was a *constant* error probability.

Using the PCP property of Φ we deduce that it is satisfiable. This completes the proof of lemma 4 (the soundness of the reduction). ■

4 g-CVP is NP-hard

We begin by defining the Closest Vector Problem (CVP), and its gap version g-CVP. We then define an intermediate problem called Shortest Integer Solution (SIS), and show a reduction from g-SIS to g-CVP. We then show the simple reduction from g-$\mathcal{SSAT}$ to g-SIS and therefore to g-CVP.

4.1 g-CVP

A lattice $L = L(v_1, .., v_n)$, for vectors $v_1, .., v_n \in R^n$ is the set of all integral linear combinations of $v_1, .., v_n$, $L = \{\sum a_i v_i \mid a_i \in Z\}$.
The closest-vector problem is defined as follows:

CVP. Given (L, y) where L is a lattice and y a vector in R^n, find the lattice vector closest to y.

Approximating CVP to within factor $g = g(n)$ means finding a vector whose distance from y is no more than g times the minimal distance. The gap version of CVP is a decision problem as follows,

g-CVP. Given (L, y) for a lattice L and a vector $y \in R^n$, distinguish between the following two cases:

[Yes] The closest lattice vector to y is of distance d or less.

[No] All lattice vectors are of distance at least $g \cdot d$ from y.

Proving that g-CVP is NP-hard means that having an approximation algorithm to within factor g were to imply $P = NP$.

4.2 Shortest Integer Solution - SIS

Definition of SIS and g-SIS

We define a variant of CVP named Shortest Integer Solution (SIS), its gap version referred to as g-SIS. We then show a simple reduction from g-SIS to g-CVP.

SIS: Given (B, t) for an integer matrix B with columns $b_1, .., b_n$ and a target vector $t \in L(b_1, ..., b_n)$, find integer coefficients a_i such that $\sum a_i b_i = t$ (we assume such a_i exist), and such that $\sum |a_i|$ is minimal. In other words, find the shortest integer solution for the linear system $B \cdot x = t$.

The gap version of SIS is as follows,

g-SIS: Given (B, t) as before, distinguish between the following two cases:

Yes: The shortest integer solution is of length d or less.

No: The shortest integer solution is of length at least $g \cdot d$.

Reducing g-SIS to g-CVP

Given an instance of g-SIS, (B, t), we efficiently construct a lattice L and a target vector y such that 'yes' instances of g-SIS are translated into 'yes' instances of g-CVP and 'no' instances are translated into 'no' instances. The lattice L is constructed by multiplying the matrix B by a very large number w, and adding a distinct 1-coordinate to each column. The vector y (that we are to approximate from within the lattice) will be t multiplied by w with zeros in the n additional coordinates:

$$L = \begin{pmatrix} & wB & \\ & & \\ 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \qquad y = \begin{pmatrix} \vdots \\ wt \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

To see that 'yes' instances map into 'yes' instances just note that any solution a, $B \cdot a = t$, gives a lattice vector $L \cdot a$ such that $||L \cdot a - y|| = ||a||$. We shall choose w so that the entries in the upper half of the matrix are all integer multiples of $g \cdot d + 1$. The next lemma will show that 'no' instances of g-SIS (where the shortest solution is of size $> g \cdot d$) map into 'no' instances of g-CVP.

Lemma 3 *If there is a lattice vector, $L \cdot a$, $r = dist(L \cdot a, y) \leq g \cdot d$, then there is an integer solution to (B, t) of size r.*

Proof: $r \leq gd$ means that $L \cdot a = y$ in the upper n coordinates, otherwise the distance r would be of size at least $g \cdot d + 1$. In other words, a is a solution to the g-SIS instance. The lower n coordinates of $L \cdot a$ are exactly equal to a, and therefore $||a|| = r$. ■

4.3 From $\mathcal{SSAT}$ to g-SIS

We shall prove that g-SIS is NP hard for $g = 2^{(\log n)^{1-\epsilon}}$ for $\epsilon = (\log \log n)^{-\alpha}$ for any $\alpha < \frac{1}{2}$ by reducing it to $\mathcal{SSAT}$.

We take this $\mathcal{SSAT}$ test system Ψ and (efficiently) construct from it an instance of g-SIS, (B, t). We then show that the 'yes' instances of $\mathcal{SSAT}$ are mapped to 'yes' instances of g-SIS and 'no' instances to 'no' instances.

We show that a natural consistent assignment to Ψ translates to a short solution for (B, t). On the other hand we show that any solution that is shorter than g, translates to a consistent super-assignment of size $\leq g$ for Ψ.

The General Construction

The matrix B will have a column for every pair of test $\psi \in \Psi$ and an assignment $r \in \mathcal{R}_\psi$ for it. We will be able to translate the shortest integer solution into a consistent super-assignment for Ψ. The upper rows of B will take care of consistency, and the lower rows will take care of non-triviality.

Non-Triviality Rows. There will be a row designated to each test. In the row of ψ all of ψ's columns will have a 1, and all other columns will have zero.

Consistency Rows. We shall have $|\mathcal{F}|$ rows for each pair of tests ψ_i and ψ_j that depend on a mutual variable x. The columns of tests other than ψ_i and ψ_j will have zeros in all of these rows. These rows serve as a consistency-ensuring gadget and only the vectors of ψ_i and ψ_j will have non-zero values in these rows. The gadget will ensure that the super-assignments to ψ_i and ψ_j *are consistent* on their mutual variable x.

The **target vector** will be an all-1 vector.

We now turn to describe the structure of the gadget itself. This will complete the description of the g-SIS instance.

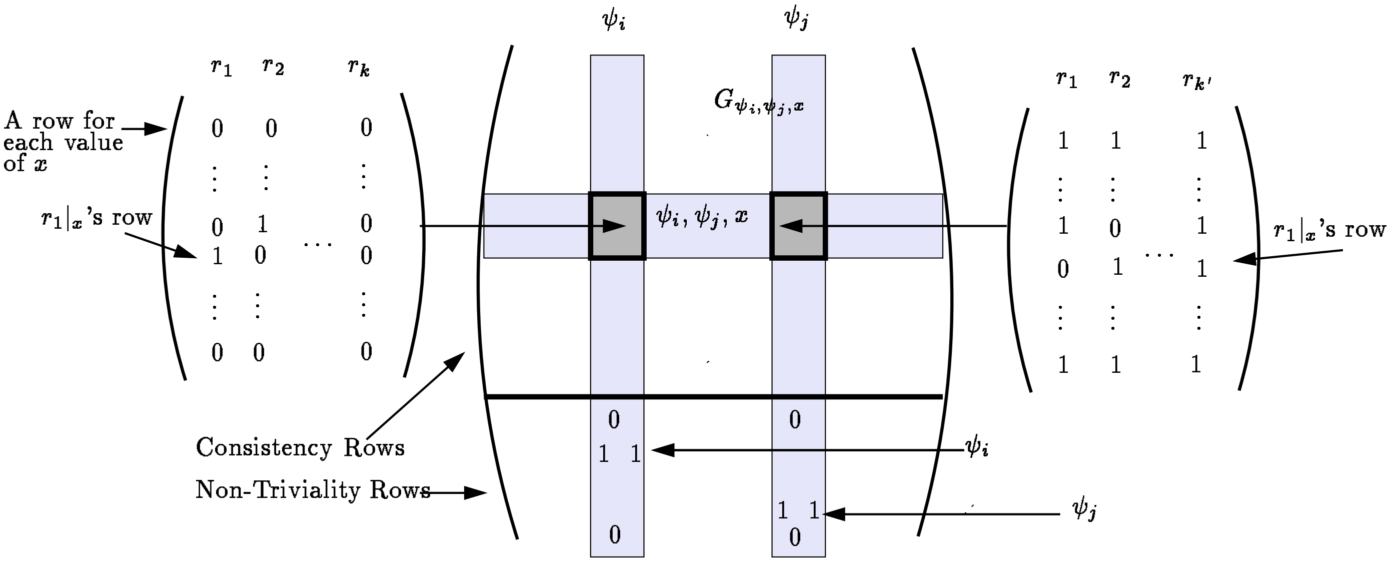

Figure 1: The SIS matrix B

The Gadget. Let's concentrate on the gadget for the pair of tests ψ_i and ψ_j with mutual variable x. This is a pair of matrices G_1 of dimension $(|\mathcal{F}| \times |\mathcal{R}_{\psi_i}|)$ and G_2 of dimension $(|\mathcal{F}| \times |\mathcal{R}_{\psi_j}|)$. Let $r \in \mathcal{R}_{\psi_i}$ be a satisfying assignment for ψ_i and $r' \in \mathcal{R}_{\psi_j}$ be a satisfying assignment for ψ_j. The column in G_1 corresponding to r is a unit vector with a 1 in the $r|_x$-th coordinate. The column in G_2 corresponding to r' is the negation of a unit vector (all ones except for one 0) with a zero in the $r'|_x$-th coordinate (see figure 1).

Proving Correctness

We will now show that 'yes' instances of the $\mathcal{SSAT}$ map to 'yes' instances of the g-SIS.

Lemma 4 *If there is a consistent natural assignment, then there is a solution of length $|\Psi|$ to the above g-SIS instance.*

Proof: We take the consistent natural assignment M and construct from it a solution to the g-SIS. We will concatenate the vectors $M(\psi_1)M(\psi_2)$... to obtain our alleged solution to g-SIS. The target vector is reached in the non-triviality rows because M is natural i.e. it assigns a $+1$ coefficient to exactly one column of every test.

To show that the target vector is reached in the consistency rows, consider the set of $|\mathcal{F}|$ rows belonging to an arbitrary pair of tests ψ_i and ψ_j with mutual variable x. Suppose $M(\psi_i)[r_1], M(\psi_j)[r_2]$ are the single 1's in $M(\psi_i), M(\psi_j)$ respectively (M is natural). M is consistent so $r_1|_x = r_2|_x$. By the construction of B we see that

$$\begin{array}{cc} & r_1 \\ \begin{array}{c} 1 \\ \vdots \\ r_1|_x \\ \vdots \\ |\mathcal{F}| \end{array} & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \end{array} + \begin{array}{cc} & r_2 \\ \begin{array}{c} 1 \\ \vdots \\ r_2|_x \\ \vdots \\ |\mathcal{F}| \end{array} & \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \end{array} = \begin{array}{c} \\ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \end{array}$$

and the target vector is reached in these rows.

The length of the solution is the sum of the lengths of the $M(x)$'s, and since $||M|| = 1$, it's size is $|\Psi|$. ■

We will now show that 'no' instances of the $\mathcal{SSAT}$ map to 'no' instances of the g-SIS by showing that if we ended up with an instance that isn't a 'no' instance, then we must have started with a non-'no' instance.

Lemma 5 *Let s be a solution to the above g-SIS instance, $||s|| \leq g\,|\Psi|$ then exists a consistent super-assignment M of size $\leq g$ for the $\mathcal{SSAT}$ instance.*

Proof: We show how to construct M from s: we 'break' s into $|\Psi|$ pieces, one for each test $\psi \in \Psi$.

For any arbitrary $\psi \in \Psi$, the target vector is reached in the ψ-th row of the non-triviality rows. This implies that

$$\sum_{r \in \mathcal{R}_\psi} M(\psi)[r] = 1 \tag{2}$$

and in particular M is non-trivial.

Let $\psi_i, \psi_j \in \Psi$ be arbitrary tests with a mutual variable x. We shall show that $\pi_x(M(\psi_i)) = \pi_x(M(\psi_j))$. Consider the $|\mathcal{F}|$ rows that correspond to ψ_i, ψ_j, x. In each of these rows the sum of the vectors is 1, in other words, for any $f \in \mathcal{F}$,

$$\sum_{r\,:\,r|_x = f} M(\psi_i) + \sum_{r\,:\,r|_x \neq f} M(\psi_j) = 1 \tag{3}$$

Subtracting (2) for ψ_j from (3) gives,

$$\sum_{r\,:\,r|_x = f} M(\psi_i) = \sum_{r\,:\,r|_x = f} M(\psi_j)$$

which, by definition of the projection means $\pi_x(M(\psi_i)) = \pi_x(M(\psi_j))$. We hence have a consistent super-assignment of size $\frac{1}{|\Psi|}||s|| \leq g$.

■

The two above lemmas complete the reduction of $\mathcal{SSAT}$ to g-SIS.

4.4 CVP over Z_p

We can actually show that g-CVP over a finite field Z_p (for any prime p) is NP-hard. This problem (with $p = 2$) was referred to as 'Nearest-Codeword' in

[ABSS93] and shown to be quasi-NP-hard to approximate to within a factor of $2^{(\log n)^{1-\epsilon}}$ for any constant $\epsilon > 0$.

We first need to define a variant of $\mathcal{SSAT}$, $\mathcal{SSAT}_{mod\ p}$ – where consistency is defined as equality of projections *modulo p* – and show that it too is NP-hard. The NP-hardness proof is carried on almost word for word if we notice that a variable was considered ambiguous if any two of its assigned values collided, disregarding the value of the coefficient.

The result for Nearest Codeword easily follows, using the same reduction from $\mathcal{SSAT}_{mod\ p}$ to $CVP_{mod\ p}$.

5 The Consistency Lemma

Lemma 2 (Consistency Lemma) *Let $\mathcal{T}$ be a set of t-tuples. Let $\mathcal{D}$ be a domain, and let $\mathcal{S}_\mathcal{T}$ be the set of $(t+3)$-cubes of $\mathcal{D}$ that contain the points of at least one tuple in $\mathcal{T}$. Let $\mathcal{M} : \mathcal{S}_\mathcal{T} \to SLDF_{r,t+3}$ be a super-assignment; $||\mathcal{M}|| \leq s < |\mathcal{F}|^{\frac{1}{100}}$. Let m be the underlying super-assignment to the points. Denote by $G \subset \mathcal{S}_\mathcal{T}$, the set of all cubes $\mathcal{C}$ for which $\mathcal{M}(\mathcal{C})$ agrees with m on at least $1-\alpha$ of $\mathcal{C}$'s points.*

If at least $1-\alpha$ of the cubes in $\mathcal{S}_\mathcal{T}$ are in G then there exists a global super-polynomial $\mathcal{G}$ of degree r on $\mathcal{D}$, with $||\mathcal{G}|| \leq 2s$ and

$$\Pr_{\mathcal{C}\in_R G}(\mathcal{M}(\mathcal{C}) = \pi_\mathcal{C}(\mathcal{G})) > 1 - |\mathcal{F}|^{-\frac{1}{2}}(2s+1)$$

Before we prove the Consistency Lemma, let us state and prove the polynomial extraction lemma, which we shall use in the proof of the Consistency Lemma.

Lemma 6 (Polynomial Extraction) *Let $\mathcal{M}$,G, s and α be as before. If $\beta > 3\alpha$ of the points are assigned non-trivial values by m (the underlying point super-assignment), then there exist a polynomial $P \in LDF_{r,d}$ and a coefficient c_p such that P's restrictions to $1 - |\mathcal{F}|^{-\frac{1}{2}}$ of the cubes $\mathcal{C} \in G$ appear in $\mathcal{M}(\mathcal{C})$ with coefficient c_p.*

We use the polynomial extraction lemma to extract a polynomial from $\mathcal{M}$ and m. We then "peal off" this polynomial and repeat the process until we extract the whole super-polynomial from $\mathcal{M}$ and m, and obtain the consistency lemma, lemma 2. We first prove lemma 6.

5.1 Proof of the Polynomial Extraction Lemma

The proof proceeds by four propositions.

Proposition 10 *Most* $(1 - \frac{4}{\beta|\mathcal{F}|})$ *cubes in G are non-trivial.*

Proof: We know that the assignment for a β fraction of the points is non-trivial. We now state a hitting lemma that shows that most of the cubes must hit a non-negligible fraction of these points.

Lemma 7 (Hitting Lemma) *Let $0 < \beta < 1$. Let $N \subset \mathcal{D}$ be a set of points, $|N| \geq \beta\,|\mathcal{D}|$. Most $(1 - \frac{2}{\beta|\mathcal{F}|})$ cubes in $\mathcal{S}_{\mathcal{T}}$ have at least $\frac{\beta}{2}$ of their points in N.*

The lemma is easly to obtain using simple probabilistic methods. We deduce that for $(1 - \frac{2}{\beta|\mathcal{F}|})$ of the cubes, $\frac{\beta}{2}$ of their points are non-trivial. G consists of more than half of the cubes, hence $1 - \frac{4}{\beta|\mathcal{F}|}$ of G's cubes have this property. Since $\frac{\beta}{2} > \frac{3\alpha}{2} > \alpha$ we deduce that $\mathcal{M}$ on the cube must be consistent with m on a non-trivial point - and must itself be non-trivial. ■

Proposition 11 *There exists an LDF P and a coefficient c_p such that*

$$\Pr_{C \in G}(P \textit{ appears in } C \textit{ with } c_p) > \delta \stackrel{def}{=} \frac{1}{O(s^{2 \cdot 6})}$$

Proof: Consider the following procedure:

1. For each non-trivial point $x \in \mathcal{D}$ choose a random value that appears in x.
2. For each non-trivial $\mathcal{C} \in G$ choose a random LDF that appears in it.
3. Choose a random cube $\mathcal{C} \in_R \mathcal{S}$ and a random point $x \in_R \mathcal{C}$.

We consider all pairs $(\mathcal{C}, x)$ of cube $\mathcal{C}$ and point $x \in \mathcal{C}$, and eliminate some of these pairs,

- Pairs $(\mathcal{C}, x)$ where $\mathcal{C} \notin G$ (α fraction).
- Pairs containing trivially assigned cubes ($\frac{2}{\beta|\mathcal{F}|}$ fraction by proposition 10).
- Pairs containing cubes with norm $> 10 \cdot s$ (no more than $\frac{1}{10}$ fraction).
- Inconsistent pairs $(\mathcal{C}, x)$ where $\pi_x(\mathcal{M}(\mathcal{C})) \neq m(x)$. (no more than α fraction).
- Ambiguous pairs $(\mathcal{C}, x)$ where $\mathcal{M}(\mathcal{C})$ is ambiguous on x. (no more than $amb(r, t+3, s) < |\mathcal{F}|^{-\frac{1}{2}}$).

With probability $\geq (1 - \alpha - \frac{2}{\beta|\mathcal{F}|} - \frac{1}{10} - \alpha - |\mathcal{F}|^{-\frac{1}{2}}) \geq \frac{1}{2}$ we have a pair of point and cube such that the norm of the cube is $\leq 10s$, and they agree non-ambiguously. For any value randomly chosen for a point, there must be a "matching value" from the cube. This value is chosen from the cube with probability at least $\frac{1}{10s}$. The norm of the cube is $\leq 10s$, there are $20s$ possible coefficients with which an *LDF* may appear. For a pair of a cube and a point that agree, we look at the coefficient of the agreed value in both of them (it is equal). There exists a coefficient c_p such that at least $\frac{1}{10s} \cdot \frac{1}{20s} = \frac{1}{200s^2}$ of the pairs not only agree but also have coefficient c_p.

We consider the procedure a success if the values are equal and appear with coefficient c_p. The success probability is therefore $\geq \frac{1}{2} \cdot \frac{1}{200s^2} = \frac{1}{O(s^2)}$. Using the cube vs. point lemma of [DFK$^+$98] (corollary of [RS96]), there exists an *LDF* P that agrees with $\geq \delta = \frac{1}{O(s^2)^6}$ of the cubes and their chosen values. ■

We have found a polynomial P that appears (with c_p) in a non-negligible fraction of the cubes. We now show that P, in fact, appears in most of the points with c_p.

Proposition 12 *P appears in most points with c_p*

Proof: Denote by N the set of points where P does not appear with coefficient c_p. We shall prove that $\mu \overset{def}{=} \frac{|N|}{|\mathcal{D}|} < \frac{1}{2}$. Using the hitting lemma (lemma 7), we have that $1 - \frac{4}{\mu|\mathcal{F}|}$ of the cubes in G have $\frac{\mu}{2}$ of their points from N. If $\frac{\mu}{2} > \alpha + amb(r, s, t)$ then every such cube must agree non-ambiguously with at least one point from N. This implies that P does not appear in these cubes with c_p, and hence $\delta < \frac{4}{\mu|\mathcal{F}|}$ (because of proposition 11). Altogether we have that

$$\mu \leq \max\left(2(amb(r, s, t) + \alpha), \frac{4}{\delta\,|\mathcal{F}|}\right) < \frac{1}{2}$$

■

Having P appearing in most points, we now show that P appears in most cubes with c_p.

Proposition 13 *P appears in $1 - |\mathcal{F}|^{-\frac{1}{2}}$ of the cubes of G with coefficient c_p.*

Proof: Denote by I the set of points where P appears with coefficient c_p. I has, by proposition 12, $\geq \frac{1}{2}$ fraction of the points. According to the hitting lemma, all except $\frac{4}{|\mathcal{F}|}$ of the cubes ($\frac{8}{|\mathcal{F}|}$ of G, since $|G| > \frac{1}{2}$), have $\frac{1}{4}$ of their points from I.

By the Markov inequality, at most $\frac{8rd}{|\mathcal{F}|} \cdot s$ of the cubes in G are assigned more than $\frac{|\mathcal{F}|}{8rd}$ polynomials. Therefore $1 - \frac{8srd}{|\mathcal{F}|} - \frac{8}{|\mathcal{F}|} \geq 1 - |\mathcal{F}|^{-\frac{1}{2}}$ of the cubes in G have $\frac{1}{4}$ of their points from I, and are assigned no more than $\frac{|\mathcal{F}|}{8rd}$ LDFs. Denote these cubes $G(P)$.

Let $\mathcal{C}$ be a cube in $G(P)$. The fraction of points of $\mathcal{C}$ which agree with $\mathcal{C}$ non-ambiguously and belong to I is at least $\frac{1}{4} - \alpha - amb(r, t+3, s) > \frac{1}{5}$. For each such point $x \in \mathcal{C}$, $\mathcal{M}(\mathcal{C})$ has an LDF Q, $Q(x) = P(x)$ such that the coefficient of Q is c_p. For every such point there are no more than $\frac{|\mathcal{F}|}{8rd}$ candidates, hence there is at least one *LDF* Q in $\mathcal{M}(\mathcal{C})$, which has the coefficient c_p, and is equal to P on at least

$$\frac{1}{5} \cdot \frac{8rd}{|\mathcal{F}|} > \frac{rd}{|\mathcal{F}|}$$

of $\mathcal{C}$'s points. This polynomial is therefore equal to P.

We have shown that the assignment of each cube in $G(P)$ contains P with coefficient c_p. Therefore all the good cubes but a $|\mathcal{F}|^{-\frac{1}{2}}$ fraction of the cubes are assigned P with coefficient c_p. ■

■

5.2 Proof of the Consistency Lemma (Lemma 2).

Lemma 2 (Consistency Lemma) *Let $\mathcal{T}$ be a set of t-tuples. Let $\mathcal{D}$ be a domain, and let $\mathcal{S}_\mathcal{T}$ be the set of $(t+3)$-cubes of $\mathcal{D}$ that contain the points of at least one tuple in $\mathcal{T}$. Let $\mathcal{M} : \mathcal{S}_\mathcal{T} \to SLDF_{r,t+3}$ be a super-assignment; $||\mathcal{M}|| \leq s < |\mathcal{F}|^{\frac{1}{100}}$. Let m be the underlying super-assignment to the points. Denote by $G \subset \mathcal{S}_\mathcal{T}$, the set of all cubes $\mathcal{C}$ for which $\mathcal{M}(\mathcal{C})$ agrees with m on at least $1-\alpha$ of $\mathcal{C}$'s points.*

If at least $1-\alpha$ of the cubes in $\mathcal{S}_\mathcal{T}$ are in G then there exists a global super-polynomial $\mathcal{G}$ of degree r on $\mathcal{D}$, with $||\mathcal{G}|| \leq 2s$ and

$$\Pr_{\mathcal{C}\in_R G}(\mathcal{M}(\mathcal{C}) = \pi_\mathcal{C}(\mathcal{G})) > 1 - |\mathcal{F}|^{-\frac{1}{2}}(2s+1)$$

Proof: The proof proceeds by induction. The inductive hypothesis for k is that the theorem is true for $s \leq \frac{k}{2}$.

The Base of the Induction ($k=1$). Assume $s < \frac{1}{2}$. Then no more than $\beta = \max(3\alpha, \frac{2}{|\mathcal{F}|})$ of the points have a non-trivial assignment: Otherwise, using the polynomial-extraction lemma (lemma 6) we extract a polynomial P that appears with coefficient $c_p \neq 0$ in $1 - |\mathcal{F}|^{-\frac{1}{2}}$ of the cubes, and deduce that the $||\mathcal{M}||$ is at least $|c_p|(1-|\mathcal{F}|^{-\frac{1}{2}}) > \frac{1}{2}$. Hence most of the points are trivial. Take $\mathcal{G}$ to be the trivial super-polynomial. By the hitting lemma, $1 - \frac{4}{|\mathcal{F}|} \geq 1 - |\mathcal{F}|^{-\frac{1}{2}}(2s+1)$ of the cubes in G are trivial, hence agree with $\mathcal{G}$. In addition $||\mathcal{G}|| = 0 \leq 2s$.

The Inductive Step. Assume $\frac{k-1}{2} \leq ||\mathcal{M}|| < \frac{k}{2}$. If less than β of the points have a non-trivial assignment we use the hitting lemma to deduce, as above, $||\mathcal{M}|| < \frac{1}{2}$, hence a contradiction. Otherwise, we use the extraction lemma to extract a polynomial P that appears with coefficient $c_p \neq 0$ in $1 - |\mathcal{F}|^{-\frac{1}{2}}$ of the cubes in G.

We now 'peal' P from $\mathcal{M}$ and m. Denote by $\mathcal{P}(P, c_p)$ the super-polynomial that gives a coefficient c_p to P, and zero to all other LDFs. We define new assignments to the points and cubes:

$$\mathcal{M}' \overset{def}{=} \mathcal{M} - \mathcal{P}(P, c_p), \quad m' \overset{def}{=} m - \mathcal{P}(P, c_p)$$

where this notation means that for every point or cube, the super-assignment to it is subtracted c_p in P's coefficient.

It now follows that $||\mathcal{M}'||$ is

$$\begin{aligned} ||\mathcal{M}'|| &\leq (1-|\mathcal{F}|^{-\frac{1}{2}})(||\mathcal{M}||-|c_p|)+|\mathcal{F}|^{-\frac{1}{2}}(||\mathcal{M}||+|c_p|) \\ &= ||\mathcal{M}||-(1-2\,|\mathcal{F}|^{-\frac{1}{2}})\,|c_p| \end{aligned}$$

It is easy to see that for every $\mathcal{C} \in G$, $\mathcal{M}'$ agrees with m' on at least $1-\alpha$ of $\mathcal{C}$'s points.

We therefore proceed by induction, to obtain a global super-polynomial $\mathcal{G}'$ that is consistent with m' and $\mathcal{M}'$. We claim that $\mathcal{G} \overset{def}{=} \mathcal{G}' + \mathcal{P}(P, c_p)$ is the desired super-polynomial:

1. The norm of $\mathcal{G}$, using the inductive hypothesis is,

$$\begin{aligned} ||\mathcal{G}|| &\leq ||\mathcal{G}'||+|c_p| \\ &\leq 2||\mathcal{M}'||+|c_p| \\ &\leq 2||\mathcal{M}||-2\cdot\frac{1}{2}\,|c_p|+|c_p| \\ &\leq 2||\mathcal{M}|| \end{aligned}$$

2. By the inductive hypotheses, $\mathcal{G}'$ agrees with $\mathcal{M}'$ in $1-(2s+1)\,|\mathcal{F}|^{-\frac{1}{2}}$ of the cubes. P appears in $\mathcal{M}$ with c_p in $1-|\mathcal{F}|^{-\frac{1}{2}}$ of the cubes. In the worst case, these cubes are disjoint, and $\mathcal{G}$ agrees with $\geq 1-(2(s-\frac{1}{2})+1)\,|\mathcal{F}|^{-\frac{1}{2}}-|\mathcal{F}|^{-\frac{1}{2}} \geq 1-(2s+1)\,|\mathcal{F}|^{-\frac{1}{2}}$ cubes.

■

References

[ABSS93] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes and linear equations. In *Proc. 34th IEEE Symp. on Foundations of Computer Science*, pages 724–733, 1993.

[AD96] M. Ajtai and S. Dwork. A public-key cryptosystem with worst-case average-case equivalence. ECCC, TR6-065, December 1996.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. In *Proc. 28th ACM Symp. on Theory of Computing*, 1996.

[Ajt97] M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions. manuscript, May 1997.

[ALM$^+$92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 13–22, 1992.

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 2–13, 1992.

[Bab86] L. Babai. On Lovász's lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–14, 1986.

[CN97] J. Y. Cai and A. Nerukar. Approximating the SVP to within a factor $1 + \frac{1}{dim^\epsilon}$ is NP-hard under randomized reductions. manuscript, 1997.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd ACM Symp. on Theory of Computing*, pages 151–158, 1971.

[DFK$^+$98] I. Dinur, E. Fischer, G. Kindler, R. Raz, and S. Safra. A near optimal PCP characterization of NP. manuscript, 1998.

[GG] O. Goldreich and S. Goldwasser. On the limits of non-approximability of lattice problems. ECCC, TR97-031.

[LLL82] A.K. Lenstra, H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:513–534, 1982.

[LLS90] J. Lagarias, H.W. Lenstra, and C.P. Schnorr. Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10:333–348, 1990.

[Mic98] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. In *Proc. 39th IEEE Symp. on Foundations of Computer Science*, 1998.

[RS96] R. Raz and S. Safra. A sub-constant error-probability PCP characterization of NP; part II: The consistency-test. Manuscript, 1996.

[Sch85] C.P. Schnorr. A hierarchy of polynomial-time basis reduction algorithms. In *Proceedings of Conference on Algorithms, Pécs (Hungary)*, pages 375–386. North-Holland, 1985.

[vEB81] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81–04, Math. Inst. Univ. Amsterdam, 1981.