2014 IEEE Annual Symposium on Foundations of Computer Science

Hardness of Coloring 2-Colorable 12-Uniform Hypergraphs with $2^{(\log n)^{\Omega(1)}}$ Colors

Subhash Khot*
New York University
New York, USA
Email: khot@cims.nyu.edu

Rishi Saket
IBM Research
Bangalore, India
Email: rissaket@in.ibm.com

***Abstract*—We show that it is quasi-NP-hard to color 2-colorable 12-uniform hypergraphs with $2^{(\log n)^{\Omega(1)}}$ colors where n is the number of vertices. Previously, Guruswami et al. [1] showed that it is quasi-NP-hard to color 2-colorable 8-uniform hypergraphs with $2^{2^{\Omega(\sqrt{\log \log n})}}$ colors. Their result is obtained by composing a standard Outer PCP with an Inner PCP based on the Short Code of super-constant degree. Our result is instead obtained by composing a new Outer PCP with an Inner PCP based on the Short Code of degree two.**

I. Introduction

A k-uniform hypergraph is a collection of vertices and hyperedges where each hyperedge is a subset of k vertices. An independent set in a hypergraph is a subset of vertices that does not contain any hyperedge completely inside it. A hypergraph is said to be q-colorable if the vertices can be partitioned into q disjoint independent sets, or equivalently if the vertices can be colored with q colors so that every edge is non-monochromatic. Coloring a hypergraph using *few* colors is one of the most well studied problems in combinatorics and theoretical computer science.

On graphs (i.e. $k = 2$), there is an efficient algorithm to determine 2-colorability, i.e. bipartiteness. A series of results – [2], [3], [4], [5], [6] and [7] – give efficient algorithms to color 3-colorable graphs with n^{β} colors, where the current best value of β is ≈ 0.2038. On the other hand, it is known to be NP-hard to color 3-colorable graphs with 4 colors [8], [9]. For q-colorable graphs with sufficiently large q, a lower bound of $2^{\Omega(q^{1/3})}$ colors was recently shown by Huang [10], improving upon an earlier bound of $q^{\Omega(\log q)}$ by Khot [11]. Dinur, Mossel and Regev [12] propose a variant of the Unique Games Conjecture referred to as the α-Conjecture and show hardness of coloring 3-colorable graphs with any constant number of colors under this conjecture.

Our understanding is much better for the problem of coloring q-colorable k-uniform hypergraphs with $k \geq 3$ (in this case, even determining 2-colorability is NP-hard). From the algorithmic side, the problem becomes only harder, so the best known algorithms still require $n^{\Omega(1)}$ colors, see Krivelevich et al. [13], Chen and Frieze [14], Kelsen et al. [15] and Kawarabayashi and Thorup [16]. From the hardness side, there has been steady progress on obtaining stronger and stronger results. We avoid giving a long list of all the known results for different values of q and k and instead refer to the respective papers [17], [18], [19], [1], [20], [21]. Here we focus on the case where q and k are allowed to be (preferably small) constants and the concern is obtaining quantitatively strong lower bounds on the number of colors used by efficient algorithms. Guruswami, Håstad and Sudan [22] proved the first superconstant bound, showing hardness of coloring 2-colorable 4-uniform hypergraphs with $\Omega\left(\frac{\log \log n}{\log \log \log n}\right)$ colors. Subsequently, Khot [23] showed the first poly-logarithmic bound, showing hardness of coloring q-colorable 4-uniform hypergraphs with $(\log n)^{\Omega(q)}$ colors where $q \geq 7$. In recent work, Guruswami et al. [1] obtained the first super-polylogarithmic bound, showing hardness of coloring 2-colorable 8-uniform hypergraphs with $2^{2^{\Omega(\sqrt{\log \log n})}}$ colors. The main result of this work is a further "exponential" improvement:

Theorem I.1. *For some absolute constant $c > 0$, it is* quasi-NP-hard[1] *to find an independent set of relative size $2^{-(\log n)^c}$ in an n-vertex 2-colorable 12-uniform hypergraph. Hence, it is* quasi-NP-hard *to color a 2-colorable 12-uniform hypergraph with $2^{(\log n)^c}$ colors. In particular, any $c < \frac{1}{20}$ works.*

We note that all results quoted above, with the exception of [18], also show hardness of finding an independent set of relative size $\delta(n)$ which in turn implies hardness of coloring with $1/\delta(n)$ colors. Our result takes us another step closer to the $n^{\Omega(1)}$ bound, which might perhaps be the truth. We further note that significantly stronger results are known for the case of *almost* coloring: a hypergraph is almost q-colorable if the removal of a small fraction of its vertices and incident hyperedges makes it q-colorable. Given an almost q-colorable graph with $q \geq 3$, it is known to be NP-hard to find an independent set of relative size $q^{-\lfloor \log_2 q \rfloor - 1}$ [24], [25] and of relative size $2^{-\frac{q}{2}}$ [26]. Given an almost 2-colorable 4-uniform hypergraph, it is known to be quasi-NP-hard to

*Research partly supported by NSF Expeditions grant CCF-0832795 and NSF Waterman Award.

[1]A problem is said to be *quasi-NP-hard* if it admits a DTIME($N^{\text{poly}(\log N)}$) reduction from 3SAT.

0272-5428/14 $31.00 © 2014 IEEE
DOI 10.1109/FOCS.2014.30

find an independent set of relative size $2^{-(\log n)^{1-o(1)}}$ [20].

Hardness results (including ours) are typically obtained by constructing a *probabilistically checkable proof* (PCP), letting the proof locations be vertices of a hypergraph and letting the tests (or rather the set of proof locations queried in a run of the test) be the hyperedges. Guruswami et al. [22] related the hardness of hypergraph coloring problem to the *covering complexity* of a PCP with the Not-All-Equal predicate. The covering complexity is k if (in the NO case) one needs at least k proofs so that every constraint is satisfied in at least one proof. The number of colors required to color the hypergraph is then 2^k. Dinur and Kol [27] study the covering complexity of general predicates. It is easily observed that the covering complexity is at most $O(\log n)$ where the PCP proof has size n and has $\mathsf{poly}(n)$ constraints. This is because if $O(\log n)$ random proofs were constructed, then with high probability over the choice of the proofs, every constraint is satisfied. In terms of covering complexity, ours is the first result to achieve a PCP with covering complexity that is polynomial in $\log n$, specifically $(\log n)^c$. This holds for the Not-All-Equal predicate of arity 12 (optimizing the exponent c and the arity 12 is not the focus of the paper; however the current techniques face a natural barrier of $\frac{1}{2}$ for the exponent c). We consider a new notion called *super-position complexity* of PCPs. Though it resembles the notion of covering complexity, there is no obvious upper bound better than n for the super-position complexity of a PCP. We work with this new notion for most of the paper and in the end show a hardness result for hypergraph coloring problem that amounts to a $(\log n)^c$ covering complexity result.

A. Overview of the Proof

Our hardness result follows from a long sequence of reductions, the initial steps presented as Theorems III.1, III.2 and III.7, with the remainder omitted due to lack of space. The reader is referred to the full version of this paper [28]. It is infeasible to give an overview of all the steps here, so we present only a high level view of some of the steps and emphasize some aspects in which our approach differs from the prior ones, in particular from that of Guruswami et al [1].

As mentioned before, hardness results (including ours) are typically obtained by constructing a probabilistically checkable proof and letting the proof locations be vertices of a hypergraph and letting the tests be the hyperedges. The PCP is typically viewed as a *composition* of an *outer verifier* with an *inner verifier*. The quantitative strength of the hardness result depends (mainly) on the efficiency of the inner verifier and in particular, on the efficiency (= length) of the *encoding scheme* used by the inner verifier. Several results – such as [22], [17], [23], [19], [21] – have been obtained using inner verifiers based on the *Long Code*. The Long Code of an m-bit string is a string of length 2^{2^m} and this leads to a large proof (= hypergraph) size, limiting the hardness result to a poly-logarithmic number of colors. At the other end of the spectrum, the *Hadamard Code* of an m-bit string is a string of length 2^m. Using an inner verifier based on the Hadamard Code, Khot and Saket [20] obtain a hardness result with $2^{(\log n)^{1-o(1)}}$ colors.[2] However, Hadamard Code can only incorporate (via a technique called *folding*) linear constraints and one is forced to use an underlying NP-hard problem with linear constraints. This forces the PCP to have *imperfect completeness* and one obtains a hardness result only for the *almost* coloring version of the problem. Recently, Barak et al. [29] proposed a new encoding scheme referred to as the *Short Code* that has length intermediate between the Hadamard Code and the Long Code. To encode an m-bit string u, the Hadamard Code writes down the value of all linear functions on u, whereas the Long Code writes down the value of all functions on u. The Short Code takes an intermediate route and writes down the value of all degree d functions on u for some constant d. The length of the encoding is $\approx 2^{m^d}$ and even though much less than the Long Code, it does increase rapidly for higher degree d. For $d \geq 2$, it allows one to incorporate (via folding) non-linear constraints and hence a PCP with *perfect completeness* is potentially feasible. In a recent work, Dinur and Guruswami [30] were indeed able to construct an inner verifier based on the Short Code and obtain hardness results for a variant of the hypergraph coloring problem. Guruswami et al. [1] were then able to adapt this Short Code based inner verifier for the hypergraph coloring problem, leading to the $2^{2^{\Omega(\sqrt{\log \log n})}}$ bound mentioned before. Their outer verifier is a standard one[3] and its composition with the inner verifier requires using a high degree d, limiting the quantitative bound to $2^{2^{\Omega(\sqrt{\log \log n})}}$ as stated.

Our key idea is to use, at the inner level, a *Quadratic Code* which is same as the Short Code with degree $d = 2$. This leads to a significant saving in the encoding length and we are able to obtain a $2^{(\log n)^{\Omega(1)}}$ bound. However, as we elaborate below, the composition now requires a much stronger guarantee from the outer verifier. The guarantee from the outer verifier is usually in terms of *low soundness*, but we need an additional guarantee that we refer to as the *high super-position complexity* (see below). Much of our effort is then invested in constructing such an outer verifier. We now describe the testing primitive used by the inner verifier and how its analysis motivates (and necessitates) the idea of super-position complexity.

We intend to use the Quadratic Code that encodes an m-bit input $u \in \mathbb{F}[2]^m$ by writing down the values of all quadratic functions on u. This is same as defining an

[2]This *almost polynomial factor* is a well-known barrier and should be considered as the best possible bound via the current technology.

[3]By a standard outer verifier we mean the 2-Prover-1-Round Game, a.k.a. Label Cover, instance obtained by parallel repetition of a *clause versus variable* game constructed from a Gap3SAT instance [31], [32], [33].

$m \times m$ matrix $M = u \otimes u$ and writing down the values of all linear functions on M (i.e. the Hadamard Code of M). The Quadratic Code is indexed by the set of all $m \times m$ matrices X and the value at location X is given by the entry-wise inner product $\langle M, X \rangle$. We describe a 6-query test to check whether a supposed code is indeed a Quadratic Code (in a loose, *list decoding* sense). It can be adapted, without much additional effort, to a 12-query test of an inner verifier, leading to a hardness result for coloring 12-uniform hypergraphs. This involves reading 6 queries each from two supposed codes and in addition to checking that these are indeed codewords, also checking that these are *consistent*.

The test is as follows. Pick matrices $X, Y, Z \in \mathbb{F}[2]^{m \times m}$ and vectors $a, b \in \mathbb{F}[2]^m$ uniformly and independently at random. Let $\mathsf{Diag}(a)$ be the diagonal matrix with a as the diagonal. Test whether,

$$[C(X) + C(X + \mathsf{Diag}(a))] \cdot [C(Y) + C(Y + \mathsf{Diag}(b))] = C(Z) + C(Z + a \otimes b).$$

It is easy to check that if C is the Quadratic Code of some $u \in \mathbb{F}[2]^m$, then the test always accepts. Indeed, letting $M = u \otimes u$, the right hand side of the equation is ($\langle u, a \rangle$ denotes the inner product over $\mathbb{F}[2]^m$)

$$\langle M, Z \rangle + \langle M, Z + a \otimes b \rangle = \langle M, a \otimes b \rangle = \langle u \otimes u, a \otimes b \rangle = \langle u, a \rangle \cdot \langle u, b \rangle,$$

whereas the left hand side evaluates to the same value:

$$\langle M, \mathsf{Diag}(a) \rangle \cdot \langle M, \mathsf{Diag}(b) \rangle = \langle u, a \rangle \cdot \langle u, b \rangle.$$

On the other hand, it can be shown, by an elementary Fourier analysis, that if the test passes with probability $\frac{1}{2} + 2^{-O(k)}$, then the given C-table can be decoded (by simply outputting a Fourier coefficient with significant magnitude) to a symmetric rank k matrix $\tilde{M}$. Writing $\tilde{M}$ as a super-position (i.e. sum) of k symmetric rank one matrices $\tilde{M} = \sum_{\ell=1}^{k} u^{(\ell)} \otimes u^{(\ell)}$, this amounts to decoding the C-table to a bounded list $u^{(1)}, \ldots, u^{(k)} \in \mathbb{F}[2]^m$ of inputs.[4]

Typically, the inner verifier also needs to check that the input u satisfies a constraint. In our setting, the constraint will be given as a quadratic equation, say $h(u) = 0$ for some quadratic polynomial h (assume for the ease of this overview that h has no constant term). Let's write the constraint as

$$\sum_{i,j=1}^{m} h_{i,j} u_i u_j = 0.$$

This amounts to a linear constraint $\langle H, M \rangle$ on the matrix $M = u \otimes u$ where $H = (h_{i,j})$ is also a matrix. If C is the Quadratic Code of u such that $h(u) = 0$, then it satisfies $C(X + H) = C(X)$ for every index X. We can ensure that the supposed code always satisfies this property by identifying the proof locations corresponding to $X + H$ and X for every index X. Also, since $M = u \otimes u$ is symmetric, one expects $C(X) = C(X^{\mathsf{T}})$ and this property can be ensured similarly. This trick is known as folding and its consequence is that the decoded matrix $\tilde{M}$ (as described above, by outputting a significant Fourier coefficient of the given C-table) is symmetric and satisfies the constraint $\langle H, M \rangle = 0$. Since $\tilde{M} = \sum_{\ell=1}^{k} u^{(\ell)} \otimes u^{(\ell)}$, this amounts to saying that

$$\sum_{i,j=1}^{m} h_{i,j} \left(\sum_{\ell=1}^{k} u_i^{(\ell)} u_j^{(\ell)} \right) = 0. \tag{1}$$

We say that the quadratic equation $h = 0$ is satisfied *in super-position* by the k inputs $u^{(1)}, \ldots, u^{(k)}$. In summary, the analysis of the inner verifier furnishes a short list of inputs that together satisfy the quadratic equation $h = 0$ in super-position, in the sense of Equation (1). This is an aspect in which our PCP differs from all earlier ones. In earlier PCPs, the inner verifier furnishes a short list of inputs such that *every* input in the list satisfies the relevant constraint whereas in our case, the constraint is only satisfied in super-position. To accommodate this weaker guarantee furnished by the inner verifier, the outer verifier needs a correspondingly stronger guarantee, which we refer to as the high super-position complexity.

We hope it is now clear why we need the outer verifier to have both the low soundness and high super-position complexity. We elaborate further on the latter property. As is standard, the outer verifier can be viewed as a 2-prover-1-round game where the first prover's answer is $u \in \mathbb{F}[2]^m$ and the second prover's answer is $v \in \mathbb{F}[2]^r$ (where $r \leq m$). The verifier accepts if $\pi(u) = v$ for some *projection map* $\pi : \mathbb{F}[2]^m \mapsto \mathbb{F}[2]^r$ that happens to be linear in our setting. In addition, the answer u must satisfy a quadratic equation $h(u) = 0$ for the verifier to accept. In the YES case, the provers have a strategy that makes the verifier accept with probability 1. In the NO case, the verifier accepts with negligible probability even under a looser criterion for acceptance. The provers are now allowed to furnish a short list $u^{(1)}, \ldots, u^{(k)}$ and $v^{(1)}, \ldots, v^{(k)}$ of answers respectively and the verifier accepts if $\pi(u^{(\ell)}) = v^{(\ell)}$ $\forall \ell \in \{1, \ldots, k\}$ and that $u^{(1)}, \ldots, u^{(k)}$ satisfy the constraint $h = 0$ in super-position. Once we have an outer verifier with such a guarantee, it is straightforward to compose it with the inner verifier described above.

The bulk of our paper is devoted to the construcion of the outer verifier which follows via a sequence of reductions (= PCPs), the initial steps presented as Theorems III.1, III.2 and III.7. We focus on constraint satisfaction problems where the constraints are quadratic equations over $\mathbb{F}[2]$. The super-position complexity of a CSP instance is the minimum number of assignments that satisfy every constraint in super-

[4]To express a symmetric rank k matrix as a sum of symmetric rank one matrices needs up to $\frac{3k}{2}$ summands, see Lemma II.1. We ignore this small issue here.

position in the sense of Equation (1). We start by showing that it is NP-hard to distinguish whether a CSP has super-position complexity of 1 or at least k (we choose the parameter k to be poly-logarithmic in the instance size though the result also holds for much higher settings of the parameter). This appears as Theorems III.1 and III.2. Interestingly, we do use some of the techniques from Dinur and Guruswami [30] here, specifically Lemma II.3 which in turn is based on techniques from [34] to test Reed-Muller codes over $\mathbb{F}[2]$. However, we emphasize that Dinur and Guruswami [30] employ these techniques in the analysis of the inner verifier whereas for us, these serve as a starting point in a long sequence of reductions.[5] We then use the ingredients used to prove the PCP Theorem (sum-check protocol, low degree test etc) to simultaneously reduce the *arity* of the constraints and to achieve *low soundness*, while preserving the high super-position complexity at every step. In the last step, the constraints are those given by a point-versus-surface low degree test and is naturally viewed as a 2-prover-1-round game, i.e. as the outer verifier. As mentioned before, the inner PCP is then based on the Quadratic Code. Its analysis is elementary and does not use any of the machinery required to analyze the Short Code. Subsequent to this work, Varma [35] has shown that the outer verifier can be combined with inner PCPs from [1] leading to similar inapproximability for the 2-colorable 8-uniform and 4-colorable 4-uniform cases.

II. Preliminaries

This section describes some useful tools that are used in the hardness reduction.

A. Tensor Decomposition of Symmetric Matrices

The following lemma shows a canonical way to write a symmetric matrix as a sum of symmetric rank one matrices. We only consider matrices over a field $\mathbb{F}[q]$ of characteristic 2.

Lemma II.1. *Given a symmetric matrix $A \in \mathbb{F}[q]^{m\times m}$ of rank k over a field $\mathbb{F}[q]$ of characteristic 2, there are k linearly independent vectors $\overline{z}_1, \ldots, \overline{z}_k \in \mathbb{F}[q]^m$ from the column space of A such that,*

$$A = \sum_{i=1}^{s} \overline{z}_i \otimes \overline{z}_i + \sum_{j=1}^{t} \left(\overline{z}_{s+2j-1} \otimes \overline{z}_{s+2j} + \overline{z}_{s+2j} \otimes \overline{z}_{s+2j-1} \right), \quad (2)$$

where $k = s + 2t$ for some non-negative integers s and t. This implies,

$$A = \sum_{i=1}^{s} \overline{z}_i \otimes \overline{z}_i + \sum_{j=1}^{t} \Big(\overline{z}_{s+2j-1} \otimes \overline{z}_{s+2j-1} + \overline{z}_{s+2j} \otimes \overline{z}_{s+2j} + (\overline{z}_{s+2j-1} + \overline{z}_{s+2j}) \otimes (\overline{z}_{s+2j-1} + \overline{z}_{s+2j}) \Big), \quad (3)$$

and that A is a sum of at most $\frac{3k}{2}$ symmetric rank one matrices.

Proof: Note that the second equation in the statement of the lemma follows from the first by observing that $\overline{a}\otimes\overline{b}+\overline{b}\otimes\overline{a} = \overline{a}\otimes\overline{a}+\overline{b}\otimes\overline{b}+(\overline{a}+\overline{b})\otimes(\overline{a}+\overline{b})$. So we focus on obtaining the decomposition as in the first equation. If $A = 0$, there is nothing to prove. If $A = (a_{ij}) \neq 0$, then we consider two cases and in each case, we give a decomposition of A into a single term in Equation (2) and a matrix of lower rank A'. The lemma then follows by an inductive argument on A'. We use a crucial fact that in a field $\mathbb{F}[q]$ of characteristic 2, every element is a square. In particular, for any $a \in \mathbb{F}[q]$, $a \neq 0$, the element $\frac{1}{\sqrt{a}}$ exists.

Case (i): Consider the case when A has a non-zero diagonal element, i.e. $a_{ii} \neq 0$ for some $i \in \{1, \ldots, m\}$. Let $\overline{a}_i$ be the i^{th} column of A and let $\overline{b}_i = \frac{1}{\sqrt{a_{ii}}} \cdot \overline{a}_i$. Consider the symmetric matrix,

$$A' = A + \overline{b}_i \otimes \overline{b}_i.$$

It is easy to see that the i^{th} column as well as row of A' is zero. This implies that $\overline{b}_i$ is linearly independent of the columns of A' and $\mathsf{rank}(A') = \mathsf{rank}(A) - 1$. We can then inductively decompose A' keeping in mind that the decomposition will involve vectors that are linearly independent of $\overline{b}_i$.

Case (ii): Now consider the case when all diagonal elements of A are zero, but there are indices $i \neq j$ such that $a_{ij} = a_{ji} \neq 0$. As before, let $\overline{b}_i = \frac{1}{\sqrt{a_{ij}}}\cdot\overline{a}_i$ and $\overline{b}_j = \frac{1}{\sqrt{a_{ij}}}\cdot\overline{a}_j$. Since $a_{ii} = a_{jj} = 0$, we have $\overline{b}_i \neq \overline{b}_j$. Consider the symmetric matrix

$$A' = A + \overline{b}_i \otimes \overline{b}_j + \overline{b}_j \otimes \overline{b}_i.$$

The i^{th} and the j^{th} columns as well as rows of A' are zero. This implies that $\overline{b}_i$ and $\overline{b}_j$ are linearly independent of the columns of A' and $\mathsf{rank}(A') = \mathsf{rank}(A) - 2$. We can then inductively decompose A' keeping in mind that the decomposition will involve vectors that are linearly independent of $\overline{b}_i$ and $\overline{b}_j$. ∎

B. Representations of Monomial Assignments

This and the next section describe the basic setup used by Dinur and Guruswami [30] for analyzing their inner verifier.

[5] One may view Dinur and Guruswami reduction as a sequence of four steps: NP-hardness of 3SAT (= Cook-Levin Theorem), NP-hardness of Gap3SAT (= the PCP Theorem), the Outer PCP and the inner PCP. With this viewpoint, the techniques referred to, are used by Dinur and Guruswami at the inner PCP level whereas we use them to prove the analogue of the Cook-Levin Theorem. We then naturally proceed to prove the analogue of the PCP Theorem.

Their verifier relies on the Short Code (we do not define it here since we won't be using it) that was proposed and analyzed by Barak et al. [29].

Let $x_1, \ldots, x_m$ be variables over $\mathbb{F}[2]$. Fix a degree parameter $d \geq 0$ and let $\mathcal{S}_d$ be the set of all monomials $\prod_{i \in S} x_i$ corresponding to subsets $S \subseteq [m]$ of size at most d. An assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$ is referred to as a *monomial assignment*. One can naturally extend assignment σ to all polynomials of degree at most d by linearity, i.e. if $q(x) = c + \sum_{S \subseteq [m], 1 \leq |S| \leq d} c_S \prod_{i \in S} x_i$ is a polynomial, then

$$\sigma(q) = c \cdot \sigma(\emptyset) + \sum_{S \subseteq [m], 1 \leq |S| \leq d} c_S \cdot \sigma\left(\prod_{i \in S} x_i\right),$$

where $\sigma(\emptyset)$ denotes the assignment given by σ to the empty monomial. We say that a monomial assignment σ satisfies an equation $q(x) = 0$, if $\sigma(q) = 0$.

Lemma II.2. *For any monomial assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$, there is a subset $\beta \subseteq \mathbb{F}[2]^m$ such that for all polynomials $q(x)$ of degree at most d,*

$$\sigma(q) = \sum_{a \in \beta} q(a). \tag{4}$$

Proof: Let $\mathcal{P}_d$ be the linear vector space of all polynomials $q(x)$ of degree at most d. The dimension of this space equals the number of monomials (including the empty monomial), i.e. $\sum_{i=0}^{d} \binom{m}{i}$. Let $\mathcal{A}$ be the set of all inputs $a \in \mathbb{F}[2]^m$ with Hamming weight at most d so that $|\mathcal{A}| = \sum_{i=0}^{d} \binom{m}{i}$ is same as the dimension of $\mathcal{P}_d$. For every fixed $a \in \mathbb{F}[2]^m$, the map $q(x) \mapsto q(a)$ is a linear map on $\mathcal{P}_d$. We will show that these maps are linearly independent and hence form a basis for the space of all linear maps on $\mathcal{P}_d$ and in particular, the linear map σ can be expressed as their linear combination, proving the lemma. In order to show the linear independence of the maps $\{q(x) \mapsto q(a) \mid a \in \mathcal{A}\}$, it suffices to show that if a degree (at most) d polynomial $q(x)$ vanishes on all inputs in $\mathcal{A}$, then it vanishes identically. Indeed, if on the contrary, $q(x) \neq 0$, then $q(x) = \prod_{i \in S} x_i + \sum_{S' \neq S} c_{S'} \prod_{j \in S'} x_j$ where $\prod_{i \in S} x_i$ is a monomial of lowest degree that has a non-zero coefficient in $q(x)$. Clearly, for the input $a \in \mathbb{F}[2]^m$ whose non-zero co-ordinates are precisely on the set S, we have $q(a) \neq 0$ reaching a contradiction. ∎

Note that the subset β guaranteed by Lemma II.2 need not be unique. For a monomial assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$, let β_σ denote a minimum sized subset β satisfying the conclusion of the lemma (i.e. Equation (4)).

C. A Useful Tool from Dinur and Guruswami [30]

We now state (a minor variant of) the main tool we borrow from Dinur and Guruswami [30] paper. Let $\mathcal{F}_m$ be the space of all functions $f : \mathbb{F}[2]^m \mapsto \mathbb{F}[2]$. For a subset $\beta \subseteq \mathbb{F}[2]^m$, define the character $\chi_\beta : \mathcal{F}_m \mapsto \{-1, 1\}$ as:

$$\begin{aligned}\chi_\beta(f) = (-1)^{\sum_{x \in \beta} f(x)} &= (-1)^{\sum_{x \in \mathbb{F}[2]^m} \mathbb{1}_\beta(x) f(x)} \\ &= (-1)^{\langle \mathbb{1}_\beta, f \rangle},\end{aligned}$$

where $\mathbb{1}_\beta$ denotes the indicator function of that subset. If β_σ is a (minimum sized) subset corresponding to a monomial assignment σ as defined earlier, then for any polynomial g of degree at most d,

$$\chi_{\beta_\sigma}(g) = (-1)^{\sum_{x \in \beta_\sigma} g(x)} = (-1)^{\sigma(g)}.$$

The following is a minor variant of a theorem proved in [30]. The ideas in its proof go back to the analysis of testing Reed-Muller codes in [34].

Lemma II.3. *Let $\beta = \beta_\sigma$ be a (minimum sized) subset corresponding to some monomial assignment σ such that $|\beta| \geq 2^{d/2}$ and $\alpha, \gamma \subseteq \mathbb{F}[2]^m$ are arbitrary. Then*

$$\left|\mathbb{E}_{g,h}\left[\chi_\beta(gh)\chi_\gamma(g)\chi_\alpha(h)\right]\right| \leq 2^{-2^{d/4-2}},$$

where g is a uniformly random polynomial of degree at most $3d/4$ and h is a uniformly random polynomial of degree at most $d/4$.

Proof: The expectation can be upper bounded by

$$\mathbb{E}_h\left[\left|\mathbb{E}_g\left[\chi_\beta(gh)\chi_\gamma(g)\right]\right|\right]. \tag{5}$$

The inner expectation is same as

$$\mathbb{E}_g\left[\chi_\beta(gh)\chi_\gamma(g)\right] = \mathbb{E}_g\left[(-1)^{\langle \mathbb{1}_\beta \cdot h + \mathbb{1}_\gamma, g \rangle}\right]. \tag{6}$$

We use the fact that the space of polynomials of degree at most $m - 3d/4 - 1$ is precisely the orthogonal space of the space of polynomials of degree at most $3d/4$. Thus the expectation in Equation (6) is 1 if $\mathbb{1}_\beta \cdot h + \mathbb{1}_\gamma$ is a polynomial of degree at most $m - 3d/4 - 1$ and zero otherwise. Hence the expression in Equation (5) is same as

$$\Pr_h[\mathbb{1}_\beta \cdot h + \mathbb{1}_\gamma \text{ is a polynomial of degree at most } m - 3d/4 - 1],$$

where h is a random polynomial of degree at most $d/4$. By Lemma II.5, this probability is upper bounded by $2^{-2^{d/4-2}}$. ∎

Lemma II.5 is an immediate consequence of a similar lemma in [30].

Lemma II.4. *For a uniformly random polynomial h of degree at most $d/4$ and β such that $|\beta| \geq 2^{d/2}$,*

$$\Pr_h[\mathbb{1}_\beta \cdot h \text{ is a polynomial of degree at most } m - 3d/4 - 1] \leq 2^{-2^{d/4-2}}.$$

Lemma II.5. *For a uniformly random polynomial h of degree at most $d/4$ and any γ, β such that $|\beta| \geq 2^{d/2}$,*

$$\Pr_h[\mathbb{1}_\beta \cdot h + \mathbb{1}_\gamma \text{ is a polynomial of degree at most } m - 3d/4 - 1] \leq 2^{-2^{d/4-2}}.$$

Proof: If there is no h such that $\mathbb{1}_\beta \cdot h + \mathbb{1}_\gamma$ is a polynomial of degree at most $m - 3d/4 - 1$ then we are done. Otherwise the set of all such h is an affine subspace and translating it to include the origin yields the subspace (of the same size) of h' such that $\mathbb{1}_\beta \cdot h'$ is a polynomial of degree at most $m - 3d/4 - 1$. An appeal to Lemma II.4 completes the proof. ∎

D. Arora-Sudan Analysis of the Low Degree Test

Let $\mathbb{F}[q]$ be a field and d, m be positive integers. Suppose we are given a table of values of a function $f : \mathbb{F}[q]^m \mapsto \mathbb{F}[q]$ that is supposed to be a degree d polynomial. Suppose, in addition, we are given, for every line ℓ in the space $\mathbb{F}[q]^m$, a univariate degree d polynomial f_ℓ that is supposed to be the restriction of the supposed global polynomial f to that line.[6] For a point $\overline{v}$ on the line ℓ, we denote by $f_\ell(\overline{v})$ the value given by f_ℓ at the point $\overline{v}$. The following theorem was proved by Arora and Sudan [36].

Theorem II.6. *There are constants $c_0, c_1, c_2, c_3 > 0$ such that the following holds. For any parameter $\delta > 0$ such that $q \geq c_0(dm/\delta)^{c_1}$, let $\{P_1, \ldots, P_t\}$ be the set of degree d polynomials that agree with f at δ^{c_2}/c_3 fraction of the points. Then, taking the probability over a random line ℓ and random point $\overline{v}$ on the line,*

$$\Pr_{\ell, \overline{v}}[f(\overline{v}) \notin \{P_1(\overline{v}), \ldots, P_t(\overline{v})\} \text{ and } f_\ell(\overline{v}) = f(\overline{v})] \leq \delta.$$

Also, by coding theoretic bounds $t \leq 2c_3/\delta^{c_2}$.

E. Super-position Complexity

Definition II.7. *Let $a^{(1)}, \ldots, a^{(t)} \in \mathbb{F}[2]^m$ be t assignments and $q(x) = 0$ be a quadratic equation in m boolean variables with $q(x) = c + \sum_{i=1}^m c_i x_i + \sum_{1 \leq i < j \leq m} c_{ij} x_i x_j$. We say that the t assignments satisfy the equation $q(x) = 0$ in super-position if*

$$c + \sum_{i=1}^m c_i \left(\sum_{\ell=1}^t a_i^{(\ell)} \right) + \sum_{1 \leq i < j \leq m} c_{ij} \left(\sum_{\ell=1}^t a_i^{(\ell)} a_j^{(\ell)} \right) = 0.$$

Note that for $t = 1$, this is same as saying that $q(a^{(1)}) = 0$, i.e. that $a^{(1)}$ satisfies the equation (in the standard sense). Also, if $q(x)$ is linear, this is same as saying that the assignment $a = \sum_{\ell=1}^t a^{(\ell)}$ satisfies the equation (in the standard sense).

[6] A line is a set $\ell(t) = \overline{\alpha} + t\overline{\beta}$ parameterized by $t \in \mathbb{F}[q]$ for some $\overline{\alpha}, \overline{\beta} \in \mathbb{F}[q]^m$.

Definition II.8. *Given a system of quadratic equations $\{q_i(x) = 0\}_{i=1}^L$, its super-position complexity is the minimum number t, if it exists, such that there are t assignments $a^{(1)}, \ldots, a^{(t)} \in \mathbb{F}[2]^m$ that satisfy every equation $q_i(x) = 0$, $i \in \{1, \ldots, L\}$ in super-position. Otherwise, one may define the super-position complexity to be ∞ (but we will not encounter this scenario).*

III. Starting Point for Our PCPs

In this section, we describe a set of results that serve as the starting point for our PCPs. The main theorem is Theorem III.2 that provides a *super-position gap* for constraint satisfaction problems with constraints that are quadratic equations over $\mathbb{F}[2]$. The theorem states that given an instance of such a CSP, it is NP-hard to distinguish whether it has a satisfying assignment (i.e. has super-position complexity of 1) or has high super-position complexity. Theorem III.1 is a preparatory step towards the main Theorem III.2. For subsequent applications, we need certain strengthenings of the main theorem stated as Theorem III.5 and III.7.

A. CSPs with High Degree Equations

Recall that given n boolean variables $x_1, \ldots, x_n$ and the degree parameter d, $\mathcal{S}_d$ denotes the set of all monomials of size at most d over the n variables. Given a monomial assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$, one can extend it naturally to all polynomials of degree at most d by linearity. Moreover there exists a set $\beta_\sigma \subseteq \mathbb{F}[2]^n$ (of minimal size, by definition) such that for all polynomials $q(x)$ of degree at most d, $\sigma(q) = \sum_{s \in \beta_\sigma} q(s)$. A monomial assignment σ is said to satisfy a system of degree d polynomial equations $\{q_i(x) = 0\}_{i=1}^m$ if $\sigma(q_i) = 0$ for every $i \in \{1, \ldots, m\}$. We prove the following theorem in this section.

Theorem III.1. *For any $d \geq 3$, there is a DTIME($n^{O(d)}$) reduction from 3SAT to a system $\mathcal{B}$ of degree d equations over $\mathbb{F}[2]$ such that,*

YES Case: If the 3SAT instance is satisfiable then there is an assignment that satisfies (all equations in) $\mathcal{B}$.

NO Case: If the 3SAT instance is unsatisfiable then for any monomial assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$ that satisfies (all equations in) $\mathcal{B}$, one must have $|\beta_\sigma| \geq 2^{d-3}$.

Proof: Suppose the 3SAT instance consists of n boolean variables $x_1, \ldots, x_n$ and m clauses. For $i = 1, \ldots, m$, the i^{th} clause can be written as an equation $p_i(x) = 0$ where $p_i(x)$ is a polynomial of degree at most 3. It depends on at most 3 variables, but this will not be relevant to us. Let $d \geq 3$ be as in the statement of the theorem. We construct a system $\mathcal{B}$ of equations as desired by adding the equation

$$\left(\prod_{i \in S} x_i \right) p_i(x) = 0,$$

for all monomials $\prod_{i\in S} x_i$ of degree at most $d-3$ and $i = 1,\ldots,m$. Note that every equation in $\mathcal{B}$ has degree at most d. In the YES case, if the 3SAT instance has a satisfying assignment, then clearly the same assignment satisfies all equations in $\mathcal{B}$. So we focus on the NO case. Let a monomial assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$ be given that satisfies all equations in $\mathcal{B}$ and let $\beta_\sigma \subseteq \mathbb{F}[2]^n$ be the corresponding set. Note that for any polynomial $q(x)$ of degree at most $d-3$ and any $i \in \{1,\ldots,m\}$, the equation $q(x)p_i(x) = 0$ is a linear combination of equations in $\mathcal{B}$ and hence must be satisfied by σ, i.e. $\sigma(q \cdot p_i) = 0$.

Assume for the sake of contradiction that $|\beta_\sigma| < 2^{d-3}$. Fix an arbitrary $a \in \beta_\sigma$. By Lemma 2.13 of [1], there exists a polynomial $q(x)$ of degree at most $d-3$ such that $q(a) = 1$ and $\forall b \in \beta_\sigma, b \neq a, q(b) = 0$. Since the 3SAT instance is unsatisfiable, the assignment a fails on some, say j^{th}, clause, i.e. $p_j(a) = 1$. We reach a contradiction by observing that

$$\begin{aligned}\sigma(q \cdot p_j) &= \sum_{s\in\beta_\sigma} q(s)p_j(s) \\ &= q(a)p_j(a) + \sum_{s\in\beta_\sigma, s\neq a} q(s)p_j(s) = p_j(a) = 1.\end{aligned}$$

■

B. Quadratic CSP with Superposition Gap

We recall Definition II.7 and prove our main theorem in this section.

Theorem III.2. *There is a reduction from* 3SAT *to an instance $\mathcal{A}$ of quadratic equations such that,*

YES Case. *If the* 3SAT *instance is satisfiable then there is an assignment to $\mathcal{A}$ that satisfies all the equations.*

NO Case. *If the* 3SAT *instance is unsatisfiable then there are no t assignments to $\mathcal{A}$ that satisfy all the equations simultaneously in super-position for any $1 \leq t \leq k$. Here k is a parameter and the reduction runs in time $N^{O(\log k)}$ where N is the size of the* 3SAT *instance.*

Proof: We first reduce 3SAT to a system of degree d equations $\mathcal{B}$ as given by Theorem III.1. The size of instance $\mathcal{B}$ is $N^{O(d)}$ where N is the size of the 3SAT instance. Let $x_1,\ldots,x_n$ be the variables of the instance $\mathcal{B}$ and the degree parameter d will be set later. Note that in the YES case, the instance $\mathcal{B}$ has a satisfying assignment $a \in \mathbb{F}[2]^n$ whereas in the NO case, for any assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$ that satisfies $\mathcal{B}$, it must be that $|\beta_\sigma| \geq 2^{d-3}$ (to recall again, $\mathcal{S}_d$ is the set of all monomials over variables $x_1,\ldots,x_n$ of degree at most d). We construct the desired system $\mathcal{A}$ of quadratic equations as follows.

- For every $A \subseteq [n]$, $1 \leq |A| \leq d$ we have a variable y_A. This variable is supposed to represent the monomial $\prod_{i\in A} x_i$ and in the YES case, it takes the same value as this monomial under a satisfying assignment to $\mathcal{B}$. Note that we have variables corresponding only to the non-empty monomials.
- Add all the equations of $\mathcal{B}$ replacing each non-empty monomial $\prod_{i\in A} x_i$ by the corresponding variable y_A. These equations are linear in the variables $\{y_A \mid 1 \leq |A| \leq d\}$ (so this is simply a linearization of $\mathcal{B}$).
- For every pair $A, B \subseteq [n]$ such that $1 \leq |A|, |B|, |A \cup B| \leq d$, add the quadratic equation $y_A y_B = y_{A\cup B}$. Note that this quadratic equation is indeed satisfied in the YES case since the variables y_A have values same as the corresponding monomials under an assignment to the variables $x_1,\ldots,x_n$.

This completes the construction of the instance $\mathcal{A}$. In the YES case, taking the satisfying assignment $a \in \mathbb{F}[2]^n$ to instance $\mathcal{B}$ and assigning to every variable y_A the value $\prod_{i\in A} a_i$ satisfies all equations of instance $\mathcal{A}$.

In the NO case, we wish to show that no t assignments $\sigma_1,\ldots,\sigma_t : \{y_A \mid 1 \leq |A| \leq d\} \mapsto \mathbb{F}[2]$ can satisfy all equations of $\mathcal{A}$ in super-position for any $1 \leq t \leq k$. We can assume that t is odd by adding (if necessary) the assignment which maps each y_A to 0. This increases t by at most 1 and makes no difference to the quantitative bounds. Assume for a contradiction that such a set of t assignments exists, for an odd t. Note that any assignment σ_i is naturally also a monomial assignment $\sigma_i : \mathcal{S}_d \mapsto \mathbb{F}[2]$, by extending it to the empty monomial as $\sigma_i(\emptyset) = 1$.

Letting $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$ be the monomial assignment given by $\sigma = \sum_{i=1}^t \sigma_i$ we have the following lemma.

Lemma III.3. *The monomial assignment $\sigma : \mathcal{S}_d \mapsto \mathbb{F}[2]$ satisfies all equations of $\mathcal{B}$.*

Proof: Let $q(x) = 0$ be an equation in $\mathcal{B}$ where $q(x) = c + \sum_{A\subseteq[n],1\leq|A|\leq n} c_A \cdot \left(\prod_{i\in A} x_i\right)$. Its linearization in $\mathcal{A}$ is,

$$c + \sum_{A\subseteq[n],1\leq|A|\leq n} c_A y_A = 0.$$

As the above equation is satisfied in super-position by the assignments $\{\sigma_i\}_{i=1}^t$ to the variables y_A,

$$c + \sum_{A\subseteq[n],1\leq|A|\leq n} c_A \cdot \left[\sum_{i=1}^t \sigma_i(y_A)\right] = 0.$$

Viewing σ_i as a monomial assignment and observing that since t is odd, $\sum_{i=1}^t \sigma_i(\emptyset) = \sum_{i=1}^t 1 = 1$, the above implies,

$$c\cdot\left(\sum_{i=1}^t \sigma_i(\emptyset)\right) + \sum_{A\subseteq[n],1\leq|A|\leq n} c_A \cdot \left[\sum_{i=1}^t \sigma_i\left(\prod_{i\in A} y_A\right)\right] = 0.$$

By the definition of σ we have,

$$c\cdot\sigma(\emptyset) + \sum_{A\subseteq[n],1\leq|A|\leq n} c_A \cdot \sigma\left(\prod_{i\in A} y_A\right) = \sigma(q) = 0.$$

■

From Lemma II.2 there exist subsets $\beta_{\sigma_i} \subseteq \mathbb{F}[2]^n$ corresponding to the monomial assignments σ_i for $i = 1, \ldots, t$ and a subset β_σ corresponding to σ. As proved in Lemma III.3 above σ satisfies all equations in $\mathcal{B}$. Hence, by the guarantee offered by the NO case of Theorem III.1, we have $|\beta_\sigma| \geq 2^{d-3}$. We now show that $|\beta_\sigma|$ being large implies that $\sigma_1, \ldots, \sigma_t : \{y_A \mid 1 \leq |A| \leq d\} \mapsto \mathbb{F}[2]$ cannot simultaneously satisfy the equations $y_A y_B = y_{A \cup B}$ in super-position. Assume on the contrary that this is the case, i.e. for all A, B such that $1 \leq |A|, |B|, |A \cup B| \leq d$,

$$\sum_{i=1}^{t} \sigma_i(y_{A \cup B}) = \sum_{i=1}^{t} \sigma_i(y_A)\sigma_i(y_B).$$

Since σ_i are also thought of as monomial assignments $\sigma_i : \mathcal{S}_d \mapsto \mathbb{F}[2]$, the above amounts to saying that

$$\sum_{i=1}^{t} \sigma_i(gh) = \sum_{i=1}^{t} \sigma_i(g)\sigma_i(h), \tag{7}$$

where g, h are non-empty monomials in variables $x_1, \ldots, x_n$ such that the sizes of g, h, gh are all upper bounded by d. Further, as monomial assignments defined above, $\sigma_i(\emptyset) = 1$ for all $i = 1, \ldots, t$ which impies that Equation (7) holds for *all* monomials g, h such that the sizes of g, h and gh are bounded by d. In particular, this holds whenever g and h are monomials of degree at most $3d/4$ and $d/4$ respectively (assume d is divisible by 4). We observe that by linearity, Equation (7) holds also when g is a polynomial of degree at most $3d/4$ and h is a polynomial of degree at most $d/4$.

We switch from values over $\mathbb{F}[2]$ to real values in $\{-1, 1\}$, i.e. replace $\sigma_i(g)$ by $(-1)^{\sigma_i(g)}$. Noting that $\sigma = \sum_{i=1}^{t} \sigma_i$, we get

$$(-1)^{\sigma(gh)} = \prod_{i=1}^{t} \left((-1)^{\sigma_i(g)} \wedge (-1)^{\sigma_i(h)} \right).$$

Note that addition over $\mathbb{F}[2]$ now becomes multiplication over signs $\{-1, 1\}$ and multiplication over $\mathbb{F}[2]$ now becomes the operation $a \wedge b = (1 + a + b - ab)/2$ over signs $\{-1, 1\}$. Since $(-1)^{\sigma_i(g)} = \chi_{\beta_{\sigma_i}}(g)$, we get that

$$\chi_{\beta_\sigma}(gh) \left[\prod_{i=1}^{t} (\chi_{\beta_{\sigma_i}}(g) \wedge \chi_{\beta_{\sigma_i}}(h)) \right] = 1, \tag{8}$$

whenever g is a polynomial of degree at most $3d/4$ and h is a polynomial of degree at most $d/4$. We reach a contradiction by showing that if g and h are chosen as random polynomials of the kind prescribed, the expectation of the left hand side of Equation (8) is nearly zero. Indeed, replacing each expression $a \wedge b$ by $(1 + a + b - ab)/2$ and expanding the product into a sum of 4^t terms, the left hand side of Equation (8) is a sum of 4^t terms of type

$$\left(\frac{1}{2^t} \right) \chi_{\beta_\sigma}(gh)\chi_\gamma(g)\chi_\alpha(h),$$

for some $\gamma, \alpha \subseteq \mathbb{F}[2]^n$. The sets γ, α are related to the sets β_{σ_i}, but this is not relevant for the argument. We finish the proof by showing that the expectation of the term above is negligible and hence the sum of the expectations of the 4^t terms is negligible too. The claim follows by Lemma III.4 below. It is enough to take $d = O(\log k)$. ∎

Lemma III.4. *For $\beta_\sigma \subseteq \mathbb{F}[2]^n$, $|\beta_\sigma| \geq 2^{d-3}$, $d \geq 6$ and arbitrary $\gamma, \alpha \subseteq \mathbb{F}[2]^n$, we have*

$$|\mathbb{E}_{g,h}[\chi_{\beta_\sigma}(gh)\chi_\gamma(g)\chi_\alpha(h)]| \leq 2^{-2^{d/4-2}},$$

where g is a random polynomial of degree at most $3d/4$ and h is a random polynomial of degree at most $d/4$.

Proof: For $d \geq 6$, we have $2^{d-3} \geq 2^{d/2}$. The proof follows from Lemma II.3. ∎

C. Strengthening of Theorem III.2

We will need to consider quadratic equations over $\mathbb{F}[q]$ that is an extension field of $\mathbb{F}[2]$. In particular we need analogue of Theorem III.2 where the conclusion holds even for $\mathbb{F}[q]$-valued assignments. In this section, while considering quadratic equations over $\mathbb{F}[q]$, we only consider equations that have $\mathbb{F}[2]$ coefficients and no linear terms, i.e. equations of the form $c + \sum_{1 \leq i \leq j \leq m} c_{ij} x_i x_j = 0$ where $c, c_{ij} \in \mathbb{F}[2]$. The notion of satisfying an equation in super-position is similar as before. Assignments $a^{(1)}, \ldots, a^{(t)} \in \mathbb{F}[q]^m$ are said to satisfy an equation $c + \sum_{1 \leq i \leq j \leq m} c_{ij} x_i x_j = 0$ in super-position if,

$$c + \sum_{1 \leq i \leq j \leq m} c_{ij} \left(\sum_{\ell=1}^{t} a_i^{(\ell)} a_j^{(\ell)} \right) = 0.$$

Theorem III.2 easily implies the theorem below.

Theorem III.5. *Let $\mathbb{F}[q]$ be an extension field of $\mathbb{F}[2]$ with $q = 2^r$. There is a reduction from* 3SAT *to an instance $\mathcal{C}$ of quadratic equations over $\mathbb{F}[q]$ such that*

- *The equations have $\mathbb{F}[2]$ coefficients and no linear terms.*
- YES Case. *If the* 3SAT *instance is satisfiable then there is an assignment to $\mathcal{C}$ that satisfies all the equations. In fact there is such an assignment that is $\mathbb{F}[2]$ valued.*
- NO Case. *If the* 3SAT *instance is unsatisfiable then there are no t assignments to $\mathcal{C}$ that are $\mathbb{F}[q]$ valued and satisfy all the equations simultaneously in super-position for any $1 \leq t \leq k$. Here k is a parameter and the reduction runs in time $N^{O(r \log k)}$ where N is the size of the* 3SAT *instance.*

Proof: The instance $\mathcal{C}$ is essentially the same as the instance $\mathcal{A}$ given by Theorem III.2. The only difference is that every linear term x_i is replaced by a quadratic term x_i^2. Specifically, an equation $c + \sum_{i=1}^{m} c_i x_i + \sum_{1 \leq i \leq j \leq m} c_{ij} x_i x_j = 0$ in instance $\mathcal{A}$ is now written as $c + \sum_{1 \leq i \leq j \leq m} c_{ij} x_i x_j = 0$ in instance $\mathcal{C}$ where $c_{ii} = c_i$.

The claim in the YES case follows from the analogous claim in Theorem III.2, so we focus on the NO case.

We show that if there are t assignments over $\mathbb{F}[q]$ that satisfy all equations in the instance $\mathcal{C}$ in super-position, then there are $t \cdot s$ assignments over $\mathbb{F}[2]$ that satisfy all equations in the instance $\mathcal{A}$ in super-position and $s \leq 2r$. Let a typical equation in the instance $\mathcal{C}$ be $c + \sum_{1 \leq i \leq j \leq m} c_{ij} x_i x_j = 0$, where $c, c_{ij} \in \mathbb{F}[2]$. Suppose there are $\mathbb{F}[q]$-valued assignments $a^{(1)}, \ldots, a^{(t)} \in \mathbb{F}[q]^m$ that satisfy the equation in super-position, i.e.

$$c + \sum_{1 \leq i \leq j \leq m} c_{ij} \left(\sum_{\ell=1}^{t} a_i^{(\ell)} a_j^{(\ell)} \right) = 0.$$

The computations above are over $\mathbb{F}[q]$. Fixing an arbitrary representation of $\mathbb{F}[q]$ as a r-dimensional vector space over $\mathbb{F}[2]$, the above equation must hold in the last bit of the vector representation, i.e. in the notation of Lemma III.6,

$$c + \sum_{1 \leq i \leq j \leq m} c_{ij} \left(\sum_{\ell=1}^{t} (a_i^{(\ell)} a_j^{(\ell)})_{\mathsf{last}} \right) = 0.$$

However by Lemma III.6, there are vectors $\lambda_1, \ldots, \lambda_s \in \mathbb{F}[2]^r$, that *capture* the computation of the last bit of a product of two elements in $\mathbb{F}[q]$. Hence, the above equation can be written as

$$c + \sum_{1 \leq i \leq j \leq m} c_{ij} \left(\sum_{\ell=1}^{t} \sum_{p=1}^{s} \langle a_i^{(\ell)}, \lambda_p \rangle \cdot \langle a_j^{(\ell)}, \lambda_p \rangle \right) = 0.$$

Since all values now are in $\mathbb{F}[2]$, we can separate the diagonal terms and re-write them as linear terms (note $c_i = c_{ii}$), i.e.

$$c + \sum_{1 \leq i \leq m} c_i \left(\sum_{\ell=1}^{t} \sum_{p=1}^{s} \langle a_i^{(\ell)}, \lambda_p \rangle \right) + \sum_{1 \leq i < j \leq m} c_{ij} \left(\sum_{\ell=1}^{t} \sum_{p=1}^{s} \langle a_i^{(\ell)}, \lambda_p \rangle \cdot \langle a_j^{(\ell)}, \lambda_p \rangle \right) = 0.$$

This is same as saying that the $t \cdot s$ many $\mathbb{F}[2]$-valued assignments given by $\langle a^{(\ell)}, \lambda_p \rangle$ for $\ell \in [t], p \in [s]$ satisfy the corresponding equation in the instance $\mathcal{A}$ in super-position. Noting that the choice of the equation is arbitrary, the theorem follows by the guarantee on the NO case in Theorem III.2. ∎

Lemma III.6. *Let $\mathbb{F}[q]$ be an extension field of $\mathbb{F}[2]$ with $q = 2^r$. Any $x \in \mathbb{F}[q]$ can be thought of as a (row) vector in $\mathbb{F}[2]^r$ in some fixed representation of $\mathbb{F}[q]$ as a r-dimensional vector space over $\mathbb{F}[2]$. For $x \in \mathbb{F}[q]$, let $(x)_{\mathsf{last}}$ denote the last bit of the corresponding vector. Then there exist vectors $\lambda_1, \ldots, \lambda_s \in \mathbb{F}[2]^r$, $s \leq 2r$ such that*

$$\forall x, y \in \mathbb{F}[q] \qquad (xy)_{\mathsf{last}} = \sum_{i=1}^{s} \langle x, \lambda_i \rangle \cdot \langle y, \lambda_i \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product over $\mathbb{F}[2]^r$ and while computing the expression $\langle x, \lambda_i \rangle$, x is being thought of as a vector in $\mathbb{F}[2]^r$.

Proof: The map $(x, y) \mapsto (xy)_{\mathsf{last}}$ can be thought of as a symmetric bilinear map $\mathbb{F}[2]^r \times \mathbb{F}[2]^r \mapsto \mathbb{F}[2]$. Hence there is a $r \times r$ symmetric matrix Λ over $\mathbb{F}[2]$ such that

$$\forall x, y \in \mathbb{F}[q] \qquad (xy)_{\mathsf{last}} = x \cdot \Lambda \cdot y^T.$$

The matrix Λ can be written as $\sum_{i=1}^{s} \lambda_i \otimes \lambda_i$ for some $s \leq 2r$ and $\lambda_i \in \mathbb{F}[2]^r$ by Lemma II.1. The same s and λ_i satisfy the conclusion of the lemma. ∎

The conclusion in the NO case (i.e. *soundness*) of Theorem III.5 can be boosted via a standard trick, so that a constant fraction of equations must fail instead of at least one equation failing. Suppose the instance $\mathcal{C}$ in Theorem III.5 has L equations written as $E_1 = 0, \ldots, E_L = 0$. One can take a $M \times L$ matrix Γ over $\mathbb{F}[2]$, $M = O(L)$, that is a generator matrix of a linear code of constant relative distance, say 0.10, and construct a new system $\mathcal{C}'$ of equations

$$\sum_{j=1}^{L} \Gamma_{ij} E_j = 0 \qquad i = 1, \ldots, M.$$

Clearly, a satisfying assignment to $\mathcal{C}$ is also a satisfying assignment to $\mathcal{C}'$. On other other hand, if no t assignments satisfy *all equations* in $\mathcal{C}$ in super-position, then no t assignments satisfy even 0.90 fraction of the equations in $\mathcal{C}'$ in super-position. With this observation, we re-state Theorem III.5 as:

Theorem III.7. *Let $\mathbb{F}[q]$ be an extension field of $\mathbb{F}[2]$ with $q = 2^r$. There is a reduction from* 3SAT *to an instance $\mathcal{C}$ of quadratic equations over $\mathbb{F}[q]$ such that,*

- *The equations have $\mathbb{F}[2]$ coefficients and no linear terms.*
- YES Case. *If the* 3SAT *instance is satisfiable then there is an assignment to $\mathcal{C}$ that satisfies all the equations. In fact there is such an assignment that is $\mathbb{F}[2]$ valued.*
- NO Case. *If the* 3SAT *instance is unsatisfiable then there are no t assignments to $\mathcal{C}$ that are $\mathbb{F}[q]$ valued and satisfy 0.90 fraction of the equations in super-position for any $1 \leq t \leq k$. Here k is a parameter and the reduction runs in time $N^{O(r \log k)}$ where N is the size of the* 3SAT *instance.*

Note however that the equations in the instance above have unbounded arity, i.e. a typical equation may depend on almost all the variables.

References

[1] V. Guruswami, J. Håstad, P. Harsha, S. Srinivasan, and G. Varma, "Super-polylogarithmic hypergraph coloring hardness via low-degree long codes," in *Proc. STOC*, 2014, pp. 614–623.

[2] A. Wigderson, "Improving the performance guarantee for approximate graph coloring," *Journal of the ACM*, vol. 30, no. 4, pp. 729–735, 1983.

[3] A. Blum, "New approximation algorithms for graph coloring," *Journal of the ACM*, vol. 41, no. 3, pp. 470–516, 1994.

[4] D. R. Karger, R. Motwani, and M. Sudan, "Approximate graph coloring by semidefinite programming," *Journal of the ACM*, vol. 45, no. 2, pp. 246–265, 1998.

[5] A. Blum and D. R. Karger, "An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs," *Information Processing Letters*, vol. 61, no. 1, pp. 49–53, 1997.

[6] S. Arora, E. Chlamtac, and M. Charikar, "New approximation guarantee for chromatic number," in *Proc. STOC*, 2006, pp. 215–224.

[7] K. Kawarabayashi and M. Thorup, "Combinatorial coloring of 3-colorable graphs," in *Proc. FOCS*, 2012, pp. 68–75.

[8] S. Khanna, N. Linial, and S. Safra, "On the hardness of approximating the chromatic number," *Combinatorica*, vol. 20, no. 3, pp. 393–415, 2000.

[9] V. Guruswami and S. Khanna, "On the hardness of 4-coloring a 3-colorable graph," *SIAM Journal of Discrete Mathematics*, vol. 18, no. 1, pp. 30–40, 2004.

[10] S. Huang, "Improved hardness of approximating chromatic number," in *APPROX-RANDOM*, 2013, pp. 233–243.

[11] S. Khot, "Improved inaproximability results for MaxClique, chromatic number and approximate graph coloring," in *Proc. FOCS*, 2001, pp. 600–609.

[12] I. Dinur, E. Mossel, and O. Regev, "Conditional hardness for approximate coloring," *SIAM Journal of Computing*, vol. 39, no. 3, pp. 843–873, 2009.

[13] M. Krivelevich, R. Nathaniel, and B. Sudakov, "Approximating coloring and maximum independent sets in 3-uniform hypergraphs," *Journal of Algorithms*, vol. 41, no. 1, pp. 99–113, 2001.

[14] H. Chen and A. M. Frieze, "Coloring bipartite hypergraphs," in *Proc. IPCO*, 1996, pp. 345–358.

[15] P. Kelsen, S. Mahajan, and R. Hariharan, "Approximate hypergraph coloring," in *Proc. SWAT*, 1996, pp. 41–52.

[16] K. Kawarabayashi and M. Thorup, "Coloring 3-colorable graphs with $o(n^{1/5})$ colors," in *Proc. STACS*, 2014, pp. 458–469.

[17] J. Holmerin, "Vertex cover on 4-regular hyper-graphs is hard to approximate within 2 - ε," in *Proc. CCC*, 2002.

[18] I. Dinur, O. Regev, and C. D. Smyth, "The hardness of 3-uniform hypergraph coloring," *Combinatorica*, vol. 25, no. 5, pp. 519–535, 2005.

[19] S. Khot, "Hardness results for coloring 3-colorable 3-uniform hypergraphs," in *Proc. FOCS*, 2002, pp. 23–32.

[20] S. Khot and R. Saket, "Hardness of finding independent sets in 2-colorable and almost 2-colorable hypergraphs," in *Proc. SODA*, 2014, pp. 1607–1625.

[21] R. Saket, "Hardness of finding indpendent sets in 2-colorable hypergraphs and of satisfiable CSPs," in *Proc. CCC*, 2014, pp. 78–89.

[22] V. Guruswami, J. Håstad, and M. Sudan, "Hardness of approximate hypergraph coloring," *SIAM Journal of Computing*, vol. 31, no. 6, pp. 1663–1686, 2002.

[23] S. Khot, "Hardness results for approximate hypergraph coloring," in *Proc. STOC*, 2002, pp. 351–359.

[24] I. Dinur, S. Khot, W. Perkins, and M. Safra, "Hardness of finding independent sets in almost 3-colorable graphs," in *Proc. FOCS*, 2010, pp. 212–221.

[25] S. Khot and R. Saket, "Hardness of finding independent sets in almost q-colorable graphs," in *Proc. FOCS*, 2012, pp. 380–389.

[26] S. O. Chan, "Approximation resistance from pairwise independent subgroups," in *Proc. STOC*, 2013, pp. 447–456.

[27] I. Dinur and G. Kol, "Covering CSPs," in *Proc. CCC*, 2013, pp. 207–218.

[28] S. Khot and R. Saket, "Hardness of coloring 2-colorable 12-uniform hypergraphs with $2^{(\log n)^{\Omega(1)}}$ colors," *ECCC*, vol. 21, p. 51, 2014. [Online]. Available: http://eccc.hpi-web.de/report/2014/051

[29] B. Barak, P. Gopalan, J. Håstad, R. Meka, P. Raghavendra, and D. Steurer, "Making the Long Code shorter," in *Proc. FOCS*, 2012, pp. 370–379.

[30] I. Dinur and V. Guruswami, "PCPs via low-degree long code and hardness for constrained hypergraph coloring," in *Proc. FOCS*, 2013.

[31] S. Arora, L. Babai, J. Stern, and Z. Sweedyk, "The hardness of approximate optima in lattices, codes, and systems of linear equations," *J. Comput. Sys. Sci*, vol. 54, no. 2, pp. 317–331, 1997.

[32] M. Bellare, O. Goldreich, and M. Sudan, "Free bits, PCPs, and nonapproximability-towards tight results," *SIAM Journal of Computing*, vol. 27, no. 3, pp. 804–915, 1998.

[33] J. Håstad, "Some optimal inapproximability results," *Journal of the ACM*, vol. 48, no. 4, pp. 798–859, 2001.

[34] A. Bhattacharyya, S. Kopparty, G. Schoenebeck, M. Sudan, and D. Zuckerman, "Optimal testing of Reed-Muller codes," in *Proc. FOCS*, 2010, pp. 488–497.

[35] G. Varma, "A note on reducing uniformity in Khot-Saket hypergraph coloring hardness reductions," *CoRR*, vol. abs/1408.0262, 2014. [Online]. Available: http://arxiv.org/abs/1408.0262

[36] S. Arora and M. Sudan, "Improved low-degree testing and its applications," *Combinatorica*, vol. 23, no. 3, pp. 365–426, 2003.