2013 IEEE 54th Annual Symposium on Foundations of Computer Science

PCPs via low-degree long code and hardness for constrained hypergraph coloring

Irit Dinur
Dept. of Applied Math and Computer Science
The Weizmann Institute of Science
Rehovot, Israel
Email: irit.dinur@weizmann.ac.il

Venkatesan Guruswami
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
Email: guruswami@cmu.edu

***Abstract*—We develop new techniques to incorporate the recently proposed "short code" (a low-degree version of the long code) into the construction and analysis of PCPs in the classical "Label Cover + Fourier Analysis" framework. As a result, we obtain more size-efficient PCPs that yield improved hardness results for approximating CSPs and certain coloring-type problems.**

In particular, we show a hardness for a variant of hypergraph coloring (with hyperedges of size 6), with a gap between 2 and $\exp(2^{\Omega(\sqrt{\log\log N})})$ number of colors where N is the number of vertices. This is the first hardness result to go beyond the $O(\log N)$ barrier for a coloring-type problem. Our hardness bound is a doubly exponential improvement over the previously known $O(\log\log N)$-coloring hardness for 2-colorable hypergraphs, and an exponential improvement over the $(\log N)^{\Omega(1)}$-coloring hardness for $O(1)$-colorable hypergraphs. Stated in terms of "covering complexity," we show that for 6-ary Boolean CSPs, it is hard to decide if a given instance is perfectly satisfiable or if it requires more than $2^{\Omega(\sqrt{\log\log N})}$ assignments for covering all of the constraints.

While our methods do not yield a result for conventional hypergraph coloring due to some technical reasons, we also prove hardness of $(\log N)^{\Omega(1)}$-coloring 2-colorable 6-uniform hypergraphs (this result relies just on the long code).

A key algebraic result driving our analysis concerns a very low-soundness error testing method for Reed-Muller codes. We prove that if a function $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$ is $2^{\Omega(d)}$ far in absolute distance from polynomials of degree $m-d$, then the probability that $\deg(\beta g) \leqslant m - 3d/4$ for a random degree $d/4$ polynomial g is *doubly exponentially* small in d.

***Keywords*-PCP, Hardness of approximation, hyper graph coloring, short code**

I. Introduction

Hardness of approximating constraint satisfaction problems is an area that has seen a great deal of progress in recent years. Following the pioneering works [3], [9], the standard framework for proving inapproximability has been via a combination of Label Cover (or special cases such as Unique Games [13]) and the long code. For proving constant gap inapproximability, the relative inefficiency of the long code is negligible. However, it becomes a serious bottleneck for non-constant parameter settings, most obviously, for proving hardness of approximate coloring. For this set of problems, there is an exponential or doubly exponential gap between the best known approximation algorithms (which require $n^{\Omega(1)}$ colors for n-vertex (hyper)graphs) and the best known hardness results (which at best only rule out efficient $o(\log n)$-coloring)

A very intriguing object called the "short code" was introduced and studied in [2]. This is a puncturing of the long code to locations indexed by low-degree polynomials, and to better reflect this, in this work we refer to the short code as the *low-degree long code*. This code was introduced in [2] as a "derandomization" of the long code, where it was used it to establish exponentially stronger integrality gaps for Unique Games, construct small set expanders whose Laplacians have many small eigenvalues, and obtain a more efficient version of the KKMO alphabet reduction [14] for Unique Games.

In this work we develop new techniques to use the low-degree long code in reductions from Label Cover and obtain the following (quasi-)NP-hardness results. Our main results are

- A hardness for a variant of approximate hypergraph coloring, with a gap between 2 and $\exp(2^{\Omega(\sqrt{\log\log N})})$ number of colors (where N is the number of vertices). This is the *first* inapproximability result to go beyond the logarithmic barrier for a coloring-type problem.
- A hardness for gap$(1, \frac{15}{16} + \varepsilon)$-4SAT for $\varepsilon = \exp(-2^{\Omega(\sqrt{\log\log N})})$. This improves upon Håstad's result [9] where $\varepsilon = 1/(\log N)^c$ for some constant $c > 0$.
- A hardness for approximate hypergraph coloring, with a gap between 2 and $(\log N)^{\Omega(1)}$ colors.

Adapting a long-code test into the low-degree long code setting turns out to be non-trivial, and there seems to be no general recipe (as of yet) for doing so. For instance, while it is straightforward to import Håstad's classic gap$(1-\varepsilon, 1/2+\delta)$-3LIN result to the low-degree long code setting, the above results require a more carefully tailor-made construction.

Both authors' research is supported in part by a BSF grant. Irit Dinur's research is also supported by an ERC grant numer 239985.

0272-5428/13 $26.00 © 2013 IEEE
DOI 10.1109/FOCS.2013.44

For certain PCPs in Håstad's work, such as 3SAT and 4-set splitting, we do not yet know how to adapt them to work with the low-degree long code. We comment that invariance-principle based analysis [15] is very powerful for analyzing dictatorship tests, and was used by [2] for analyzing their constructions. Nevertheless, for obtaining strong parameters we find that working directly with the Fourier expressions gives us a better handle on the kind of noise analysis that is needed.

For proving these results, we develop a "folding" mechanism for the low-degree long code that works with available label cover constraints. One of the important components of any long-code test is the noise, which becomes especially subtle when aiming for perfect completeness. The degree restriction in the low-degree long code makes it harder to control the correlations between various functions via appropriately chosen noise. Finally, to analyze some of the noise expressions in our tests, and especially to be able to get stronger parameters, we prove some new results on local testing Reed Muller codes, which we discuss next.

A. Local testing of Reed Muller codes over $\mathbb{F}_2$

One of the key insights in [2] was a connection between the analysis of the low-degree long code and Reed-Muller testing. Let us denote by $P(m,r)$ the functions $\mathbb{F}_2^m \to \mathbb{F}_2$ that have degree $\leqslant r$. For functions $\beta, g : \mathbb{F}_2^m \to \mathbb{F}_2$, denote $\chi_\beta(g) = (-1)^{\sum_{x\in\mathbb{F}_2^m} \beta(x)g(x)}$. Specifically, given a β that is far from $P(m, m-d-1)$ polynomials, they noted that one can bound the expectation $|\mathbb{E}_\mu[\chi_\beta(\mu)]|$ for a random *low-weight* μ by appealing to a powerful result of [4] about testing Reed-Muller codes. This is formally stated in Proposition 14. Using such a noise μ enables attenuating the contribution of large weight Fourier coefficients; however, it causes the test to have imperfect completeness. To obtain our low-degree long code based constructions with perfect completeness, we prove a new result concerning testing Reed-Muller codes, stated below.

Theorem 1. *Let d be a multiple of 4. Let $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$ be $2^{d/2}$-far from $P(m, m-d-1)$. Then for uniformly random polynomials $g \in P(m, d/4)$ and $h \in P(m, 3d/4)$, we have $\mathbb{E}_g\Big[\ \big|\mathbb{E}_h[\chi_\beta(gh)]\big|\Big] \leqslant 2^{-4\cdot 2^{d/4}}$.*

The key quantitative aspect of the above result is the *doubly exponential* decay in d. To obtain such a bound, we observe that the set of "bad" choices of g, for which βg has degree $m - 3d/4 - 1$ (i.e., one lower than what one expects), is a *subspace* of $P(m, d/4)$. We then lower bound the co-dimension of this subspace by $2^{\Omega(d)}$. We do this via a recursive approach to pass to *two* similar problems in dimension $(m-1)$, by making use of the main technical ingredient in [4] which argues the abundance of hyperplanes A such that $\beta_{|A}$ is $2^{d/2-2}$-far from polynomials of degree $m-d-1$ on one less variable.

We note that a "robust" version of the above theorem, which argues that βg will also likely be *far* from $P(m, m-3d/4-1)$, would be nice to have (as an interesting algebraic statement in itself). One can deduce such a claim from the above-mentioned result of [4] which proves such a robust version for $g \in P(m,1)$, but this will only give an upper bound of $2^{-O(d)}$.

B. Inapproximability Results

To describe our results let us first briefly recall the notion of covering CSPs from [6]. A q-ary φ-CSP is given by a q-uniform hypergraph where each hyperedge is associated with a constraint φ. The *covering number* of a CSP is the minimal number of assignments to the vertices so that each hyperedge is covered by at least one assignment, see also Definition 1. If one views a hypergraph coloring instance as a not-all-equal CSP, then the covering number is exactly log of the coloring number. This was the motivation of [8] and later [6] for studying the notion of covering.

In light of the lack of progress on hardness of approximate coloring for both graphs and hypergraphs, [6] suggested studying the hardness of gap covering problem, in the hope of approaching a potentially optimal gap-covering hardness result of 1 vs. $\Omega(\log N)$, which corresponds to a hardness gap of $O(1)$ vs. a polynomial number of colors. Given the current state of the art, they mentioned that even obtaining a gap of $O(1)$ vs. $\omega(\log\log N)$ would be interesting.

Theorem 2. *Assume that* NP *does not admit $n^{2^{O(\sqrt{\log\log n})}}$ time algorithms (note that this runtime is $n^{o(\log n)}$). Given a 6-ary CSP of size N, no polynomial-time algorithm can decide if it is perfectly satisfiable, or if its covering number is at least $2^{\Omega(\sqrt{\log\log N})}$.*

Prior to this work the best known gap-covering hardness was $O(1)$ vs. $O(\log\log N)$ (implicit in [11]) and 1 vs. $O(\log\log\log N)$ (implicit in [8]). Both these results in [8], [11] in fact applied to coloring (4-uniform) hypergraphs. It remains to be seen if a result similar to Theorem 2 can be obtained for hypergraph coloring. This would be a major quantitative jump, breaking the barrier of $O(\log N)$ colors.

We remark that the result above is obtained for a 6-ary constraint that is the disjunction of three inequality constraints. Since inequality makes sense over any alphabet size, one can think of this problem directly as a coloring-type problem, instead of a covering problem. This is always possible when the constraints are so-called "equality constrained languages" [5], and we give an alternative formulation of this theorem as a coloring problem in Theorem 22.

Along the way to proving Theorem 2, we establish the following inapproximability result for 4SAT with perfect completeness. We present this result first to illustrate our techniques in the basic setting of 4SAT, before applying them to a covering 6-CSP to deduce Theorem 2.

Theorem 3. *Assume that* NP *does not admit $n^{O(\log n)}$ time algorithms. Given an instance of 4SAT of size N, there is no polynomial time algorithm to distinguish between the following two cases:*

- *The instance is satisfiable.*
- *Every assignment satisfies at most a fraction $\frac{15}{16} + 2^{-2^{\Omega(\sqrt{\log\log N})}}$ of the clauses.*

We remark a similar result but without the perfect completeness would have been significantly easier to prove. A direct adaptation of the perfect completeness tests seems less forthcoming due to the limitation on the noise imposed by working with the short code. It is worth mentioning that even for long code based constructions, perfect completeness tends to be significantly more difficult to ensure, often requiring additional technical elements, such as smoothness of Label Cover projections [12], and/or picking functions whose bias itself is sampled from carefully chosen distributions as in [9, Sections 6,7], [10].

Fortunately, for 4SAT one can establish hardness avoiding the more complicated technical elements [9, Thm. 6.2] (this would yield an inapproximability factor $\frac{15}{16} + \frac{1}{(\log N)^c}$ for some small absolute constant $c > 0$). Even so, adapting this to the low-degree long code setting involves some careful design choices, as multiplying two functions, which seems like an essential component when perfect completeness is desired, increases the degree. This necessitates restricting certain functions in the test to be of smaller degree. In order to ensure that this doesn't bias the query pattern to a small portion of the low-degree long code, we query the smaller degree functions in a *separate* low-degree long code of smaller degree. This "multipartite" structural restriction is what precludes us from extending our result for covering 6-CSP (Theorem 2 to a result about hypergraph coloring. (Clearly, if the variables of every constraint straddles two or more parts, then the associated hypergraph is trivially 2-colorable.)

Finally, we also include a result on the hardness of hypergraph coloring. This result does not rely on the low-degree long code and is just based on techniques in Håstad's 1997 paper [9]. However, as the result statement is not explicit in the literature, we include it here and defer the proof to the full version. (Also, this test paved the way for the version with the low-degree long code stated in Theorem 2.)

Theorem 4. *Assume that* NP *does not admit $n^{O(\log\log n)}$ time algorithms. There is an absolute constant $c > 0$ such that the following holds. Given a 6-uniform hypergraph on N vertices, there is no polynomial time algorithm to distinguish between the following two cases:*

- *The hypergraph can be colored with 2 colors so that every hyperedge is bichromatic.*
- *The hypergraph does not have an independent set with $N/(\log N)^c$ vertices, and in particular any coloring of the vertices with $(\log N)^c$ colors will have a monochromatic hyperedge.*

We note that $(\log N)^{\Omega(1)}$ colors is currently the strongest quantitative bound on hardness for hypergraph coloring. Khot obtained a similar result using the "split code" for coloring 7-colorable 4-uniform hypergraphs [11]. The above statement is incomparable as it applies to 2-colorable hypergraphs, albeit of larger uniformity. For 3-uniform hypergraphs, hardness of $O(\sqrt[3]{\log\log N})$-coloring 2-colorable hypergraphs is shown in [7], and a super-constant hardness for 3-colorable case is shown in [12].

C. Organization

We begin in Section II with background information on label cover and CSPs, the low-degree long code and its connection to Reed-Muller testing, and describe our folding mechanism for the low-degree long code. Our new algebraic result on testing Reed-Muller codes (Theorem 1) is proved in Section III. In Section IV, we prove Theorem 3 on the hardness of approximating satisfiable instances of 4SAT. We prove the result for covering 6-CSP (Theorem 2) in Section V.

II. Preliminaries

A. Label Cover and its hardness

A label cover instance is given by a bipartite graph $G = (U, V, E)$, two alphabets Σ_U and Σ_V and a projection constraint $\pi_{uv} : \Sigma_U \to \Sigma_V$ per edge $uv \in E$. The goal is to assign labels to the vertices in a way that maximizes the number of satisfied constraints.

We next state a theorem about the NP-hardness of label cover, where the label cover has a concrete structure that is convenient for use with the low-degree long code.

Theorem 5 (Hardness of Label Cover)**.** *Let $\ell \in \mathbb{N}$ be a parameter. There is a polynomial-time reduction from a 3SAT instance of size n to a label cover instance of size $n^{O(\ell)}$ that is specified by*

- *A constraint graph $G = (U, V, E)$, $\Sigma_U = \mathbb{F}_2^{3\ell}$ and $\Sigma_V = \mathbb{F}_2^{\ell}$.*
- *Every $u \in U$ carries ℓ functions $f_1^{(u)}, \ldots, f_\ell^{(u)} : \mathbb{F}_2^3 \to \mathbb{F}_2$.*
- *Every edge $uv \in E$ carries a projection mapping defined by a subset $\pi_{uv} \subset [3\ell]$, $|\pi_{uv}| = \ell$, that contains exactly one element in each triple of indices $(3i{+}1, 3i{+}2, 3i{+}3)$, for $i = 0, \ldots, \ell{-}1$. The constraint on an edge is said to be satisfied by $a \in (\mathbb{F}_2^3)^\ell$ and $b \in \mathbb{F}_2^\ell$ if*

$$f_1^{(u)}(a_1) = \ldots = f_\ell^{(u)}(a_\ell) = 0 \qquad \text{and} \qquad \pi_{uv}(a) = b\,.$$

The label cover instance has the following completeness and soundness conditions:

- *If the 3SAT instance is satisfiable, then there is an assignment for the label cover instance satisfying every constraint.*
- *If the 3SAT instance is unsatisfiable, then every assignment for the label cover instance satisfies at most $2^{-\Omega(\ell)}$ fraction of the constraints.*

This theorem is obtained from standard techniques: start with an NP-hard instance of gap-3SAT, and then perform ℓ-parallel repetition [1]. The functions $f_1^{(u)}, \ldots, f_\ell^{(u)}$ associated with an ℓ-tuple u of clauses check that the clauses are satisfied.

B. CSPs, Covering CSPs, and coloring problems

Let $X = \{x_1, ..., x_n\}$ be a set of n boolean variables and $\varphi : \{0,1\}^q \to \{0,1\}$ be a predicate. A *φ-constraint* over X is an equation of the form $\varphi(x_{i_1}, \ldots, x_{i_t}) = 1$, where $i_1, \ldots, i_t \in [n]$. A *φ-CSP instance* C is a set of φ-constraints over X.

It is standard to denote by 4SAT the CSP where each constraint is defined by a disjunction of four variables or their negations, and by 3LIN the CSP where each constraint is defined by a linear equation over three variables modulo 2.

Let $A_1, \ldots, A_k \in \{0,1\}^n$ be a set of assignments for X. We say that $A_1, \ldots, A_k$ *cover* the instance C if for every constraint in C, there exists $i \in [k]$ such that A_i satisfies the constraint. The *covering number* of C, denoted $\nu(C)$, is smallest number k of assignments for X such that each constraint is satisfied by at least one of the assignments. We denote by cover-φ the problem of finding the covering number of a given CSP. The gap problem is defined as follows

Definition 1 (gap-cover-φ)**.** *Let $c < s \in \mathbb{N}$, and let φ be a predicate. Given a φ-CSP instance C, decide between*

- ***Yes case:** $\nu(C) \leq c$. I.e., there exists a set of at most c assignments that covers C.*
- ***No case:** $\nu(C) \geq s$. I.e., no set of at most s assignments covers C.*

C. The low-degree long code

Notation. We denote the field with two elements by $\mathbb{F}_2$. For a positive integer m, we denote by $\mathcal{F}_m$ the $\mathbb{F}_2$-vector space of functions $\mathbb{F}_2^m \to \mathbb{F}_2$. We can equip $\mathcal{F}_m$ with the Hamming metric by defining for $g, h \in \mathcal{F}_m$, their distance $\Delta(g,h)$ to be the number of $x \in \mathbb{F}_2^m$ such that $g(x) \neq h(x)$. For a subset $A \subseteq \mathbb{F}_2^m$, we denote by $g_{|A}$ be the function g restricted to A. The distance between $g_{|A}$ and $h_{|A}$, $\Delta(g_{|A}, h_{|A})$, is the number of $\mathbf{x} \in A$ such that $g(\mathbf{x}) \neq h(\mathbf{x})$.

For $g \in \mathcal{F}_m$ and $\mathcal{H} \subseteq \mathcal{F}_m$, we define $\Delta(g, \mathcal{H}) = \min_{h \in \mathcal{H}} \Delta(g,h)$. We say g is Δ-far from a subset $\mathcal{H} \subseteq \mathcal{F}_m$ is $\Delta(g, \mathcal{H}) > \Delta$; otherwise we say g is Δ-close to $\mathcal{H}$.

Every function $f \in \mathcal{F}_m$ can be uniquely expressed as a multilinear polynomial over $\mathbb{F}_2$ of degree at most m. We will be interested in those functions which have much lower degree.

Definition 2 (Reed-Muller code)**.** *We denote by $P(m,d)$ the space of all functions $f : \mathbb{F}_2^m \to \mathbb{F}_2$ that have degree at most d. The evaluations of the polynomials in $P(m,d)$ at all points in $\mathbb{F}_2^m$ gives the binary m-variate Reed-Muller code of degree d, usually denoted as $\mathsf{RM}(m,d)$.*

Note that $P(m,d)$ is a subspace of $\mathcal{F}_m$. It is well-known and easy to see that the dual subspace of $P(m,d)$, denoted $P(m,d)^\perp$, is the subspace $P(m, m-d-1)$ of $\mathcal{F}_m$ consisting of polynomials of degree less than $m-d$.

We will now define the low-degree long code first introduced in [2], where it is called the "short code."

Definition 3. *Let $m \geqslant d$ be positive integers, and let $\mathbf{a} \in \mathbb{F}_2^m$. For integers m, d, the (m-variate degree-d) low-degree long code of $\mathbf{a}$, denoted $\mathsf{SC}_{m,d}(\mathbf{a})$, is a function from $P(m,d)$ to $\mathbb{F}_2$ defined by*

$$\mathsf{SC}_{m,d}(a)(g) = (-1)^{g(a)} \qquad \text{for } g \in P(m,d)\ .$$

When m, d are clear from context, we will refer to the low-degree long code as $\mathsf{SC}(a)$.

For $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$, the *weight* of β, denoted $\mathsf{wt}(\beta)$, is the number of $\mathbf{x} \in \mathbb{F}_2^m$ such that $\beta(\mathbf{x}) = 1$. In other words, $\mathsf{wt}(\beta) = \Delta(\beta, \mathbf{0})$ is the distance of β from the zero polynomial.

Definition 4 (Character set)**.** *For positive integers $m \geqslant d$, we define by $\Lambda(m,d)$ the set of functions $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$ which are the minimum weight functions (ties broken arbitrarily) in the cosets of $P(m, m-d-1)$ in $\mathcal{F}_m$.*[1]

By definition, for each $\beta \in \Lambda(m,d)$, the closest polynomial (in Hamming distance) of degree at most $m-d-1$ to β is the zero polynomial. The functions in $\Lambda(m,d)$ correspond to the "Voronoi cell" of the zero polynomial for the set of points $P(m, m-d-1)$, under the metric $\Delta(\cdot,\cdot)$.

For functions $\beta, g : \mathbb{F}_2^m \to \mathbb{F}_2$, we define the "character mapping" $\chi_\beta(g)$ by$\chi_\beta(g) = (-1)^{\sum_{\mathbf{x} \in \mathbb{F}_2^m} \beta(\mathbf{x}) g(\mathbf{x})}$.

The following are easy consequences of $P(m,d)^\perp$ being equal to $P(m, m-d-1)$.

Fact 6. *For $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$, we have* $\mathbb{E}_g[\chi_\beta(g)] = \begin{cases} 1 & \text{if } \beta \in P(m, m-d-1) \\ 0 & \text{otherwise} \end{cases}$

[1] Since $P(m,d)^\perp = P(m, m-d-1)$, one has $|\Lambda(m,d)| = |\mathcal{F}_m| / |P(m, m-d-1)| = |P(m,d)| = 2^{\sum_{j=0}^{d} \binom{m}{j}}$.

where the expectation is taken over a random $g \in P(m,d)$.

Fact 7. *For $\beta_1, \beta_2 \in \Lambda(m,d)$, we have $\mathbb{E}_g[\chi_{\beta_1}(g)\chi_{\beta_2}(g)] = \begin{cases} 1 & \text{if } \beta_1 = \beta_2 \\ 0 & \text{otherwise} \end{cases}$ where the expectation is taken over a random $g \in P(m,d)$.*

By well-known facts from the character theory of finite abelian groups, we have:

Fact 8. *Every function $A : P(m,d) \to \mathbb{R}$ admits the "Fourier" expansion*

$$A(g) = \textstyle\sum_{\beta \in \Lambda(m,d)} \widehat{A}(\beta)\chi_\beta(g),$$

where the Fourier coefficients are given by the inversion formula $\widehat{A}(\beta)) = \mathbb{E}_g\big[A(g)\chi_\beta(g)\big]$, with the expectation taken over a uniformly random $g \in P(m,d)$.

Finally, we consider two functions over different-dimension domains, $A : P(m,d) \to \{-1,1\}$ and $B : P(\ell,d) \to \{-1,1\}$ where $m > \ell$. Suppose we have a projection $\pi : \mathbb{F}_2^m \to \mathbb{F}_2^\ell$ defined by $\pi(x_1,\ldots,x_m) = (x_{i_1},\ldots,x_{i_\ell})$ for some indices $1 \leqslant i_1 < \cdots < i_\ell \leqslant m$. The projection π allows us to lift a polynomial $f \in P(\ell,d)$ to the larger domain without changing its degree, defining $f \circ \pi \in P(m,d)$ by $f \circ \pi(x) = f(\pi(x))$. Now, for $\beta : \mathbb{F}_2^\ell \to \mathbb{F}_2$, $\chi_\beta(f \circ \pi) = (-1)^{\sum_{x \in \mathbb{F}_2^m} (f\circ\pi)(x)\cdot\beta(x)} = (-1)^{\sum_{y \in \mathbb{F}_2^\ell} f(y)\cdot\sum_{x\in\pi^{-1}(y)}\beta(x)} = (-1)^{\sum_{y\in\mathbb{F}_2^\ell} f(y)\cdot\pi_2(\beta)(y)} = \chi_{\pi_2(\beta)}(f)$, where we define $\pi_2(\beta) : \mathbb{F}_2^\ell \to \mathbb{F}_2$ by $\pi_2(\beta)(y) = \sum_{x\in\pi^{-1}(y)}\beta(x) \mod 2$.

Fact 9. *Let $\beta \in \Lambda(m,d)$ and let $\alpha \in \Lambda(\ell,d)$. Then $\mathbb{E}_{f\in P(\ell,d)}[\chi_\beta(f\circ\pi)\chi_\alpha(f)]$ equals 1 if $\alpha = \pi_2(\beta)$ and zero otherwise.*

D. Folding properties of low-degree long code

Folding over constraints.: Let $p_1,\ldots,p_k \in P(m,3)$ be given. Let

$$I = \langle p_1,\ldots,p_k\rangle = \left\{ \sum_{i=1}^k p_i q_i \;\middle|\; q_i \in P(m,d-3) \right\},$$

clearly a linear space. We define $P(m,d)/I$ to be the collection of cosets of I in $P(m,d)$, and we denote by $p+I$ the coset of $p \in P(m,d)$.

Definition 5 (Folding)**.** *A function $A : P(m,d) \to \mathbb{R}$ is folded over $I = \langle p_1,\ldots,p_k\rangle$ if*

$$\forall p,p' \in P(m,d), \qquad p - p' \in I \quad \Rightarrow \quad A(p) = A(p').$$

A is folded over $\{-1,1\}$ if $A(g) = -A(1+g)$ for all $g \in P(m,d)$.

Fact 10. *Let $\mathbf{a} \in \mathbb{F}_2^m$. If $A = \mathsf{SC}(\mathbf{a})$ and $p_i(\mathbf{a}) = 0$ for all $i \in [k]$, then A is folded over $\langle p_1,\ldots,p_k\rangle$ and over $\{-1,1\}$.*

We next show that a function folded over I cannot have weight on small Fourier coefficients that are non-zero on I.

Claim 11. *Let $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$ have $\mathsf{wt}(\beta) < 2^{d-3}$, and suppose there is an element $x \in \mathbb{F}_2^m$ with $\beta(x) = 1$ for which there is some p_i such that $p_i(x) \neq 0$. Then if A is folded over I then $\widehat{A}(\beta) = E_g[\chi_\beta(g)A(g)] = 0$*

Proof: Let $X = \{x \in \mathbb{F}_2^m \mid \beta(x) = 1 \text{ and } \exists i, p_i(x) \neq 0\}$. Choose some $a \in X$ and let i be such that $p_i(a) = 1$. Let $p = qp_i \in I$ where q is a polynomial that vanishes on all points of X except a. q has degree at most $d-3$ as long as $|X| \leqslant \mathsf{wt}(\beta) < 2^{d-3}$. Pair each function $g \in P(m,d)$ with $g+p$. By folding, $A(g) = A(g+p)$, but $\chi_\beta(g+p) = \chi_\beta(g)\chi_\beta(p) = -\chi_\beta(g)$, so $\widehat{A}(\beta) = 0$. ∎

Folding over "true".: Let us denote by $P'(m,d)$ the set obtained by choosing exactly one function out of each pair $g, 1+g \in P(m,d)$. Similarly, denote by $P'(m,d)/I$ the set obtained by choosing exactly one coset out of each pair $g+I, 1+g+I \in P(m,d)/I$.

Given a function $A' : P'(m,d) \to \{-1,1\}$ it can be naturally extended to $A : P(m,d) \to \{-1,1\}$ by setting $A(1+g) = -A'(g)$. A function $A : P(m,d) \to \{-1,1\}$ is said to be folded over $\{-1,1\}$ if $A(g) = -A(1+g)$ for all g. If A is folded over $\{-1,1\}$ then for any β with even $\mathsf{wt}(\beta)$, $\widehat{A}(\beta) = 0$. In particular, $\widehat{A}(0) = 0$.

Fact 12. *Given a function $\tilde{A} : P'(m,d)/I \to \mathbb{R}$, there is a unique function $A : P(m,d) \to \mathbb{R}$ that is folded over $\{-1,1\}$ and folded over I and for all $g \in P(m,d)$, $\tilde{A}(g+I) = A(g)$.*

E. Reduction from Label Cover using the low-degree long code

All of our inapproximability results will follow the same general framework [3], [9] combining label cover with the long code adapted to the low-degree variant in the following way. Start from a label cover instance G as in Theorem 5. For each $v \in V$ place a block of variables corresponding to $P(\ell,d)$. For each $u \in U$, let $I^{(u)} = \langle f_1^{(u)},\ldots,f_\ell^{(u)}\rangle$ where $f_1^{(u)},\ldots,f_\ell^{(u)}$ are the degree-3 functions that are associated with u. For each u place a block of variables corresponding to $P(3\ell,d)/I^{(u)}$.

Note that an assignment to these variables is equivalent to a collection of functions for all u and v,

$$A^{(v)} : P(\ell,d) \to \{-1,1\} \text{ and } B^{(u)} : P(3\ell,d) \to \{-1,1\}$$

such that for each $u \in U$, $B^{(u)}$ is folded over $I^{(u)}$. Sometimes we will also need the tables to be folded over $\{-1,1\}$, in which case the block of variables (from which we extend $A^{(v)}$ to all of $P(\ell,d)$) will be restricted to $P'(\ell,d)$, and similarly $B^{(u)}$ will be extended from $P'(3\ell,d)/I^{(u)}$.

Our reductions, as usual, are described by a PCP verifier that randomly queries the functions $A^{(v)}$ and $B^{(u)}$. If φ is the acceptance predicate of the PCP verifier, then together with the query pattern this describes a φ-CSP. To analyze

the reduction, one writes Fourier expressions that describe the probability of acceptance. The following lemma is an adaptation to the low-degree long code of Håstad's technique for converting certain Fourier expressions into a label cover strategy. One subtle point below is that we need $\mathsf{wt}(\beta)$ to be bounded to ensure that every element in the support of β is a valid assignment to u, i.e., one that satisfies $f_1^{(u)}, \ldots, f_\ell^{(u)}$.

Lemma 13. *If $K \leqslant 2^{d-3}$ and*

$$\mathop{\mathbb{E}}_{uv}\Big[\sum_{\substack{\beta:\mathsf{wt}(\beta)<K\\ \pi_2(\beta)\neq 0}} \widehat{A^{(v)}}(\pi_2(\beta))^2 \widehat{B^{(u)}}(\beta)^2\Big] \geqslant \delta, \qquad (1)$$

then there is an assignment for the label cover satisfying at least δ/K of the constraints.

Proof: Define a randomized assignment as follows. For each $u \in U$ choose a random $\beta \in \Lambda(3\ell, d)$ with probability proportional to $\widehat{B^{(u)}}(\beta)^2$ and then assign u with a random element $b \in \beta^{-1}(1)$. Similarly, for each $v \in V$, choose a random $\alpha \in \Lambda(\ell, d)$ with probability proportional to $\widehat{A}(\alpha)^2$ and then assign u with a random element $a \in \alpha^{-1}(1)$. Since $\sum_\beta \widehat{B}(\beta)^2 \leqslant 1$, the probability of picking a certain β is at least $\widehat{B}(\beta)^2$, and similarly for α.

The left hand side of (1) lower bounds the probability that u was assigned through β, and v was assigned through $\alpha = \pi_2(\beta)$. If that happened, then for each choice of $a \in \alpha^{-1}(1)$ there is at least one matching $b \in \beta^{-1}(1)$, which is chosen with probability at least $1/K$. It remains to observe the key fact that b is a valid assignment for u because of Claim 11 and the fact that $\mathsf{wt}(\beta) < K \leqslant 2^{d-3}$. ∎

F. Local testing of Reed-Muller codes

From Fact 7 we have, for $\beta \in \Lambda(m,d)$, $\mathbb{E}_g[\chi_\beta(g)] = \begin{cases} 1 & \text{if } \beta = 0 \\ 0 & \text{if } \beta \in \Lambda(m,d)\setminus\{0\} \end{cases}$ when the expectation is taken over a random $g \in P(m,d)$. Thus, orthogonality (over $\mathbb{F}_2$) with a random degree d polynomial $g \in P(m,d)$ serves as a perfect test for whether $\beta \in \Lambda(m,d)$ is the zero polynomial or not (or equivalently, if $\beta \in \mathcal{F}_m$ belongs to $P(m, m-d-1)$ or not). The next result, which follows from [4], shows that when $\beta \in \Lambda(m,d)$ has large weight (or equivalently, if $\beta \in \mathcal{F}_m$ is far from $P(m, m-d-1)$), the above expectation is bounded away from 1 even when g is chosen *pseudorandomly*, corresponding to the minimum weight codewords of $\mathsf{RM}(m,d)$ (i.e., products of d linearly independent affine forms). Specifically, let $L(m,d) \subseteq P(m,d)$ be the subset of degree d polynomials which are the product of exactly d linearly independent affine forms. Then we have the following claim which we will use in our warm-up 3LIN PCP (but not for any other PCP construction).

Proposition 14. *There exists an absolute constant $\rho_0 < 1$ such that for all $\beta \in \Lambda(m,d)$,*

$$\mathop{\mathbb{E}}_{\mu\in L(m,d)}\Big[\chi_\beta(\mu)\Big] \leqslant \rho = \max\Big\{1 - \frac{\mathsf{wt}(\beta)}{2^d}, \rho_0\Big\}\,. \qquad (2)$$

Moreover, if we choose $\mu_1, \ldots, \mu_t$ independently at random from $L(m,d)$ then

$$\mathop{\mathbb{E}}_{\mu_1,\ldots,\mu_t\in L(m,d)}\Big[\chi_\beta(\mu_1+\cdots+\mu_t)\Big] \leqslant \rho^t\,, \qquad (3)$$

Proof: Consider the test for membership of β in $P(m, m-d-1)$ that proceeds by picking a random $\mu \in L(m,d)$ and checking that $\sum_{\mathbf{x}\in\mathbb{F}_2^m} \beta(\mathbf{x})\mu(\mathbf{x}) = 0$. Then $\mathbb{E}_{\mu\in L(m,d)}[\chi_\beta(\mu)] = 1 - 2\mathsf{Rej}(\beta)$ where $\mathsf{Rej}(\beta)$ is probability that the test rejects β. Theorem 1 in [4], applied for m variables and degree $m-d-1$, implies that $\mathsf{Rej}(\beta) \geqslant \min\{\frac{\mathsf{wt}(\beta)}{2^{d+1}}, \epsilon_1\}$ for some absolute constant $\epsilon_1 > 0$. The bound (2) follows by setting $\rho_0 = 1 - 2\epsilon_1$. The bound (3) follows by noting that $\mathbb{E}[\chi_\beta(\mu_1+\cdots+\mu_t)] = \mathbb{E}[\chi_\beta(\mu_1)]\cdots\mathbb{E}[\chi_\beta(\mu_t)]$. ∎

III. A new low-error tester for Reed-Muller codes

In this section, our goal is to prove the following result, which will be used in the analysis of our low-degree long code based PCPs to show that the "high frequency" terms in the Fourier expansion make a negligible contribution.

Theorem 15. *Let d be a multiple of 4. Let $\beta \in \mathcal{F}_m$ be $2^{d/2}$-far from $P(m, m-d-1)$, and let $g \in P(m, d/4)$ and $h \in P(m, 3d/4)$ be uniformly random polynomials from their respective domains. Then*

$$\mathop{\mathbb{E}}_{g}\Big[\ \Big|\mathop{\mathbb{E}}_{h}[\chi_\beta(gh)]\Big|\Big] \leqslant 2^{-4\cdot 2^{d/4}}\,. \qquad (4)$$

Fix a $\beta \in \mathcal{F}_m$. Appealing to Fact 6 we know $\mathbb{E}_{h\in P(m,3d/4)}[\chi_\beta(gh)] = \mathbb{E}_{h\in P(m,3d/4)}[\chi_{\beta\cdot g}(h)] = \begin{cases} 1 & \text{if } \beta g \in P(m, m-3d/4-1) \\ 0 & \text{otherwise} \end{cases}$. Therefore, the expectation in (4) equals

$$\mathop{\mathbb{E}}_{g}\Big[\ \Big|\mathop{\mathbb{E}}_{h}[\chi_\beta(gh)]\Big|\Big] = \Pr_{g\in P(m,d/4)}[\beta g \in P(m, m-3d/4-1)]\,. \qquad (5)$$

The following simple observation shows that estimating the above probability is really a linear-algebraic problem of bounding the dimension of a certain subspace. This is the subspace of polynomials g for which the degree of βg is strictly smaller than the product of the degrees.

Observation 16. *Fix any $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$. For an integer $k \leqslant d$, the set*

$$B_{d,k}^{(m)}(\beta) \stackrel{\text{def}}{=} \{g \in P(m,k) \mid \beta g \in P(m, m-d-1+k)\} \qquad (6)$$

is a subspace of $P(m,k)$.

Combining the above with Equation (5), we see that the expectation in (4) is given by

$$\mathop{\mathbb{E}}_{g}\Big[\ \Big|\mathop{\mathbb{E}}_{h}[\chi_\beta(gh)]\Big|\Big] = 2^{\dim(B^{(m)}_{d,d/4}(\beta))-\dim(P(m,d/4))}\ .$$

Theorem 15 now follows from the following result.

Theorem 17. *For all positive integers m, d, k satisfying $m > d$ and $4|d$, the following holds. If $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$ has distance more than $2^{d/2}$ from $P(m, m-d-1)$, then the subspace $B^{(m)}_{d,d/4}(\beta)$ defined in* (6) *has co-dimension (as a subspace of $P(m, d/4)$) at least $2^{d/4-2}$.*

The rest of the section will be devoted to proving Theorem 17. For positive integers d, k, let us define the function $\Phi_{d,k} : \mathbb{N} \to \mathbb{N}$ as follows. If $d < k$, then $\Phi_{d,k}$ is identically 0. Otherwise, for $d \geqslant k$,

$$\Phi_{d,k}(D) = \min\Big\{\ \dim(P(m,k)) - \dim(B^{(m)}_{d,k}(\beta))\ \Big\}\ , \quad (7)$$

where the minimum is taken over all $m > d$ and $\beta \in \mathcal{F}_m$ such that $\Delta(\beta, P(m, m-d-1)) \geqslant D$ and where $B^{(m)}_{d,k}(\beta)$ is as defined in (6).

We note that Theorem 17 will follow if we prove that

$$\Phi_{d,d/4}(2^{d/2}) \geqslant 2^{d/4-2}\ . \quad (8)$$

We begin with the following claim which gives us the base case showing a lower bound when the distance $D = 1$.

Claim 18. *For $d \geqslant k$, $\Phi_{d,k}(1) \geqslant 1$.*

Proof: The claim can be restated as follows: If $\beta \notin P(m, m-d-1)$, then $B^{(m)}_{d,k}(\beta)$ is a proper subspace of $P(m,k)$, or in other words there exists $\nu \in P(m,k)$ such that $\beta\nu \notin P(m, m-d-1+k)$. We now prove this fact. As the dual space of $P(m, m-d-1)$ in $\mathcal{F}_m$ is $P(m,d)$, when $\beta \notin P(m, m-d-1)$, there must exist $\xi \in P(m,d)$ such that $\sum_{x\in\mathbb{F}_2^m} \beta(x)\xi(x) = 1$, or equivalently $\beta\xi \notin P(m, m-1)$. We may assume that ξ is a monomial $\xi = x_{i_1}x_{i_2}\cdots x_{i_l}$ with $l \leqslant d$ as such monomials form a basis of $P(m,d)$. If $l \leqslant k$, then ξ itself serves as the witness ν such that $\beta\nu \notin P(m, m-d-1+k)$. Otherwise, we can take $\nu = x_{i_1}x_{i_2}\cdots x_{i_k}$ and $\beta\nu$ can't have degree at most $m-d-1+k$ as that would imply $\beta\xi$ has degree at most $m-d-1+l \leqslant m-1$, a contradiction. ∎

The following lemma will be used in the recursive step when proving Theorem 17. It is based on a similar statement proved in [4].

Lemma 19. *Let $m > d$ be integers, and let $40 < D < 2^d$. If $\beta : \mathbb{F}_2^m \to \mathbb{F}_2$, which we think of as a polynomial in variables $x_1, x_2, \ldots, x_m$, is D-far from $P(m, m-d-1)$, then there exists a nonzero linear form $L = L(x_1, \ldots, x_m) \in P(m,1)$ such that $\beta_{|L=0}$ and $\beta_{|L=1}$ are* both *$D/4$-far from polynomials of degree $m-d-1$.*

Proof: (of Theorem 17) Our goal is to establish the lower bound (8) on $\Phi_{d,d/4}(D)$ for $D = 2^{d/2}$. By Claim 18, we may assume $d \geqslant 12$. Let $\beta \in \mathcal{F}_m$ be a polynomial in $x_1, x_2, \ldots, x_m$ that is D-far from $P(m, m-d-1)$. We need to prove $\dim(B^{(m)}_{d,d/4}(\beta)) \leqslant P(m, d/4) - 2^{d/4-2}$.

By Lemma 19, we may assume, after applying a linear transformation on the coordinates, that $\beta_{x_m=0}$ and $\beta_{x_m=1}$ are both $D/4$-far from $P(m-1, m-d-1)$. Let us write the polynomial β in the form $\beta = x_m a(x_1, \ldots, x_{m-1}) + b(x_1, \ldots, x_{m-1})$. In other words, $\beta_{x_m=0} = b$ and $\beta_{x_m=1} = a+b$ where a, b are polynomials in $x_1, \ldots, x_{m-1}$. We know $\Delta(b, P(m-1, m-d-1)) \geqslant D/4$ and $\Delta(a+b, P(m-1, m-d-1)) \geqslant D/4$.

Define $r = m-d-1$. We need to understand when $\nu \in P(m,k)$ is such that $\beta\nu \in P(m, r+k)$. Let us write the polynomial $\nu \in P(m,k)$ as $\nu = x_m p + q$ where $p \in P(m-1, k-1)$ and $q \in P(m-1, k)$ are polynomials in $x_1, \ldots, x_{m-1}$ of degree at most $k-1$ and k respectively. We have the following claim.

Claim 20. *If $\nu \in B^{(m)}_{d,k}(\beta)$, then $q \in B^{(m-1)}_{d-1,k}(b)$, i.e., $qb \in P(m-1, r+k)$, and $p(a+b) \in qa + P(m-1, r+k-1)$.*

Proof: (of Claim) Indeed, $\beta\nu = qb + x_m((a+b)p + qa)$. The terms in qb, which is a polynomial in $x_1, \ldots, x_{m-1}$, cannot be canceled by any terms in $x_m((a+b)p+qb)$. So if $\beta\nu$ has degree at most $r+k$, qb must also have degree at most $r+k$. Also, if $\beta\nu$ has degree at most $r+k$, the polynomial $p(a+b)+qa$ must have degree at most $r+k-1$, which is the same thing as $p(a+b) \in qa + P(m-1, r+k-1)$. ∎

By the above claim, the choice of ν in the subspace $B^{(m)}_{d,k}(\beta)$ amounts to picking an arbitrary q in the subspace $B^{(m-1)}_{d-1,k}(b)$ of $P(m-1,k)$, and then p from a coset of the subspace $B^{(m-1)}_{d-1,k-1}(a+b) = \{\tilde{\nu} \in P(m-1, k-1) \mid (a+b)\tilde{\nu} \in P(m-1, r+k-1)\}$ of $P(m-1, k-1)$. Therefore,

$$\dim(B^{(m)}_{d,k}(\beta)) \leqslant \dim(B^{(m-1)}_{d-1,k}(b)) + \dim(B^{(m-1)}_{d-1,k-1}(a+b))\ . \quad (9)$$

Combining (7), (9), and the equality $\dim(P(m,k)) = \dim(P(m-1,k)) + \dim(P(m-1,k-1))$, we can conclude the following for all $d \geqslant k$ and $D < 2^d$:

$$\Phi_{d,k}(D) \geqslant \Phi_{d-1,k}(D/4) + \Phi_{d-1,k-1}(D/4)\ . \quad (10)$$

When $D = 2^{d/2} = 4^{d/4}$ and $k = d/4$, recursively applying the above for a depth of $d/4-2$ (to reduce D geometrically from $4^{d/4}$ to 16), and using Claim 18, we can lower bound $\Phi_{d,d/4}(2^{d/2}) \geqslant 2^{d/4-2}$, giving us (8). ∎

IV. PCP CHECKING 4SAT USING THE LOW-DEGREE LONG CODE

In this section, our goal is to give a low-degree long code based PCP that has perfect completeness. The smallest

number of queries for which we are able to do so is 4 queries. The predicate tested by the PCP will be 4SAT (actually we can test a slightly stronger arity 4 predicate $x \vee y \vee (z \neq w)$). As a result we will prove Theorem 3 on the inapproximability of 4SAT stated in the introduction. Our construction is inspired by Håstad's tight inapproximability result for satisfiable instances of 4SAT [9, Theorem 6.2]. The analysis here is more subtle due to the restriction of using the low-degree long code. Our main motivation here is to illustrate these techniques in the simple setting of 4SAT, before applying them to show hardness for covering CSP later on.

As explained in Section II-E, we will describe the PCP verifier as a randomized test that checks if a Label Cover instance is satisfiable, or highly unsatisfiable, in the sense of Theorem 5. The verifier will have access to tables $A^{(v)}$ and $B^{(u)}$ of purported low-degree long codes of the labels of the nodes $u \in U$ and $v \in V$ of the Label Cover instance.

However, there will be some key differences here. First, the table for the "smaller" side will be a low-degree long code for smaller degree ($3d/4$ as opposed to d). Second, there will be *two* tables for the nodes on the "larger" side, with one being a low-degree long code of smaller degree. This structure seems technically necessary as we need to restrict the degree of some of the functions to be smaller than d, and in this case the analysis necessitates making them from a separate low-degree long code so that they will be well-distributed amongst the coordinates of that low-degree long code. Let us proceed with the formal description of the PCP construction.

Let $G = (U, V, E)$ be a Label Cover instance with parameter ℓ as promised in Theorem 5. The integer d will be a degree parameter that we will choose later.

For each $v \in V$ we add a block of variables corresponding to $P'(\ell, d)$ (recall that $P'(\ell, d)$ contains for each $g \in P(\ell, d)$ exactly one of g and $1 + g$). For each $u \in U$, we add *two* blocks of variables, one corresponding to $P'(3\ell, d/4)$ and another corresponding to $P'(3\ell, d)/I^{(u)}$ (where $I^{(u)}$ denotes the same ideal corresponding to node u as in Section **??**).

Let us denote $m = 3\ell$. An assignment for the variables is described by a collection of functions $A^{(v)} : P'(\ell, d) \to \{-1, 1\}$ for each $v \in V$, and functions $C^{(u)} : P'(m, d/4) \to \{-1, 1\}$, $B^{(u)} : P'(m, d)/I^{(u)} \to \{-1, 1\}$. We can extend the functions in the natural way to assume we have access to functions $A^{(v)} : P(\ell, d) \to \{-1, 1\}$, $C^{(u)} : P(m, d/4) \to \{-1, 1\}$ that are folded over $\{-1, 1\}$, and a function $B^{(u)} : P(m, d) \to \{-1, 1\}$ that is folded over $\{-1, 1\}$ and $I^{(u)}$.

We now describe our PCP, which we call 4SAT-PCP

1) Choose a random edge (u, v) in the label cover instance, and let $\pi_{uv} : \mathbb{F}_2^m \to \mathbb{F}_2^\ell$ be the associated projection.
   For notational simplicity, we denote $\pi = \pi_{uv}$, $A = A^{(v)}$, $B = B^{(u)}$ and $C = C^{(u)}$.
2) Sample functions $f \in P(\ell, 3d/4)$, $g \in P(m, d/4)$, $\tilde{g} \in P(m, d)$ and $h \in P(m, 3d/4)$, where each function is chosen independently at random from its respective domain.
3) Denote $g' = \tilde{g} + gh + (1 + g)(1 + f \circ \pi)$ and note that $g' \in P(m, d)$.
   Accept iff at least one of $A(f)$, $C(g)$, $B(\tilde{g})$, and $B(g')$ equals -1.

A. Completeness

We first establish the perfect completeness of the test which will also explain the logic behind the test.

Lemma 21. *If G is satisfiable, then there are tables $A^{(v)}$, $B^{(u)}$, and $C^{(u)}$ for which the test* 4SAT-PCP *accepts with probability* 1. *In particular, there are tables so that the four bits read by the verifier are never all equal to* 1.

Proof: Given a perfectly satisfying assignment for G, let us assign each $A^{(v)}$ to be $\mathsf{SC}_{\ell,3d/4}(a)$, the degree-$3d/4$ low-degree long code of a, where $a \in \mathbb{F}_2^\ell$ is the label for v. Similarly, define $B^{(u)} = \mathsf{SC}_{m,d}(b)$ and $C^{(u)} = \mathsf{SC}_{m,d/4}(b)$ where b is the label for u. For the choice of edge (u, v), the condition checked by the test amounts to $f(a) = 1 \vee g(b) = 1 \vee \tilde{g}(b) = 1 \vee g'(b) = 1$. To prove that this holds, let us assume $f(a) = g(b) = 0$ and then argue that in this case $\tilde{g}(b) \neq g'(b)$ (which in particular means one either $\tilde{g}(b)$ or $g'(b)$ equals 1). Indeed $\tilde{g}(b) + g'(b) = g(b)h(b) + (1 + g(b))(1 + f(a)) = 1$ when $f(a) = g(b) = 0$. Note that we have shown the more stringent condition $A(f) = -1$ or $C(g) = -1$ or $B(\tilde{g}) \neq B(g')$ always holds in the completeness case. ∎

We omit the soundness analysis due to space considerations. The parameters are chosen as follows. Picking $\ell = 2^{\lfloor\sqrt{\log\log n}\rfloor/4}$ and $d = \lfloor\sqrt{\log\log n}\rfloor$, the size of the instance produced will be at most polynomial in $N \leqslant n^{3\ell}2^{(3\ell)^d} \leqslant n^{2^{O(\sqrt{\log\log n})}}$, and the reduction will run in $N^{O(1)}$ time. As a function of N, we have $\ell \geqslant 2^{\Omega(\sqrt{\log\log N})}$.

V. 6-QUERY COVERING PCP USING LOW-DEGREE LONG CODE

In this section, we prove Theorem 2, showing it is hard to decide if a given instance of a φ-CSP has covering number 1 or at least $k = 2^{O(\sqrt{\log\log n})}$, where the predicate φ is defined by $\varphi(a, b, c, d, e, f) = (a \neq b) \vee (c \neq d) \vee (e \neq f)$.

Before moving to the proof, let us mention that since this predicate involves monotone Boolean operations over inequality constraints on the variables, it makes sense to assign variables with any number of colors (rather than Boolean values only). Given a φ-CSP instance over variables X, such that the variables occur without negations, we say it is c-colorable if there is a coloring of the variables $\psi : X \to \{1, 2, \ldots, c\}$ such that every constraint is satisfied.

It is easy to generalize the connection in [8] to this case, showing that the logarithm of this version of the chromatic number is equal to its covering number. Thus, an equivalent statement of Theorem 2 is the following.

Theorem 22. *Assume that* NP *does not admit* $n^{2^{O(\sqrt{\log\log n})}}$ *time algorithms (note that this runtime is* $n^{o(\log n)}$*). Given a* φ*-CSP instance with* N *vertices, there is no polynomial time algorithm to distinguish between the following two cases:*

- *The instance can be colored with* $c = 2$ *colors.*
- *The instance cannot be colored even with* $2^{2^{\Omega(\sqrt{\log\log N})}}$ *colors.*

For 4SAT, there are trivially two assignments such that each constraint is satisfied by one of them, so its covering number is always at most 2. So 4SAT-PCP from the previous section cannot give the desired coloring hardness. However, we will show that a small change to the test gives us the desired PCP with a total of 6 queries. Specifically, we will replace the condition $A(f) = -1$ with the check $A(f_1) \neq A(f_1 + f)$, and the condition $C(g) = -1$ with the check $C(g_1) \neq C(g_1 + g)$.

As in Section IV we begin with a label cover instance $G = (U, V, E)$, and place low-degree long code tables for the vertices of G. Namely, for each $v \in V$, a table $A^{(v)} : P(\ell, 3d/4) \to \{-1, 1\}$, and for each $u \in U$, two tables $C^{(u)} : P(m, d/4) \to \{-1, 1\}$ and $B^{(u)} : P(m, d)/I^{(u)} \to \{-1, 1\}$ (where $m = 3\ell$). We will *not* assume that any of these tables are folded over $\{-1, 1\}$, and this implies that the generated CSP instance will have no negations. Once again, we will extend $B^{(u)}$ to all of $P(m, d)$ by defining $B(h) = B(h + I^{(u)})$, and assume that $B^{(u)} : P(m, d) \to \{-1, 1\}$ is folded over $I^{(u)}$.

We now describe our PCP, which we call 6-NE-PCP

1) Choose a random edge (u, v) in the label cover instance, and let $\pi_{uv} : \mathbb{F}_2^m \to \mathbb{F}_2^\ell$ be the associated projection.
   For notational simplicity, we denote $\pi = \pi_{uv}$, $A = A^{(v)}$, $B = B^{(u)}$ and $C = C^{(u)}$.
2) Sample functions $f, f_1 \in P(\ell, 3d/4)$, $g, g_1 \in P(m, d/4)$, $\tilde{g} \in P(m, d)$ and $h \in P(m, 3d/4)$, where each function is chosen independently at random from its respective domain. Denote $g' = \tilde{g} + gh + (1+g)(1+f\circ\pi)$ and note that $g' \in P(m, d)$.
3) Accept iff

$$(A(f_1) \neq A(f_1+f)) \vee (C(g_1) \neq C(g_1+g)) \vee (B(\tilde{g}) \neq B(g')) . \tag{11}$$

Perfect Completeness. By an argument identical to Lemma 21, we can show that when there is a labeling satisfying every edge of G, there are tables that 6-NE-PCP accepts with probability 1.

Soundness analysis. As in Section IV, it can be proved that when the Label Cover instance G is highly unsatisfiable, no choice of tables will make the 6-NE-PCP test accept with probability more than $7/8$ (again random tables will be accepted with this probability, so this bound is tight). Given our interest in covering soundness we will show that even a large number of proofs cannot cover every test made by the verifier. The formal statement follows.

Theorem 23. *If every assignment of labels to the Label Cover instance* G *satisfies at most a fraction* $2^{-\Omega(\ell)}$ *of the edges, and* $d = 4\lceil \log \ell \rceil$*, then there exists* $k = \Omega(\ell)$ *such that for every set of* k *tables there is some check* (11) *that is violated by all of them.*

Proof: Suppose there are k proofs such that every check (11) accepts at least one of them. Let $\rho = 1/2^k$. Then, viewing these k proofs as a 2^k-coloring, we can choose a subset consisting of a fraction ρ of the locations of each of the $A^{(v)}$-tables, and similarly for the $B^{(u)}$ and $C^{(u)}$-tables, such that no check (11) has all 6 queries amongst the chosen locations. (To see this simply take the most popular color class in each of the tables.) To express this analytically, let $F^{(v)} : P(\ell, 3d/4) \to \{0, 1\}$ be the indicator function of this subset restricted to $A^{(v)}$, and similarly define indicator functions $G^{(u)} : P(m, d/4) \to \{0, 1\}$ and $H^{(u)} : P(m, d) \to \{0, 1\}$ corresponding to the tables $C^{(u)}$ and $B^{(u)}$ respectively. Further, $H^{(u)}$ can be assumed to be folded over $I^{(u)}$. By construction, we have for every u, v

$$\mathop{\mathbb{E}}_{f}[F^{(v)}(f)] = \mathop{\mathbb{E}}_{g}[G^{(u)}(g)] = \mathop{\mathbb{E}}_{h}[H^{(u)}(h)] = \rho \ . \tag{12}$$

and $\delta \stackrel{\text{def}}{=} \mathbb{E}_{u,v}\Big[\mathbb{E}\big[F^{(v)}(f_1)F^{(v)}(f_1+f)G^{(u)}(g_1)G^{(u)}(g_1+g)H^{(u)}(\tilde{g})H^{(u)}(\tilde{g}+gh+\overline{g}\ \overline{f\circ\pi_{uv}})\big]\Big] = 0$, where the inner expectation is over the choice of all the functions $f, f_1, g, g_1, \tilde{g}, h$. Our goal is to prove that the above equations imply $\rho \leqslant 2^{-\Omega(\ell)}$. We will analyze the inner expectation in the definition of δ for a fixed (u, v), call it $\Gamma_{(u,v)}$. We will use the shorthand $F = F^{(v)}$, $G = G^{(u)}$, $H = H^{(u)}$ and $\pi = \pi_{uv}$. Let us define the "self-corrected" versions $\tilde{F}$ and $\tilde{G}$ of the tables F and G as $\tilde{F}(f) = \mathbb{E}_{f_1}[F(f_1)F(f_1+f)]$ and $\tilde{G}(g) = \mathbb{E}_{g_1}[G(g_1)G(g_1+g)]$ respectively. Note that the tables $\tilde{F}$ and $\tilde{G}$ take values in the interval $[0, 1]$.

Using Fourier expansion, the expectation $\Gamma_{(u,v)}$ can be written as the sum

$$\sum \widehat{H}(\beta)^2 \underbrace{\mathop{\mathbb{E}}_{g}\Big[\tilde{G}(g)\ \mathop{\mathbb{E}}_{h}\big[\chi_\beta(gh)\big]\ \mathop{\mathbb{E}}_{f}\big[\tilde{F}(f)\chi_\beta(\overline{gf\circ\pi})\big]\Big]}_{\Upsilon_g} \tag{13}$$

over $\beta \in \Lambda(m, d)$. Note that the $\beta = \gamma = 0$ term equals $\widehat{H}(0)^2\ \mathbb{E}_g[\tilde{G}(g)]\ \mathbb{E}_f[\tilde{F}(f)] = \Big(\mathbb{E}_h[H(h)]\Big)^2\ \Big(\mathbb{E}_g[G(g)]\Big)^2\ \Big(\mathbb{E}_f[F(f)]\Big)^2 = \rho^6$ using (12).

Our goal is to prove that the rest of the terms (for $\beta \neq 0$)

in (13) have a very small contribution. First, the terms in (13) with $\mathsf{wt}(\beta) \geqslant 2^{d/2}$ contribute at most $2^{-2^{d/4}}$ in absolute value. For terms with $\mathsf{wt}(\beta) < 2^{d/2}$, note that $\mathbb{E}_h[\chi_\beta(gh)] = \mathbb{E}_h[\chi_{\beta g}(h)] = 0$ unless $\beta g = 0$. This follows from Fact 6 because $\mathsf{wt}(\beta g) < 2^{d/2}$ and so βg cannot be a nonzero polynomial of degree $P(m, m - 3d/4 - 1)$. Expanding $\tilde{F}(f) = \sum_\alpha \widehat{F}(\alpha)^2 \chi_\alpha(f)$, we can simplify the expected value Υ_g in (13) as

$$\begin{aligned}
\Upsilon_g &= \mathop{\mathbb{E}}_g\Big[\tilde{G}(g)\ \mathbf{1}[\beta g = 0]\ \mathop{\mathbb{E}}_f\Big[\sum_\alpha \widehat{F}(\alpha)^2 \chi_\alpha(\overline{f})\chi_{\beta\overline{g}}(f \circ \pi)\Big]\Big] \\
&= \mathop{\mathbb{E}}_g\Big[\tilde{G}(g)\ \mathbf{1}[\beta g = 0]\ (-1)^{\mathsf{wt}(\pi_2(\beta))}\widehat{F}(\pi_2(\beta))^2\Big] \\
&\geqslant \begin{cases} 0 & \text{when } \mathsf{wt}(\beta) \text{ is even} \\ -\widehat{F}(\pi_2(\beta))^2 & \text{when } \mathsf{wt}(\beta) \text{ is odd} \end{cases} \qquad (14)
\end{aligned}$$

where in the last step we use the fact that $\mathsf{wt}(\beta)$ and $\mathsf{wt}(\pi_2(\beta))$ have the same parity.

Combining the above we can lower bound δ as

$$\delta \geqslant \rho^6 - 2^{-2^{d/4}} - \sum_{\substack{\beta : \mathsf{wt}(\beta) < 2^{d/2} \\ \mathsf{wt}(\beta)\ \text{odd}}} \widehat{F}(\pi_2(\beta))^2 \widehat{H}(\beta)^2 \ .$$

Appealing to Lemma 13, the sum in the above expression is at most $2^{d-\Omega(\ell)}$ when the Label Cover instance G is at most $2^{-\Omega(\ell)}$-satisfiable. Recalling $\delta = 0$ and $\rho = 1/2^k$, we conclude $k \geqslant \Omega(\ell)$ when $d = \Theta(\log \ell)$. ∎

Picking parameters as in Section IV we get a proof of Theorem 2 (alternatively stated as Theorem 22 at the beginning of this section).

VI. Concluding remarks

Our work raises several open questions, some of which we mention below: Can one remove the multipartite structural bottleneck of our low-degree long code based PCP constructions and prove improved hardness results for hypergraph coloring?

Can one prove a gap-covering result of 1 vs. $\exp(\Omega(\sqrt{\log \log N}))$ (or at least $O(1)$ vs. $\omega(\log \log N)$) with fewer than 6 queries? In particular, can one give a low-degree long code based version of Håstad's 4-set splitting PCP [9, Sec. 7]?

Can one derandomize the long code further and move closer to $N^{\Omega(1)}$ (or at least $2^{(\log N)^{\Omega(1)}}$) hardness for hypergraph coloring or related problems?

References

[1] S. Arora and C. Lund. *Approximation Algorithms for NP-hard Problems*, chapter Hardness of Approximations. PWS Publishing, 1996.

[2] B. Barak, P. Gopalan, J. Håstad, R. Meka, P. Raghavendra, and D. Steurer. Making the long code shorter. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 370–379, 2012.

[3] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and nonapproximability: Towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.

[4] A. Bhattacharyya, S. Kopparty, G. Schoenebeck, M. Sudan, and D. Zuckerman. Optimal testing of reed-muller codes. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, pages 488–497, 2010.

[5] M. Bodirsky and J. Kára. The complexity of equality constraint languages. *Theory Comput. Syst.*, 43(2):136–158, 2008.

[6] I. Dinur and G. Kol. Covering CSPs. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:88, 2012. Extended abstract to appear in CCC 2013.

[7] I. Dinur, O. Regev, and C. D. Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, 2005.

[8] V. Guruswami, J. Håstad, and M. Sudan. Hardness of approximate hypergraph coloring. *SIAM J. Comput.*, 31(6):1663–1686, 2002.

[9] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[10] J. Håstad. On the NP-hardness of Max-Not-2. In *15th International Workshop on Approximation, Randomization, and Combinatorial Optimization (APPROX)*, pages 170–181, 2012.

[11] S. Khot. Hardness results for approximate hypergraph coloring. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing*, pages 351–359, 2002.

[12] S. Khot. Hardness results for coloring 3 -colorable 3 -uniform hypergraphs. In *Proceedings of 43rd Symposium on Foundations of Computer Science*, pages 23–32, 2002.

[13] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, 2002.

[14] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

[15] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. *Ann. Math.*, 171(1):295–341, 2010.