Hardness Results for Approximate Hypergraph Coloring

Subhash Khot *
Princeton University, NJ-08544
khot@cs.princeton.edu

ABSTRACT

Guruswami et al [6] show the hardness of coloring 2-colorable 4-uniform hypergraphs on n vertices with $\Omega(\frac{\log\log n}{\log\log\log n})$ colors assuming NP $\not\subseteq$ DTIME($n^{O(\log\log n)}$). We obtain a stronger hardness result for approximate coloring of p-colorable 4-uniform hypergraphs for any fixed integer $p \geq 7$. We prove that there exists an absolute constant $c > 0$ such that for every fixed integer $p \geq 7$, it is hard to color a p-colorable 4-uniform hypergraph with $(\log n)^{cp}$ colors assuming NP $\not\subseteq$ DTIME($2^{(\log n)^{O(1)}}$).

This work builds on the idea of "covering complexity" of probabilistically checkable proof systems (PCPs) developed in [6] and we introduce some new techniques as well. Firstly, we define a new code which we call the Split Code. This is a variation of the Long Code, but much shorter in length and it reduces the proof size significantly. Split Codes enable us to exploit the special structure of the "outer PCP verifier" constructed via Raz's Parallel Repetition Theorem [18]. Secondly, we make a novel use of the Split Codes over the domain $GF(p)$ for a prime p. Working over non-boolean domain in fact makes our proof technically simpler than the proof of Guruswami at al [6].

*This work was partly supported by Sanjeev Arora'a David and Lucile Packard Fellowship and NSF Grant CCR-0098180

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
STOC'02, May 19-21, 2002, Montreal, Quebec, Canada.
Copyright 2002 ACM 1-58113-495-9/02/0005 ...$5.00.

1. INTRODUCTION

Hypergraph coloring is a natural generalization of graph coloring. The edges of a q-uniform hypergraph are subsets of size q of the set of vertices and the goal is to color the vertices so that for every edge at least two of its vertices have different colors. Hypergraph coloring has been considered in literature for long, both from combinatorial and algorithmic point of view (see [1], [13], [14], [15], [16], [17]).

In this paper we consider the problem of *approximate coloring* of hypergraphs, which means coloring a hypergraph with a small number of colors when the hypergraph is guaranteed to be colorable with a fixed constant number of colors.

There hasn't been much progress on this problem when restricted to graphs (as opposed to hypergraphs). Khot [12] shows that for all sufficiently large constants k, it is NP-hard to color k-colorable graphs with $k^{\Omega(\log k)}$ colors. There is a huge gap between this hardness result and the best algorithmic result due to Karger et al [11] who give an algorithm to color k-colorable graphs with $\tilde{O}(n^{1-\frac{3}{k+1}})$ colors. It remains an open problem whether k-colorable graphs (in particular $k = 3$) are hard to color with constantly many colors or even $\Omega(\log n)$ colors.

In light of the failure to prove hardness results for approximate graph coloring, it is natural to ask whether one can say anything about hypergraph coloring. Indeed Guruswami et al [6] were able to obtain strong hardness results for coloring 4-uniform hypergraphs. They showed that it is hard to color 2-colorable 4-uniform hypergraphs with $\Omega(\frac{\log\log n}{\log\log\log n})$ colors assuming NP $\not\subseteq$ DTIME($n^{O(\log\log n)}$). However this result is still very far from the algorithmic result of Krivelevich and Sudakov [13] which colors 2-colorable 4-uniform hypergraphs with $\tilde{O}(n^{3/4})$ colors in polynomial time.

In this paper, we prove a much stronger hardness result for coloring p-colorable 4-uniform hypergraphs for every integer $p \geq 7$. We prove that

THEOREM 1. *There exists an absolute constant $c > 0$ such that for every integer $p \geq 7$, it is hard to color p-colorable 4-uniform hypergraphs with $(\log n)^{cp}$ colors assuming NP$\not\subseteq$DTIME($2^{(\log n)^{O(1)}}$).*

Note that the function $(\log n)^{cp}$ exceeds every polylog function as p increases. In comparison, the known algorithms color (hyper-)graphs with a fixed constant chromatic number with $n^{\Omega(1)}$ colors. (see [5], [11], [1], [13]).

The proof of Theorem 1 builds on the idea of *covering complexity* of probabilistically checkable proof systems (PCPs) introduced by Guruswami et al [6]. In the standard notion of PCPs, a probabilistic verifier for a language L, on an input x of size n, uses a random string of length $r = r(n)$, queries $q = q(n)$ bits from a proof Π and has the following two properties : (Completeness) If $x \in L$, there exists a proof Π which is accepted with probability 1. (Soundness) If $x \notin L$, no proof is accepted with probability more than s where $s < 1$ is the *soundness parameter*. Recall the PCP Theorem ([3], [2]) that NP has (polynomial time) PCP verifiers with $r = O(\log n), q = O(1)$ and $s = \frac{1}{2}$. In the new model of PCPs introduced in [6], one requires k proofs $\Pi_1, \Pi_2, \ldots, \Pi_k$ such that for every choice of the random string, the verifier accepts at least one proof Π_i. The minimum number of proofs k required to satisfy this property is called the *covering complexity* of the PCP. The verifier has the following properties : (Completeness) If $x \in L$, there exists a proof Π which the verifier accepts on every random string. (Soundness) If $x \notin L$, at least k proofs are needed to make the verifier accept on every random string, where k is called the covering complexity of the PCP.

Guruswami et al [6] show that PCPs with high covering complexity (and a specific acceptance predicate) imply good hardness results for hypergraph coloring. We prove Theorem 1 by constructing a PCP with high covering complexity with proofs over the alphabet $GF(p)$.

Our techniques : We introduce a new code which we call the Split Code. It is a variation of the Long Codes which have been extensively used in PCP constructions (see [4], [7], [8], [6]). The Split Codes are much shorter in length than the Long Codes and significantly reduce the proof size and consequently the size of the hypergraph produced. The Split Codes are designed to exploit the structure of the so called "outer PCP verifier" (see [6]) . Specifically we exploit the fact that the outer verifier is constructed from Raz's Parallel Repetition Theorem [18].

Using the Split Codes in our construction requires us to employ Split Codes over large domain i.e. $GF(p)$. We make a novel and essential use of the fact that we are working over non-boolean domain. In fact this makes our proof simpler and more self-contained than the proof of [6]. In particular we avoid using much of the technical machinery from Håstad's paper [8].

The Split Codes can be seen as a general technique for reducing proof size. Håstad and Srinivasan [9] use Split Codes to show the following result for the problem 3-*Linear-Equations* where one is given a system of linear equations modulo 2, each equation containing 3 variables and the goal is to find an assignment that satisfies the maximum number of equations.

THEOREM 2. *[9] There exists a constant $\gamma > 0$ such that it is hard to distinguish between instances of 3-Linear-Equations where there exists an assignment that satisfies $1 - \frac{1}{(\log n)^{\gamma}}$ fraction of equations or no assignment can satisfy more than $\frac{1}{2} + \frac{1}{2^{(\log n)^{\gamma}}}$ fraction of equations if $NP \not\subseteq DTIME(2^{(\log n)^{O(1)}})$.*

Split Codes can also be used to give an alternate proof of Khot's [12] result that it is hard to approximate MaxClique within factor $\frac{n}{2^{(\log n)^{1-\gamma}}}$ for some constant $\gamma > 0$.

Other results in the paper : One can consider the following natural maximization problem : color the vertices of a 4-uniform hypergraph with p colors so as to maximize the number of edges which are non-monochromatic. Note that for a random p-coloring of the vertices, $1 - \frac{1}{p^3}$ fraction of the edges will be non-monochromatic. The following result is implicit in this paper.

THEOREM 3. *For every prime $p \geq 7$ and arbitrarily small $\epsilon > 0$, it is NP-hard to color the vertices of a p-colorable 4-uniform hypergraph with p colors such that more than $1 - \frac{1}{p^3} + \epsilon$ fraction of the edges are non-monochromatic.*

Guruswami et al [6] give a reduction from 4-uniform hypergraphs to q-uniform hypergraphs for any $q \geq 5$. One can use this reduction and extend Theorem 1 to :

THEOREM 4. *There exists an absolute constant $c > 0$ such that for fixed integers $p \geq 7$ and $q \geq 5$, it is hard to color p-colorable q-uniform hypergraphs with $(\log n)^{cp}$ colors assuming $NP \not\subseteq DTIME(2^{(\log n)^{O(1)}})$.*

The following theorem is a straightforward consequence of Theorem 1. This result does not follow from Guruswami et al's result [6]. An analogous result for graphs is an open problem.

THEOREM 5. *If $NP \not\subseteq DTIME(2^{(\log n)^{O(1)}})$, for a sufficiently large constant p, for all constants $\delta > 0$, it is hard to find an independent set of size δn in a 4-uniform hypergraph which is guaranteed to contain an independent set of size $\frac{n}{p}$.*

Remark : Holmerin [10] in fact proves that it is NP-hard to find an independent set of size δn in a 4-uniform hypergraph that contains an independent set of size $\frac{n}{2}$. His result has an additional advantage that it holds w.r.t. Karp reduction whereas hardness in Theorem 5 holds w.r.t. Turing reduction. (If there exists a polynomial time algorithm that finds an independent set of size δn is a p-colorable 4-uniform hypergraph, then this algorithm can be repeatedly used $O(\log n)$ times to color the hypergraph with $O(\log n)$ colors. This contradicts Theorem 1 that for a sufficiently large constant p, it is hard to color a p-colorable 4-uniform hypergraph with $(\log n)^2$ colors).

Overview of the paper : Section 2 describes the preliminary background including the idea of covering complexity of PCPs, Split Codes and the basic paradigm for PCP constructions. Section 3 gives our PCP construction. The analysis, which is rather long and involved, is presented in Section 4. Section 5 concludes with some remarks and open problems.

2. PRELIMINARIES

In this section we introduce some of the techniques used in this paper.

2.1 Covering Complexity of PCPs

The following idea was introduced by Guruswami et al [6].

Definition 1. A PCP verifier with parameters (r, q, k, p) for a language L

- Uses $r = r(n)$ random bits on input x of size n.
- Queries q symbols from a proof Π where the proof is over the alphabet $GF(p)$.
- Satisfies the following two properties :

1. (Completeness) : If $x \in L$, there exists a proof Π which is accepted with probability 1.
2. (Soundness) : If $x \notin L$, at least $k = k(n)$ proofs $\Pi_1, \ldots, \Pi_k$ are needed so that for every choice of the random string, the verifier accepts at least one proof $\Pi_i, 1 \leq i \leq k$.

We call k the covering complexity of the verifier. Here q and p are assumed to be fixed integers.

Guruswami et al [6] make the following observation :

THEOREM 6. *Suppose NP has a PCP verifier with parameters (r, q, k, p) such that*

- *The verifier runs in time $2^{O(r)}$.*
- *The verifier reads q symbols from the proof and accepts iff these q symbols are not all equal.*

Then it is hard to color p-colorable q-uniform hypergraphs with p^{k-1} colors assuming $NP \not\subseteq DTIME(2^{O(r)})$.

PROOF. The idea behind this theorem is simple. We construct a hypergraph whose vertices correspond to the positions in the proof. For every choice of the random string by the verifier, there is an edge consisting of the q positions which the verifier reads. This is clearly a q-uniform hypergraph.

In the completeness case, there is a proof Π that is always accepted. For every vertex in the hypergraph we assign a color which is the symbol in the proof Π at the corresponding position. (Recall that the proof is over the alphabet $GF(p)$). The fact that Π is always accepted and the acceptance predicate of the verifier is Not-all-equal on the symbols it reads implies that the hypergraph can be colored with p colors.

In the soundness case, suppose the hypergraph can be colored with p^{k-1} colors. We identify these colors with column vectors in $GF(p)^{k-1}$. Imagine that the vertices of the hypergraph are arranged in a row and the color of every vertex is a column vector in $GF(p)^{k-1}$. Now for every $1 \leq i \leq k-1$, construct a proof Π_i with alphabet $GF(p)$ as follows. For every position in the proof Π_i, let this position contain the i^{th} coordinate of the color assigned to the vertex corresponding to this position. Now the fact that for every edge, its vertices are not all of the same color in $GF(p)^{k-1}$ implies that for every choice of the random string of the verifier, for at least one proof Π_i, the symbols in Π_i at the positions corresponding to this edge are not all equal. In other words, for every choice of the random string, at least one proof Π_i is accepted. This contradicts the fact that the covering complexity is k. $\square$

From Theorem 6 we see that for proving Theorem 1 it suffices to build a PCP verifier that queries 4 symbols over $GF(p)$, accepting iff not all 4 are equal and has high enough covering complexity. The first step in constructing such a verifier is the gap-version of Label Cover problem which we define next.

2.2 Hardness of Label Cover Problem

Let Label Cover be the following problem : We are given a bipartite graph $G = ((V, W), E)$ such that all vertices in V have the same degree and all vertices in W have the same degree. There is a set of labels $M = M_1 \times M_2 \ldots \times M_t$ for vertices in W and a set of labels $N = N_1 \times N_2 \ldots \times N_t$ for vertices in V. For every edge $(v, w) \in E$, there is a projection function $\pi^{v,w} : M \mapsto N$ which is given by componentwise projections $\pi^{v,w} = (\pi^{v,w,1}, \ldots, \pi^{v,w,t})$ where $\pi^{v,w,i} : M_i \mapsto N_i$ and

$$\pi^{v,w}(b_1, b_2, \ldots, b_t) = (\pi^{v,w,1}(b_1), \ldots, \pi^{v,w,t}(b_t))$$

$$\forall\ (b_1, b_2, \ldots, b_t) \in M$$

The goal is to assign labelings $L_V : V \mapsto N$ and $L_W : W \mapsto M$ such that

$$\forall\ (v, w) \in E, \quad \pi^{v,w}(L_W(w)) = L_V(v)$$

We say that an edge is "satisfied" if this condition holds for that edge. The optimal value of the Label Cover problem is the maximum fraction of the edges that can be satisfied by any labeling. The following is a consequence of the PCP Theorem ([3], [2]) and Raz's Parallel Repetition Theorem [18].

THEOREM 7. *There exist absolute constants d_0, e_0 such that for all integers u, t, there exists a reduction from a 3-SAT formula ϕ of size n to a Label Cover instance $\mathcal{L}$ such that*

1. *$|V|, |W| \leq n^{d_0 ut}$ and the reduction runs in time $n^{O(ut)}$.*
2. *$|M_i|, |N_i| \leq 2^{e_0 u}$ $\forall$ $1 \leq i \leq t$.*
3. *If ϕ is satisfiable, $OPT(\mathcal{L}) = 1$.*
4. *If ϕ is unsatisfiable, $OPT(\mathcal{L}) \leq 2^{-ut}$.*

Note : The important idea here is to exploit the fact that for any constant $t \geq 1$, Raz's Parallel Repetition Theorem gives a Label Cover instance where sets M and N have the cartesian product structure $M = M_1 \times M_2 \times \ldots M_t$ and $N = N_1 \times N_2 \times \ldots N_t$ and every projection $\pi^{v,w}$ is actually a combination of componentwise projections $\pi^{v,w,1}, \ldots, \pi^{v,w,t}$. Previous PCP constructions do not exploit this property. The purpose of the Split Codes defined in the next section is precisely to exploit this property of the Raz's Parallel Repetition Theorem.

2.3 The Split Codes

Following the standard paradigm (see [4], [8], [6]), our PCP verifier reduces a 3-SAT instance to the Label Cover instance $G((V, W), E)$ given by Theorem 7 and expects the proof to contain encodings of the labels of the vertices in V and W. The previous PCP constructions use the Long Code, whereas we use the "Split Code" which we define next. (Note : The reduction to the gap-version of Label Cover problem has been called the "outer verifier" (see [4], [6])).

Let ω be the basic p^{th} root of unity, i.e. $\omega = e^{2\pi i/p}$. In this paper, the variables β, g, b stand for tuples and variables β_i, g_i, b_i stand for the i^{th} components of these tuples respectively.

Definition 2. The Split Code (over $GF(p)$) on a set

$$M = M_1 \times M_2 \ldots \times M_t$$

is indexed by all tuples $g = (g_1, g_2, \ldots, g_t)$ where g_i is a function $g_i : M_i \mapsto \{1, \omega, \omega^2, \ldots, \omega^{p-1}\}$.

The Split Code B of an element $b \in M$, $b = (b_1, b_2, \ldots, b_t)$, $b_i \in M_i$, is defined as

$$B(g) = \prod_{i=1}^{t} g_i(b_i)$$

The set of all functions $g_i : M_i \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ is denoted by $\mathcal{G}_i$.

Note that the length of the Split Code on the set M is $p^{|M_1|+|M_2|+\ldots+|M_t|}$. On the other hand, the length of the Long Code on the set M will be much larger i.e. $p^{|M|} = p^{|M_1|\cdot|M_2|\cdot\ldots\cdot|M_t|}$.

2.4 Fourier Analysis of Split Codes

Let $\mathcal{G}_1, \ldots, \mathcal{G}_t$ be as in Definition 2. Consider the complex vector space of all functions

$$B : \mathcal{G} = \mathcal{G}_1 \times \mathcal{G}_2 \times \ldots \times \mathcal{G}_t \mapsto \mathbf{C}$$

where the addition of two functions is defined as pointwise addition. This is a complex vector space of dimension $|\mathcal{G}| = p^{|M_1|+|M_2|+\ldots+|M_t|}$. Define an inner product on this space as

$$< B_1, B_2 > = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} B_1(g)\overline{B_2(g)} = E_g[B_1(g)\overline{B_2(g)}]$$

We will identify an orthonormal basis for this vector space. Let β_i be a function

$$\beta_i : M_i \mapsto GF(p)$$

and $\beta = (\beta_1, \beta_2, \ldots, \beta_t)$ be a tuple. The character χ_β is a function $\chi_\beta : \mathcal{G} \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ defined as

$$\text{For } g = (g_1, \ldots, g_t), \quad \chi_\beta(g) = \prod_{i=1}^{t} \prod_{x_i \in M_i} g_i(x_i)^{\beta_i(x_i)}$$

It is easy to verify that the characters χ_β form am orthonormal basis.

LEMMA 1. *The characters χ_β form an orthonormal basis.*

PROOF. We clearly have $< \chi_\beta, \chi_\beta > = 1$. For $\beta = (\beta_1, \ldots, \beta_t)$, $\gamma = (\gamma_1, \ldots, \gamma_t)$ and $\beta \neq \gamma$, there exists an index i_0 and an element $b_{i_0} \in M_{i_0}$ such that $\beta_{i_0}(b_{i_0}) \neq \gamma_{i_0}(b_{i_0})$. Hence

$$\begin{aligned} < \chi_\beta, \chi_\gamma > &= E_{g_1,\ldots,g_t}\left[\prod_{i=1}^{t} \prod_{x_i \in M_i} g_i(x_i)^{\beta_i(x_i)-\gamma_i(x_i)}\right] \\ &= \prod_{i=1}^{t} \prod_{x_i \in M_i} E_{g_i}\left[g_i(x_i)^{\beta_i(x_i)-\gamma_i(x_i)}\right] \end{aligned}$$

The inner expectation is zero when $i = i_0$ and $x_i = b_{i_0}$. $\square$

Hence every function $B : \mathcal{G} \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ can be written as

$$B = \sum_\beta \widehat{B}_\beta \chi_\beta \quad \text{with the Parseval's identity} \quad \sum_\beta |\widehat{B}_\beta|^2 = 1$$

2.5 Equality Folding of Split Codes

Recall that the proof for the PCP verifier is supposed to contain the Split Codes of the labels of all the vertices in V and W in the Label Cover instance given by Theorem 7.

It turns out that we can identify pairs of indices (g, g') such that for every correct Split Code B on a set M we have $B(g) = B(g')$. So we can identify the indices g and g' together which we call the “equality folding”. The positions in the Split Code are eventually going to be the vertices of a hypergraph and this identification corresponds to merging the vertices for positions g and g' into one vertex.

Definition 3. For a function $\beta_i : M_i \mapsto GF(p)$, define

$$weight(\beta_i) = \sum_{x \in M_i} \beta_i(x) \quad (in\ GF(p))$$

For a tuple $\beta = (\beta_1, \ldots, \beta_t)$, let $weight(\beta) = \sum_{i=1}^{t} weight(\beta_i)$.

Definition 4. We call a function $B : \mathcal{G} \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ equality folded if for $e_1, \ldots, e_t \in GF(p)$

$$e_1+e_2+\ldots+e_t = 0 \implies B(g_1, \ldots, g_t) = B(\omega^{e_1}g_1, \ldots, \omega^{e_t}g_t)$$

Note that a Split Code is always equality folded. The notion of equality folding forces only equality constraints and it is allright for coloring results. There are other notions of folding (see [8]) which cannot be used in coloring results, and in fact this turns out to be the biggest difficulty in proving hardness of hypergraph coloring. The following is a crucial consequence of equality folding :

LEMMA 2. *If a function $B : \mathcal{G} \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ is equality folded then $\forall\ \beta = (\beta_1, \ldots, \beta_t)$,*

$$\widehat{B}_\beta \neq 0 \implies weight(\beta_1) = weight(\beta_2) = \ldots = weight(\beta_t)$$

PROOF. Let $e = (e_1, \ldots, e_t)$ be a vector chosen at random such that $e_1 + e_2 + \ldots + e_t = 0$. There are p^{t-1} choices for e. We have

$$\begin{aligned} \widehat{B}_\beta &= E_{g=(g_1,\ldots,g_t)}\left[B(g_1, \ldots, g_t)\overline{\chi_\beta(g_1, \ldots, g_t)}\right] \\ &= \tfrac{1}{p^{t-1}} E_g\left[\textstyle\sum_e B(\omega^{e_1}g_1, \ldots, \omega^{e_t}g_t)\overline{\chi_\beta(\omega^{e_1}g_1, \ldots, \omega^{e_t}g_t)}\right] \\ &= \tfrac{1}{p^{t-1}} E_g\left[\textstyle\sum_e B(g)\overline{\chi_\beta(g)}\ \omega^{-\sum_{i=1}^{t} e_i \cdot weight(\beta_i)}\right] \end{aligned}$$

As e ranges over all vectors with $e_1 + e_2 + \ldots + e_t = 0$, the inner sum is zero unless $weight(\beta_1) = weight(\beta_2) = \ldots = weight(\beta_t)$. $\square$

3. THE PCP CONSTRUCTION

Now we are ready to define the PCP test that proves the desired hardness result for approximate hypergraph coloring. The verifier takes $t = \lfloor \frac{p-1}{6} \rfloor$ and expects the proof to contain Split Codes over $GF(p)$ of the labels of all the vertices $w \in W$ in the Label Cover instance $\mathcal{L}$ given by Theorem 7. (Note : The verifier does not need the encodings of the labels of the vertices in V). These codes are assumed to be equality folded. The verifier proceeds as follows :

1. Pick a vertex $v \in V$ at random and two of its neighbors $w, w' \in W$ at random. Let B be the supposed Split Code of the supposed label of w and C be the supposed Split Code of the supposed label of w'.

2. Pick random functions $g_i, h_i : M_i \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ and $f_i : N_i \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ for $1 \leq i \leq t$. Let $g = (g_1, \ldots, g_t)$, $h = (h_1, \ldots, h_t)$ and $f = (f_1, \ldots, f_t)$.

3. Pick functions $\mu_i, \mu_i' : M_i \mapsto \{\omega, \omega^2, \omega^3\}$ by defining independently for every $x \in M_i$

$$\mu_i(x) = \begin{cases} \omega & \text{with} \quad \text{probability} \;\; 0.2 \\ \omega^2 & \text{with} \quad \text{probability} \;\; 0.6 \\ \omega^3 & \text{with} \quad \text{probability} \;\; 0.2 \end{cases}$$

   The functions μ_i' are defined similarly. Let $\mu = (\mu_1, \ldots, \mu_t)$ and $\mu' = (\mu_1', \ldots, \mu_t')$.

4. Define function $g_i' : M_i \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ by defining for every $x \in M_i$,

$$g_i'(x) = g_i(x) f_i(\pi^{v,w,i}(x)) \; \mu_i(x)$$

   Let $g' = (g_1', \ldots, g_t')$. We will use the shortform $g' = g \cdot (f \circ \pi^{v,w}) \cdot \mu$ where $\circ$ denotes composition of functions.

5. In a similar way, (but note the differences) we define functions $h_i' : M_i \mapsto \{1, \omega, \ldots, \omega^{p-1}\}$ by defining for every $x \in M_i$,

$$h_i'(x) = h_i(x) f_i(\pi^{v,w',i}(x)) \; \overline{\mu_i'(x)}$$

   Let $h' = (h_1', \ldots, h_t')$. We will use the shortform $h' = h \cdot (f \circ \pi^{v,w'}) \cdot \overline{\mu'}$.

6. Read $B(g), B(g'), C(h), C(h')$ from the proof and accept iff all these are not equal.

The way we pick the functions μ, μ' is the most novel aspect of this construction (see the next subsection and the proof of Lemma 3).

3.1 Completeness and the Number of Random Bits Used

It is easy to see that the test always accepts a correct proof. In a correct proof, B is the Split Code of the label of w, say $b = (b_1, \ldots, b_t)$ and C is the Split Code of the label of w', say $c = (c_1, \ldots, c_t)$. Also v has a label $a = (a_1, \ldots, a_t)$ and we have correct projections i.e. $a_i = \pi^{v,w,i}(b_i) = \pi^{v,w',i}(c_i)$ for $1 \leq i \leq t$. Hence $B(g) = g(b)$,

$$B(g') = g'(b) = g(b) f(\pi^{v,w}(b)) \mu(b) = g(b) f(a) \mu(b),$$

$C(h) = h(c), C(h') = h(c) f(a) \overline{\mu'(c)}$. So the test will accept provided $\mu(b) \neq \overline{\mu'(c)}$, i.e. provided

$$\prod_{i=1}^{t} \mu_i(b_i) \neq \prod_{i=1}^{t} \overline{\mu_i'(c_i)}$$

Note that μ_i, μ_i' take values in the set $\{\omega, \omega^2, \omega^3\}$. Hence the left hand side is ω^r for some $t \leq r \leq 3t$ and the right hand side is ω^{-s} for some $t \leq s \leq 3t$. Since $p \geq 6t + 1$, the two sides are unequal !

Clearly the number of random bits used is at most $O(ut \log n + t \; 2^{e_0 u} \log p)$.

4. THE COVERING COMPLEXITY ANALYSIS

In this section, we prove that the PCP has high covering complexity. We show that if there are k proofs such that the verifier accepts at least one proof on every random string, then these proofs can be "decoded" to define a labeling to the Label Cover instance $\mathcal{L}$ with a "good" value of $OPT(\mathcal{L})$. However this will be a contradiction if $\mathcal{L}$ was chosen to have a sufficiently small value of $OPT(\mathcal{L})$ to begin with (see Theorem 7). We prove the following theorem :

THEOREM 8. *If there are k proofs such that the probability that the verifier accepts at least one proof is $1 - \frac{1}{p^{3k}} + \delta$, then there exists a labeling to the Label Cover instance $\mathcal{L}$ with $OPT(\mathcal{L}) \geq 2^{-O(t)} p^{-4t} \delta$.*

Note : A set of k random proofs will make the verifier accept with probability $1 - \frac{1}{p^{3k}}$. So the theorem says that doing even slightly better than a set of random proofs enables us to define a "good" labeling to $\mathcal{L}$.

First we state a couple of easy lemmas which we prove in Appendix A and B.

Definition 5. For a function $\beta_i : M_i \mapsto GP(p)$, let $|\beta_i| = |\{x \in M_i : \; \beta_i(x) \neq 0\}|$.

LEMMA 3. *For $\beta = (\beta_1, \ldots, \beta_t)$,*

$$\begin{aligned} E_{\mu=(\mu_1,\ldots,\mu_t)}[\chi_\beta(\mu)] &= C(\beta) \; \omega^{2 \; \sum_{i=1}^{t} weight(\beta_i)} \\ &= C(\beta) \; \omega^{2 \cdot weight(\beta)} \end{aligned}$$

where $C(\beta)$ is a positive real number satisfying

$$C(\beta) \leq (1 - \Omega(\frac{1}{p^2}))^{|\beta_1| + |\beta_2| + \ldots |\beta_t|}$$

LEMMA 4. *If $x, y, z, w \in \{1, \omega, \ldots, \omega^{p-1}\}$, then*

$$\frac{1}{p^3} \sum_{r_1, r_2, r_3, r_4 \in GF(p) \; : \; r_1 + r_2 + r_3 + r_4 = 0} x^{r_1} y^{r_2} z^{r_3} w^{r_4}$$

is 1 when x, y, z, w are all equal and 0 otherwise.

Now we will prove Theorem 8. Suppose there are k proofs that satisfy the hypothesis of Theorem 8. Let $(B_i, C_i)_{i=1}^{k}$ be the tables in these k proofs. Consider the following expression :

$$1 - \frac{1}{p^{3k}} \prod_{i=1}^{k} \left(\sum_{\substack{r_{i1}+r_{i2}+ \\ r_{i3}+r_{i4}=0}} B_i(g)^{r_{i1}} B_i(g')^{r_{i2}} C_i(h)^{r_{i3}} C_i(h')^{r_{i4}} \right) \tag{1}$$

where $r_{ij} \in GF(p)$.

From Lemma 4, if $(B_i(g), B_i(g'), C_i(h), C_i(h'))$ are all equal for every i, this expression is 0. If for some i, these are not all equal then this expression is 1. Thus the probability of accepting at least one proof is the expectation of this expression over the choices of vertices v, w, w', and functions g, g', h, h'.

We will expand out this expression and rearrange the terms into different groups. Each group can be bounded separately. This idea of bounding a group as a whole rather than trying to bound each term individually comes from Guruswami et al's paper [6]. We follow the general outline of their proof, though there are subtleties arising due to the peculiar choice of μ, μ' and the equality folding.

Let us first write $S_j = (r_{1j}, r_{2j}, \ldots, r_{kj})$ for $1 \leq j \leq 4$. Note that $S_1, S_2, S_3, S_4 \in GF(p)^k$ and $S_1 + S_2 + S_3 +$

$S_4 = 0$. For a vector $S = (r_1, \ldots, r_k) \in GF(p)^k$ and tables $B_1, \ldots, B_k$, let us define a table B_S as

$$B_S(g) = \prod_{l=1}^{k} B_l(g)^{r_l}$$

With this definition, the expectation of expression (1) can be written as

$$1 - \frac{1}{p^{3k}} \sum_{S_1+S_2+S_3+S_4=0} E\left[B_{S_1}(g)B_{S_2}(g')B_{S_3}(h)B_{S_4}(h')\right] \tag{2}$$

Let us now rearrange terms in expression (2) into groups. Define

$$\mathrm{Term(S)} = \sum_{T,T' \in GF(p)^k} E\left[B_T(g)B_{S-T}(g')C_{-T'}(h)C_{-S+T'}(h')\right]$$

Term(S) consists of all terms from (2) with $S_1 + S_2 = S$. Thus the acceptance probability is

$$\Pr[\mathrm{Acc}] = 1 - \frac{1}{p^{3k}} \sum_{S} \mathrm{Term(S)}$$

We distinguish between $S = 0$ and $S \neq 0$. The acceptance probability is :

$$\Pr[\mathrm{Acc}] = 1 - \frac{1}{p^{3k}} - \frac{1}{p^{3k}} \sum_{S\neq 0} \mathrm{Term(S)}$$

$$-\frac{1}{p^{3k}} \sum_{T\neq 0, T'\neq 0} E\left[B_T(g)B_{-T}(g')C_{-T'}(h)C_{T'}(h')\right]$$

$$-\frac{1}{p^{3k}} \sum_{T\neq 0} E\left[B_T(g)B_{-T}(g')\right] - \frac{1}{p^{3k}} \sum_{T'\neq 0} E\left[C_{-T'}(h)C_{T'}(h')\right]$$

$$= 1 - \frac{1}{p^{3k}} - \frac{1}{p^{3k}} \sum_{S\neq 0} \mathrm{Term(S)}$$

$$-\frac{1}{p^{3k}} X - \frac{1}{p^{3k}} \sum_{T\neq 0} Y(T) - \frac{1}{p^{3k}} \sum_{T'\neq 0} Z(T') \tag{3}$$

We first show that $Y(T), Z(T')$ are nonnegative real numbers. We often use the following lemma which is a trivial consequence of the fact that the tables B_T and B_{-T} are complex conjugates of each other.

LEMMA 5. $\widehat{B}_{T,\beta} = \overline{\widehat{B}}_{-T,-\beta}$

LEMMA 6. *$Y(T), Z(T')$ are nonnegative reals.*

PROOF. We prove the claim for $Y(T)$. The claim for $Z(T')$ can be proved similarly. Using Fourier expansions,

$$\begin{aligned} Y(T) &= E\left[B_T(g)B_{-T}(g(f \circ \pi^{v,w})\mu)\right] \\ &= E\left[\sum_{\beta,\gamma} \widehat{B}_{T,\beta}\widehat{B}_{-T,\gamma} \;\; \chi_{\beta+\gamma}(g)\; \chi_\gamma(f \circ \pi^{v,w})\; \chi_\gamma(\mu)\right] \end{aligned}$$

where $\beta = (\beta_1, \ldots, \beta_t)$, $\gamma = (\gamma_1, \ldots, \gamma_t)$ and $\beta + \gamma = (\beta_1 + \gamma_1, \ldots, \beta_t + \gamma_t)$. The expectation over g is zero unless $\beta + \gamma = 0$, i.e. every function $\beta_i + \gamma_i : M_i \mapsto GF(p)$ is identically zero. For function $\gamma_i : M_i \mapsto GF(p)$, and the projection $\pi^{v,w,i} : M_i \mapsto N_i$, let $\pi_p^{v,w,i}(\gamma_i)$ be a function $\pi_p^{v,w,i}(\gamma_i) : N_i \mapsto GF(p)$ defined by

$$\pi_p^{v,w,i}(\gamma_i)(a) = \sum_{x \in M_i : \pi^{v,w,i}(x) = a} \gamma_i(x) \quad \forall\, a \in N_i$$

Let

$$\pi_p^{v,w}(\gamma) = (\pi_p^{v,w,1}(\gamma_1), \pi_p^{v,w,2}(\gamma_2), \ldots, \pi_p^{v,w,t}(\gamma_t))$$

It is easy to verify that $\chi_\gamma(f \circ \pi^{v,w}) = \chi_{\pi_p^{v,w}(\gamma)}(f)$.

For expectation over f to be nonzero, we must have $\pi_p^{v,w}(\gamma) = 0$, i.e. for every coordinate i, the function $\pi_p^{v,w,i}(\gamma_i)$ is identically zero. From Lemma 3, we have $\chi_\gamma(\mu) = C(\gamma)\, \omega^{2\cdot weight(\gamma)}$. Also, Lemma 5 gives $\widehat{B}_{T,\beta} = \overline{\widehat{B}}_{-T,-\beta} = \overline{\widehat{B}}_{-T,\gamma}$. Hence

$$\begin{aligned} Y(T) &= \sum_{\gamma \;:\; \pi_p^{v,w}(\gamma)=0} |\widehat{B}_{-T,\gamma}|^2\; C(\gamma)\; \omega^{2\cdot weight(\gamma)} \\ &= \sum_{\gamma \;:\; \pi_p^{v,w}(\gamma)=0} |\widehat{B}_{-T,\gamma}|^2\; C(\gamma) \end{aligned}$$

since $\pi_p^{v,w}(\gamma) = 0$ implies that $weight(\gamma) = 0$. This proves that $Y(T)$ is a nonnegative real. □

4.1 Analyzing Term(S) and X

Now we show that if the real part of Term(S) or X is too negative, then we can extract labels for the Label Cover instance $\mathcal{L}$ giving a good value of $OPT(\mathcal{L})$. We analyze Term(S) for some fixed S. The analysis of X is similar. (X is same as Term(S) with $S = 0$, but the summation runs only over $T \neq 0, T' \neq 0$). Note that Term(S) is expectation over $(v, w, w', g, h, f, \mu, \mu')$. Omitting the $E[\cdot]$ notation for convenience,

$$\mathrm{Term(S)} = \sum_{T,T'} B_T(g)\; B_{S-T}(g(f \circ \pi^{v,w})\mu)\; \cdot$$

$$C_{-T'}(h)\; C_{-S+T'}(h(f \circ \pi^{v,w'})\overline{\mu'})$$

$$= \sum_{T,T'} \sum_{\beta,\phi,\gamma,\psi} \widehat{B}_{T,\beta}\; \widehat{B}_{S-T,\phi}\; \widehat{C}_{-T',-\gamma}\; \widehat{C}_{-S+T',\psi}\; \cdot$$

$$\chi_\beta(g)\; \chi_\phi(g(f \circ \pi^{v,w})\mu)\chi_{-\gamma}(h)\; \chi_\psi(h(f \circ \pi^{v,w'})\overline{\mu'})$$

Taking expectation over g, h, f, we see that terms are nonzero only if $\phi = -\beta$, $\psi = \gamma$, $\pi_p^{v,w}(\phi) + \pi_p^{v,w'}(\psi) = 0$, *i.e.* $\pi_p^{v,w}(\beta) = \pi_p^{v,w'}(\gamma)$. Also Lemma 3 gives $E_\mu\left[\chi_\phi(\mu)\right] = C(\phi)\; \omega^{2\cdot weight(\phi)} = C(\beta)\; \omega^{-2\cdot weight(\beta)}$ and $E_{\mu'}\left[\chi_\psi(\overline{\mu'})\right] = C(\gamma)\; \omega^{-2\; weight(\gamma)}$. Thus

$$\mathrm{Term(S)} = \sum_{\beta,\gamma \;:\; \pi_p^{v,w}(\beta) = \;\pi_p^{v,w'}(\gamma)} \widehat{F}_\beta \widehat{G}_\gamma C(\beta) C(\gamma)\; \cdot$$

$$\omega^{-2\; weight(\beta)}\; \omega^{-2\; weight(\gamma)} \tag{4}$$

where we define the "super-coefficients" $\widehat{F}_\beta$ and $\widehat{G}_\gamma$ as

$$\widehat{F}_\beta = \sum_{T} \widehat{B}_{T,\beta}\; \widehat{B}_{S-T,-\beta} \qquad \text{and}$$

$$\widehat{G}_\gamma = \sum_{T'} \widehat{C}_{-T',-\gamma}\; \widehat{C}_{-S+T',\gamma}$$

A crucial point here is that all the tables $B_i, 1 \le i \le k$ are equality folded and hence so are the tables $B_S\ \forall\ S \in GF(p)^k$. Thus $\widehat{F}_\beta \neq 0$ only if $weight(\beta_1) = weight(\beta_2) = \ldots = weight(\beta_t)$. Also, $weight(\beta) = \sum_{i=1}^t weight(\beta_i) = t \cdot weight(\beta_1)$. We distinguish the cases $weight(\beta) = 0$ and $weight(\beta) \neq 0$. In the latter case, $weight(\beta_i) \neq 0\ \forall\ i$ which implies that $\pi_p^{v,w,i}(\beta_i) \neq 0\ \forall\ i$.

LEMMA 7. *The expressions for $\widehat{F}_\beta$ and $\widehat{G}_\gamma$ are "roughly" complex conjugates of each other (they are conjugates if $B = C$ and $\beta = \gamma$).*

PROOF. Provided $B = C$ and $\beta = \gamma$,

$$\widehat{G}_\gamma = \sum_{T'} \widehat{C}_{-T',-\gamma} \widehat{C}_{-S+T',\gamma} \quad =$$

$$\sum_{T'} \overline{\widehat{C}}_{T',\gamma} \overline{\widehat{C}}_{S-T',-\gamma} = \overline{\sum_{T'} \widehat{C}_{T',\gamma} \widehat{C}_{S-T',-\gamma}} \quad = \overline{\widehat{F}_\beta}$$

□

Consider the terms in (4) for which $weight(\beta) = weight(\gamma) = 0$. These terms contribute

$$\sum_\alpha \sum_{\substack{\beta,\gamma\ :\ \alpha=\pi_p^{v,w}(\beta)=\pi_p^{v,w'}(\gamma), \\ weight(\beta)=0}} \widehat{F}_\beta \widehat{G}_\gamma C(\beta) C(\gamma)$$

Fixing the vertex $v \in V$ of the Label Cover instance and taking expectation over the choice of the vertices w, w', the previous expression can be written as

$$\sum_\alpha E_w \left[\sum_{\substack{\beta:\ \alpha=\pi_p^{v,w}(\beta), \\ weight(\beta)=0}} \widehat{F}_\beta C(\beta) \right] \cdot E_{w'} \left[\sum_{\substack{\gamma\ :\ \alpha=\pi_p^{v,w'}(\gamma), \\ weight(\gamma)=0}} \widehat{G}_\gamma C(\gamma) \right]$$

Since w, w' are identically distributed and using Lemma 7, it can be seen that the second expectation is just the complex conjugate of the first expectation. Thus the expression evaluates to the non-negative quantity

$$\sum_\alpha \left| E_w \left[\sum_{\beta\ :\ \alpha=\pi_p^{v,w}(\beta),\ weight(\beta)=0} \widehat{F}_\beta\ C(\beta) \right] \right|^2$$

The terms in (4) where $weight(\beta) = weight(\gamma) \neq 0$ can be used to extract labels. We would like to emphasize again that in this case $\pi_p^{v,w,i}(\beta_i) = \pi_p^{v,w,i}(\gamma_i) \neq 0$ for **every** coordinate $1 \le i \le t$.

4.2 Proof of Theorem 8

If $\Pr[\text{Acc}] \ge 1 - \frac{1}{p^{3k}} + \delta$, then from expression (3) and Lemma 6, either there exists $S \neq 0$ such that

$$\text{Real part}\ \ (\ \text{Term(S)}\ ) \le -p^{2k}\delta \qquad \text{or}$$

$$\text{Real part}\ \ (X) \le -p^{2k}\delta$$

Since the analyses for Term(S) and X are similar, lets assume the former case holds. From the discussion in the previous section it follows that

$$\text{Real part} \left[\sum_{\substack{\pi_p^{v,w}(\beta)=\pi_p^{v,w'}(\gamma), \\ weight(\beta)\neq 0}} \widehat{F}_\beta \widehat{G}_\gamma\ C(\beta) C(\gamma) \cdot \omega^{-2\ weight(\beta)}\ \omega^{-2\ weight(\gamma)} \right] \le -p^{2k}\delta$$

Taking absolute values and noting that

$$C(\beta) \le \prod_{i=1}^t (1 - \Omega(\frac{1}{p^2}))^{|\beta_i|} \le \prod_{i=1}^t O(p^2) \frac{1}{|\beta_i|}$$

and a similar inequality for γ, we see that

$$E_{v,w,w'} \left[\sum_{\substack{\beta,\gamma\ :\ \pi_p^{v,w}(\beta)=\pi_p^{v,w'}(\gamma), \\ weight(\beta)\neq 0}} \frac{|\widehat{F}_\beta|}{p^k} \frac{|\widehat{G}_\gamma|}{p^k} \prod_{i=1}^t \frac{1}{|\beta_i|} \frac{1}{|\gamma_i|} \right] \ge 2^{-O(t)} p^{-4t}\delta \quad (5)$$

We note that $\sum_\beta |\widehat{F}_\beta| \le p^k$ and $\sum_\gamma |\widehat{G}_\gamma| \le p^k$. Hence $|\widehat{F}_\beta|/p^k$ and $|\widehat{G}_\gamma|/p^k$ are valid probability distributions. Now we extract labels in a (by now) standard way.

For every vertex $w \in W$, we pick $\beta = (\beta_1, \ldots, \beta_t)$ with probability $|\widehat{F}_\beta|/p^k$ and define label of w to be $(b_1, \ldots, b_t)$ where each b_i is a randomly chosen element $x_i \in M_i$ for which $\beta_i(x_i) \neq 0$. For every vertex $v \in V$, we pick its random neighbor $w' \in W$, pick $\gamma = (\gamma_1, \ldots, \gamma_t)$ with probability $|\widehat{G}_\gamma|/p^k$, define c_i to be a randomly chosen element $y_i \in M_i$ for which $\gamma_i(y_i) \neq 0$ and finally define label of v to be $(\pi^{v,w',1}(c_1), \ldots, \pi^{v,w',t}(c_t))$.

We claim that the fraction of edges satisfied by this labeling equals the expression (5). Note that the summation runs only over β, γ such that for all i, $\beta_i \neq 0, \gamma_i \neq 0$, $\pi_p^{v,w,i}(\beta_i) = \pi_p^{v,w',i}(\gamma_i) \neq 0$. Hence there always exists $x_i \in M$ such that $\beta_i(x_i) \neq 0$ and $y_i \in M$ such that $\gamma_i(y_i) \neq 0$. Also, with probability $\prod_{i=1}^t \frac{1}{|\beta_i|} \frac{1}{|\gamma_i|}$, it holds that

$$\pi^{v,w,i}(b_i) = \pi^{v,w',i}(c_i)\ \ \forall\ i$$

In other words $\pi^{v,w}(b_1, \ldots, b_t) = \pi^{v,w'}(c_1, \ldots, c_t)$, i.e.

$$\pi^{v,w}(label(w)) = label(v)$$

which is precisely the condition for satisfying an edge (v, w). Hence the claim follows.

The lower bound of $2^{-O(t)} p^{-4t}\delta$ on expression (5) proves Theorem 8. Theorem 1 follows easily from Theorem 8 and it is proved in Appendix C.

5. CONCLUSION

We now have strong hardness results for approximate coloring of k-uniform hypergraphs for $k \ge 4$. However it seems difficult to extend these techniques for proving hardness results for graph coloring or 3-uniform hypergraph coloring. Theorem 5 points out a fundamental difference between graphs and hypergraphs. An analogous theorem for graphs would imply the existence of a PCP with zero free bits, completeness $\Omega(1)$ and arbitrarily low soundness, which is an open problem. But Theorem 5 does not imply existence of such PCPs. Thus hardness results for graph coloring may require new techniques.

Another open problem is whether it is hard to color a p-colorable 4-uniform hypergraph with $f(n)$ colors where $f(n)$ exceeds every polylog function and p remains bounded.

6. ACKNOWLEDGEMENT

I sincerely thank Venkatesan Guruswami, Johan Håstad and Madhu Sudan for providing me the journal version of their paper. I also thank Sanjeev Arora and Venkatesan Guruswami for their valuable comments on an earlier version of this paper.

7. REFERENCES

[1] N. Alon, P. Kelsen, S. Mahajan, and H. Ramesh. Coloring 2-colorable hypergraphs with a sublinear number of colors. *Nordic Journal of Computing*, 3:425–439, 1996.

[2] S. Arora, C. Lund, R. Motawani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[3] S. Arora and S. Safra. Probabilistic checking of proofs : A new characterization of np. *Journal of the ACM*, 45(1):70–122, 1998.

[4] M. Bellare, O. Goldreich, and M. Sudan. Free bits, pcps and non-approximability. *Electronic Colloquium on Computational Complexity, Technical Report TR95-024*, 1995.

[5] A. Blum and D. Karger. An $\tilde{O}(n^{3/14})$ coloring algorithm for 3-colorable graphs. *Information Processing Letters*, 61:49–53, 1997.

[6] V. Guruswami, J. Håstad, and M. Sudan. Hardness of approximate hypergraph coloring. In *Proc. of the 41st IEEE Symposium on Foundations of Computer Science*, pages 149–158, 2000. (The references are to the journal version of this paper which is in preparation).

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

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

[9] J. Håstad and V. Srinivasan. On the advantage over a random assignment. In *Proc. of the 34thth Annual ACM Symposium on Theory of Computing*, 2002.

[10] J. Holmerin. Vertex cover on 4-regular hyper-graphs is hard to approximate within $2-\epsilon$. In *Proc. of the 34th Annual ACM Symposium on Theory of Computing*, 2002.

[11] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. In *Proc. of the 35th IEEE Symposium on Foundations of Computer Science*, 1994.

[12] S. Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science*, 2001.

[13] M. Krivelevich and B. Sudakov. Approximate coloring of uniform hypergraphs. In *Proc. of European Symposium on Algorithms*, 1998.

[14] L. Lovász. Coverings and colorings of hypergraphs. In *Proc. of 4th Southeastern Conf. on Combinatorics, Graph Theory and Computing*, pages 3–12. Utilitas Mathematica Publishing, Winnipeg, 1973.

[15] C. McDiarmid. A random recoloring method for graphs and hypergraphs. *Combinatorics, Probability and Computing*, 2:363–365, 1993.

[16] C. McDiarmid. Hypergraph coloring and the lovász local lemma. *Discrete Mathematics*, 167/168:481–486, 1997.

[17] J. Radhakrishnan and A. Srinivasan. Improved bounds and algorithms for hypergraph coloring. In *Proc. of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 684–693, 1998.

[18] R. Raz. A parallel repetition theorem. *SIAM J. of Computing*, 27(3):763–803, 1998.

APPENDIX

A. PROOF OF LEMMA 3

First we prove another lemma.

LEMMA 8. *For a function $\beta_i : M_i \mapsto GF(p)$ and a function μ_i chosen as in Section 3*

$$E_{\mu_i}[\prod_{x\in M_i} \mu_i(x)^{\beta_i(x)}] = C(\beta_i)\ \omega^{2\cdot weight(\beta_i)}$$

where $C(\beta_i)$ is a positive real number satisfying $C(\beta_i) \leq (1-\Omega(\frac{1}{p^2}))^{|\beta_i|}$.

PROOF. We note that the given expectation is

$$\prod_{x\in M_i} E_{\mu_i}[\mu_i(x)^{\beta_i(x)}]$$

We show that for every x such that $\beta_i(x) \neq 0$, the inner expectation is $\omega^{2\cdot\beta_i(x)}$ times a positive number $\leq 1-\Omega(\frac{1}{p^2})$. Let $r = \beta_i(x)$. Because of the way $\mu_i()$ is defined, the inner expectation is

$$0.2\ \omega^r + 0.6\ \omega^{2r} + 0.2\ \omega^{3r} = \omega^{2r}(0.6 + 0.2\ \omega^r + 0.2\ \omega^{-r})$$

This complex number is ω^{2r} times a positive real number which is maximized when $r = 1$ and this maximum value is $0.6 + 0.4\cos(\frac{2\pi}{p}) = 1 - \Omega(\frac{1}{p^2})$. □

Now we prove Lemma 3. Note that

$$\begin{aligned} E_\mu[\chi_\beta(\mu)] &= E_\mu\left[\prod_{i=1}^{t}\prod_{x_i\in M_i}\mu_i(x_i)^{\beta_i(x_i)}\right] \\ &= \prod_{i=1}^{t} E_{\mu_i}\left[\prod_{x_i\in M_i}\mu_i(x_i)^{\beta_i(x_i)}\right] \end{aligned}$$

Lemma 3 now follows from Lemma 8, with

$$C(\beta) = C(\beta_1)C(\beta_2)\cdots C(\beta_t)$$

B. PROOF OF LEMMA 4

When $x = y = z = w$, the sum is

$$\frac{1}{p^3}\sum_{r_1+r_2+r_3+r_4=0} x^{r_1+r_2+r_3+r_4} = \frac{1}{p^3}\sum_{r_1+r_2+r_3+r_4=0} x^0 = 1$$

In the second case, assuming w.l.o.g. that $z \neq w$, the sum is $\frac{1}{p^3}$ times

$$\sum_{r_1,r_2} x^{r_1} y^{r_2} \sum_{r_3} z^{r_3} w^{-r_1-r_2-r_3} = \sum_{r_1,r_2} x^{r_1} y^{r_2} w^{-r_1-r_2} \sum_{r_3} (zw^{-1})^{r_3}$$

and the inner sum is 0 since $zw^{-1} \neq 1$.

C. PROOF OF THEOREM 1

We have $t = \lfloor \frac{p-1}{6} \rfloor$. Let $k = \alpha \log\log n$ for a sufficiently large constant α. Let $\delta = \frac{1}{p^{3k}}$ and let u be chosen such that

$$2^{-ut} = \frac{1}{p^{4k}} < 2^{-O(t)} p^{-4t} \delta, \qquad \text{i.e.} \qquad u = \frac{4k \log p}{t} \quad (6)$$

The number of random bits used by the verifier is

$$O(ut \log n + t\ 2^{e_0 u} \log p) \leq 2^{2e_0 u}$$

The hypergraph is constructed as described in Section 2.1. The size N of the hypergraph is at most exponential in the number of random bits used by the verifier. So

$$N \leq 2^{2^{2e_0 u}} = 2^{(\log n)^{O(1)}}, \quad \log N \leq 2^{2e_0 u}$$

Suppose on the contrary that the covering complexity of the verifier is k, i.e. there exist k proofs such that on every random string the verifier accepts at least one proof. Thus the hypothesis of Theorem 8 holds with $\delta = \frac{1}{p^{3k}}$ which implies that there exists a labeling of the Label Cover instance $\mathcal{L}$ with $OPT(\mathcal{L}) \geq 2^{-O(t)} p^{-4t} \delta$. However u is chosen large enough so that Theorem 7 implies $OPT(\mathcal{L}) < 2^{-O(t)} p^{-4t} \delta$. This contradiction proves that the covering complexity of the PCP is at least $k+1$. From Theorem 6 it is hard to color the hypergraph with p^k colors. We note that

$$p^k = 2^{ut/4} \geq (\log N)^{cp}$$

for some absolute constant c.