Hardness of Approximate Hypergraph Coloring

Venkatesan Guruswami* Johan Håstad† Madhu Sudan* ‡

Abstract

We introduce the notion of covering complexity of a probabilistic verifier. The covering complexity of a verifier on a given input is the minimum number of proofs needed to "satisfy" the verifier on every random string, i.e., on every random string, at least one of the given proofs must be accepted by the verifier. The covering complexity of PCP verifiers offers a promising route to getting stronger inapproximability results for some minimization problems, and in particular, (hyper)-graph coloring problems. We present a PCP verifier for NP statements that queries only four bits and yet has a covering complexity of one for true statements and a super-constant covering complexity for statements not in the language. Moreover, the acceptance predicate of this verifier is a simple Not-all-Equal check on the four bits it reads. This enables us to prove that for any *constant c, it is NP-hard to color a 2-colorable 4-uniform hypergraph using just c colors, and also yields a super-constant inapproximability result under a stronger hardness assumption.*

1 Introduction

In this paper we study a variant of the standard notion of a probabilistically checkable proof (PCP). In the standard notion, the probabilistic verifier is provided restricted oracle access to a proof, is allowed some probability of error, and the goal is to find a proof that *maximizes* the *acceptance probability* of the verifier (on any given input). For integer valued functions $r(\cdot)$ and $q(\cdot)$, the verifier is said to be (r, q)-restricted if it tosses at most $r(n)$ coins and queries the proof for at most $q(n)$ bits, on inputs that are n bits long. A language L belongs to the class $\mathrm{PCP}_{c,s}[r, q]$ if an (r, q)-restricted verifier accepts the language with completeness c and soundness s. I.e., for instances in the language there exist proofs that are accepted by the verifier with probability at least c, while for instances not in the language no proof is accepted with probability more than s.

In the variant we consider here, we allow multiple proofs, say $\Pi_1, \ldots, \Pi_k$, to be provided to the verifier. We require that for every random string used by the verifier, at least one of the proofs Π_i must be accepted by the verifier. The goal now is to find the smallest set of proofs that satisfy this property and the cardinality of this set is said to be the *covering complexity* of the verifier on this input. Analogous to the class PCP, we may define the class $\mathrm{cPCP}_{c,s}[r, q]$ to be the class of all languages for which there exist (r, q)-restricted verifiers that satisfy the following conditions: (Completeness) If $x \in L$, the covering complexity of V on x is at most $1/c$. (Soundness) If $x \notin L$ then the covering complexity of V on x is at least $1/s$.

The class cPCP arises naturally in the study of certain minimization problems, and in particular in the study of the approximability of graph coloring. Traditionally, however the class has not been focussed on explicitly. Instead all previous (PCP based) results on graph coloring [21, 18, 10] have implicitly relied on the obvious containment $\mathrm{PCP}_{1,s}[r, q] \subseteq \mathrm{cPCP}_{1,s}[r, q]$. Thus it sufficed to prove strong containments of NP in PCP to get hardness result for graph coloring.

This approach was quite successful in proving strong (and in fact essentially tight) inapproximability of graph coloring for general graphs [10], but for graphs whose chromatic number is a small constant, however, the known hardness results are much weaker. For example, for 3-colorable graphs the best known hardness result only rules out coloring using 4 colors [18, 14]. This paper is motivated by the quest for strong (super-constant) inapproximability for coloring graphs whose chromatic number is a small constant, and the kind of PCP constructions that this question motivates. A necessary (but not sufficient condition) for such a result is a containment of NP in $\mathrm{cPCP}_{c,o(1)}[O(\log n), q]$ for $c > 0$ and constant q. However such a result can not be obtained by passing through PCP, since it is known that if NP $\subseteq \mathrm{PCP}_{c,s}[O(\log n), q]$ then $s \geq c2^{-q}$ (and hence $s = \Omega(1)$ as well). Moreover, while the existence of "good" cPCP's is implied by a strong hardness result for coloring

[*] Laboratory for Computer Science, MIT, 545 Technology Square, Cambridge, MA 02139. Email: {venkat@theory.lcs.,madhu@}mit.edu

[†] Department of Numerical Analysis and Computer Science, Royal Institute of Technology, SE-100 44 Stockholm, Sweden. Email: johanh@nada.kth.se

[‡] Supported in part by an MIT-NEC Research Initiation Award, a Sloan Foundation Fellowship and NSF Career Award CCR-9875511.

(for example the hardness of c-coloring 3-colorable graphs for every constant c), such a result is not known to be true for PCP's (see [14] for related discussions). In light of these facts, in order to get the stronger inapproximability results for coloring, it may be better to study cPCP directly, and we do so in this paper.

Our Results. Our main result is a containment of NP in the class $\mathrm{cPCP}_{1,\epsilon}[O(\log n), 4]$, for every $\epsilon > 0$. If the randomness is allowed to be slightly super-logarithmic, then the soundness can be reduced to some explicit $o(1)$ function. Technically, this result is of interest in that it overcomes the qualitative limitation described above of passing through standard PCPs. Furthermore, our proof shows how to apply the (by now) standard Fourier-analysis based techniques to the studying of covering complexity as well. Thus it lays out the hope for applying such analysis to other cPCP's as well.

Unfortunately, the resulting cPCP fails to improve inapproximability of graph coloring. In part, this is due to the rather fragile nature of covering complexity, which makes the utility of cPCP's to be closely tied to the actual predicates used by the verifier in deciding its actions. In standard PCPs one can use gadgets to transform the predicates used by the verifier, thus allowing one to transform hardness results among different problems. In covering PCPs such transformations typically completely destroy the properties of the PCP. For example, to design a covering PCP appropriate for use in hardness results for 3-colorable graphs, the verifier must be restricted to working with proofs that are strings from $\{0, 1, 2\}^*$ and the verifiers actions are only allowed to read two elements of the proof and verify they are unequal.

Keeping this finicky nature of covering PCPs in mind, we design a different verifier (whose query complexity is also 4 bits), but whose acceptance predicate just checks if not all of the 4 bits read are equal, and thus corresponds *directly* to coloring of 4-uniform hypergraphs. Recall that a 4-uniform hypergraph H is given by a set of vertices V and a collection E of 4-element subsets of V called hyperedges. (In a general hypergraph there is no restriction on the number of vertices in any hyperedge.) A k-coloring is a map from V to the set $\{1, 2, \dots, k\}$ such that in every edge at least two vertices are assigned distinct colors, i.e., no edge is *monochromatic*. The goal here is to find the chromatic number of H, which is the smallest k such that a k-coloring of the given hypergraph exists.

Hypergraph coloring has been studied in the literature from both the combinatorial and algorithmic angle. In contrast with graphs, deciding if a given hypergraph is 2-colorable is NP-hard, even for 3-uniform hypergraphs [20]. The property of hypergraph 2-colorability, also called *Property B*, has been studied in the extremal combinatorics literature for long and much work has been done on proving hypergraph families 2-colorable and the corresponding algorithmic questions [9, 5, 6, 22, 23, 26, 24]. It has also been studied by computer scientists due to its connections to the graph coloring and satisfiability problems. Inspired in part by the work of [17] on approximate graph coloring, several authors [1, 8, 19] have provided approximation algorithms for coloring 2-colorable hypergraphs. The best known result for 2-colorable 4-uniform hypergraphs is a polynomial time coloring algorithm that uses $\tilde{O}(n^{3/4})$ colors [1, 8] where n is the number of vertices. No non-trivial hardness results seem to be known, and in fact it was not known prior to our work if 3-coloring a 2-colorable 4-uniform hypergraph is NP-hard. Our result yields a super-constant lower bound on coloring 2-colorable 4-uniform hypergraphs: we prove that c-coloring such hypergraphs is NP-hard for any constant c (Theorem 4.4), and moreover there exists a constant $c_0 > 0$ such that, unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{O(\log\log n)})$, there is no polynomial time algorithm to color a 2-colorable 4-uniform hypergraph using $c_0 \log\log\log n$ colors (Theorem 4.5). A similar hardness result also holds for coloring 2-colorable k-uniform hypergraphs for any $k \geq 5$ by reduction from the case of 4-uniform hypergraphs (the details of this reduction are omitted here and will appear in the full version).

There is also a natural maximization version of hypergraph 2-coloring: color the vertices with two colors so that a maximum number of hyperedges are non-monochromatic. For k-uniform hypergraphs, this is clearly the same problem as Max k-Set Splitting. For $k = 4$ (the case we study here), a tight hardness result of $7/8 + \varepsilon$ is known [16] — thus the problem is "approximation resistant" and a random 2-coloring is the best one can do. Obtaining a hardness for the minimization version as always turns out to be more difficult. For $k = 3$, a tight hardness result is not known even for the maximization version (see [13]). In fact, algorithms that do (much) better than a random 2-coloring are known for this case [11], and thus the problem is not "approximation resistant". We believe this indicates that getting a strong inapproximability for coloring 3-uniform hypergraphs similar to our result here is likely to be even harder, and the same applies for coloring 3-colorable graphs as well.

2 Preliminaries

We first repeat the formal definition of a covering PCP.

Definition 1 (Covering PCP) *A language L belongs to the class $\mathrm{cPCP}_{c,s}[r, q]$ if there is an (r, q)-restricted verifier V such that on input x: (i) if $x \in L$ then there is a set of at most $1/c$ proofs such that V accepts at least one of them for* any *random choice it makes, and (ii) if $x \notin L$, for any set of k proofs $\Pi_1, \Pi_2, \dots, \Pi_k$ with $k < 1/s$, there is random string for which V rejects* every *Π_i, $1 \leq i \leq k$.*

2.1 Covering PCPs and Graph Coloring

We now verify our intuition that "good" covering PCPs (i.e., those which have a large gap in covering complexity between the completeness and soundness cases) are necessary for strong lower bounds on the approximating the chromatic number. As usual, for a graph G, we denote by $\chi(G)$ its chromatic number, i.e., the minimum number of colors required in a proper coloring of G.

Proposition 2.1 *Suppose for functions $f, g : \mathbb{Z}^+ \to \mathbb{Z}^+$, given a graph G on n vertices, it is NP-hard to distinguish between the cases $\chi(G) \leq f(n)$ and $\chi(G) \geq g(n)$. Then* $\mathrm{NP} \subseteq \mathrm{cPCP}_{\lceil \log f(n) \rceil^{-1}, \lceil \log g(n) \rceil^{-1}} \big[O(\log n), 2\big]$.

Proof: Let the vertex set of G be $V = \{v_1, v_2, \ldots, v_n\}$. The covering PCP will consist as proofs $\Pi_1, \Pi_2, \ldots, \Pi_k$ which correspond to "cuts" $\Gamma_1, \ldots, \Gamma_k$ of G, i.e., each Π_i will be n-bits long, with the j^{th} bit being 1 or 0 depending on which side of the cut Γ_i contains v_j. The verifier will simply pick two vertices v_{j_1} and v_{j_2} at random such that they are adjacent in G, and then check if the j_1^{th} and j_2^{th} bits differ in *any* of the k proofs. The minimum number k of proofs required to satisfy the verifier for all its random choices is clearly the *cut cover number* $\kappa(G)$ of G, i.e., the minimum number of cuts that cover all edges of G. It is easy to see that $\kappa(G) = \lceil \log \chi(G) \rceil$, and therefore the claimed result follows. □

The logarithms in the covering completeness and soundness above (as are all logs in this paper) are to the base 2. One can get a similar result for any base q, by letting the proofs be q-ary strings and the verifier read two q-ary symbols from the proof. In light of this, we get the following.

Corollary 2.2 *If there exists an $\varepsilon > 0$ such that it is NP-hard to n^{ε}-color a 3-colorable graph, then* $\mathrm{NP} \subseteq \mathrm{cPCP}_{1,(\varepsilon \log_3 n)^{-1}} \big[O(\log n), 2\big]$ *where the covering PCP is over a ternary alphabet, and the verifier reads two ternary symbols from the proof.*

In light of the above Corollary, very powerful covering PCP characterizations of NP are necessary in order to get strong hardness results for coloring graphs with small chromatic number. A result similar to Proposition 2.1, with an identical proof, also holds for hypergraph coloring, and thus motivates us to look for good covering PCP characterizations of NP in order to prove hardness results for coloring 2-colorable hypergraphs.

Proposition 2.3 *If there exists a function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that $f(n)$-coloring a 2-colorable r-uniform hypergraph is NP-hard, then* $\mathrm{NP} \subseteq \mathrm{cPCP}_{1, \frac{1}{\log f(n)}} \big[O(\log n), r\big]$. *In particular, if c-coloring 2-colorable r-uniform hypergraphs is NP-hard for every constant c, then* $\mathrm{NP} \subseteq \mathrm{cPCP}_{1, \frac{1}{k}} \big[O(\log n), r\big]$ *for every constant $k \geq 1$.*

2.2 Preliminaries on Long Code

We now describe a very redundant error-correcting code, called the *long code*. The long code was first used by [7], and has been very useful in all PCP constructions since. We first develop some notation.

We represent boolean values by the set $\{1, -1\}$ with 1 standing for FALSE and -1 for TRUE. This representation has the nice feature that XOR just becomes multiplication. For any domain D, denote by $\mathcal{F}_D$ stands for the space of all boolean functions $f : D \to \{1, -1\}$. The long code of an element x in a domain D, denoted $\mathsf{LONG}(x)$, is simply the evaluations of all the $2^{|D|}$ boolean functions in $\mathcal{F}_D$ at x. If A is the long code of a, then we denote by $A(f)$ the coordinate of A corresponding to function f, so that $A(f) = f(a)$.

Folding of Long Codes: A Discussion. A function $A : \mathcal{F}_D \to \{1, -1\}$ is said to be *folded* if $A(f) = -A(-f)$ for all $f \in \mathcal{F}_D$ [7]. A codeword of the long code is clearly folded (since $A(f) = f(a) = -(-f(a)) = -A(-f)$). One can assume that the proofs which are purportedly long codes are folded since for any $A : \mathcal{F}_D \to \{1, -1\}$, one can define a new function A' by: $A'(f) = A(f)$ if $f(\alpha_0) = 1$ and $A'(f) = -A(-f)$ if $f(\alpha_0) = -1$, where α_0 is some fixed element of D, and now A' is clearly folded.

Thus for several applications one can assume access to folded proofs, and this turns out to be essential for several PCP constructions. Tight results for certain applications call for working without the folding assumption though, a good example is set splitting [16]. Folding illustrates one of many natural things that could go wrong in the analysis of covering soundness, since even though for our (first) cPCP construction (Theorem 3.5 of Section 3) we can assume the proof tables are folded, our analysis has to deal with tables that are not folded. The discussion following Lemma 3.2 of Section 3 further brings out this point.

3 PCP Construction I

In this section, for any constant k, we describe a (covering) PCP construction that uses logarithmic randomness, makes 4 queries (and reads 4 bits from these locations), has perfect completeness and covering soundness at most $1/k$. By allowing slightly super-logarithmic randomness, we can even achieve an $o(1)$ covering soundness, for some explicit $o(1)$ function.

The PCP construction is based on the one in [16] for proving a tight hardness result for 4-Set Splitting. Our analysis, however, is different, and proves that no k proofs can together satisfy all the predicates tested by the PCP verifier. (In contrast the analysis in [16] would prove that this PCP has perfect completeness and soundness $3/4 + \varepsilon$, for $\varepsilon > 0$ as small as desired. The perfect completeness implies perfect covering completeness, but the soundness analysis has to be different in our case.) We provide below a high-level description of the PCP construction; this is not meant to be complete, but should give some sense of the ideas used in the construction.

3.1 Preliminaries on Proof Composition

Our PCP constructions (also) follow the paradigm of *proof composition*, by composing an "outer verifier" with an "inner verifier". In its most modern and easy to apply form, one starts with an *outer proof system* which is a *2-Prover 1-Round proof system* (2P1R) construction for NP.

Label Cover. We abstract the 2P1R by a graph-theoretic optimization problem called LABEL COVER. The specific version of LABEL COVER we refer to is the maximization version $\mathrm{LabelCover}_{\max}$ discussed in [2] (see [2] for related versions and the history of this problem). A $\mathrm{LabelCover}_{\max}$ instance $\mathcal{LC}$ consists of a bipartite graph $H = (V, W, F)$ with vertex set $V \cup W$ and edge set F, "label sets" L_V, L_W which represent the possible *labels* that can be given to vertices in V, W respectively, and *projection functions* $\pi_{v,w} : L_W \to L_V$ for each $v \in V$ and $w \in W$ such that $(v, w) \in F$. The optimization problem we consider is to assign a label $\ell(v) \in L_v$ (resp. $\ell(w) \in L_W$) to each $v \in V$ (resp. $w \in W$) such that the fraction of edges $e = (v', w')$ with $\ell(v') = \pi_{v',w'}(\ell(w'))$ (call such an edge "satisfied") is maximized. The optimum value of a $\mathrm{LabelCover}_{\max}$ instance $\mathcal{LC}$, denoted $\mathsf{OPT}(\mathcal{LC})$, is the maximum fraction of "satisfied" edges in any label assignment. In the language of $\mathrm{LabelCover}_{\max}$, the PCP theorem [4, 3] together with the parallel repetition theorem of Raz [25] yields the following (the proof is standard and we omit it; see the Remark following the statement of the Theorem though).

Theorem 3.1 *There exist absolute constants $d_0, e_0 > 0$ such that for any δ, $0 < \delta < 1$, there is a polynomial time transformation mapping instances φ of* SAT *to instances $\mathcal{LC} = (V, W, F, L_V, L_W, \{\pi_{v,w} | (v, w) \in F\})$ of* $\mathrm{LabelCover}_{\max}$ *such that*

(i) $|V|, |W| \le n^{d_0 \log \delta^{-1}}$ where n is the size of the SAT *instance φ.*

(ii) $|L_V|, |L_W| \le \delta^{-e_0}$.

(iii) If φ is satisfiable then $\mathsf{OPT}(\mathcal{LC}) = 1$, while if φ is not satisfiable then $\mathsf{OPT}(\mathcal{LC}) \le \delta$.

(iv) The projection functions are "smooth", i.e., map large subsets of their domain to large subsets of their range. More specifically, there is an absolute constant c, $0 < c < 1$, such that for each $w \in W$ and every $\beta \subseteq L_W$,

$$\Pr_{v \in_R N(w)} [\, |\pi_{v,w}(\beta)| \ge |\beta|^c \,] \ge 1 - |\beta|^{-c} \qquad (1)$$

where $N(w) = \{v \in V | (v, w) \in F\}$.

Remark: Conditions (i) to (iii) are standard for $\mathrm{LabelCover}_{\max}$. We require Condition (iv) for some technical aspects which arise in the proof, and it follows from a combinatorial Lemma in [16] that this condition can also be met for the projection functions in the above Theorem.

Constructing a "Composed" PCP. Note that the above Theorem implies a PCP where the proof is simply the labels of all vertices in V, W of the $\mathrm{LabelCover}_{\max}$ instance and the verifier picks an edge $e = (v, w) \in F$ at random and checks if the labels of v and w are "consistent", i.e., $\pi_{v,w}(\ell(w)) = \ell(v)$. By the properties guaranteed in the Theorem, this PCP uses $O(\log n \log \delta^{-1})$ randomness, has perfect completeness and soundness at most δ. While the soundness is excellent, the number of bits it reads from the proof in total (from the two "locations" it queries) is huge ($O(\log \delta^{-1})$). In order to improve the query complexity, one "composes" this "outer" verification with an "inner" verification procedure. The inner verifier is given as input a projection function $\pi : L_W \to L_V$, and has oracle access to purported encodings, via the encoding function Enc of some error-correcting code, of two labels $a \in L_V$ and $b \in L_W$, and its aim is to check that $\pi(b) = a$ (with "good" accuracy) by making very few queries to $\mathsf{Enc}(a)$ and $\mathsf{Enc}(b)$. The inner verifiers we use have a slightly different character: they are given input two projections π_1 and π_2 and have oracle access to purported encodings $\mathsf{Enc}(b)$ and $\mathsf{Enc}(c)$ of two labels $b, c \in L_W$, and the aim is to test whether $\pi_1(b) = \pi_2(c)$. This interesting feature was part of and necessary for Håstad's construction for set splitting [16], and our PCPs also inherit this feature.

In our final PCP system, the proof is expected to be the encodings of the labels $\ell(w)$ of all vertices $w \in W$ using the encoding Enc. For efficient constructions the code used is the *long code* of [7], i.e., $\mathsf{Enc} \overset{\mathrm{def}}{=} \mathsf{LONG}$. We denote the portion of the (overall) proof that corresponds to w by $\mathsf{LP}(w)$, and in a "correct" proof $\mathsf{LP}(w)$ would just be $\mathsf{LONG}(\ell(w))$ (the notation LP stands for "long proof").

The construction of a PCP now reduces to the construction of a good *inner verifier* that given a pair of strings B, C which are purportedly long codes, and projection functions π_1 and π_2, checks if these strings are the long codes of two "consistent" strings b and c whose respective projections agree (i.e., $\pi_1(b) = \pi_2(c)$). Given such an inner verifier IV, one can get a "composed verifier" V_{comp} using standard techniques as follows (given formula φ the verifier first computes the $\mathrm{LabelCover}_{\max}$ instance $\mathcal{LC}$ in polynomial time and then proceeds with the verification):

1. Pick $v \in V$ at random and $w, w' \in N(v)$ at random
2. Run the inner verifier with input $\pi_{v,w}$ and $\pi_{v,w'}$ and oracle access to $\mathsf{LP}(w)$ and $\mathsf{LP}(w')$.
3. Accept iff the inner verifier IV accepts

We denote by $V_{\text{comp}}(\mathsf{IV})$ the composed verifier obtained using inner verifier IV. The (usual) soundness analysis of the composed PCP proceeds by saying that if there is a proof that causes the verifier V_{comp} to accept with large, say $(s+\varepsilon)$, probability, where s is the soundness we are aiming for, then this proof can be "decoded" into labels for $V \cup W$ that "satisfy" more than a fraction δ of the edges in the $\text{LabelCover}_{\max}$ instance, and by Theorem 3.1 therefore the the original formula φ was satisfiable. In our case, we would like to make a similar argument and say that if at most k proofs together satisfy all tests of V_{comp}, then these proofs can be "decoded" into labels for $V \cup W$ that satisfy more than δ fraction of edges of $\mathcal{LC}$.

3.2 The Inner Verifier

We now delve into the specification of our first "inner verifier", which we call Basic-IV4. This inner verifier is essentially the same as the one for 4-set splitting in [16], but has a different acceptance predicate. Recall the inner verifier is given input two projections functions $\pi_1, \pi_2 : L_W \to L_V$ and has oracle access to two tables $B, C : \mathcal{F}_{L_W} \to \{1, -1\}$, and aims to check that B (resp. C) is the long code of b (resp. c) which satisfy $\pi_1(b) = \pi_2(c)$.

```
Inner Verifier Basic-IV4$_p^{B,C}$ $(\pi_1, \pi_2)$
  Choose uniformly at random $f \in \mathcal{F}_{L_V}, g_1, h_1 \in \mathcal{F}_{L_W}$
  Choose at random $g', h' \in \mathcal{F}_{L_W}$ such that $\forall b \in L_W$,
    $\mathbf{Pr}[g'(b) = 1] = p$ and $\mathbf{Pr}[h'(b) = 1] = p$
  Set $g_2 = -g_1(f \circ \pi_1 \wedge g')$;  $h_2 = -h_1(-f \circ \pi_2 \wedge h')$.
  Accept iff $(B(g_1) \neq B(g_2)) \vee (C(h_1) \neq C(h_2))$
```

For a technical reason, as in [16, 15, 12], the final inner verifier needs to run the above inner verifier for the bias parameter p chosen at random from an appropriate set of values. The specific distribution we use is the one used by Håstad [16] (the constant c used in its specification is the constant from Equation (1)).

```
Inner Verifier IV4$_\gamma^{B,C}$ $(\pi_1, \pi_2)$
  Set $t = \lceil 1/\gamma \rceil$, $\varepsilon_1 = \gamma^2$ and $\varepsilon_i = \varepsilon_{i-1}^{4/c}$ for $1 < i \le t$.
  Choose $p \in \{\varepsilon_1, \ldots, \varepsilon_t\}$ uniformly at random.
  Run Basic-IV4$_p^{B,C}(\pi_1, \pi_2)$.
```

Note that the inner verifier above has perfect completeness. Indeed when B, C are long codes of b, c where $\pi_1(b) = \pi_2(c) = a$ (say), then for each $f \in \mathcal{F}_{L_V}$, if $f(a) = 1$ then $B(g_1) = g_1(b)$ while $B(g_2) = B(-g_1(f \circ \pi_1 \wedge g')) = -g_1(b)$ and so these are not equal, and similarly for the case when $f(a) = -1$.

3.3 Covering Soundness analysis

Let $X(\gamma)$ be the indicator random variable for the rejection of a particular proof $\Pi = \{\mathsf{LP}(w) : w \in W\}$ by the composed verifier $V_{\text{comp}}(\text{IV4}_\gamma)$ (henceforth $V_1(\gamma)$). The probability that $V_1(\gamma)$ rejects Π taken over its random choices is clearly the expectation

$$\mathbf{E}[X(\gamma)] = \mathbf{E}\left[\left(\frac{1 + B(g_1)B(g_2)}{2}\right)\left(\frac{1 + C(h_1)C(h_2)}{2}\right)\right]. \tag{2}$$

taken over the random choices of $v, w, w', p, f, g_1, h_1, g_2$ and h_2. (Here B, C are tacitly understood to stand for $\mathsf{LP}(w)$ and $\mathsf{LP}(w')$ respectively and will equal $\mathsf{LONG}(\ell(w))$ and $\mathsf{LONG}(\ell(w'))$ respectively in a "correct" proof.) We wish to say that no k proofs can together satisfy all the tests which $V_1(\gamma)$ performs. Now, if $X_k(\gamma)$ is the indicator random variable for the rejection of a set of k proofs $\{\mathsf{LP}_i(w) : w \in W\}$, $1 \le i \le k$, by the verifier $V_1(\gamma)$, then the overall probability that $V_1(\gamma)$ rejects all these k proofs, taken over its random choices, is exactly

$$\begin{aligned}\mathbf{E}[X_k(\gamma)] \;&=\; \frac{1}{4^k}\Big(\mathbf{E}\Big[\prod_{i=1}^{k}\big(1 + B_i(g_1)B_i(g_2)\big) \\ &\qquad\qquad \big(1 + C_i(h_1)C_i(h_2)\big)\Big]\Big)\end{aligned} \tag{3}$$

where the expectation is once again taken over $v, w, w', p, f, g_1, h_1, g_2$ and h_2. We will now argue (see Lemma 3.2 below) that if this rejection probability is much smaller than 4^{-k}, then there is a way to obtain labels $\ell(u)$ for $u \in V \cup W$ by "decoding" Π_1 such that more than δ fraction of the edges (v, w) are satisfied by this labeling, i.e., $\ell(v) = \pi_{v,w}(\ell(w))$. Together with Theorem 3.1, this implies that the rejection probability (from Equation (3)) for any set of k proofs for a false claim of satisfiability (of φ), can be made arbitrarily close to $\frac{1}{4^k}$, and in particular is non-zero, and thus the covering soundness of the composed verifier is less than $1/k$.

Lemma 3.2 *There is an absolute constant $a' > 0$ such that for every integer $k \ge 1$, every ε, $0 < \varepsilon < 4^{-k}$, and all $\gamma \le \varepsilon/8$, if $\mathbf{E}[X_k(\gamma)] < \frac{1}{4^k} - \varepsilon$, then $\mathsf{OPT}(\mathcal{LC}) > 2^{-2^{a'\gamma^{-1}}}$.*

Before presenting the formal proof of Lemma 3.2, we first highlight the basic approach. The power of arithmetizing the rejection probability for a set of k proofs as in Equation (3) is that one can expand out the product and analyze the overall expectation as a sum of expectations of terms of the form $B_S(g_1)B_S(g_2)$, $C_T(h_1)C_T(h_2)$ or $B_S(g_1)B_S(g_2)C_T(h_1)C_T(h_1)$, for $S, T \subseteq \{1, 2, \ldots, k\}$ where $B_S = \prod_{i \in S} B_i$ and $C_T = \prod_{i \in T} C_i$, and analyze the terms individually. We can now imagine two new proofs $\tilde{B} = B_S$ and $\tilde{C} = C_T$ which are exclusive-ors of subsets of the k given proofs. (Note that even if our original B_i's are assumed to be folded, this is no longer true for the tables $\tilde{B}$ and $\tilde{C}$, and thus we need to perform our analysis with tables that are *not* folded. This is why we started with IV4 which can be analyzed without folding [16].) Now one can apply existing techniques from [16] to analyze terms involving the tables $\tilde{B}$ and $\tilde{C}$ and show that $\tilde{B}(g_1)\tilde{B}(g_2)$ and $\tilde{C}(h_1)\tilde{C}(h_2)$ cannot be too negative, and similarly if the expectation of $\tilde{B}(g_1)\tilde{B}(g_2)\tilde{C}(h_1)\tilde{C}(h_2)$ is too much below zero, then in fact $\mathsf{OPT}(\mathcal{LC})$ is quite large. (In short, at a high level, we are saying that if there exist k proofs such that the verifier accepts at least one of them with good probability, then some exclusive-or of these proofs is also accepted by the verifier with good probability, and we know this cannot happen by the soundness analysis of [16] for the case of a single proof.) This is formalized in the following two Lemmas which are Lemmas 7.9 and 7.12 from [16] (we have changed the statements slightly from those in [16]).

Lemma 3.3 ([16]) *For every $\gamma > 0$ and for all $B : \mathcal{F}_{L_W} \to \{1, -1\}$, and all $w \in W$*

$$\mathop{\mathbf{E}}_{p,v \in N(w), f, g, g'} \left[B(g_1)B(g_2)\right] \geq -4\gamma ,$$

where the distribution of p, f, g_1, g_2 is the same as the one in IV4$_\gamma$.

Lemma 3.4 ([16]) *For every $\gamma > 0$ and all proof tables $\{B_w\}$ and $\{C_w\}$ (indexed by $w \in W$) where each $B_w, C_w : \mathcal{F}_{L_W} \to \{1, -1\}$, we have $\mathbf{E}\left[B_w(g_1)B_w(g_2)C_{w'}(h_1)C_{w'}(h_2)\right]$ is at least*

$$-7\gamma - \mathsf{OPT}(\mathcal{LC})2^{2^{O(\gamma^{-1})}} ,$$

where the expectation is taken over $p, v, w, w', f, g_1, g_2, h_1, h_2$, and where the distribution of p, f, g_1, g_2, h_1, h_2 is the same as the one in IV4$_\gamma$.

Proof of Lemma 3.2: The proof is actually simple given Lemmas 3.3 and 3.4. We pick a $\gamma > 0$ that satisfies $\gamma < \frac{\varepsilon}{8}$. By Equation (3), if $\mathbf{E}[X_k(\gamma)] < 4^{-k} - \varepsilon$, then there exist subsets S_1, S_2 of $\{1, 2, \ldots, k\}$, $S_1 \cup S_2 \neq \emptyset$, such that

$$\mathbf{E}\left[B_{S_1}(g_1)B_{S_1}(g_2)C_{S_2}(h_1)C_{S_2}(h_2)\right] < -\varepsilon \qquad (4)$$

where B_{S_1} (resp. C_{S_2}) denotes $\Pi_{j \in S_1} B_j$ (resp. $\Pi_{j \in S_2} C_j$).

Suppose one of S_1, S_2 is empty, say $S_2 = \emptyset$. Lemma 3.3 applied to B_{S_1} (which is a function mapping $\mathcal{F}_{L_W} \to \{1, -1\}$), gives $\mathbf{E}[B_{S_1}(g_1)B_{S_1}(g_2)] \geq -4\gamma$ which together with Equation (4) above yields $\gamma > \frac{\varepsilon}{4}$, a contradiction since $\gamma \leq \varepsilon/8$.

Now suppose both S_1 and S_2 are non-empty. Now we apply Lemma 3.4 to B_{S_1} and C_{S_2} to get that the expectation in Equation (4) is at least $-7\gamma - \mathsf{OPT}(\mathcal{LC})2^{2^{O(\gamma^{-1})}}$. Together with Equation (4) this yields (using $\varepsilon \geq 8\gamma$)

$$\mathsf{OPT}(\mathcal{LC}) > \gamma 2^{-2^{O(\gamma^{-1})}} > 2^{-2^{a'\gamma^{-1}}}$$

for some *absolute* constant $a' > 0$. □

We are now ready to state and prove the main Theorem of this section.

Theorem 3.5 *For every constant k,* $\mathrm{NP} \subseteq \mathrm{cPCP}_{1,\frac{1}{k}}[\log, 4]$.

Proof: The proof will follow easily from Lemma 3.2 and Theorem 3.1. Let $\varepsilon = \frac{1}{2} \cdot 4^{-k}$ and $\gamma = \varepsilon/8$, and pick $\delta > 0$ small enough so that $2^{-2^{a'\gamma^{-1}}} > \delta$. By Lemma 3.2 we have $\mathbf{E}[X_k(\gamma)] < \frac{1}{4^k} - \varepsilon = \frac{1}{2 \cdot 4^k}$ implies $\mathsf{OPT}(\mathcal{LC}) > \delta$. Consider the PCP with verifier $V_{\mathrm{comp}}(\mathrm{IV4}_\gamma)$. Using Theorem 3.1, we get that if the input formula φ is not satisfiable, the verifier $V_{\mathrm{comp}}(\mathrm{IV4}_\gamma)$ rejects any k proofs with probability at least $\frac{1}{2 \cdot 4^k}$. Since it clearly has perfect completeness and makes only 4 queries, the claimed result follows. □

Remark on tightness of the analysis: In fact, Lemma 3.2 can be used to show that for any $\varepsilon > 0$, there exists a (covering) PCP verifier that makes 4 queries, has perfect completeness and which rejects any set of k proofs with probability at least $\frac{1}{4^k} - \varepsilon$. Note that this analysis is in fact *tight* for the verifier $V_{\mathrm{comp}}(\mathrm{IV4})$ since a random set of k proofs is accepted with probability $1 - 4^{-k}$.

4 PCP Construction II and Hardness of Hypergraph Coloring

In the previous section we gave a PCP construction which made only 4 queries into the proof and had covering soundness smaller than any desired constant. This is already interesting in that it highlights the power of taking the covering soundness approach (since as remarked in the Introduction one cannot achieve arbitrarily low soundness using classical PCPs with perfect completeness that make some fixed constant number of queries). We next turn to applying this to get a strong inapproximability result for hypergraph coloring.

The predicate tested by the inner verifier IV4$_\gamma$ is $F(x, y, z, w) = (x \neq y) \vee (z \neq w)$, and to get a hardness result for hypergraph coloring, we require the predicate to be $\mathrm{NAE}(x, y, z, w)$ which is true unless all of x, y, z, w are equal. Note that $\mathrm{NAE}(x, y, z, w)$ is true whenever $F(x, y, z, w)$ is true, so one natural approach is to simply replace the predicate F tested by IV4$_\gamma$ by NAE without losing perfect completeness. The challenge of course is to prove that the covering soundness does not suffer in this process, and this is exactly what we accomplish, though the proof gets much more complicated. Let us call the new inner verifier, obtained by changing the predicate tested by IV4$_\gamma$, as IV-NAE4$_\gamma$ (we hide the dependence on γ when no confusion can arise).

4.1 Soundness Analysis: Intuition

Note that for a particular random choice of functions (f, g_1, g_2, h_1, h_2) the inner verifier rejects all k proofs $\{LP_i(w) : w \in W\}$ exactly when $B_i(g_1) = B_i(g_2) = C_i(h_1) = C_i(h_2)$ for every i, $1 \le i \le k$. As in Lemma 3.2, we wish to argue that if the probability of this (rejection) happening is small then there is an assignment of labels to the vertices in $\mathcal{LC}$ that satisfy a "good" fraction of its edges.

It is possible to arithmetize the probability that the verifier $V_{\rm comp}$(IV-NAE4) rejects all k proofs (over its random coin tosses) similar to expression (3) in the analysis of the previous section. In the case of (3) we were able to "bound" all the terms that arose from expanding out the product. The arithmetization of the NAE predicate is a little more complicated, and a *tight* analysis in the spirit of the previous section seems difficult and there are terms in the expansion of the arithmetization which we are unable to bound or argue about directly.

Instead we take a "two-step" approach. We know from the analysis of the previous section that the probability that $B_i(g_1) = B_i(g_2)$ holds for all i, $1 \le i \le k$, simultaneously, is (roughly) 2^{-k}, and similarly for $C_i(h_1)$ and $C_i(h_2)$. We now wish to say that we will in addition also have $B_i(g_1) = C_i(h_1)$ for every i with reasonably large probability, so that the verifier with NAE predicate will also reject all k proofs with good probability. To prove this, note that B and C are really only different names for the same "tables" and the distinction is only that (g_1, g_2) is chosen differently from (h_1, h_2) (once v, f are picked). For a fixed v, f, denote by $\Delta_{v,f}$ the distribution of the $2k$ bits $\{B_i(g_1), B_i(g_2)\}_{i=1}^k \in \{1,-1\}^{2k}$ given that the verifier IV-NAE4$_\gamma$ picked v, f. (The distribution $\Delta_{v,f}$ is governed by the random choices of $w \in W$, the "bias parameter" p, and $g_1, g_2 \in \mathcal{F}_{L_W}$ as in verifier IV4$_\gamma$. The distribution thus depends on the parameter γ though we hide this for notational convenience.) It is also easy to check that once v, f is picked, the distribution of the bits $\{C_i(h_1), C_i(h_2)\}_{i=1}^k \in \{1,-1\}^{2k}$ that the verifier reads is exactly $\Delta_{v,-f}$. Hence, if the distributions $\Delta_{v,f}$ and $\Delta_{v,-f}$ are *nearly the same*, then $B_i(g_1) = B_i(g_2) = C_i(h_1) = C_i(h_2)$ holds for all i with good probability (this is shown in Lemma 4.6), and therefore the verifier rejects with good probability as well. We will also show that if there is a significant difference between the distributions $\Delta_{v,f}$ and $\Delta_{v,-f}$, then there is a way to "decode" this difference between the distributions into labels for the vertices of the LabelCover$_{\max}$ instance $\mathcal{LC}$ that satisfy a good fraction of edges (this is Lemma 4.7). In either situation we get the desired result.

4.2 The actual soundness analysis

We now proceed to the formal analysis. We need a few definitions. For each fixed (v, f) (here $v \in V$ and $f \in \mathcal{F}_{L_V}$ as usual), we will use the distribution $\Delta_{v,f}$ on $\{1,-1\}^{2k}$ defined above. Define

$$M \overset{\rm def}{=} \{\vec{y} = (y_1, y_2, \ldots, y_{2k}) \in \{1,-1\}^{2k} : y_1 = y_2 \wedge \wedge\, y_3 = y_4 \wedge \cdots \wedge y_{2k-1} = y_{2k}\}.$$

Note that the action of the verifier $V_{\rm comp}$(IV-NAE4$_\gamma$) in question given k proofs can be viewed as picking $v \in V$ and $f \in \mathcal{F}_{L_V}$ at random, and then picking x, x' randomly and independently from $\{1,-1\}^{2k}$ according to the distributions $\Delta_{v,f}$ and $\Delta_{v,-f}$ respectively, and finally rejecting if and only if *all* k proofs are "wrong", i.e., if $x, x' \in M$ and $x = x'$. Thus the probability that the verifier $V_{\rm comp}$(IV-NAE4$_\gamma$) rejects a set of k proofs $\{LP_i(w) : w \in W\}_{i=1}^k$ is precisely $\mathop{\mathbf{Pr}}_{v,f,x,x'}[x = x' \wedge x \in M]$. The lemma below is similar in spirit to Lemma 3.2 and states that if the verifier rejects some set of k proofs with low probability, then in fact $\mathsf{OPT}(\mathcal{LC})$ is quite high. The Lemma is proved in Section 4.4.

Lemma 4.1 *There is an absolute constant $b' > 0$ such that for every integer $k \ge 1$ and all sufficiently small $\gamma > 0$, if* $\mathop{\mathbf{Pr}}_{v,f,x \in \Delta_{v,f}, x' \in \Delta_{v,-f}}\big[x = x' \wedge x \in M\big] \le 2^{-(4k+7)}$, *then* $\mathsf{OPT}(\mathcal{LC}) > 2^{-2^{b'2^k}}$.

Theorem 4.2 *For every constant k,* NP $\subseteq$ cPCP$_{1,\frac{1}{k}}$[log, 4], *where moreover the predicate verified by the PCP upon reading bits x, y, z, w is* NAE(x, y, z, w).

Proof: Similar to the proof of Theorem 3.5 (using Lemma 4.1 in place of Lemma 3.2). □

4.3 Hardness results for hypergraph coloring

Since the predicate used by the PCP of Theorem 4.2 is that of 4-set splitting, we get the following Corollary.

Corollary 4.3 *For every constant $k \ge 2$, given an instance of 4-set splitting, it is NP-hard to distinguish between the case when there is a partition of the universe that splits all the 4-sets, and when for every set of k partitions there is at least one 4-set which is is not split by any of the k partitions.*

The above hardness can be naturally translated into a hardness result for coloring 4-uniform hypergraphs.

Theorem 4.4 (Main Theorem) *For any constant $c \ge 2$, it is NP-hard to color a 2-colorable 4-uniform hypergraph using c colors such that there is no monochromatic 4-set.*

Proof: Follows from the above Corollary since a 4-set splitting instance can be naturally identified with a 4-uniform hypergraph whose hyperedges are the 4-sets, and it is easy to see that the minimum number of partitions k needed to split all 4-sets equals $\lceil \lg c \rceil$ where c is the minimum number of colors to color the hypergraph such that no hyperedge is monochromatic. □

Theorem 4.5 *Assume* NP $\not\subseteq$ DTIME$(n^{O(\log\log n)})$. *Then there exists an absolute constant* $c_0 > 0$ *such that there is no polynomial time algorithm that can color a* 2*-colorable* 4*-uniform hypergraph using* $c_0 \log\log\log n$ *colors, where* n *is the number of vertices in the hypergraph.*

Proof: This follows since the covering soundness of the PCP in Theorem 4.2 can be made an explicit $o(1)$ function. Indeed, to have a covering soundness of $1/\log g(n)$, combining Lemma 4.1 with Theorem 3.1, the proof size we need is $n^{O(\log\delta^{-1})}2^{\delta^{-O(1)}}$ where $\delta = 2^{-2^{O(g(n))}}$. We can thus have $n^{O(\log\log n)}$ size proofs by letting $\delta^{-1} = (\log n)^{O(1)}$ and $g(n) = O(\log\log\log n)$. Similarly to Theorem 4.4, this implies $g(n)$-coloring a 2-colorable 4-uniform hypergraph is hard unless NP $\subseteq$ DTIME$(n^{O(\log\log n)})$. □

4.4 Proof of Lemma 4.1

The Proof: The proof comprises of several intermediate steps. We will not be concerned with getting the best possible bounds in an attempt not to obscure the proof. Lemma 4.1 follows from the following two lemmas. The first one states that if the distributions $\Delta_{v,f}$ and $\Delta_{v,-f}$ are close to each other, then the probability that the verifier rejects all k proofs is large. The proof of this Lemma is omitted here and can be found in the full version, but it is quite standard and follows since x, x' drawn according to $\Delta_{v,f}$ and $\Delta_{v,-f}$ are equal with large probability if the distributions are close to each other, and we know that $\mathop{\mathbf{Pr}}_{x\in\Delta_{v,f}}[x \in M]$ is large from the analysis of the previous section.

Lemma 4.6 *For every integer* $k \geq 1$*, and for all* $\gamma \leq 2^{-(k+4)}$*, if* $\mathop{\mathbf{Pr}}_{v,f,x\in\Delta_{v,f},x'\in\Delta_{v,-f}}\big[x = x' \wedge x \in M\big] \leq 2^{-(4k+7)}$ *(recall that the distributions* $\Delta_{v,f}$ *and* $\Delta_{v,-f}$ *depend upon* γ*), then*

$$\mathop{\mathbf{E}}_{v,f}\Big[\sum_{y\in\{1,-1\}^{2k}} |\Delta_{v,f}(y) - \Delta_{v,-f}(y)|\,\Big] > 2^{-(4k+6)}\,. \quad (5)$$

Lemma 4.7 *There are absolute constants* $d', e' > 0$ *such that for every* $\varepsilon > 0$*, every integer* $k \geq 1$ *and every* $\gamma > 0$*, if* $\mathop{\mathbf{E}}_{v,f}\big[\sum_{y\in\{1,-1\}^{2k}} |\Delta_{v,f}(y) - \Delta_{v,-f}(y)|\,\big] > \varepsilon$*, then* $\mathsf{OPT}(\mathcal{LC}) > (\varepsilon 2^{-2k})^{e'} 2^{-2^{d'\gamma^{-1}}}$.

Lemma 4.1 now follows since combining Lemma 4.7 with the Condition (5), we get $\mathsf{OPT}(\mathcal{LC}) > 2^{-O(k)}2^{-2^{O(2^k)}}$, and this clearly implies that $\mathsf{OPT}(\mathcal{LC}) > 2^{-2^{b'2^k}}$ for some absolute constant $b' > 0$. □ *(Lemma 4.1)*

Proof of Lemma 4.7: We are given that

$$\mathop{\mathbf{E}}_{v,f}\Big[\sum_{y\in\{1,-1\}^{2k}} \big|\Delta_{v,f}(y) - \Delta_{v,-f}(y)\big|\Big] > \varepsilon\,.$$

Now consider the Fourier expansion of $\Delta_{v,f}$ as $\Delta_{v,f}(y) = \sum_{\alpha\in\{0,1\}^{2k}} \hat{\Delta}_{v,f,\alpha}\ell_\alpha(y)$ where $\ell_\alpha(y) = \prod_{j:\alpha_j=1} y_j$, and similarly for the function $\Delta_{v,-f}$. Then using the above condition ε is less than

$$\begin{aligned}\mathop{\mathbf{E}}_{v,f}\Big[\sum_{y\in\{1,-1\}^{2k}} \big|\sum_{\alpha\in\{0,1\}^{2k}} (\hat{\Delta}_{v,f,\alpha} - \hat{\Delta}_{v,-f,\alpha})\ell_\alpha(y)\big|\Big] \\ \leq\; 2^{2k}\mathop{\mathbf{E}}_{v,f}\Big[\sum_{\alpha}|\hat{\Delta}_{v,f,\alpha} - \hat{\Delta}_{v,-f,\alpha}|\Big]\,,\end{aligned}$$

and this implies that there exists an $\alpha \in \{0,1\}^{2k}$ such that

$$\mathop{\mathbf{E}}_{v,f}\big[\,|\hat{\Delta}_{v,f,\alpha} - \hat{\Delta}_{v,-f,\alpha}|\big] > \frac{\varepsilon}{2^{4k}}\,. \quad (6)$$

We will use any (fixed) such α to define "proof tables" A_v, D_w, E_w for every $v \in V$ and $w \in W$ where $A_v : \mathcal{F}_{L_V} \to \{1,-1\}$ and $D_w, E_w : \mathcal{F}_{L_W} \to \{1,-1\}$. For any $v \in V$, the table $A = A_v$ (we will omit the subscript v though it should be treated as implicit) is defined as follows: For $f \in \mathcal{F}_{L_V}$, $A(f) = \mathrm{sign}(\hat{\Delta}_{v,f,\alpha} - \hat{\Delta}_{v,-f,\alpha})$ where $\mathrm{sign}(x)$ is the sign function that takes value 1 if $x > 0$ and -1 if $x < 0$. Note that clearly $A(f) = -A(-f)$[1]; so that the A-table is *folded*.

To define D_w, E_w, first, set $\alpha_1 \in \{0,1\}^k$ (resp. $\alpha_2 \in \{0,1\}^k$) to be the projection of α on the odd coordinates $\{1, 3, \ldots, 2k-1\}$ (resp. even coordinates $\{2, 4, \ldots, 2k\}$). (Here α_1 and α_2 "correspond" to the $B_i(g_1)$ and $B_i(g_2)$ coordinates respectively.) For any $g \in \mathcal{F}_{L_W}$, we define $D(g) = D_w(g) = \prod_{i:\alpha_1(i)=1} B_i(g)$ and similarly $E(g) = E_w(g) = \prod_{i:\alpha_2(i)=1} B_i(g)$, where B_i stands for $\mathsf{LP}_i(w)$. We will omit the subscript on D, E for notational convenience, and it should always be treated as implicit. The key property satisfies by these tables is captured by the following two Claims about the properties of the tables A, D, E defined above. Proofs of these claims are sketched at the end of this section.

Claim 1: $\mathop{\mathbf{E}}_{v,w,p,f,g_1,g_2}\big[A(f)D(g_1)E(g_2)\big] = 2^{2k-1}\mathop{\mathbf{E}}_{v,f}\Big[\;\big|\hat{\Delta}_{v,f,\alpha} - \hat{\Delta}_{v,-f,\alpha}\big|\;\Big]$ *where the distribution of* p, f, g_1, g_2 *is the same as the one used by the inner verifier IV4.*

Claim 2: *For every* $\zeta > 0$*, and every* $\gamma > 0$*, if* $\mathop{\mathbf{E}}_{v,w,p,f,g_1,g_2}\big[A(f)D(g_1)E(g_2)\big] > \zeta$ *then there is a constant* δ *depending only on* ζ *and* γ*, with* $\delta = \zeta^{O(1)}2^{-2^{O(\gamma^{-1})}}$*, such that* $\mathsf{OPT}(\mathcal{LC}) > \delta$.

Combining the result of Claim 1 with Equation (6) we get

$$\mathop{\mathbf{E}}_{v,w,p,f,g_1,g_2}\big[A(f)D(g_1)E(g_2)\big] > \varepsilon 2^{-(2k+1)}\,, \quad (7)$$

[1] When $\hat{\Delta}_{v,f,\alpha} = \hat{\Delta}_{v,-f,\alpha}$, we assume that $A(f)$ is defined to be $f(\ell_0)$ for some fixed $\ell_0 \in L_V$, so that $A(f) = -A(-f)$ holds even in this case.

and the proof of Lemma 4.7 is now complete using Claim 2 together with the above Equation (7). □ *(Lemma 4.7)*

Proof of Claim 1: Observe that for each fixed (v, f), $\mathop{\mathbf{E}}_{p,w,g_1,g_2}\left[D(g_1)E(g_2)\right]$ equals

$$\sum_{y\in\{1,-1\}^{2k}} \left(\mathbf{Pr}\left[(B_i(g_1)B_i(g_2))_{i=1}^{k} = y\right]\cdot \prod_{i:\ \alpha_i=1} y_i\right)$$
$$= \sum_{y\in\{1,-1\}^{2k}} \Delta_{v,f}(y)\ell_\alpha(y) = 2^{2k}\hat{\Delta}_{v,f,\alpha}\,.$$

Now $\mathop{\mathbf{E}}_{v,w,p,f,g_1,g_2}\left[A(f)D(g_1)E(g_2)\right]$ equals

$$\frac{1}{2}\Big(\mathop{\mathbf{E}}_{v,w,p,f,g_1,g_2}\left[A(f)D(g_1)E(g_2)\right] - \mathop{\mathbf{E}}_{v,w,p,-f,g_1,g_2}\left[A(f)D(g_1)E(g_2)\right]\Big)$$
$$= \frac{2^{2k}}{2}\Big(\mathop{\mathbf{E}}_{v,f}\left[\mathrm{sign}(\hat{\Delta}_{v,f,\alpha}-\hat{\Delta}_{v,-f,\alpha})\hat{\Delta}_{v,f,\alpha}\right] - \mathop{\mathbf{E}}_{v,f}\left[\mathrm{sign}(\hat{\Delta}_{v,f,\alpha}-\hat{\Delta}_{v,-f,\alpha})\hat{\Delta}_{v,-f,\alpha}\right]\Big)$$
$$= 2^{2k-1}\mathop{\mathbf{E}}_{v,f}\left[\left|\hat{\Delta}_{v,f,\alpha}-\hat{\Delta}_{v,-f,\alpha}\right|\right]$$

where in the first step we used that $A(f) = -A(-f)$, and the second step follows since the distribution of (g_1, g_2) given $-f$ was picked is identical to the distribution of (h_1, h_2) given f was picked. □ *(Claim 1)*

Proof of Claim 2 (Sketch): The proof follows along the lines of the proof of Lemma 6.10 in [16]. Recall that p is picked uniformly at random from $\{\varepsilon_1, \ldots, \varepsilon_t\}$ where $t = \lceil \gamma^{-1}\rceil$, $\varepsilon_1 = \gamma^2$ and $\varepsilon_j = \varepsilon_{j-1}^{4/c}$ for every j, $1 < j \le t$. Clearly there exists a j, $1 \le j \le t$ such that $\mathop{\mathbf{E}}_{v,w,f,g_1,g_2}\left[A(f)D(g_1)E(g_2)|p = \varepsilon_j\right] > \zeta$. In the rest of the proof, we fix p to equal this ε_j. Note that $p = \varepsilon_j \ge \varepsilon_t \ge 2^{-2^{O(\gamma^{-1})}}$. It turns out to be useful to express $A(f), D(g_1)$ and $E(g_2)$ using their Fourier expansion as $A(f) = \sum_{\alpha\subseteq L_V} \hat{A}_\alpha \ell_\alpha(f)$ where $\ell_\alpha(f) = \prod_{y\in\alpha} f(y)$, and similarly for $D(g_1)$ and $E(g_2)$. Now using an analysis similar to Lemma 6.10 of [16], one can show that if $\mathop{\mathbf{E}}_{v,w,f,g_1,g_2}\left[A(f)D(g_1)E(g_2)\right] > \zeta$, then for all $K \ge \left(\frac{16}{p\zeta}\right)^{1/c}$,

$$\mathop{\mathbf{E}}_{v,w}\Big[\sum_{\substack{\beta:|\beta|\le K\\ \alpha\subseteq\pi_{v,w}(\beta)}} \hat{A}_\alpha^2|\hat{D}_\beta\hat{E}_\beta|\Big] > \frac{\zeta}{4}\cdot\frac{\zeta}{2} = \frac{\zeta^2}{8}\,. \tag{8}$$

Since $p \ge 2^{-2^{O(\gamma^{-1})}}$, this implies we can pick $K = \zeta^{-O(1)}2^{2^{O(\gamma^{-1})}}$ in the above inequality. Note that Condition (8) forces the Fourier spectrum of A, D, E to have (somewhat) large support on low-weight coefficients (corresponding to small $|\alpha|, |\beta|$), and this will enable us to define "good" labels for vertices in $V \cup W$ and prove the claimed lower bound on $\mathsf{OPT}(\mathcal{LC})$. We now describe a probabilistic procedure to define labels which satisfies a good fraction of edges in the instance $\mathcal{LC}$ in expectation, and this will clearly give a lower bound on $\mathsf{OPT}(\mathcal{LC})$.

We define $\ell(v) \in L_V$ for a vertex $v \in V$ as follows. Let $A = A_v$ and pick a set $\alpha \subseteq L_V$ with probability $\hat{A}_\alpha^2$. By Parseval's identity this is a valid probability distribution. Then pick an element $a \in \alpha$ at random and set $\ell(v) = a$. An important point here is that $\hat{A}_\emptyset = 0$ since $A(f) = -A(-f)$ for all $f \in L_V$ [16], and thus we never get "stuck" by picking $\alpha = \emptyset$.

Next, we define $\ell(w) \in L_W$ for $w \in W$ as follows. Let $D = D_w$ and $E = E_w$. Pick a set $\beta \subseteq L_W$ with probability *proportional* to $|\hat{D}_\beta\hat{E}_\beta|$. Note that $\sum_\beta |\hat{D}_\beta\hat{E}_\beta| \le \left(\sum_\beta \hat{D}_\beta^2\right)^{1/2}\left(\sum_\beta \hat{E}_\beta^2\right)^{1/2} = 1$ by Cauchy-Schwartz, so that a set β is picked with probability *at least* $|\hat{D}_\beta\hat{E}_\beta|$. If $\beta = \emptyset$, set $\ell(w)$ to be some fixed element $b_0 \in L_W$, else set $\ell(w)$ equal to a *random* element of β.

Let $X_{v,w}$ be a random variable which takes on value 1 when the edge $(v, w) \in E$ is satisfied by the above randomized experiment, i.e., $X_{v,w} = 1$ if $\ell(v) = \pi_{v,w}(\ell(w))$, and equals 0 otherwise. The expected fraction of satisfied edges is $\mathop{\mathbf{E}}_{v,w}\left[X_{v,w}\right]$ is at least

$$\mathop{\mathbf{E}}_{v,w}\Big[\sum_{\substack{\alpha,\beta\\ \alpha\cap\pi_{v,w}(\beta)\neq\emptyset}} \hat{A}_\alpha^2|\hat{D}_\beta\hat{E}_\beta|\frac{1}{|\alpha|}\frac{1}{|\beta|}\Big]$$
$$\ge \mathop{\mathbf{E}}_{v,w}\Big[\sum_{\substack{\beta:|\beta|\le K\\ \alpha\subseteq\pi_{v,w}(\beta)}} \hat{A}_\alpha^2|\hat{D}_\beta\hat{E}_\beta|\frac{1}{K^2}\Big] > \frac{\zeta^2}{8K^2}$$

using Equation (8). The first step above is valid since $\hat{A}_\emptyset = 0$ and thus any term with non-zero $\hat{A}_\alpha$ with $\alpha \subseteq \pi_{v,w}(\beta)$ also satisfies $\alpha \cap \pi_{v,w}(\beta) \neq \emptyset$. Recalling that we picked $K = \zeta^{-O(1)}2^{2^{O(\gamma^{-1})}}$, we have also $\mathsf{OPT}(\mathcal{LC}) > \zeta^{O(1)}2^{-2^{O(\gamma^{-1})}}$ (with slightly larger constant in the O-notation), and the claim follows. □ *(Claim 2)*

5 Concluding Remarks

We gave a 4-query PCP verifier for languages in NP with $o(1)$ covering soundness and whose acceptance predicate was $(x \neq y) \vee (z \neq w)$. In order to obtain our hardness result for hypergraph coloring, we needed to tailor the acceptance predicate of the PCP to correspond exactly to the one for hypergraph coloring (i.e., $\mathrm{NAE}(x, y, z, w)$), and then analyze the covering soundness of the resulting PCP. This is necessary to obtain hardness results for minimization problems using this approach. Gadgets, which are useful in transforming PCPs in the usual setting, are useless here. Indeed, say we "implement" a constraint f using several other constraints $\sigma_1, \ldots, \sigma_s$, and two proofs Π_1 and Π_2 suffice to satisfy all the σ_i. The constraints σ_1 and σ_2 for example might be satisfied by two different proofs, and thus one cannot conclude that one of Π_1 and Π_2 indeed satisfies the original constraint f. Thus the standard approach of reduction between various constraint families completely breaks down. As a concrete example, suppose we reduce $\mathrm{NAE}(x, y, z, w)$ into 4-SAT clauses $(x \vee y \vee z \vee w)$ and $(\bar{x} \vee \bar{y} \vee \bar{z} \vee \bar{w})$. We know, by our main result Theorem 4.4, that the NAE constraints (of even an instance that is satisfiable by a single assignment) are NP-hard to satisfy using any constant number of assignments, where as any 4-SAT instance is trivially satisfiable using just two assignments, namely any assignment and its complement!

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 (1996), pp. 425-439.

[2] S. Arora and C. Lund. Hardness of Approximations. In *Approximation Algorithms for NP-hard Problems*, (Dorit Hochbaum, ed.), PWS, 1996.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in *Proceedings of FOCS'92*.

[4] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in *Proceedings of FOCS'92*.

[5] J. Beck. On 3-chromatic hypergraphs. *Discrete Mathematics*, 24 (1978), pp. 127-137.

[6] J. Beck. An algorithmic approach to the Lovász Local Lemma. *Random Structures and Algorithms*, 2 (1991), pp. 343-365.

[7] M. Bellare, O. Goldreich and M. Sudan. Free bits, PCP's and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804-915, 1998. Preliminary version in *Proc. of FOCS'95*.

[8] H. Chen and A. Frieze. Coloring bipartite hypergraphs. *Proc. of 5th IPCO*, 1996, pp. 345-358.

[9] P. Erdös. On a combinatorial problem I. *Nordisk Mat. Tidskrift*, 11 (1963), pp. 5-10.

[10] U. Feige and J. Kilian. Zero-knowledge and the chromatic number. In *Proceedings of the 11th Annual Conference on Computational Complexity*, 1996.

[11] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115-1145, 1995.

[12] V. Guruswami. *Query-efficient Checking of Proofs and Improved PCP Characterizations of NP*. S.M Thesis, MIT, May 1999.

[13] V. Guruswami. The Approximability of set splitting problems and satisfiability problems with no mixed clauses. *Proc. of the 3rd Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2000)*, to appear.

[14] V. Guruswami and S. Khanna. On the hardness of 4-coloring a 3-colorable graph. *Proc. of Complexity 2000*, pp. 188-197.

[15] V. Guruswami, D. Lewin, M. Sudan and L. Trevisan. A tight characterization of NP with 3 query PCPs. *ECCC Technical Report* TR98-034, 1998. Preliminary Version in *Proc. of FOCS'98.*

[16] J. Håstad. Some optimal inapproximability results. *Technical Report TR97-37*, Electronic Colloquium on Computational Complexity, 1997. Preliminary version in *Proc. of STOC'97.*

[17] D. R. Karger, R. Motwani and M. Sudan. Approximate graph coloring using semidefinite programming. *Journal of the ACM*, 45 (1998), pp. 246-265.

[18] S. Khanna, N. Linial and S. Safra. On the hardness of approximating the chromatic number. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems, ISTCS*, pp. 250-260, IEEE Computer Society Press, 1993.

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

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

[21] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41:960-981, 1994.

[22] C. McDiarmid. A random recoloring method for graphs and hypergraphs. *Combinatorics, Probability and Computing*, 2 (1993), pp. 363-365.

[23] C. McDiarmid. Hypergraph coloring and the Lovász Local Lemma. *Discrete Mathematics*, 167/168 (1997), pp. 481-486.

[24] J. Radhakrishnan and A. Srinivasan. Improved bounds and algorithms for hypergraph two-coloring. *Proc. of* 39^{th} *FOCS*, (1998), pp. 684-693.

[25] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998. Preliminary version in *Proc. of STOC'95.*

[26] J. H. Spencer. Coloring n-sets red and blue. *J. Combinatorial Theory, Series A*, 30 (1981), pp. 112-113.