Electronic Colloquium on Computational Complexity, Report No. 73 (2018)

NP-hardness of coloring 2-colorable hypergraph with poly-logarithmically many colors

Amey Bhangale [*]

April 21, 2018

Abstract

We give very short and simple proofs of the following statements: Given a 2-colorable 4-uniform hypergraph on n vertices,

1. It is NP-hard to color it with $\log^{\delta} n$ colors for some $\delta > 0$.
2. It is $quasi$-NP-hard to color it with $O\left(\log^{1-o(1)} n\right)$ colors.

In terms of NP-hardness, it improves the result of Guruswami, Håstad and Sudan [GHS02], combined with Moshkovitz-Raz [MR10], by an 'exponential' factor. The second result improves the result of Saket [Sak14] which shows $quasi$-NP-hardness of coloring a 2-colorable 4-uniform hypergraph with $O\left(\log^{\gamma} n\right)$ colors for a sufficiently small constant $1 \gg \gamma > 0$.

Our result is the first to show the NP-hardness of coloring a c-colorable k-uniform hypergraph with poly-logarithmically many colors, for *any* constants $c \geq 2$ and $k \geq 3$.

1 Introduction

A k-uniform hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$ is a collection of vertices $\mathcal{V}$ and a family $\mathcal{E}$ of subsets of size k of the vertices. Let n denote the number of vertices $|\mathcal{V}|$. A coloring of $\mathcal{H}$ with c colors is a mapping $\chi : \mathcal{V} \to [c]$. A coloring χ is said to be a valid c-coloring if for every $e \in \mathcal{E}$, χ assigns at least two different colors to the vertices in e. The *chromatic number* of a hypergraph is the minimum c for which a valid c-coloring exists.

It is easy to find a 2-coloring of a 2-colorable graph. Deciding whether a graph is 3-colorable or not is a well known NP-complete problem [GJ76]. In an approximate graph (or hypergraph) coloring problem, given a graph which is c colorable, the algorithm is allowed to use more than c colors to properly color the given graph. The best known polynomial-time algorithms to color a 3-colorable graph require $n^{\Omega(1)}$ colors. However, the known NP-hardness results can only rule out coloring a 3-colorable graph with 4 colors [KLS93, GK04]. This also implies that coloring a t-colorable graph with $t + 2\lfloor t/3 \rfloor - 1$ colors is NP-hard. Garey and Johnson [GJ76] proved NP-hardness of $(2t - 5)$-coloring a t-colorable graph, for all $t \geq 6$. Recently, Brakensiek-Guruswami [BG16] showed that for all $t \geq 3$, it is NP-hard to find a c-coloring when $c \leq 2t - 2$.

[*] Weizmann Institute of Science, Rehovot, Israel. Email: amey.bhangale@weizmann.ac.il. Research supported by Irit Dinur's ERC-CoG grant 772839.

ISSN 1433-8092

For hypergraphs with uniformity 3 and more, the situation is completely different when the hypergraph is 2-colorable. The best known approximation algorithm for coloring a 2-colorable hypergraph is with n^ϵ color for some explicit $\epsilon \in (0,1)$. It is easy to see that if t-coloring of c-colorable k-uniform hypergraphs is hard, then it also implies a similar hardness result for k'-uniform hypergraphs for any $k' > k$ up to polynomial-time reductions i.e. coloring a c-colorable k'-uniform hypergraph with t colors is also hard. Therefore, the wishful inapproximability result is to get n^ϵ coloring hardness for 2-colorable 3-uniform hypergraphs for some $\epsilon > 0$.

Two directions have been pursued in literature. One is getting better and better hardness results in terms of colorability, where the uniformity of a hypergraph is any constant k and the guarantee is that the hypergraph is colorable with constantly many colors. The other is getting hardness results for smaller values of k . Both the directions are equally interesting and have been looked into for several years. The table in Figure 1 summarizes the known inapproximability results: For *e.g.* Guruswami *et al.* [GHS02] (the first row) showed that given a 2-colorable 4-uniform hypergraph, there is no polynomial-time algorithm to color it with $O\left(\frac{\log\log n}{\log\log\log n}\right)$ many colors unless $\mathsf{NP} \subseteq \mathsf{DTIME}\left(n^{O(\log\log n)}\right)$. We would like to point out that the condition $\mathsf{NP} \subseteq \mathsf{DTIME}\left(n^{O(\log\log n)}\right)$ from [GHS02] can be replaced with $\mathsf{P} \neq \mathsf{NP}$, if one carries out the reduction of [GHS02] starting with the PCP of [MR10].

We prove the following results:

Theorem 1.1. *Assuming $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log\log n)})$, there is no polynomial-time algorithm that colors a 2-colorable 4-uniform hypergraph on n vertices with $O\left(\frac{\log n}{\log\log n}\right)$ colors.*

Although an explicit constant in the exponent of $\log n$ was not mentioned in [Sak14], it is less than 0.01. Therefore, Theorem 1.1 improves the best known inapproximability result by Saket [Sak14] by a large polynomial factor, in terms of coloring with few colors. However, both the results are incomparable (See Remark 1.3).

Our second result shows NP-hardness of coloring a 2-colorable 4-uniform hypergraph.

Theorem 1.2. *Assuming $\mathsf{P} \neq \mathsf{NP}$, there is no polynomial-time algorithm that colors a 2-colorable 4-uniform hypergraph on n vertices with $\log^\delta n$ colors for some $\delta > 0$.*

Theorem 1.2 improves the best known inapproximability result by Guruswami *et al.*[GHS02] by an‘exponential’ factor, in terms of coloring with few colors. In fact, our result is the first to show the NP-hardness of coloring a c-colorable k-uniform hypergraph with $\texttt{poly}(\log n)$ colors for any constants $c \geq 2$ and $k \geq 3$.

An independent set in a hypergraph is defined as a set of vertices such that no hyperedge lies entirely within the set. In a valid coloring of a hypergraph, the vertices colored with the same color form an independent set. Thus, saying that a hypergraph does not contain an independent set of size n/c is stronger than saying that it cannot be colored with c colors.

In terms of approximating the size of the maximum independent set, Khot-Saket [KS14b] showed that given an *almost* 2-colorable 4-uniform hypergraph, it is quasi-NP-hard to find an independent set of fractional size $2^{(\log n)^{1-\gamma}}$, for any $\gamma > 0$. More concretely, they show that it is quasi-NP-hard to distinguish between cases when a 4-uniform hypergraph has

	Completeness	Soundness	Assumption
Guruswami *et al.* [GHS02, MR10]	**2-colorable 4-uniform hypergraph**	$\Omega\left(\frac{\log\log n}{\log\log\log n}\right)$	$\mathsf{P} \neq \mathsf{NP}$
Khot [Kho02a]	q-colorable 4-uniform hypergraph, $q \geq 5$	$(\log n)^{cq}$ for some $c > 0$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{(\log n)^{O(1)}}\right)$
Khot [Kho02b]	3-colorable 3-uniform hypergraph	$(\log\log n)^{1/9}$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{(\log n)^{O(1)}}\right)$
**Dinur *et al.* [DRS02]	2-colorable 3-uniform hypergraph	$\Omega(\sqrt[3]{\log\log n})$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{(\log n)^{O(1)}}\right)$
Saket [Sak14]	**2-colorable 4-uniform hypergraph**	$\Omega(\log^{\delta} n)$ for $0 < \delta \ll 1$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{O(\log\log n)}\right)$
Guruswami *et al.* [GHH+14]	2-colorable 8-uniform hypergraph 4-colorable 4-uniform hypergraph	$2^{2^{\Omega(\sqrt{\log\log n})}}$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{2^{O(\sqrt{\log\log n})}}\right)$
Guruswami *et al.* [GHH+14]	3-colorable 3-uniform hypergraph	$2^{\Omega(\frac{\log\log n}{\log\log\log n})}$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{2^{O(\frac{\log\log n}{\log\log\log n})}}\right)$
Khot-Saket [KS14a]	2-colorable 12-uniform hypergraph	$2^{(\log n)^{\Omega(1)}}$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{(\log n)^{O(1)}}\right)$
Varma [Var15]	2-colorable 8-uniform hypergraph 4-colorable 4-uniform hypergraph	$2^{(\log n)^{\Omega(1)}}$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{(\log n)^{O(1)}}\right)$
**This paper	**2-colorable 4-uniform hypergraph**	$\Omega\left(\frac{\log n}{\log\log n}\right)$	$\mathsf{NP} \not\subseteq \mathsf{DTIME}\left(n^{O(\log\log n)}\right)$
**This paper	**2-colorable 4-uniform hypergraph**	$\Omega(\log^{\delta} n)$ for $0 < \delta \ll 1$	$\mathsf{P} \neq \mathsf{NP}$

Figure 1: Known inapproximability results for hypergraph coloring (for ** see Remark 1.3).

an independent set of size at least $\left(\frac{n}{2} - o(n)\right)$ vs. when it has no independent set of size $n/2^{(\log n)^{1-\gamma}}$, for any $\gamma > 0$.

Remark 1.3. *With the exception of results from this paper and Dinur et al.(denoted by ** in Figure 1), all the remaining results give stronger inapproximability results than showing non c-colorability in* soundness. *Namely, they show it is hard to distinguish between cases when a hypergraph satisfies the property in the* completeness *column vs. when there does not exist an independent set of size at least $1/s$, where s is the quantity in the* soundness *column.*

1.1 Proof Overview

Our inapproximability result is obtained by constructing a *probabilistically checkable proof* where the locations are the vertices of a hypergraph and every check denotes a hyperedge of the hypergraph. Typically, the PCP is constructed by composing the so-called *outer verifier* with the *inner verifier*. Our starting point is a basic outer verifier that one gets from the PCP Theorem [FGL+96, ALM+98, AS98], along with Raz's parallel repetition theorem [Raz98]. We can view it as an instance of a *Label Cover*. A Label Cover instance consists of a bipartite graph $G(U, V, E)$ where every edge is associated with a projection constraint $\pi_e : [R] \to [L]$ (See section 2.1). The inner verifier consists of encoding of labels of the vertices in a specific format and performing a test on the encoding.

A typical hardness of approximation result uses the *Long Code* encoding where the encoding of a k bit binary (or q-ary) string has an entry for every $f : \{0,1\}^k \to \{0,1\}$ (or $f : [q]^k \to [q]$ for $q \geq 2$). This blows up a string of length k into a string of length 2^{2^k}. Such an encoding is used in the construction of PCP by Guruswami *et al.* [GHS02] and they managed to show a $\approx \log\log n$ lower bound on the inapproximability of coloring a 2-colorable 4-uniform hypergraph. Khot [Kho02a] used a different encoding based on *Split Code* to get the hardness of poly-logarithmic number of colors, but for q-colorable 4-uniform hypergraph where $q \geq 5$. Saket [Sak14] improved the result of Guruswami *et al.* [GHS02] and showed the inapproximability of coloring with $O((\log n)^\delta)$ colors for a very small constant $0 < \delta \ll 1$. Saket also used the Long Code encoding, but the hypergraph is formed using a different inner verifier compared to [GHS02]. In a more recent series of works [GHH+14, KS14a, Var15], efficient encodings were designed based on so-called *Short Code* which was used to break the poly-logarithmic barrier. It is not clear how to use this Short Code encoding to prove better inapproximability for 2-colorable 4-uniform hypergraphs. In all the previous constructions of PCPs based on Short Code, either the alphabet size is at least 3 (the guarantee on the hypergraph colorability in the completeness case) or the number of queries (the uniformity of the constructed hypergraph) is at least 6 if the alphabet size is restricted to 2 (See Figure 1).

Our reduction is along the lines of the reduction in [DRS02] which showed that coloring a 2-colorable 3-uniform hypergraph with $(\log\log n)^{1/3}$ colors is quasi-NP-hard. We give a very brief outline of their reduction first: This reduction also encodes a label in the outer verifier with a (rather large) subset of the Long Code encoding of size $2^{2^k - \texttt{poly}(k)}$. They were able to construct a PCP over a binary alphabet which only queries 3 bits, giving the hardness for the above mentioned coloring problem. The encoding consists of all the locations in the Long Code encoding which correspond to a *slice* of $\{0,1\}^{2^k}$ of hamming weight m, where m is roughly $2^k/2$. The reduction crucially used two important properties of a specific graph, known as *Kneser graph*, on these locations. The first is that the chromatic number of the graph is *large* and the second is that in all the colorings of this graph with strictly fewer colors than the chromatic number, there is a *large* color class containing a monochromatic edge. The reduction also needed a multi-layered version of Label Cover, which gives the hardness of $(\log\log n)^{1/3}$ colors based on $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{(\log n)^{O(1)}})$.

In our construction of the PCP, we start with a basic outer verifier and use a very short encoding of the labels of the outer verifier. More specifically, to get an inapproximability of c-colorability, we use an encoding of size roughly 2^{kc}, which is much shorter than the Long Code encoding 2^{2^k} or the encoding used in [DRS02] for $c \ll 2^k$. This short encoding is the reason for improvement in the inapproximability. Of course, the whole analysis needs to work for such a short encoding. For the analysis to work, we need to query one more bit from the composed PCP, giving a hardness result for 4-uniform hypergraphs.

The subset of locations in the Long Code that we use corresponds to the vertex set of the *Schrijver graph* (See section 2.2 for the definition). It has much fewer number of vertices than the Kneser graph on the same slice. Moreover, the Schrijver graph also has a property that its chromatic number is large. Unfortunately, it is known to be a *vertex-critical graph* [1]. Thus, the property analogous to the second property of the Kneser graph mentioned previously no longer holds for the Schrijver graph. However, if we are allowed to query one more bit, then given an efficient coloring of the constructed 4-uniform hypergraph, it is possible to list-decode, from the coloring, a small list of possible labelings to the vertices of the outer verifier. The structure of the hyperedges in our 4-uniform hypergraph ensures that the lists are *consistent* with each other i.e. there is a way to assign labels to the vertices of the Label Cover instance from the list, which satisfies many constraints of the Label Cover instance.

1.1.1 $\log^{\delta} n$ NP-hardness

As highlighted earlier, the size of the encoding we use is 2^{kc} and if we start with the NP-hardness of the Label Cover given by [MR10], which has better parameters in the sub-constant soundness regime, we can run the whole reduction in polynomial-time if we set $c = \log^{\delta} n$ for small enough $\delta > 0$. Therefore, we get the required inapproximability result proving Theorem 1.2.

1.1.2 $\Omega\left(\frac{\log n}{\log\log n}\right)$ $quasi$-NP-hardness

The soundness parameter of the Label Cover instance given by [MR10] has an *inverse exponential* dependence on the alphabet size of the instance (See Theorem 2.3). To get the improved $\log^{1-o(1)} n$ coloring hardness, we start with the usual Label Cover instance derived from the combination of the PCP Theorem and the parallel repetition theorem. In this case, there is an *inverse polynomial* dependence on the soundness parameter and the alphabet size which enables us to prove a better hardness factor. One drawback of using this Label Cover instance is that the instance size grows polynomially with the number of parallel repetitions. More concretely, an instance of size n becomes an instance of size $n^{O(r)}$ with r repetitions. Thus, it restricts us to assume NP $\not\subseteq$ DTIME $\left(n^{O(\log\log n)}\right)$ instead of P $\neq$ NP, as we need $r = \Omega(\log\log n)$ repetitions.

1.2 $\log^{\delta} n$ NP-hardness via Independent set analysis?

We discuss here informally why the previous approaches could not reach $\texttt{poly}(\log n)$ factor NP-hardness for approximating hypergraph coloring. Since all the previous approaches (except [DRS02]) get the inapproximability result by showing the hardness of approximating an independent set, we focus on those works here. To get NP-hardness of $\texttt{poly}(\log n)$-coloring by showing the NP-hardness of approximating an independent set by $1/\texttt{poly}(\log n)$

[1] A vertex-critical graph is a graph G in which every vertex is a critical element, that is, if its deletion decreases the chromatic number of G.

factor, usually the soundness of the outer verifier (or the Label Cover instance) that one needs is at most $1/\texttt{poly}(\log n)$. The outer verifier of [MR10] hasan exponential dependence on the label set size and the inverse of the soundness parameter. Thus, the label set size which is required for such reductions is at least $2^{\texttt{poly}(\log n)}$. The analysis of [GHH$^+$14] uses the Short Code encoding where the degree is $\omega(1)$. Therefore, the blowup in the reduction size is $2^{\texttt{poly}(\log n)^{\omega(1)}}$, which is super polynomial.

One can use a Short Code encoding where the degree is constant so that the blow-up is limited to $2^{\texttt{poly}(\log n)^{O(1)}}$. [KS14a] used a Quadratic Code where the degree is 2. One can possibly use the Quadratic Code encoding or Short Code encoding of constant degree and get a hypergraph of polynomial size starting with a Label Cover instance with $1/\texttt{poly}(\log n)$ soundness. Unfortunately, the analysis of [KS14a] crucially needed an outer verifier with certain properties. It is not yet known how to get a Label Cover instance by a polynomial-time reduction starting from a 3SAT instance, with similar properties and with $1/\texttt{poly}(\log n)$ soundness (the latter was done in [MR10]).

2 Preliminaries

We denote a set $\{1, 2, \ldots, n\}$ by $[n]$. Bold face letters $\mathbf{a}, \mathbf{b}, \mathbf{c} \ldots$ are used to denote strings and subscripts are used to denote the elements at the respective locations in a string. By an abuse of notation, we will use $\mathbf{a} \in \{0, 1\}^n$ as a binary string of length n as well as a subset of $[n]$ given by $\{i \in [n] \mid \mathbf{a}_i = 1\}$. We will denote by $2^{[n]}$ the set of all subsets of $[n]$ and by $\binom{[n]}{k}$ the set of all subsets of $[n]$ of size k. We denote the quantity $O(n^k)$, where $k \in \mathbb{R}^+$ by $\texttt{poly}(n)$.

2.1 Label Cover

Now, we define a Label Cover instance which is the starting point of our reduction.

Definition 2.1 (Label Cover)**.** *A Label Cover instance $\mathcal{L} = (U, V, E, [R], [L], \phi)$ consists of a bi-regular bipartite graph on two sets of variables $U \cup V$. The range of variables in U is denoted by $[R]$ and that in V by $[L]$. Every $(x, y) \in E$, has a constraint $\phi_{x \to y} : [R] \to [L]$. Moreover, every constraint between a pair of variables is a projection constraint i.e. a labeling to x uniquely defines a labeling to y that satisfies the constraint $\phi_{x \to y}$.*

For brevity, we say $x \sim y$, or 'x is a neighbor of y' if $\phi_{x \to y} \in \phi$. We say that a Label Cover instance is ϵ-satisfiable, if there exists a labeling to the variables which satisfies at least ϵ fraction of the constraints between U and V.

We have the following NP-hardness result which follows from the PCP Theorem [FGL$^+$96, ALM$^+$98, AS98] along with Raz's parallel repetition theorem [Raz98].

Theorem 2.2. *For any parameter $l \in \mathbb{N}$, there exists a reduction from a 3-SAT instance of size n to a Label Cover instance with $n^{O(\ell)}$ variables over a range of size $2^{O(\ell)}$. The Label Cover instance has the following completeness and soundness conditions:*

- *If the 3-SAT instance is satisfiable, then there exists an assignment to the Label Cover instance that satisfies all the constraints.*

- *If the 3-SAT instance is not satisfiable, then every assignment to the Label Cover instance satisfies at most $2^{-\Omega(\ell)}$ fraction of the constraints.*

Moreover, the reduction runs in time $n^{O(\ell)}$.

For our NP-hardness result, we need the following reduction from a 3-SAT instance to a Label Cover instance by [MR10], which gives better parameters for *subconstant soundness*.

Theorem 2.3. *There exist absolute constants $c', c'' > 1$ such that for every n and $\epsilon > 0$ (ϵ can be any function of n), there exists a reduction from a 3-SAT instance of size n to a Label Cover instance with $n^{1+o(1)} \cdot \left(\frac{1}{\epsilon}\right)^{c'}$ variables over a range of size $2^{\left(\frac{1}{\epsilon}\right)^{c''}}$. The Label Cover instance has the following completeness and soundness conditions:*

- *If the 3-SAT instance is satisfiable, then there exists an assignment to the Label Cover instance that satisfies all the constraints.*
- *If the 3-SAT instance is not satisfiable, then every assignment to the Label Cover instance satisfies at most ϵ fraction of the constraints.*

Moreover, the reduction runs in time $\texttt{poly}(n, \frac{1}{\epsilon})$.

2.2 Schrijver Graphs

In this section, we define a Schrijver graph of order n, k, denoted by $SG(n,k)$, where $n \geq 3, k < n$ and $n-k$ is even, which we use as a gadget in our reduction. The vertex set of this Schrijver graph, denoted by $V_{SG}(n,k)$, is a subset of $\{0,1\}^n$. An element $\mathbf{a} \in \{0,1\}^n$ belongs to $V_{SG}(n,k)$ iff

1. The hamming weight of $\mathbf{a}$ is $\frac{n-k}{2}$.
2. There exists no $i \in [n]$ such that both $\mathbf{a}_i = 1$ and $\mathbf{a}_{(i+1) \mod (n+1)} = 1$.

In other words, if we denote a cycle graph on $\{1, 2, \ldots, n\}$ by C_n, then the vertex set $V(n,k)$ corresponds to all independent sets in C_n of size $\frac{n-k}{2}$. The edge set of $SG(n,k)$ is as follows:

$$E_{SG}(n,k) = \{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a} \cap \mathbf{b} = \emptyset\}.$$

In other words, $\mathbf{a} \sim \mathbf{b}$ in $SG(n,k)$ iff their supports are disjoint.

We need two properties of the Schrijver graph $SG(n,k)$, first of which is easy to prove.

Claim 2.4 ([DRS02])**.** $|V_{SG}(n,k)| \leq \binom{n}{k}$.

Proof: Map every vertex $\mathbf{x} \in V_{SG}(n,k)$ to $\tilde{\mathbf{x}} \in \{0,1\}^n$ by setting $\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_{i+1 \mod (n+1)} = 1$ iff $\mathbf{x}_i = 1$ and rest of the coordinates to 0. Clearly, this is an injective map from $V_{SG}(n,k)$ to the set $\binom{[n]}{n-k}$. ■

The second property of $SG(n,k)$ is its chromatic number. The following theorem from [Sch78] gives the exact chromatic number of $SG(n,k)$ which builds on the beautiful proofs by Lovász [Lov78] and Bárány [Bár78].

Theorem 2.5 ([Sch78])**.** *The chromatic number of $SG(n,k)$ is $k+2$.*

3 Main Reduction

We give a reduction from a Label Cover instance $\mathcal{L} = (U, V, E, [R], [L], \phi)$ with a parameter $R > L$ and a parameter k, both of which we will set later, to a 4-uniform hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$. We also assume that $R - k$ is even. The reduction is given in Figure 2.

Vertices $\mathcal{V}$. Each vertex $x \in U$ in the Label Cover instance $\mathcal{L}$ is replaced by a cloud of size $|V_{SG}(R, k)|$ denoted by $C[x] := x \times V_{SG}(R, k)$. We refer to a vertex from the cloud $C[x]$ by a pair $(x, \mathbf{a})$, where $\mathbf{a} \in V_{SG}(R, k)$. The vertex set of the hypergraph $\mathcal{H}$ is given by

$$\mathcal{V} = \cup_{x \in U} C[x].$$

Hyperedges $\mathcal{E}$. The hyperedges of $\mathcal{H}$ are given by sets $\{(x, \mathbf{a}), (x, \mathbf{b}), (y, \mathbf{c}), (y, \mathbf{d})\}$ such that the following conditions hold:

1. There exists a common neighbor z of x and y in V.
2. For every $\alpha, \beta \in [R]$ such that $\phi_{x \to z}(\alpha) = \phi_{y \to z}(\beta)$, $\{\mathbf{a}_\alpha, \mathbf{b}_\alpha, \mathbf{c}_\beta, \mathbf{d}_\beta\}$ are not all the same i.e. $|\{\mathbf{a}_\alpha, \mathbf{b}_\alpha, \mathbf{c}_\beta, \mathbf{d}_\beta\}| = 2$.

Figure 2: Reduction to a 4-uniform hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$

The completeness is easy to prove:

Lemma 3.1 (Completeness). *If the Label Cover instance is satisfiable then the hypergraph $\mathcal{H}$ is 2-colorable.*

Proof: Let $A : U \cup V \to [R] \cup [L]$ be the perfectly satisfiable labeling to the Label Cover instance $\mathcal{L}$. The 2-coloring of the hypergraph $\mathcal{H}$ is given by assigning a vertex $(x, \mathbf{a}) \in C[x]$ with a color $\mathbf{a}_{A(x)}$. We show that this is a valid 2-coloring of $\mathcal{H}$. Suppose not, in which case, there exists a monochromatic hyperedge. Let that hyperedge be $\{(x, \mathbf{a}), (x, \mathbf{b}), (y, \mathbf{c}), (y, \mathbf{d})\}$. Let z be the common neighbor responsible for adding this edge. It must be that $\phi_{x \to z}(A(x)) \neq \phi_{y \to z}(A(y))$ and hence A is not a perfectly satisfiable assignment, which is a contradiction. ■

We now show the soundness of the reduction.

Lemma 3.2 (Soundness). *If the Label Cover instance is not $\frac{1}{(k+1)k^2}$ satisfiable then the hypergraph is not even $k + 1$-colorable.*

Proof: In this case, we show that if the hypergraph is $(k + 1)$-colorable, then there exists a labeling to the Label Cover instance that satisfies at least $\frac{1}{(k+1)k^2}$ fraction of the constraints between the two layers U and V.

Suppose the hypergraph is $(k + 1)$-colorable. Fix a $k + 1$ coloring χ of the hypergraph. Consider the Schrijver Graph $SG(R, k)$ defined on the cloud $C[x]$ where $x \in U$. By Theorem 2.5, there exists a monochromatic edge i.e $\chi((x, \mathbf{a})) = \chi((x, \mathbf{b}))$ where $(\mathbf{a}, \mathbf{b}) \in E_{SG}(R, k)$. Label a cloud $C[x]$ (and hence the vertex x) $c \in [k + 1]$ if that edge is c-colored (breaking ties arbitrarily). Out of $|U|$ vertices, there exists at least $1/(k+1)$ fraction of the vertices with the same color. Let that color be c'.

From now on, we denote the c' colored vertices of U by U'. We know from the bi-regularity of the Label Cover instance that the total number of constraints between U' and V is at least $\frac{1}{k+1}$ fraction of the constraints between layers U and V. Thus, if we show that we can satisfy at least $\frac{1}{k^2}$ fraction of the constraints between U' and V, then we are done. This will satisfy $\frac{1}{(k+1)k^2}$ fraction of the constraints between layers U and V.

List-Labeling. We define the list-labeling A to the vertices in U' as follows: For $x \in U'$, in the induced Schrijver Graph on $C[x]$, there exists a monochromatic edge with color c'. Let that edge be $\{(x, \mathbf{a}), (x, \mathbf{b})\}$. Let $A(x) = [R] \setminus (\mathbf{a} \cup \mathbf{b})$.

Observation 3.1. *For every $x \in U'$, $|A(x)| = k$.*

We need a simple fact as follows:

Fact 3.3. *For every pairwise intersecting family $\mathcal{F} \subseteq \binom{[n]}{k}$, there exists $i \in [n]$ which is present in at least $1/k$ fraction of the sets in $\mathcal{F}$.*

Proof: Suppose not. Consider $A = \{a_1, a_2, \ldots, , a_k\} \in \mathcal{F}$ and the sub-families $\mathcal{A}_i = \{A' \in \mathcal{F} \mid a_i \in A'\}$ for $i \in [k]$. By assumption, we have $|\mathcal{A}_i| < |\mathcal{F}|/k$. By the union bound, $| \cup_i \mathcal{A}| \leq \sum_i |\mathcal{A}_i| < |\mathcal{F}|$. Thus, there exists $B \in \mathcal{F} \setminus (\cup_i \mathcal{A}_i)$ and $A \cap B = \emptyset$, a contradiction.

■

The following two claims finish the proof.

Claim 3.4. *For every $x, y \in U'$ that have a common neighbor z in V,*

$$\phi_{x \to z}(A(x)) \cap \phi_{y \to z}(A(y)) \neq \emptyset.$$

Proof: For contradiction, suppose there exist $x, y \in U'$ and a common neighbor z such that $\phi_{x \to z}(A(x)) \cap \phi_{y \to z}(A(y)) = \emptyset$. Let $\{(x, \mathbf{a}), (x, \mathbf{b})\}$ and $\{(y, \mathbf{c}), (y, \mathbf{d})\}$ be the monochromatic c'-colored edges that were used to define A on x and y respectively. Since, $\phi_{x \to z}(A(x)) \cap \phi_{y \to z}(A(y)) = \emptyset$, we know that $\phi_{x \to z}([R] \setminus (\mathbf{a} \cup \mathbf{b}))$ and $\phi_{y \to z}([R] \setminus (\mathbf{c} \cup \mathbf{d}))$ are disjoint. This means that for every $\alpha, \beta \in [R]$ such that $\phi_{x \to z}(\alpha) = \phi_{y \to z}(\beta)$, we have $\mathbf{a}_\alpha, \mathbf{b}_\alpha, \mathbf{c}_\beta, \mathbf{d}_\beta$, not all of which are 1 (this follows from $\mathbf{a}_\alpha, \mathbf{b}_\alpha$ not both equal to 1 as they form an edge in the Schrijver graph $SG(R, k)$). Also, not all of $\mathbf{a}_\alpha, \mathbf{b}_\alpha, \mathbf{c}_\beta, \mathbf{d}_\beta$ are 0, since otherwise $\alpha \in A(x)$ and $\beta \in A(y)$ and hence $\phi_{x \to y}(A(x)) \cap \phi_{y \to z}(A(y)) \neq \emptyset$. Thus, $\{(x, \mathbf{a}), (x, \mathbf{b}), (y, \mathbf{c}), (y, \mathbf{d})\}$ is a valid hyperedge in $\mathcal{H}$, which is monochromatic with color c'. This contradicts the fact that χ is a valid $k+1$ coloring of $\mathcal{H}$. ■

Now, consider the following randomized labeling B:

$$\forall z \in V, \qquad B(z) \leftarrow \operatorname*{\texttt{arg}\,max}_{i \in [L]} \left|\{x \mid (x \in U') \wedge (z \sim x) \wedge i \in \phi_{x \to z}(A(x))\}\right|$$

$$\forall x \in U', \qquad B(x) \leftarrow \text{ A uniformly random label from } A(x).$$

In other words, $B(z)$ for $z \in V$ is a label $i \in [L]$ which is the most common projection (w.r.t. $\phi_{x \to z}$) of the labels $A(x)$, where $x \sim z$ and $x \in U'$.

Claim 3.5. *The randomized labeling B satisfies at least $\frac{1}{k^2}$ fraction of the constraints between U' and V in expectation.*

Proof: Fix $z \in V$ and let $X = \{x_1, x_2, \ldots, x_t\}$ be the neighbors of z in U' which satisfy $B(z) \in \phi_{x_i \to z}(A(x_i))$. Thus, X constitutes at least $1/k$ fraction of z's neighbors in U' using Claim 3.4 and Fact 3.3. The edge (x_i, z) is satisfied by the labeling $(B(x_i), B(z))$ with probability $1/|A(x_i)| = 1/k$ using Observation 3.1. Thus, in expectation, the randomized labeling satisfies at least $1/k^2$ fraction of the constraints between z and U'. Finally, by linearity of expectation, B satisfies at least $1/k^2$ fraction of the constraints between U' and V. ■

Thus, the partial labeling B satisfies at least $\frac{1}{(k+1)k^2}$ fraction of the constraints between U and V in expectation. ■

3.1 Setting of parameters

Proof of Theorem 1.1: Starting with a 3-SAT instance of size n, we first reduce it to a Label Cover instance given by Theorem 2.2 with parameter $\ell = c_0 \log\log n$ for a large constant $c_0 > 1$, so that the soundness of the Label Cover instance is $2^{-\Omega(\ell)} \ll \Omega(\log^{-3} n)$. We will set $k = \log n$. Thus, the number of vertices in the hypergraph $\mathcal{H}$ is $N = n^{O(\ell)} \cdot 2^{O(\ell)\cdot k} = n^{O(\log\log n)}$ and the reduction runs in time $n^{O(\log\log n)}$. Therefore, assuming $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log\log n)})$, there is no polynomial-time algorithm to color a 2-colorable 4-uniform hypergraph on N vertices with $O\left(\frac{\log N}{\log\log N}\right)$ colors. ■

Proof of Theorem 1.2: Starting with a 3-SAT instance of size n, we first reduce it to a Label Cover instance given by Theorem 2.3 with parameter $\epsilon = (\log n)^{-4\delta}$ for a sufficiently small constant $\delta > 0$. Thus, the soundness of the Label Cover instance is $(\log n)^{-4\delta}$, the alphabet size is $|\Sigma| = 2^{(\log n)^{c'\delta}}$ and the size of the Label Cover instance is $n^{1+o(1)} \cdot (\log n)^{c''\delta}$ for some absolute constants $c', c'' > 1$. We will set $k = (\log n)^{\delta}$. Thus, the number of vertices in the hypergraph $\mathcal{H}$ is $N \le n^{1+o(1)} \cdot (\log n)^{c''\delta} \cdot |\Sigma|^k = n^{1+o(1)} \cdot (\log n)^{c''\delta} \cdot 2^{(\log n)^{4c'\delta+\delta}} = n^{1+o(1)}$, for small enough δ. Also, the overall reduction starting from an instance of 3-SAT of size n runs in time $\texttt{poly}(n)$. Therefore, it is NP-hard to color a 2-colorable 4-uniform hypergraph on N vertices with $\log^{\delta} N$ colors for some $\delta > 0$. ■

4 Conclusion

Long Code has been used extensively to prove inapproximability results. Many inapproximability results benefit from *puncturing/derandomization* of the Long Code constructions. Most of them are algebraic in nature *e.g.* Hadamard code, Split code, Short code etc. We made an attempt to find a puncturing of the Long Code which is more combinatorial in nature. We believe that such combinatorial puncturings might find new applications in the hardness of approximation as well as in other areas which use the Long Code or similar objects.

We conclude with a couple of obvious interesting open problems:

1. Can we show that coloring a 2-colorable 3-uniform hypergraph with $\omega(\log\log n)$ colors is NP-hard, or even $quasi$-NP-hard?
2. Can we go beyond $\texttt{poly}(\log n)$ coloring NP-hardness factor for coloring a c-colorable k-uniform hypergraph for some constants $c \geq 2$ and $k \geq 3$?

Acknowledgment: I would like to thank Irit Dinur for helpful discussions. I would also like to thank Subhash Khot and Rishi Saket for their invaluable comments on the draft. Finally, I would like to thank all the anonymous reviewers for their comments which helped in improving the presentation significantly.

References

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *Journal of the ACM*, 45(3):501–555, 1998. (Preliminary version in *33rd FOCS*, 1992).

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A new characterization of NP. *JACM*, 45(1):70–122, January 1998. (Preliminary version in *33rd FOCS*, 1992).

[Bár78] J Bárány. A short proof of Kneser's conjecture. *Journal of Combinatorial Theory, Series A*, 25(3):325–326, 1978.

[BG16] Joshua Brakensiek and Venkatesan Guruswami. New hardness results for graph and hypergraph colorings. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[DRS02] I Dinur, O Regev, and C Smyth. The hardness of 3-uniform hypergraph coloring. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 33–40. IEEE, 2002.

[FGL$^+$96] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM (JACM)*, 43(2):268–292, 1996.

[GHH$^+$14] Venkatesan Guruswami, Prahladh Harsha, Johan Håstad, Srikanth Srinivasan, and Girish Varma. Super-polylogarithmic Hypergraph Coloring Hardness via Low-degree Long Codes. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 614–623, New York, NY, USA, 2014. ACM.

[GHS02] Venkatesan Guruswami, Johan Håstad, and Madhu Sudan. Hardness of approximate hypergraph coloring. *SIAM Journal on Computing*, 31(6):1663–1686, 2002.

[GJ76] Michael R Garey and David S Johnson. The complexity of near-optimal graph coloring. *Journal of the ACM (JACM)*, 23(1):43–49, 1976.

[GK04] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. *SIAM Journal on Discrete Mathematics*, 18(1):30–40, 2004.

[Kho02a] Subhash Khot. Hardness results for approximate hypergraph coloring. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 351–359. ACM, 2002.

[Kho02b] Subhash Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 23–32. IEEE, 2002.

[KLS93] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. In *Theory and Computing Systems, 1993., Proceedings of the 2nd Israel Symposium on the*, pages 250–260. IEEE, 1993.

[KS14a] Subhash Khot and Rishi Saket. Hardness of Coloring 2-Colorable 12-Uniform Hypergraphs with exp (log^{Omega (1)} n) Colors. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 206–215. IEEE, 2014.

[KS14b] Subhash Khot and Rishi Saket. Hardness of finding independent sets in 2-colorable and almost 2-colorable hypergraphs. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1607–1625. SIAM, 2014.

[Lov78] László Lovász. Kneser's conjecture, chromatic number, and homotopy. *Journal of Combinatorial Theory, Series A*, 25(3):319–324, 1978.

[MR10] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *Journal of the ACM (JACM)*, 57(5):29, 2010.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[Sak14] Rishi Saket. Hardness of finding independent sets in 2-colorable hypergraphs and of satisfiable CSPs. In *IEEE 29th Conference on Computational Complexity (CCC)*, pages 78–89, 2014.

[Sch78] A Schrijver. Vertex-critical subgraphs of Kneser-graphs. *Nieuw Archief voor Wiskunde*, 26:454–461, 1978.

[Var15] Girish Varma. Reducing uniformity in Khot-Saket hypergraph coloring hardness reductions. *Chicago Journal OF Theoretical Computer Science*, 3:1–8, 2015.