The Hardness of 3-Uniform Hypergraph Coloring

Irit Dinur * Oded Regev * Clifford Smyth *

November 23, 2002

Abstract

We prove that coloring a 3-uniform 2-colorable hypergraph with any constant number of colors is NP-hard. The best known algorithm [20] colors such a graph using $O(n^{1/5})$ colors. Our result immediately implies that for any constants $k > 2$ and $c_2 > c_1 > 1$, coloring a k-uniform c_1-colorable hypergraph with c_2 colors is NP-hard; leaving completely open only the $k = 2$ graph case.

We are the first to obtain a hardness result for approximately-coloring a 3-uniform hypergraph that is colorable with a constant number of colors. For $k \geq 4$ such a result has been shown by [14], who also discussed the inherent difference between the $k = 3$ case and $k \geq 4$.

Our proof presents a new connection between the Long-Code and the Kneser graph, and relies on the high chromatic numbers of the Kneser graph [19, 22] and the Schrijver graph [26]. We prove a certain maximization variant of the Kneser conjecture, namely that any coloring of the Kneser graph by fewer colors than its chromatic number, has 'many' monochromatic edges.

1 Introduction

Background

A hypergraph $H = (V, E)$ with vertices V and edges $E \subset 2^V$ is 3-uniform if every edge in E has exactly 3 vertices. A legal χ-coloring of a hypergraph H is a function $f : V \to [\chi]$ such that no edge of H is monochromatic. The chromatic number of H is the minimal χ for which such a coloring exists.

During the past decade, significant progress has been made – via the PCP theorem – in understanding the complexity of many combinatorial optimization problems. It is known, for example, that it is hard to approximate the chromatic number of a graph to within factor of $n^{1-\epsilon}$ [9] (this holds for hypergraphs as well, see [21]). In numerous other cases the hardness lower bound almost matches the algorithmic upper bound, or perhaps some constant gap separates between the two. In contrast, the problem of approximate coloring (where the input is a hypergraph or a graph that is known to be colorable with *very* few colors and the target is to color it with as few colors as possible) retains perhaps the largest gap between lower and upper bounds.

The best algorithms for these problems require a polynomial number of colors: for example the best approximate coloring algorithm for 2-colorable 3-uniform hypergraphs requires $O(n^{1/5})$ colors [20], and the best coloring algorithm for 3-colorable graphs, requires $\tilde{O}(n^{3/14})$ colors [5].

*Institute for Advanced Study, Princeton, NJ. E-Mail: {iritd,odedr,csmyth}@ias.edu. Research supported by NSF grant CCR-9987845.

On the lower-bound side, not much is known. For graphs, the best hardness result states that using 4 colors to color a 3-colorable graph is NP-hard [17, 13]. It would already be a significant step to prove that coloring a 3-colorable graph with $O(1)$ colors is NP-hard.

The property of being 2-colorable is well studied in combinatorics and is also referred to as 'property B' (see [14] for further references). Nevertheless, prior to this work no hardness of approximation result was known for 3-uniform hypergraphs, and in fact it wasn't even known if it is NP-hard to color a 3-uniform 2-colorable hypergraph with 3 colors. For 4-uniform hypergraphs and upwards, Guruswami, Håstad, and Sudan [14] were able to show such a separation, i.e., they showed that it is NP-hard to color a 2-colorable 4-uniform hypergraph with any constant number of colors (this result immediately extends to any $k \geq 4$). In their work an inherent difference between the $k \geq 4$ and $k = 2, 3$ was raised, which was considered evidence that the case of $k = 3$ would be harder to understand. This difference has to do with the corresponding maximization problem called 'Set-Splitting', which is the problem of 2-coloring a k-uniform hypergraph while maximizing the number of non-monochromatic hyperedges. This problem exhibits a different behavior for $k \geq 4$ and for $k = 2, 3$. Indeed, for $k \geq 4$ there is a tight hardness result by Håstad [16] around $1 - 2^{-k+1}$ which is trivially matched by a random assignment. For $k = 2, 3$ the best approximation algorithms use semi-definite programming [11, 27] and have a constant gap from the best gadget-constructed hardness results, see [15].

Guruswami, Håstad, and Sudan [14] also showed that unless $NP \subseteq DTIME(n^{O(\log\log n)})$, there is no polynomial-time algorithm for coloring a 2-colorable 4-uniform hypergraph with $c_0 \frac{\log\log n}{\log\log\log n}$ colors for some constant $c_0 > 0$. We can show that unless $NP \subseteq DTIME(2^{poly(\log n)})$, there is no polynomial-time algorithm for coloring a 2-colorable 3-uniform hypergraph with $O(\sqrt[3]{\log\log n})$ colors. These results again extend to any $k \geq 3$.

Following our result, Khot [18] was able to prove, via different techniques yet relying on a similar layered PCP, that it is NP-hard to color a 3-colorable 3-uniform hypergraph with any constant number of colors. Although this result is already contained in ours, his construction has the extra nice property that the bad instances do not contain even a small independent set, while in our hypergraph construction, this is not the case.

Technique

The general structure of many hardness results has two main components. First, one takes a version of the PCP theorem [2, 1] (usually, after an application of the parallel repetition theorem of [25]) and applies some transformations on it to construct a PCP system with certain desired properties. Then, one applies a version of the Long-Code, tailored to the specific problem whose hardness is being studied. The 'trick' is to obtain the correct interplay between the specific PCP properties and the Long-Code variant, so as to capture the hardness of the problem. The proof of correctness divides into two parts: completeness and soundness. First one proves completeness, i.e., that if the initial PCP system was a 'yes' instance then the resulting construction is also a 'yes' instance (in our case: a 2-colorable 3-uniform hypergraph), which generally follows from the construction in a straightforward manner. Then one proves soundness, i.e., that if the initial instance was a 'no' instance then the resulting construction is also a 'no' instance (in our case: that the resulting graph has a high chromatic number). The proof of soundness, which is where the heart of the proof really lies, integrates the two components of the construction. First one applies some sort of 'list-decoding' to the Long-Code, which in our case means to translate an arbitrary coloring of the Long-Code with few colors into a short list of possible 'decoded-values'. Then one exploits the special PCP properties to ensure global consistency between these decoded values. In what follows we describe the two components of our construction, and how they are combined together.

The Layered PCP. Let us focus on PCP systems whose tests look at two variables (corresponding to 2-prover interactive proofs). In general, such PCP systems can be divided into those that have one type of variables, X, and those that have two types of variables, X and Y. PCP systems of the first kind have tests that look at two X-variables, $x_1, x_2 \in X$, and each value of $x_1 \in X$ still allows many values for $x_2 \in X$. This phenomenon does not go well with an application of the Long-Code, since encodings of these variables appear falsely consistent. PCP systems of the second kind, e.g., the parallel repetition theorem [25], have each test look at one 'large' X-variable whose value completely determines that of the 'small' Y-variable, and whose tests are essentially projection tests. Such tests do not suffer from the false-consistency problem but have an inherent 'bipartite-ness' problem in our coloring setting: coloring all the X variables red, and all the Y variables orange, will be a legal coloring regardless of whether the initial PCP system was satisfiable or not. We take the second approach, and overcome the 'bipartite-ness' problem by utilizing a layered-PCP that was constructed in [6]. This essentially extends the bipartite nature of Raz's PCP into being *multipartite*, with many 'types' of variables (rather than only X and Y). Thus, as the number of parts increases, the number of colors required to color an unsatisfiable instance increases as well.

The Long-Code and the Kneser Graph. The Long-Code [4] of a domain R is a powerful tool in numerous hardness of approximation results. We adopt a completely combinatorial viewpoint, following [7], and discover a new connection between the Long-Code and a well-known combinatorial object called the Kneser graph [19].

The Long-Code of a domain R consists of all possible subsets[1] of R. The *Long-Code Graph*, explicitly defined in [7], is the graph whose $2^{|R|}$ vertices are the subsets of R, and whose edges connect disjoint subsets. Let us consider the following encoding of an element $a \in R$: color all subsets containing a red, and the rest orange. This coloring corresponds to the *legal-encoding* of $a \in R$ via the Long-Code. It is easy to check that *in the Long-Code Graph* this coloring has no *red* monochromatic edges, yet has many orange ones, and in particular it is not a 2-coloring. Nevertheless, in our *hypergraph* this coloring corresponds to a legal 2-coloring. We prove the following interesting property of the Long-Code graph. Namely, that any coloring of the vertices by a constant number of colors contains one 'special' color that is used in a non-negligible fraction of the vertices and colors two large disjoint subsets. This property enables a list-decoding of the coloring in a manner that ensures global consistency. The property is established by proving a maximization variant of the Kneser conjecture showing that using less than the chromatic number colors to color the Kneser graph, leaves 'many' monochromatic edges. We use properties of both the Kneser graph and the Schrijver graph, which appear as subgraphs in the Long-Code graph.

[1] In the original and standard definition of the Long-Code [4], each subset of R is viewed as a Boolean function $f : R \to \{0, 1\}$; and the Long-Code is the collection of all possible Boolean functions over R – hence the name.

Combining the Two Components. The vertices of our constructed 3-uniform hypergraph will be partitioned into blocks whose vertices are roughly the vertices of the Long-Code graph. Every hyperedge will contain both ends of an edge of the Long-Code graph. This guarantees that a global 2-coloring that within a block is a legal-encoding, will never contain *red* monochromatic hyperedges. The hyperedge will contain an additional vertex always from another block, that will ensure 'inter-block consistency' intuitively via the requirement of not having any *non-red* monochromatic hyperedges.

In the soundness proof we will need to translate a legal coloring of the hypergraph (with some $\chi > 2$ colors) within each block into a short list of elements of R that are supposedly encoded by that coloring. The main difficulty arises from the need of ensuring consistency between the decoded values of distinct blocks. We will do so by highlighting one color per block that 'highly violates' the edges of the Long-Code graph, in a certain sense that will be clarified later on.

2 The Long Code and the Kneser Graph

The Long-Code graph of a domain R, explicitly defined in [7], is the graph whose $2^{|R|}$ vertices are the subsets of R, and whose edges connect disjoint subsets. In this work we analyze colorings of this graph using a constant number of colors. We discover that the induced subgraph of the Long-Code graph obtained by taking vertices that correspond to subsets of size $s = \frac{|R|}{2} - c$ for some constant c, is no other than the Kneser graph [19], whose chromatic number is well-studied. In this section we define the Kneser graph and describe some of its important properties.

For $n \geq 2s - 1$, the Kneser graph $KG_{n,s}$ has the set $\binom{[n]}{s} \stackrel{def}{=} \{S \subseteq [n] \mid |S| = s\}$ as its vertex set and two vertices S_1, S_2 are adjacent iff $S_1 \cap S_2 = \phi$. In other words, each vertex corresponds to a s-set and two vertices are adjacent if their corresponding sets are disjoint. In this paper we are mainly interested in the case where s is smaller than $n/2$ by a constant. These graphs have the important property that the chromatic number is high although large independent sets exist. For a discussion of Kneser graphs and other combinatorial problems, see an excellent book by Matoušek [24].

There exists a simple way to color this graph with $n - 2s + 2$ coloring. In 1955, Kneser [19] conjectured that there is no way to color the graph with less colors, i.e., $\chi(KG_{n,s}) = n - 2s + 2$. The first to prove this conjecture was Lovász in 1978 [22]. Many other proofs and extension are known (see, e.g., [8, 3, 23]) and the latest and simplest one is by Greene [12].

We also define the Schrijver graph $SG_{n,s,\pi}$. Given a permutation π of $[n]$, we say that an s-subset $S \in \binom{[n]}{s}$ is *π-stable* if it does not contain two π-adjacent elements modulo n, i.e., if $\pi(i) \in S$ then $\pi(i+1) \notin S$ and if $\pi(n) \in S$ then $\pi(1) \notin S$. We denote the number of stable s-sets by $\binom{n}{s}_{stab}$ (notice that it is independent of π). The graph $SG_{n,s,\pi}$ contains a vertex for each π-stable s-subset of $[n]$ and two vertices are adjacent if their corresponding sets are disjoint. Clearly, $SG_{n,s,\pi}$ is an induced subgraph of $KG_{n,s}$. Interestingly, the chromatic of the Schrijver graph is the same, i.e., $\chi(SG_{n,s,\pi}) = n - 2s + 2$ [26].

We begin with a simple bound on the number of vertices in a Schrijver graph:

Claim 2.1 $\binom{n}{s}_{stab} \leq \binom{n}{n-2s}$

Proof: Consider a stable set S according to the permutation $\pi(i) = i$. Notice that the set of all $i \in [n]$ such that $i \notin S$ and $i-1 \notin S$ (or $n \notin S$ if $i = 1$) uniquely defines S. The claim follows from the fact that this is always a set of $n - 2s$ elements. ■

The following is the main lemma of this section:

Lemma 2.2 *In any coloring of $KG_{n,s}$ by $n-2s+1$ colors there exists a monochromatic edge whose color is used in at least a $\frac{2}{(n-2s+1)\binom{n}{s}_{stab}}$-fraction of the vertices.*

Proof: Fix a coloring of $KG_{n,s}$ by $n-2s+1$ colors. For every permutation π, the induced subgraph $SG_{n,s,\pi}$ contains a monochromatic edge since $\chi(SG_{n,s,\pi}) = n - 2s + 2$. Therefore, there exists a color, say red, such that at least $\frac{1}{n-2s+1}$ of the Schrijver graphs contain a red monochromatic edge. Consider the following distribution on the vertices of $KG_{n,s}$: choose a random permutation π and then a random vertex in $SG_{n,s,\pi}$. The probability of choosing a red vertex is at least $\frac{1}{n-2s+1} \cdot \frac{2}{\binom{n}{s}_{stab}}$ since we first have to choose a Schrijver graph that contains a red monochromatic edge and then one of the two end points of the monochromatic edge. Since each $SG_{n,s,\pi}$ contains the same number of vertices and each vertex of $KG_{n,s}$ is contained in the same number of $SG_{n,s,\pi}$ this distribution is equivalent to the uniform distribution on the vertices of $KG_{n,s}$. Therefore, the fraction of red vertices is at least $\frac{2}{(n-2s+1)\binom{n}{s}_{stab}}$. ■

Corollary 2.3 *In any coloring of $KG_{n,s}$ by $n-2s+1$ colors there exists a color, say red, for which*

1. *The subset of red vertices contains at least a $\frac{2}{(n-2s+1)\binom{n}{n-2s}}$-fraction of the vertices in the graph.*
2. *There are two red vertices S_1, S_2 such that $S_1 \cap S_2 = \phi$.*

3 The Layered PCP

We use the layered PCP construction of Dinur et al. [6]. For completeness, we describe the construction and sketch the proof of its NP-hardness. In an l-layered PCP there are l sets of variables which we call layers and denote by $X_1, \ldots, X_l$. The range of variables in X_i is denoted R_i. For every $1 \leq i < j \leq l$ there is a set of tests Φ_{ij} where each test $\varphi \in \Phi_{ij}$ depends on exactly one $x \in X_i$ and one $y \in X_j$. For any two variables we denote by $\varphi_{x \to y}$ the test between them if such a test exists. Moreover, the tests in Φ_{ij} are projections from x to y, that is, for every assignment to x there is exactly one assignment to y such that the test accepts.

Theorem 3.1 *For any parameters l, u there exists a reduction from an NP-hard problem of size n to the problem of distinguishing between the following two cases in an l-layered PCP Φ with $n^{O(ul)}$ variables over a range of size $2^{O(ul)}$. Either there exists an assignment that satisfies all the tests or, for every $i < j$, not more than $2^{-\Omega(u)}$ of the tests in Φ_{ij} can be satisfied by an assignment. Moreover, for any constant $\delta > 0$ let $m = \lceil \frac{2}{\delta} \rceil$. Then, for any m layers $i_1 < \ldots < i_m$ and sets $S_j \subseteq X_{i_j}$ for $j \in [m]$ such that $S_j \geq \delta |X_{i_j}|$ we can find two sets S_j and $S_{j'}$ such that the number of tests between them is at least $\frac{\delta^2}{4}$ of the tests between the layers X_{i_j} and $X_{i_{j'}}$.*

Proof Sketch: We begin with the parallel repetition lemma [25] applied with parameter u to the 3-SAT-5 of [1]. This provides us with a reduction from an NP-hard problem of size n to a PCP Ψ over two sets of variables Y (with range R_Y of size 7^u) and Z (with range R_Z of size 2^u) where the number of variables is $n^{O(u)}$. The tests $\psi_{y,z}$ are only between a variable $y \in Y$ and a variable $z \in Z$ and are projections from R_Y to R_Z, i.e., for any given assignment to $y \in Y$ there exists exactly one consistent assignment to $z \in Z$. The problem is to decide whether there exists an assignment that satisfies all the tests or no assignment satisfies more than $2^{-\Omega(u)}$ fraction of the tests. One property of the resulting PCP that we use is its uniformity: the distribution created by uniformly taking a variable $y \in Y$ and then uniformly choosing one of the variables in $z \in Z$ with which it has a test is a uniform distribution on Z.

We construct Φ as follows. The variables X_i of layer $i \in [l]$ are the elements of the set $Z^i Y^{l-i}$, i.e., all l-tuples where the first i elements are Z variables and the last $l-i$ elements are Y variables. The variables in layer i have assignments from the set $R_i = R_Z^i R_Y^{l-i}$ corresponding to an assignment to each variable of Ψ in the l-tuple. It is easy to see that $|R_i| \leq 2^{O(ul)}$ for any $i \in [l]$ and that the total number of variables is $n^{O(ul)}$. For any $1 \leq i < j \leq l$ we define the tests in Φ_{ij} as follows. A test exists between a variable $x_i \in X_i$ and a variable $x_j \in X_j$ if they contain the same Ψ variables in the first i and the last $l-j$ elements of their l-tuples. Moreover, for any $i < k \leq j$ there should be a test in Ψ between $x_{i,k}$ and $x_{j,k}$. More formally,

$$\Phi_{ij} = \{\varphi_{x_i,x_j} \mid x_i \in X_i, x_j \in X_j, \forall k \in [l] \setminus \{i+1,\ldots,j\}$$
$$x_{i,k} = x_{j,k}, \forall k \in \{i+1,\ldots,j\}\ \psi_{x_{i,k},x_{j,k}} \in \Psi\}.$$

As promised, the tests φ_{x_i,x_j} are projections. Given as assignment a to x_i, we define the consistent assignment b to x_j as $b_k = \psi_{x_{i,k},x_{j,k}}(a_k)$ for $k \in \{i+1,\ldots,j\}$ and $b_k = a_k$ otherwise.

The completeness of Φ follows easily from the completeness of Ψ. That is, assume we are given an assignment $A : Y \cup Z \to R_Y \cup R_Z$ that satisfies all the tests. Then, the assignment $B : \bigcup X_i \to \bigcup R_i$ defined by $B(x_1,\ldots,x_l) = (A(x_1),\ldots,A(x_l))$ is a satisfying assignment. For the soundness part, assume that there exist two layers $i < j$ and an assignment B that satisfies more than $2^{-\Omega(u)}$ of the tests in Φ_{ij}. We partition X_i into classes such that two variables in X_i are in the same class iff they are identical except possibly on coordinate j. The variables in X_j are also partitioned according to coordinate j. Since more than $2^{-\Omega(u)}$ of the tests in Φ_{ij} are satisfied, it must be the case that there exist a class $x_{i,1},\ldots,x_{i,j-1},x_{i,j+1},\ldots,x_{i,l}$ in the partition of X_i and a class $x_{j,1},\ldots,x_{j,j-1},x_{j,j+1},\ldots,x_{j,l}$ in the partition of X_j between which there exist tests and the fraction of satisfied tests is more than $2^{-\Omega(u)}$. We define an assignment to Ψ as $A(y) = (B(x_{i,1},\ldots,x_{i,j-1},y,x_{i,j+1},\ldots,x_{i,l}))_j$ for $y \in Y$ and as $A(z) = (B(x_{j,1},\ldots,x_{j,j-1},z,x_{j,j+1},\ldots,x_{j,l}))_j$ for $z \in Z$. Notice that there is a one-to-one and onto correspondence between the tests in Ψ and the tests between the two chosen classes in Φ. Moreover, if the test is Φ is satisfied, then the test in Ψ is also satisfied. Therefore, A is an assignment to Ψ that satisfies more than $2^{-\Omega(u)}$ of the tests.

To prove the second part of the theorem take any m layers $i_1 < \ldots < i_m$ and sets $S_j \subseteq X_{i_j}$ for $j \in [m]$ such that $S_j \geq \delta|X_{i_j}|$. Consider a random walk beginning from a uniformly chosen variable $x_1 \in X_1$ and proceeding to a variable $x_2 \in X_2$ chosen uniformly among the variables with which x_1 has a test. The random walk continues in a similar way to a variable $x_3 \in X_3$ chosen uniformly among the variables with which x_2 has a test and so on up to a variable in X_l. Denote by E_j the indicator variable of the event that the random walk hits an S_j variable when in layer X_{i_j}. From the uniformity of Ψ it follows that for every j, $P(E_j) \geq \delta$. Moreover, by using the inclusion-exclusion principle, we get:

$$1 \geq P(\bigvee E_j) \geq \sum_j P(E_j) - \sum_{j<k} P(E_j \wedge E_k) \geq 2 - \binom{m}{2} \max_{j<k} P(E_j \wedge E_k)$$

which implies

$$\max_{j<k} P(E_j \wedge E_k) \geq 1/\binom{m}{2} \geq \frac{\delta^2}{4}.$$

Fix j and k such that $P(E_j \wedge E_k) \geq \frac{\delta^2}{4}$ and consider a shorter random walk beginning from a random variable in X_{i_j} and proceeding to the next layer and so on until hitting layer i_k. Since E_j is uniform on X_{i_j} we still have that $P(E_j \wedge E_k) \geq \frac{\delta^2}{4}$ where the probability is taken over the random walks between X_{i_j} and X_{i_k}. Also, notice that there is a one-to-one and onto mapping from the set of all random walks between X_{i_j} and X_{i_k} to the set Φ_{i_j,i_k}. Therefore, $\frac{\delta^2}{4}$ of the tests between X_{i_j} and X_{i_k} are between S_j and S_k. ■

4 The Hypergraph Construction

Theorem 4.1 (Main Theorem) *For any constant $\chi \geq 2$, it is NP-hard to color a 2-colorable 3-uniform hypergraph using χ colors.*

Proof: Let Φ be a PCP instance with layers $X_1, .., X_l$, as described in Theorem 3.1, with parameters $l = 2\chi^2$, and u to be chosen later. We present a construction of a 3-uniform hypergraph $G = (V, E)$.

Vertices. For each variable x in layer X_i we construct a block of vertices $V[x]$, which is essentially a variant of the Long-Code of x's range R_i. This block contains a vertex for each subset of R_i of size $s_i \overset{def}{=} \lfloor (|R_i| - \chi)/2 \rfloor$, i.e., $V[x] = \binom{R_i}{s_i}$. Altogether,

$$V = \bigcup_{x \in \cup X_i} V[x].$$

Throughout this section we slightly abuse notation by writing a vertex rather than the subset it represents.

Hyperedges. We construct hyperedges between blocks $V[x]$ and $V[y]$ only if there exists a test $\varphi_{x \to y} \in \Phi$. We put a hyperedge between any $v_1, v_2 \in V[x]$ and $u \in V[y]$ whenever $v_1 \cap v_2 = \phi$ and $\varphi_{x \to y}(R_i \setminus (v_1 \cup v_2)) \subseteq u$. In summation,

$$E = \bigcup_{\varphi_{x \to y} \in \Phi} \{\{v_1, v_2, u\} \mid v_1, v_2 \in V[x],\ u \in V[y],\ v_1 \cap v_2 = \phi \text{ and } \varphi_{x \to y}(R_i \setminus (v_1 \cup v_2)) \subseteq u\}.$$

The condition $v_1 \cap v_2 = \phi$ (in the terms of Section 2, that the hyperedge contain both ends of an edge from the Long-Code Graph) guarantees that any coloring of G that within one $V[x]$

is a legal-encoding of an assignment a_x to x (i.e., that colors all subsets containing a_x red and the rest orange), automatically contains no red monochromatic hyperedges. The second condition, $\varphi_{x\to y}(R_i \setminus (v_1 \cup v_2)) \subseteq u$ guarantees consistency between the encodings of the blocks of x and y in the following way. Intuitively, if both v_1 and v_2 are colored orange, and assuming the adversary colors each block with a legal-encoding, then the encoded assignment must lie outside both v_1 and v_2, i.e., in $R \setminus (v_1 \cup v_2)$. Hence, any set u containing the image under $\varphi_{x\to y}$ of this set must be colored red. A more accurate analysis is in the soundness proof.

Note that our hyperedges are always between two layers of the PCP and are 'directed' in the sense that they have two vertices from block of the bigger variable x, and one vertex in the block of the smaller variable y.

Remark. Interestingly, it is easy to see that this construction has rather large independent sets no matter what the underlying PCP is. For example, we can choose from each block all the vertices that contain a certain assignment, say the first one. Moreover, it is possible to construct two independent sets (i.e., two colors) that cover almost all of the hypergraph, leaving out only a sub-constant part. We briefly sketch this example which originally appeared in [10]. Consider any $x \in X_i$, $i \in [l]$ and let T_i be any subset of R_i of size $\frac{|R_i|}{2}$. The first independent set contains all the vertices in $V[x]$ whose intersection with T_i is of size more than $\frac{|T_i|}{2}$. Similarly, the second independent set contains all the vertices in $V[x]$ whose intersection with $R_i \setminus T_i$ is more than $\frac{|T_i|}{2}$. These are indeed independent sets because two vertices in the first independent set intersect on T_i and similarly for the second independent set. Also, most of the vertices in $V[x]$ are in one of the independent sets. Informally, this is true because the size of the intersection of a vertex in $V[x]$ and T_i has a standard deviation of $\Theta(\sqrt{|R_i|})$.

Lemma 4.2 (Completeness) *If Φ is satisfiable then G is 2-colorable.*

Proof: Let A be a satisfying assignment for Φ, i.e., A maps each variable $x \in X_i$ to an assignment in R_i such that all the tests are satisfied. In the block $V[x]$ we color all the vertices that contain the assignment $A(x)$ red and all the rest orange. There are no red monochromatic edges because two red vertices inside the same block have a non-empty intersection. Also, let $\{v_1, v_2, u\}$ be any hyperedge and choose x, y, i such that $v_1, v_2 \in V[x]$, $x \in X_i$ and $u \in V[y]$. Assume that both v_1 and v_2 are orange. Since there exists a hyperedge between them, v_1 and v_2 are disjoint. Therefore, $A(x) \in R_i \setminus (v_1 \cup v_2)$ which implies that $\varphi_{x\to y}(A(x)) \in \varphi_{x\to y}(R_i \setminus (v_1 \cup v_2)) \subseteq u$ where the last containment is again because $\{v_1, v_2, u\}$ is a hyperedge. But $\varphi_{x\to y}(A(x)) = A(y)$ since A is a satisfying assignment and therefore $A(y) \in u$ and u is red. ∎

Lemma 4.3 (Soundness) *If G is χ-colorable then Φ is satisfiable.*

Proof: Fix a coloring of the graph G. The first step in our proof is to 'list-decode' this coloring, i.e., to find a list of candidate-assignments for each variable. For any variable $x \in X_i$, $i \in [l]$, consider the vertices inside the block $V[x]$. We can map them to the vertices of the Kneser graph $KG_{|R_i|,\lfloor(|R_i|-\chi)/2\rfloor}$. According to Corollary 2.3 we can find a color c_x with the following double property:

1. The subset $U[x] \subset V[x]$ of vertices colored c_x contains at least an $\Omega(|R_i|^{-(\chi+1)})$ fraction of the vertices in $V[x]$.

2. There are two vertices $v_{x,1}, v_{x,2} \in U[x]$ such that $v_{x,1} \cap v_{x,2} = \phi$.

Let $B(x)$ be the set $R_i \setminus (v_{x,1} \cup v_{x,2})$. This is the list of 'decoded' values. Notice that it contains at most $\chi + 1$ assignments. We say that the variable x is *colored with* c_x, and denote as above by $U[x] \subseteq V[x]$ the set of vertices $V[x]$ colored with c_x.

The next step in the proof is to establish consistency. Define the color of a layer $i \in [l]$ to be the color in which most of its variables are colored. Notice that at least $\frac{1}{\chi}$ of the variables of the layer must be colored with the color of the layer. Finally, we can find $\frac{l}{\chi} = 2\chi$ layers that are colored with the same color, say red. Using the properties of the PCP with the set of red layers and the red variables within each layer, we conclude that there exist two layers X_i and X_j such that $\frac{1}{4\chi^2}$ of the tests between them are tests between red variables. Let us denote by X the red variables in X_i and by Y the red variables in X_j.

In the following we define an assignment to the variables in X and Y such that many of the tests between them are satisfied. For a variable $x \in X$ we choose a random assignment from the set $B(x)$. For a variable $y \in Y$ we choose the assignment

$$A(y) = \text{maxvar}_{a \in R_Y} |\{x \in X \mid \varphi_{x \to y} \in \Phi \text{ and } a \in \varphi_{x \to y}(B(x))\}|,$$

i.e., the assignment that is contained in the largest number of projections of $B(x)$.

In order to prove that this assignment satisfies a good fraction of the tests, we need the following simple claim:

Claim 4.4 *Let $A_1, \dots, A_n$ be a collection of n sets of size at most m such that no element is contained in more than k sets. Then, there are at least $\frac{n}{1+(k-1)m} \geq \frac{n}{km}$ disjoint sets in this collection.*

Proof: We prove by induction on n that there are at least $\frac{n}{1+(k-1)m}$ disjoint sets in the collection. The claim holds trivially for $n \leq 1 + (k-1)m$. Otherwise, consider all the sets that intersect A_1. Since no element is contained in more than k sets, the number of such sets (including A_1) is at most $1 + (k-1)m$. Removing these sets we get, by using the induction hypothesis, a collection that contains $\frac{n-1-(k-1)m}{1+(k-1)m} = \frac{n}{1+(k-1)m} - 1$ disjoint sets. We conclude the proof by adding A_1 to the disjoint sets. ■

Consider a variable $y \in Y$ and a variable x such that the test $\varphi_{x \to y}$ exists. Since the vertices $v_{x,1}, v_{x,2}$ and the vertices in $U[y]$ are colored red, there are no hyperedges between them. Therefore, all the vertices $u \in U[y]$ must not contain $\varphi_{x \to y}(B(x))$. Now consider the family of projections $\varphi_{x \to y}(B(x))$ for all the variables x such that the test $\varphi_{x \to y}$ exists. Let q denote the number of disjoint sets in this family. The following claim provides an upper bound on q:

Claim 4.5 *Let $A_1, \dots, A_q \in \binom{[n]}{s}$ be disjoint. Let $\mathcal{F} = \{F \in \binom{[n]}{k} \mid \forall i \; F \cap A_i \neq \emptyset\}$. Then if $\frac{n}{3} \leq k \leq \frac{2n}{3}$ we have*

$$q \leq 3^s \left[\frac{1}{2} \log n + c_0 - \log \left(|\mathcal{F}| / \binom{n}{k} \right) \right]$$

where c_0 is an absolute constant.

Proof: Consider the probability distribution μ on $2^{[n]}$ where each element $i \in [n]$ is chosen to be in the set with probability $\frac{k}{n}$ and out of it with probability $1 - \frac{k}{n}$. Thus, $\mu(F) = (\frac{k}{n})^{|F|}(1 - \frac{k}{n})^{n-|F|}$

and for $\mathcal{F} \subset 2^{[n]}$ we define $\mu(\mathcal{F}) = \sum_{F \in \mathcal{F}} \mu(F)$. Since all k-sets appear with equal probability, we have

$$|\mathcal{F}| / \binom{n}{k} = \mu(\mathcal{F}) / \mu\left(\binom{[n]}{k}\right).$$

¿From Stirling's formula we get $n! = \Theta(n^n e^{-n}\sqrt{n})$ and hence $\mu(\binom{[n]}{k}) = \binom{n}{k}(\frac{k}{n})^k(1-\frac{k}{n})^{n-k} = \Theta(\sqrt{n}/\sqrt{k(n-k)})$ which for $\frac{n}{3} \le k \le \frac{2n}{3}$ equals $\Theta(1/\sqrt{n})$. Since $\mu(\mathcal{F}) \le (1-(1-\frac{k}{n})^s)^q \le \exp(-q(1-\frac{k}{n})^s)$, we have $|\mathcal{F}|/\binom{n}{k} = \Theta(\sqrt{n}\exp(-q(1-\frac{k}{n})^s))$. Taking logarithms we get $q(1-\frac{k}{n})^s \le \frac{1}{2}\log(n) + c_0 - \log(|\mathcal{F}|/\binom{n}{k})$ for some constant $c_0 > 0$. Since $\frac{n}{3} \le k \le \frac{2n}{3}$ this implies the conclusion of the claim. ■

We use this claim with the family $U[y]$ for which $\log(\frac{|U[y]|}{|V[y]|}) \ge -(\chi+1)\log(|R_Y|)$. Therefore, we get that $q \le 3^{\chi+1}(\frac{1}{2}\log|R_Y| + c_0 + (\chi+1)\log(|R_Y|)) = O(\chi 3^\chi \log(|R_Y|)) = O(\chi 3^\chi ul)$. Claim 4.4 implies that there exists an assignment for y that is contained in at least a fraction $\Omega(\chi^{-1}3^{-\chi}(ul)^{-1}(\chi+1)^{-1})$ of the projections $\varphi_{x \to y}(B(x))$. Therefore, since we randomly choose one out of $\chi+1$ possible assignments to each $x \in X$, the expected fraction of tests satisfied between X and Y is at least $\Omega(\chi^{-3}3^{-\chi}(ul)^{-1})$. Recalling that tests between X and Y represent $\Omega(\chi^{-2})$ of the tests between layers X_i and X_j, we get that $\Omega(\chi^{-5}3^{-\chi}(ul)^{-1})$ of the tests between layers X_i and X_j are satisfied. Choosing u to be $c\chi$ with a big enough constant c, this is more than $2^{-\Omega(u)}$ and hence Φ is satisfiable. ■

■

Theorem 4.6 *Assuming $NP \not\subseteq DTIME(2^{poly(\log n)})$, there is no polynomial time algorithm that colors a 2-colorable 3-uniform hypergraph using $O(\sqrt[3]{\log\log N})$ colors where N is the number of vertices in the hypergraph.*

Proof: We note that in the previous proof, we can take $\chi = c\sqrt[3]{\log\log n}$ where $c > 0$ is any constant and n is the size of the $3SAT5$ instance from which the reduction begins. The parameters we chose are $l = O(\chi^2)$ and $u = O(\chi)$. Therefore, the size of the hypergraph we construct is $N = n^{O(ul)}2^{2^{O(ul)}} = 2^{poly(\log n)}$. The proof is completed by noting that $\log\log N = \log(poly(\log n)) = O(\log\log n)$. ■

5 Discussion

The construction we presented relies on the properties of the Kneser graph in a strong way. To our knowledge, this is the first time hardness of coloring is shown directly and not via the size of the maximal independent set. We believe that the Kneser graph might be useful in understanding the hardness of approximate coloring for graphs, a problem that is notoriously difficult.

6 Acknowledgements

We would like to thank Benny Sudakov for introducing us to the Kneser graph, and to both Benny Sudakov and Noga Alon, for insightful discussions.

References

[1] 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, May 1998.

[2] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

[3] I. Bárány. A short proof of Kneser's conjecture. *J. Combin. Theory Ser. A*, 25(3):325–326, 1978.

[4] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, June 1998.

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

[6] I. Dinur, V. Guruswami, S. Khot, and O. Regev. Vertex cover on k-uniform hypergraphs is NP-hard to approximate to within $k-1-\epsilon$. Manuscript, 2002.

[7] I. Dinur and S. Safra. On the importance of being biased. In *Proc. 34th ACM Symp. on Theory of Computing*, 2002.

[8] V. L. Dol′nikov. Transversals of families of sets. In *Studies in the theory of functions of several real variables (Russian)*, pages 30–36, 109. Yaroslav. Gos. Univ., Yaroslavl′, 1981.

[9] U. Feige and J. Kilian. Zero knowledge and the chromatic number. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity (CCC-96)*, pages 278–289, Los Alamitos, May 24–27 1996. IEEE Computer Society.

[10] P. Frankl and Z. Füredi. Extremal problems concerning Kneser graphs. *J. Combin. Theory Ser. B*, 40(3):270–284, 1986.

[11] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[12] J. E. Greene. A new short proof of Kneser's conjecture. Manuscript, to appear in *Amer. Math. Monthly, 2002.*

[13] V. Guruswami and S. Khanna. On the hardness of 4-coloring a 3-colorable graph. In *Proceedings of the 15th Annual IEEE Conference on Computational Colmplexity (COCO-00)*, pages 188–197, Los Alamitos, CA, July 4–7 2000. IEEE Press.

[14] V. Guruswami, J. Håstad, and M. Sudan. Hardness of approximate hypergraph coloring. In *Proc. 41st IEEE Symp. on Foundations of Computer Science*, pages 149–158. IEEE Computer Society Press, 2000.

[15] Venkatesan Guruswami. Inapproximability results for set splitting and satisfiability problems with no mixed clauses. In *APPROX*, pages 155–166, 2000.

[16] Johan Håstad. Some optimal inapproximability results. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 1–10, El Paso, Texas, 4–6 May 1997.

[17] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000.

[18] Subhash Khot. 3-colorable 3-uniform hypergraphs are hard to color with constantly many colors. 2002.

[19] M. Kneser. Aufgabe 360. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 58(2), Abteilung, S. 27, 1955.

[20] Krivelevich, Nathaniel, and Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *ALGORITHMS: Journal of Algorithms*, 41, 2001.

[21] Krivelevich and Sudakov. Approximate coloring of uniform hypergraphs. In *ESA: Annual European Symposium on Algorithms*, 1998.

[22] L. Lovász. Kneser's conjecture, chromatic number, and homotopy. *J. Combin. Theory Ser. A*, 25(3):319–324, 1978.

[23] J. Matoušek. A combinatorial proof of Kneser's conjecture. *Combinatorica*, to appear.

[24] J. Matoušek. Topological methods in combinatorics and geometry. Book in preparation.

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

[26] A. Schrijver. Vertex-critical subgraphs of Kneser graphs. *Nieuw Arch. Wisk. (3)*, 26(3):454–461, 1978.

[27] Uri Zwick. Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC'99)*, pages 679–687, New York, May 1999. Association for Computing Machinery.