Improved Linearly Ordered Colorings of Hypergraphs via SDP Rounding

Anand Louis
Indian Institute of Science, Bengaluru, India

Alantha Newman
Université Grenoble Alpes, France

Arka Ray
Indian Institute of Science, Bengaluru, India

Abstract

We consider the problem of *linearly ordered* (LO) coloring of hypergraphs. A hypergraph has an LO coloring if there is a vertex coloring, using a set of ordered colors, so that (i) no edge is monochromatic, and (ii) each edge has a unique maximum color. It is an open question as to whether or not a 2-LO colorable 3-uniform hypergraph can be LO colored with 3 colors in polynomial time. Nakajima and Živný recently gave a polynomial-time algorithm to color such hypergraphs with $\widetilde{O}(n^{1/3})$ colors and asked if SDP methods can be used directly to obtain improved bounds. Our main result is to show how to use SDP-based rounding methods to produce an LO coloring with $\widetilde{O}(n^{1/5})$ colors for such hypergraphs. We show how to reduce the problem to cases with highly structured SDP solutions, which we call *balanced* hypergraphs. Then we discuss how to apply classic SDP-rounding tools in this case to obtain improved bounds.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases hypergraph coloring, SDP rounding, promise constraint satisfaction problems

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2024.30

Related Version *Full Version*: `https://arxiv.org/abs/2405.00427` [21]

Funding *Anand Louis*: Supported in part by SERB Award CRG/2023/002896 and the Walmart Center for Tech Excellence at IISc (CSR Grant WMGT-23-0001).
Arka Ray: Supported in part by the Walmart Center for Tech Excellence at IISc (CSR Grant WMGT-23-0001).

1 Introduction

Approximate graph coloring is a well-studied "promise" optimization problem. Given a simple graph $G = (V, E)$ that is promised to be k-colorable, the goal is to find a coloring of G using the minimum number of colors. A (proper) *coloring* is an assignment of colors, which can be represented by positive integers, to the vertices of G so that for each edge ij in G, the vertices i and j are assigned different colors. The most popular case of this problem is when the input graph is promised to be 3-colorable. Even with this very strong promise, the gap between the upper and lower bounds are quite large: the number of colors used by the state-of-the-art algorithm is $\widetilde{O}\left(n^{0.19996}\right)$ [19], while it is NP-hard to color a 3-colorable graph with 5 colors [4]. There is also super constant hardness conditioned on assumptions related to the Unique Games Conjecture [11]. More generally, when we are promised that the graph G is k-colorable, it is NP-hard to color it using $\binom{k}{\lfloor k/2 \rfloor} - 1$ colors [25]. Regarding upper bounds, we note that almost all algorithms for coloring 3-colorable graphs use some combination of semidefinite programming (SDP) and combinatorial tools [17, 2, 19].

Approximate hypergraph coloring is a natural generalization of the above problem to hypergraphs. Here, we want to assign each vertex a color such that there are no monochromatic edges, while using the minimum number of colors. In the case of hypergraph coloring, we know that for every pair of constants $\ell \geqslant k \geqslant 2$, it is NP-hard to ℓ-color a k-colorable 3-uniform hypergraph [12]. Even in the special case, when the 3-uniform hypergraph is promised to be 2-colorable, there is a large gap between the best algorithm, which uses at most $\widetilde{O}\left(n^{1/5}\right)$ colors [20, 1, 10] and the aforementioned (super constant) lower bound.

In this paper, we study a variant of the hypergraph coloring problem known as *linearly ordered coloring*, introduced in several different contexts by [18, 9, 3]. A linearly ordered (LO) k-coloring of an r-uniform hypergraph assigns an integer from $\{1, \ldots, k\}$ to every vertex so that, in each edge in the hypergraph, there is a unique vertex assigned the maximum color in the (multi)set of colors for that edge. Recently, there has been a renewed interest in studying this problem. This is because this problem constitutes a gap in the understanding of the complexity of an important class of problems called *promise constraint satisfaction problems* (PCSPs). To elaborate, [13, 7] classified the complexity of all (symmetric) PCSPs on the binary alphabet, showing that these problems are either polynomial-time solvable or NP-complete. Subsequently, [3] gave a complete classification for PCSPs of the form: given a 2-colorable 3-uniform hypergraph, find a 3-coloring. Here, the notion of "coloring" can have several definitions. As highlighted by [3], the only PCSP of this type whose complexity is unresolved is that of determining whether a 3-uniform hypergraph is 2-LO colorable or is not even 3-LO colorable. In contrast, it was recently shown that it is NP-complete to decide if a 3-uniform hypergraph is 3-LO colorable or not even 4-LO colorable [14].

The work [22] addresses the corresponding optimization problem by giving an algorithm to compute an LO coloring using at most $\widetilde{O}\left(n^{1/3}\right)$ colors for a 2-LO colorable 3-uniform hypergraph. [22] leave open the question of finding an LO coloring for such a hypergraph using fewer colors. Moreover, they state that they do not know how to directly use SDP-based methods[1] and remark that SDP-based approaches seem "less suited for LO colorings". In this paper, one of our main contributions is to show how to use SDP relaxations to give an improved bound for coloring such hypergraphs. Our main result improves this bound significantly by using at most $\widetilde{O}\left(n^{1/5}\right)$ colors to LO color a 2-LO colorable 3-uniform hypergraph.

▶ **Theorem 1.** *Let H be a 2-LO colorable 3-uniform hypergraph on n vertices. Then there exists a (randomized) polynomial-time algorithm that finds an LO coloring of H using $\widetilde{O}\left(n^{1/5}\right)$ colors.*

The SDP relaxation that we use is similar to the natural SDP used in the case of 2-colorable 3-uniform hypergraphs [20]. In fact, the upper bound on the number of colors used in Theorem 1 is the same as the upper bound given by [20] to color 2-colorable 3-uniform hypergraphs. It is the same SDP used by [8] who show that a straightforward hyperplane rounding algorithm yields a solution to the PCSP (1-IN-3-SAT, NAE-3-SAT), in which we are given a satisfiable instance of the first problem and we want to find a feasible solution for the second. Notice that a satisfiable (1-IN-3-SAT) instance on all positive literals is exactly a 2-LO colorable 3-uniform hypergraph.

[1] However, they do use [15] which is an indirect use of SDP-based methods.

General Framework for (Hyper)Graph Coloring

Most algorithms for coloring graphs and hypergraphs proceed iteratively, producing a partial coloring of the remaining (uncolored) vertices at each step. This was formalized by [5], following [23]. The goal is to color a significant number of vertices with few colors in each step, ensuring that the number of iterations and therefore, the overall number of colors used, is small. Typically, in each step, the method used to color the vertices is chosen according to the degree of the graph (or hypergraph) induced on the remaining vertices. In particular, if the induced graph (or hypergraph) has a low degree, then most algorithms use an SDP-based method to find a large independent set, which can be assigned a single color [17, 6, 2]. The algorithm for LO-coloring presented in [22], as well as ours, uses this general framework, except that in [22], they did not use an SDP-based method directly, and instead used [15] to find a large independent set. The improved upper bound on the number of colors output by our algorithm comes from using an SDP and rounding methods tailored to LO coloring.

Overview of our SDP-Based Approach

As noted, we first solve a natural SDP relaxation for 2-LO coloring. Then our rounding proceeds in two steps. In the first step we look at the projection of the vectors to a particular special vector (the vector $\mathsf{v}_\emptyset$ in Proposition 2) from the solution of the SDP, which signifies the color that is unique in all edges in the promised 2-LO coloring. For each of the three vertices in an edge, all three of the corresponding vectors can have a projection onto this special vector with roughly the same value (a *balanced* edge), or they can have very different values (an *unbalanced* edge). It is also possible to classify vertices into balanced and unbalanced (see Definition 4 for formal definitions) so that balanced edges contain only balanced vertices. We use a combinatorial rounding procedure to color all the unbalanced vertices with a small number of colors, leaving only a balanced (sub)hypergraph to be colored. Since this number of colors is much smaller than the bound stated in Theorem 1, this can be viewed as a reduction of the problem to the balanced case. To the best of our knowledge, this rounding method is not present in previous works on LO-coloring and thus, this tool can be considered a main contribution of this paper. We note that [20] showed that the vectors can be "bucketed" with respect to their projection onto a special vector, and used a simple argument to show that there is a large bucket on which they can focus. Our approach allows us to focus on a single bucket containing vectors with projection $\approx -1/3$ with the special vector, which have useful geometric properties.

In the second step we color the hypergraph containing the balanced edges. In this step, we produce (following [22]) an "even" independent set or an "odd" independent set at each round. An *even independent set* is one which intersects each hyperedge two or zero times, while an *odd independent set* intersects each hyperedge one or zero times. To find an even independent set, we use the same approach used by [22]. To find an odd independent set, we use a variant of the standard threshold rounding for a coloring SDP [15, 20]. As in [20] rather than use the vectors output by the SDP solution, we use a modified set of vectors, which have properties useful to obtain better bounds from the threshold rounding. Specifically, the set consists of the normalized projections of the vectors from the SDP solution onto the space orthogonal to the special vector; in the balanced case, the special vector seems to provide no information that is useful to construct a coloring. Combining all the colorings requires some technical care, since we need to always maintain an LO coloring, but it can be done and some of the work has already been done in [22].

Update on Independent and Subsequent Work

After the initial conference submission of our paper, the work [16] appeared on the arXiv. The second version appeared after we posted our paper to arXiv and pointed out that in fact we do not need to consider the balanced case. Indeed, the observation in Section 3 of [16] can be interpreted as giving an alternative and better SDP rounding in the balanced case, directly reducing the balanced case to the unbalanced case. We discuss this more at the end of Section 4.

2 Tools for LO Coloring and Proof of the Main Theorem

In this section, we give an overview of our approach to color a 2-LO colorable 3-uniform hypergraph $H = (V, E)$ with few colors. Following [22], we assume that the input hypergraph H is a *linear hypergraph*, which is defined as follows.

▶ **Definition 2.** *A 3-uniform hypergraph is linear if every pair of edges intersects in at most one vertex.*

This is not a restriction because we can construct an equivalent 3-uniform hypergraph.

▶ **Proposition 3** (Proposition 3 in [22]). *There is a polynomial-time algorithm that, if given an 2-LO colorable 3- uniform hypergraph H, constructs an 2-LO colorable linear 3-uniform hypergraph H' with no more vertices than H such that, if given an LO k-colouring of H', one can compute in polynomial time an LO k-colouring of H.*

Given an 2-LO colorable 3-uniform hypergraph $H = (V, E)$, one can consider LO coloring it with $\{-1, +1\}$, with the natural ordering. Then we have $x_a + x_b + x_c = -1$ for each edge $\{a, b, c\} \in E$, where x_a is the color assigned to vertex $a \in V$. Relaxing this constraint to a vector program we get SDP 2.[2] SDP

$$\mathsf{v}_a + \mathsf{v}_b + \mathsf{v}_c = -\mathsf{v}_\emptyset \qquad \forall \{a, b, c\} \in E, \tag{1}$$

$$\|\mathsf{v}_a\|^2 = 1 \qquad \forall a \in V \cup \{\emptyset\}. \tag{2}$$

For any $a \in V$, we now define $\gamma_a \stackrel{\text{def}}{=} \langle \mathsf{v}_a, \mathsf{v}_\emptyset \rangle$. The values $\{\gamma_a\}_{a \in V}$ might not be integral and could even be *perfectly balanced* (i.e., $\gamma_a = \gamma_b = \gamma_c = -\frac{1}{3}$ for an edge $\{a, b, c\} \in E$). Hence, these values might not contain any information as to how the colors should be assigned to the vertices, and they might not even reveal information as to which vertex in an edge should receive the largest color. However, when all edges contain balanced vertices (i.e., $\gamma_v \approx -\frac{1}{3}$ for all vertices), threshold rounding will be used. Formally, we have the following definition.

▶ **Definition 4.** *For $\varepsilon > 0$, we say a vertex $v \in V$ is* ε-balanced *if $\gamma_v \in [-1/3 - \varepsilon, -1/3 + \varepsilon]$.*

For the rest of this paper, we fix $\varepsilon = 1/n^{100}$, where n is the number of vertices in the (fixed) hypergraph that we are trying to LO color. This is an abuse of notation, but simplifies our presentation. If a vertex is not ε-balanced, we say that it is *unbalanced*. If all vertices of a hypergraph H are ε-balanced, we say that H is an ε-balanced hypergraph.

We observe that there is a combinatorial method to color all unbalanced vertices using relatively few colors. This rounding method uses a bisection-like strategy on $\{\gamma_a\}_{a \in V}$ to color the unbalanced vertices and outputs a *partial LO coloring*, which we define as follows.

---

[2] Observe that SDP 2 can equivalently be written in terms of dot products using the following constraints: (i) $\langle \mathsf{v}_a + \mathsf{v}_b + \mathsf{v}_c + \mathsf{v}_\emptyset, \mathsf{v}_a + \mathsf{v}_b + \mathsf{v}_c + \mathsf{v}_\emptyset \rangle = 0 \quad \forall \{a, b, c\} \in E$, and (ii) $\langle \mathsf{v}_a, \mathsf{v}_a \rangle = 1 \quad \forall a \in V \cup \{\emptyset\}$.

▶ **Definition 5.** *A* partial LO coloring *of a 3-uniform hypergraph $H = (V, E)$ is a coloring of a subset of vertices $V_1 \subseteq V$ using the set of colors C such that for each edge $e \in E$, the set $e \cap V_1$ has a unique maximum color from C.*

The next lemma is proved in Section 3.

▶ **Lemma 6.** *Let $H = (V, E)$ be a 2-LO colorable 3-uniform hypergraph and let $\varepsilon > 0$. Then there exists a polynomial-time algorithm that computes a partial LO coloring of H using $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ colors that colors all unbalanced vertices.*

We remark that the previous lemma can be viewed as a reduction from LO coloring in 2-LO colorable 3-uniform hypergraphs to LO coloring in 2-LO colorable 3-uniform *balanced* hypergraphs. To formalize this, let V_U denote the vertices that are colored in a partial LO coloring produced via Lemma 6. Let $V_B = V \setminus V_U$. Notice that V_B contains only ε-balanced vertices, while V_U contains all the unbalanced vertices but might also contain some ε-balanced vertices. Thus, the induced hypergraph $H_B = (V_B, E(V_B))$ is a balanced hypergraph.[3] We now show that we can combine a partial LO coloring for $H = (V, E)$ which colors V_U and an LO coloring for $H_B = (V_B, E(V_B))$ to obtain an LO coloring of H.

▶ **Proposition 7.** *Let $H = (V_B \cup V_U, E)$ be a 2-LO colorable 3-uniform hypergraphs, let $\varepsilon > 0$. Let c_U be a partial LO coloring of H using colors from the set C_U that only assigns colors to V_U and let c_B be an LO coloring of $H_B = (V_B, E(V_B))$ using colors from the set C_B. Then we can obtain an LO coloring of H using at most $|C_U| + |C_B|$ colors.*

Proof. We assume that the colors in the set C_U are larger than the colors in the set C_B. We want to show that the given assignment of colors from C_U for vertex set V_U and C_B for vertex set V_B taken together forms a proper LO coloring of H.

Any edge $e \in E$ with $|e \cap V_B| = 3$ or $|e \cap V_U| = 3$ has a unique maximum color by assumption since c_B is an LO coloring of H_B and c_U is a partial LO coloring of H. Suppose $|e \cap V_U| = 2$. Then, by definition of partial LO coloring, it has a unique maximum in C_U and will have a unique maximum in the output coloring. If $|e \cap V_U| = 1$, then e has a unique maximum color, because all colors in C_U are larger than the colors in C_B. ◀

Thus, if our goal is to LO color 2-LO colorable 3-uniform hypergraphs with a polynomial number of colors, we can focus on LO coloring *balanced* 2-LO colorable 3-uniform hypergraphs. The next corollary follows from Lemma 6 and Proposition 7.

▶ **Corollary 8.** *Let $\alpha \in (0, 1)$. Suppose we can LO color an ε-balanced 2-LO colorable 3-uniform hypergraph H with $\widetilde{O}(n^\alpha)$ colors. Then we can LO color a 2-LO colorable 3-uniform hypergraph with $\widetilde{O}(n^\alpha)$ colors.*

Now we can focus on balanced hypergraphs. We capitalize on the promised structure to prove the next lemma, in which we show that we can find an LO coloring for a balanced hypergraph, in particular for $H_B = (V_B, E(V_B))$.

▶ **Lemma 9.** *Let $H_B = (V_B, E_B)$ be an ε-balanced 2-LO colorable 3-uniform hypergraph. Then there exists a polynomial-time algorithm that computes an LO coloring using at most $\widetilde{O}(|V_B|^{1/5})$ colors.*

[3] Note that for a hypergraph $H = (V, E)$ and $S \subset V$, we say $H' = (S, E(S))$ contains the edges *induced* on S, meaning an edge belongs to H' if all of its vertices belong to S. In other words, an induced subhypergraph of a 3-uniform must also be 3-uniform (or empty). Notice that S can contain vertices that do not belong to any edge in $E(S)$. These vertices can receive any color in a valid LO coloring of H'.

We recall our main theorem.

▶ **Theorem 1.** *Let H be a 2-LO colorable 3-uniform hypergraph on n vertices. Then there exists a (randomized) polynomial-time algorithm that finds an LO coloring of H using $\widetilde{O}\left(n^{1/5}\right)$ colors.*

The proof of Theorem 1 follows from Corollary 8 and Lemma 9. It remains to prove Lemma 9, which we discuss next.

Coloring by Finding Independent Sets

In many graph coloring algorithms, we "make progress" by finding an independent set and coloring it with a new color [5, 17, 6, 20, 22]. When LO coloring a hypergraph, a similar idea may be used, but we need to consider certain types of independent sets. With the standard notion of independent set in a 3-uniform hypergraph, in which the independent set intersects each edge of the hypergraph at most twice, it is not clear how to obtain a coloring in which each edge contains a unique maximum color. Thus, for a 3-uniform hypergraph $H = (V, E)$, following the approach of [22], we consider the following two types of independent sets.[4]

Odd Independent Set: We call $S \subseteq V$ an *odd independent set* if $|S \cap e| \leqslant 1$ for each edge $e \in E$.

Even Independent Set: We call $S \subseteq V$ an *even independent set* if $|S \cap e| \in \{0, 2\}$ for each edge $e \in E$.

In Lemma 10, we show that we can make progress by coloring an odd independent set with a "large" color or by coloring an even independent set with a "small" color. This is formally stated in a proposition from [22]. Since we modify the presentation slightly to ensure compatibility with our framework, we include the statement and the proof here for the sake of completeness.

▶ **Lemma 10** (Corollary of Proposition 5 in [22])**.** *Let $H = (V, E)$ be a hypergraph, let $S_1 \subseteq V$ be an odd independent set and let $S_2 \subseteq V$ be an even independent set. Let $H_1 = (V_1, E_1)$, $H_2 = (V_2, E_2)$ be the hypergraphs induced by $V_1 = V \setminus S_1$ and $V_2 = V \setminus S_2$, respectively. Then,*

1. *An LO coloring of H_1 using a set of colors C_1 can be extended to an LO coloring of H by assigning a color c_1 that is strictly larger than all the colors in C_1 to the vertices in S_1.*
2. *Analogously, an LO coloring of H_2 using a set of colors C_2 can be extended to an LO coloring of H by assigning c_2 to the vertices in S_2 where c_2 is strictly smaller than all the colors in C_2.*

Proof. In the proposed extension of the coloring from H_1 to H, there is no edge $e \in E_1$ where the maximum color in e occurs more than once in e; otherwise, the promised coloring of H_1 using C_1 is not valid. Consider any edge $\{u, v, w\} \in E \setminus E_1$. By definition of S_1, we have $|\{u, v, w\} \cap S_1| \leqslant 1$. Note that $|\{u, v, w\} \cap S_1| \neq 0$ as $\{u, v, w\} \notin E_1$. Therefore, we must have $|\{u, v, w\} \cap S_1| = 1$. Without loss of generality, assume that $u \in S_1$ and $v, w \notin S_1$. Then, in the proposed coloring, c_1 is only used for u, while v, w are colored using some color(s) from C_1. So, c_1 is the largest color in $\{u, v, w\}$ and occurs exactly once. Hence, for every edge, the corresponding (multi)set of colors has a unique maximum, and we conclude that the proposed coloring is a proper LO coloring of H.

[4] We remark that what [22] refer to as an "independent set" is what we refer to here as an "odd independent set".

Similarly, in the proposed extension of coloring from H_2 to H there is no edge $e \in E_2$ where the maximum color in e occurs more than once in e. Again, consider any edge $\{u, v, w\} \in E \setminus E_2$. In this case, we have $|\{u, v, w\} \cap S_2| = 2$. Without loss of generality, assume that $u, v \in S_2$ and $w \notin S_2$. Then, in the proposed coloring, c_2 is only used on u, v, while w is colored using some color c from C_2. So, c is the largest color in $\{u, v, w\}$ and it occurs exactly once. Hence, for every edge, the corresponding (multi)set of color has a unique maximum, and the proposed coloring is therefore a proper LO coloring of H. ◀

The following proposition is essentially Lemma 1 in [5] and follows in a straight-forward manner from Lemma 10.

▶ **Proposition 11** (Proposition 5 in [22]). *Let $H = (V, E)$ be an ε-balanced, 2-LO colorable 3-uniform linear hypergraph on m vertices. Suppose we can find an odd independent set of size at least $f(m)$ in H or an even independent set of size at least $f(m)$ in H (where f is* nearly-polynomial[5]*), then there exists a polynomial-time algorithm that colors any ε-balanced, 2-LO colorable 3-uniform linear hypergraph on n vertices with $n/f(n)$ colors.*

Following this standard notion of "making progress" from [5], we simply need to show that we can find an even or an odd independent set of size at least $f(m)$ in a 2-LO colorable 3-uniform ε-balanced hypergraph on m vertices. This will imply that we can color H_B with $|V_B|/f(V_B)$ colors. We will show that we can set $f(m) = \widetilde{\Theta}(m^{4/5})$, which will yield the bound in Lemma 9.

As is typical, our coloring algorithm makes progress using two different methods and chooses between the two methods depending on the degree. In the high-degree case, we use the method from [22] to find a large even independent set. The method to find a large even independent from [22] requires the input hypergraph to be a linear hypergraph, which, as discussed previously, we can assume by Proposition 3.

▶ **Proposition 12** (Proposition 11 in [22]). *Let $H = (V, E)$ be a linear 2-LO colorable 3-uniform hypergraph and Δ be such that $|E| = \Omega(\Delta\,|V|)$. Then there is a polynomial-time algorithm that finds a even independent set of size at least $\Omega(\sqrt{|V|\Delta})$.*

In the low-degree case, we show how to use an SDP based rounding method to find a large odd independent set. Here, we capitalize on the assumption that our input hypergraph is ε-balanced to obtain an improvement over the analogous lemma from [22]. In Section 4, we prove Lemma 13.

▶ **Lemma 13.** *Let $H = (V, E)$ be a $\frac{1}{|V|^{100}}$-balanced 2-LO colorable 3-uniform hypergraph $H = (V, E)$ with average degree at most Δ. Then there exists a (randomized) polynomial-time algorithm to compute an odd independent set of size at least $\Omega\left(\frac{|V|}{\Delta^{1/3}(\ln \Delta)^{3/2}}\right)$.*

Finally, we are now ready to prove Lemma 9.

Proof of Lemma 9. We need to show that on a linear 2-LO-colorable 3-uniform ε-balanced hypergraph on m vertices, we can always find either an even independent set or an odd independent set of size at least $f(m) = \widetilde{\Omega}(m^{4/5})$. By Lemma 10, this will imply we can color H_B with $|V_B|/f(|V_B|)$ colors.

[5] Definition 1 in [5]. A function $f(m) = m^{\alpha}\mathsf{polylog} m$ for $\alpha > 0$ is nearly-polynomial.

Take Δ be a parameter (fixed later) so that we say we are in the high-degree regime if the average degree is higher than Δ. Otherwise, we say that we are in the low-degree regime. In the high degree-regime, use Proposition 12 to find an even independent set S of size at least $\Omega(\sqrt{m\Delta})$. In the low-degree regime, we invoke Lemma 13 to find an odd independent set S of size at least $\widetilde{\Omega}(m/\Delta^{1/3})$. Setting $\Delta = m^{3/5}$ implies that the independent set we find has size at least $m^{4/5}$. Finally, by Proposition 11 we have the desired bound on the number of colors used. ◀

3 Combinatorial Rounding for Unbalanced Vertices

In this section, we prove Lemma 6. In other words, we show that for any $\varepsilon > 0$, Algorithm 1 outputs a partial LO coloring using $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ colors so that all the unbalanced vertices are assigned a color.

▶ **Lemma 6.** *Let $H = (V, E)$ be a 2-LO colorable 3-uniform hypergraph and let $\varepsilon > 0$. Then there exists a polynomial-time algorithm that computes a partial LO coloring of H using $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ colors that colors all unbalanced vertices.*

To prove this lemma, we give an algorithm, which given the value $\{\gamma_v\}$ for each vertex v (from SDP 2), is then combinatorial. The algorithm also takes as input the value of ε, which is the parameter we use to define ε-balanced.

Algorithm 1 Combinatorial Rounding.

Input: A 2-LO colorable 3-uniform hypergraph $H = (V, E)$, $\varepsilon > 0$, the values $\{\gamma_a\}$ for all $a \in V$ and set C of linearly ordered colors.
Output: A partial LO coloring of all unbalanced vertices in V.

1. Set $j := 0, \ell_0 := -1, u_0 := 1, I_0 := [\ell_0, u_0]$.
2. While $I_j \nsubseteq [-1/3 - \varepsilon, -1/3 + \varepsilon]$ do:
   a. If j is even then set I_{j+1} to the lower half of I_j, if j is odd then set I_{j+1} to be the upper half of I_j. More precisely, set

$$\ell_{j+1} := \begin{cases} \frac{\ell_j + u_j}{2} & j \text{ is odd} \\ \ell_j & \text{otherwise} \end{cases} \qquad u_{j+1} := \begin{cases} \frac{\ell_j + u_j}{2} & j \text{ is even} \\ u_j & \text{otherwise} \end{cases}$$

   and set $I_{j+1} := [\ell_{j+1}, u_{j+1}]$.
   b. Set $S_{j+1} := \{a \in V | \gamma_a \in I_j \setminus I_{j+1}\}$ and color S_{j+1} using the largest unused color from C.
   c. Set $j := j + 1$.

We will use the following observation.

▶ **Observation 14.** *For any $\{a, b, c\} \in E$, we have $\gamma_a + \gamma_b + \gamma_c = -1$.*

Proof. From constraint (1), we get $\gamma_a + \gamma_b + \gamma_c = \langle \mathsf{v}_a + \mathsf{v}_b + \mathsf{v}_c, \mathsf{v}_\emptyset \rangle = \langle -\mathsf{v}_\emptyset, \mathsf{v}_\emptyset \rangle = -1$. ◀

On a high level, the algorithm partitions the interval $[-1, 1]$ and assigns colors to vertices depending on where their corresponding γ_a values fall in this interval. For example, in the first iteration of the algorithm, we set S_1 to contain all vertices whose γ_a values fall into the interval $(0, 1]$. Notice that by Observation 14, at most one vertex from an edge will qualify. Now, all remaining vertices have γ_a values in the interval $[-1, 0]$. Next, we consider all vertices whose γ_a values fall into the interval $[-1,-1/2)$. Again, an edge with all three values in $[-1,0]$ can not have more than one vertex with γ_a value in $[-1,-1/2)$, and so on. We now formally analyze the algorithm.

▶ **Lemma 15.** *For even $j \geqslant 2$, the interval $[\ell_j, u_j]$ is*

$$\left[\frac{-(2^{j-1}-2)/3-1}{2^{j-1}}, \frac{-(2^{j-1}-2)/3}{2^{j-1}}\right].$$

For odd $j \geqslant 1$, the interval $[\ell_j, u_j]$ is

$$\left[\frac{-(2^{j-1}-1)/3-1}{2^{j-1}}, \frac{-(2^{j-1}-1)/3}{2^{j-1}}\right].$$

Proof. For $j=1$ the interval is $[-1,0]$ and $j=2$ the interval is $[-1/2,0]$. For odd j, we have

$$\ell_{j+1} = \frac{\ell_j+u_j}{2} = \frac{-(2^{j-1}-1)/3-1-(2^{j-1}-1)/3}{2^j} = \frac{-(2^j-2)/3-1}{2^j},$$

and

$$u_{j+1} = \frac{-(2^{j-1}-1)/3}{2^{j-1}} = \frac{-(2^j-2)/3}{2^j}.$$

For even j, we have

$$u_{j+1} = \frac{\ell_j+u_j}{2} = \frac{-(2^{j-1}-2)/3-1-(2^{j-1}-2)/3}{2^j} = \frac{-(2^j-1)/3}{2^j},$$

and

$$\ell_{j+1} = \frac{-(2^{j-1}-2)/3-1}{2^{j-1}} = \frac{-(2^j-4)/3-2}{2^j} = \frac{(-2^j+1-3)/3}{2^j} = \frac{-(2^j-1)/3-1}{2^j}.$$ ◀

As a consequence of Lemma 15, we immediately get a bound on the number of iterations in form of Corollary 16.

▶ **Corollary 16.** *For $j \geqslant \log\left(\frac{4}{3\varepsilon}\right)$, we have $I_j \subseteq [-1/3-\varepsilon, -1/3+\varepsilon]$.*

Proof. By Lemma 15 we have the following bounds on I_j. For even $j \geqslant 2$, the interval $I_j = [\ell_j, u_j]$ is

$$\left[-\frac{1}{3} - \frac{1}{3\cdot 2^{j-1}}, -\frac{1}{3} + \frac{2}{3\cdot 2^{j-1}}\right].$$

For odd $j \geqslant 1$, the interval $I_j = [\ell_j, u_j]$ is

$$\left[-\frac{1}{3} - \frac{2}{3\cdot 2^{j-1}}, -\frac{1}{3} + \frac{1}{3\cdot 2^{j-1}}\right].$$

Setting $\varepsilon = \frac{1}{3\cdot 2^{j-2}} = \frac{4}{3\cdot 2^j}$, we have $I_j \subseteq [-\frac{1}{3}-\varepsilon, -\frac{1}{3}+\varepsilon]$. Thus, $j = \log\left(\frac{4}{3\varepsilon}\right)$. ◀

In Lemma 17 we show that in each iteration Algorithm 1 colors an odd independent set. Lemma 17 also follows from Lemma 15.

▶ **Lemma 17.** *For each $j \geqslant 0$, let $H_j = (S_j, E_j)$ be a hypergraph with $E_j = \{e \in E : e \subseteq S_j\}$. Then for any $j \geqslant 0$, the set S_{j+1} is an odd independent set (i.e., we have $|S_{j+1} \cap e| \leqslant 1$ for any $e \in E_j$).*

Proof. Let $\{a, b, c\} \in E_j$. Suppose $a, b \in S_{j+1}$. If j is odd, then $\gamma_a, \gamma_b \in [\ell_j, \ell_{j+1})$. Therefore, we get $\gamma_a + \gamma_b < 2\ell_{j+1}$. This implies, by Observation 14, $\gamma_c > -1 - 2\ell_{j+1}$. Therefore, we have

$$\begin{aligned}\gamma_c > -1 - 2\ell_{j+1} &= -1 - 2\left(\frac{-(2^j-2)/3-1}{2^j}\right) = -1 + 2\left(\frac{(2^j-2)/3+1}{2^j}\right) \\ &= \frac{-3\cdot 2^j + 2(2^j-2)+6}{3\cdot 2^j} = \frac{-2^j+2}{3\cdot 2^j} = u_j,\end{aligned}$$

which is a contradiction since $\gamma_c \in [\ell_j, u_j]$.

Similarly, if j is even, then $\gamma_a, \gamma_b \in (u_{j+1}, u_j]$ as $a, b \in S_{j+1}$. Therefore, we get $\gamma_a + \gamma_b > 2u_{j+1}$. This implies, by Observation 14, $\gamma_c < -1 - 2u_{j+1}$. Therefore, we have

$$\gamma_c < -1 - 2u_{j+1} = \frac{-3\cdot 2^j + 2\cdot 2^j - 2}{3\cdot 2^j} = \frac{-2^j-2}{3\cdot 2^j} = \ell_j,$$

which is again a contradiction to the fact that $\gamma_c \in [\ell_j, u_j]$. ◀

Proof of Lemma 6. By Corollary 16, Algorithm 1 runs for $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ iterations. In each iteration, it uses exactly one color, which yields the stated bound on the number of colors used. To show that the output coloring is a partial LO coloring, we apply Lemma 17, which states that each color corresponds to an odd independent set.

Now, we need to show that any edge with at least one colored vertex will have a unique maximum color. Consider such an edge $e = \{a, b, c\}$. If only one vertex in e is colored, then we are done. First, assume exactly two vertices in e (say a, b) were colored. Let a be colored in the j_a-th iteration and b be colored in the j_b-th iteration. Assume (without loss of generality) that $j_a \geqslant j_b$. Then, by Lemma 17 we have $j_a \neq j_b$ (i.e., $j_a > j_b$). As the color used in the iteration j is the j-th largest color in C (by a simple induction) color assigned to a is strictly larger than the color assigned to b. Finally, if all the vertices, a, b, c were colored at iterations $j_a \geqslant j_b \geqslant j_c$, respectively. Then, again by the same arguments we have $j_a > j_b$ and $j_a > j_c$, so the maximum color is assigned to only a. ◀

4 SDP Rounding for Balanced Hypergraphs

In this section we show that Algorithm 2 outputs an odd independent set in an ε-balanced 2-LO colorable 3-uniform hypergraph $H_B = (V_B, E_B)$. Thus, we will prove Lemma 13.

▶ **Lemma 13.** *Let $H = (V, E)$ be a $\frac{1}{|V|^{100}}$-balanced 2-LO colorable 3-uniform hypergraph $H = (V, E)$ with average degree at most Δ. Then there exists a (randomized) polynomial-time algorithm to compute an odd independent set of size at least $\Omega\left(\frac{|V|}{\Delta^{1/3}(\ln\Delta)^{3/2}}\right)$.*

Recall that we have a solution for the SDP 2. Let u_a be the unit vector along the component orthogonal to $\mathsf{v}_\emptyset$ (if the orthogonal component is zero, then we define u_a to be any arbitrarily chosen unit vector). Therefore,

$$\mathsf{u}_a = \frac{\mathsf{v}_a - \gamma_a \mathsf{v}_\emptyset}{\|\mathsf{v}_a - \gamma_a \mathsf{v}_\emptyset\|} = \frac{\mathsf{v}_a - \gamma_a \mathsf{v}_\emptyset}{\sqrt{\|\mathsf{v}_a\|^2 + \gamma_a^2 - 2\gamma_a\langle \mathsf{v}_a, \mathsf{v}_\emptyset\rangle}} = \frac{\mathsf{v}_a - \gamma_a \mathsf{v}_\emptyset}{\sqrt{1-\gamma_a^2}}. \tag{3}$$

Let function $\bar{\Phi} : \mathsf{R} \to [0, 1]$ be defined as $\bar{\Phi}(t) \stackrel{\text{def}}{=} \mathbb{P}_{g\sim\mathcal{N}(0,1)}[g \geqslant t]$.

Algorithm 2 Randomized Rounding.

Input: H_B a ε-balanced 2-LO colorable 3-uniform hypergraph and a parameter α (see Lemma 20 for values of ε and α to be used).
Output: An odd independent set.

1. Let t be such that $\alpha = \bar{\Phi}(t)$.
2. Sample $g \sim \mathcal{N}(0,1)^{|V_B|}$ and set $S(t) := \{a \in V_B : \langle \mathsf{u}_a, g\rangle \geqslant t\}$.
3. Set $S'(t) := S(t) \setminus \left(\bigcup_{\substack{e \in E_B \\ |e \cap S(t)| \geqslant 2}} e \right)$.
4. Output $S'(t)$.

In case of an edge $\{a,b,c\}$ with perfectly balanced vertices (i.e., if we have $\gamma_a = \gamma_b = \gamma_c = -1/3$), one can observe that the component orthogonal to $\mathsf{v}_\emptyset$ of the corresponding vectors sum to 0 (i.e., we have $\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c = 0$). In Lemma 18 we show a generalization of this observation for an ε-balanced hypergraph. Recall that in an ε-balanced hypergraph, we have $\gamma_a \in [-1/3 - \varepsilon, -1/3 + \varepsilon]$ for each vertex. The proof of the next lemma can be found in Appendix C.

▶ **Lemma 18.** *Let $\{a,b,c\}$ be an edge in an ε-balanced hypergraph H_B. Then $\|\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c\|^2 \leqslant 18\varepsilon$.*

When all the vertices in $\{a,b,c\}$ are perfectly balanced then the event that both a and b belong to $S(t)$ is equivalent to $\langle \mathsf{u}_c, g\rangle \leqslant -2t$ as $\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c = 0$. Therefore, we can use bounds on Gaussians to bound the probability of the aforementioned event. Again, Lemma 19 generalizes this to ε-balanced vector for small enough ε.

▶ **Lemma 19.** *Take $\varepsilon = \frac{1}{|V_B|^{100}}$ and let a, b be adjacent vertices in H_B. Then*

$$\mathbb{P}\left[a \in S(t) \wedge b \in S(t)\right] \leqslant \bar{\Phi}(2t) + \frac{2}{|V_B|^{25}}.$$

Proof. Suppose $e = \{a,b,c\}$ is an edge in H_B containing both a and b. If both a and b belong to $S(t)$, then $\langle \mathsf{u}_a, g\rangle \geqslant t$ and $\langle \mathsf{u}_b, g\rangle \geqslant t$. Note that $\|\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c\| \leqslant 3\sqrt{2\varepsilon}$ by Lemma 18. If we additionally assume that $\|g\| \leqslant |V_B|^{25}$ (this assumption is violated with low probability) we have

$$\begin{aligned} 3\sqrt{2\varepsilon}|V_B|^{25} &\geqslant \langle \mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c, g\rangle && \text{(Cauchy-Schwarz)} \\ &= \langle \mathsf{u}_a, g\rangle + \langle \mathsf{u}_b, g\rangle + \langle \mathsf{u}_c, g\rangle \\ &\geqslant 2t + \langle \mathsf{u}_c, g\rangle && (\langle \mathsf{u}_a, g\rangle \geqslant t \text{ and } \langle \mathsf{u}_b, g\rangle \geqslant t) \end{aligned}$$

we get $\langle \mathsf{u}_c, g\rangle \leqslant -2t + 3\sqrt{2\varepsilon}|V_B|^{25}$.
Thus, we can upper bound $\mathbb{P}\left[(\langle \mathsf{u}_a, g\rangle \geqslant t) \wedge (\langle \mathsf{u}_b, g\rangle \geqslant t) \wedge \left(\|g\| \leqslant |V_B|^{25}\right)\right]$ by

$$\begin{aligned} &\mathbb{P}\left[\left(\langle \mathsf{u}_c, g\rangle \leqslant -2t + 3\sqrt{2\varepsilon}|V_B|^{25}\right) \wedge \left(\|g\| \leqslant |V_B|^{25}\right)\right] \\ &\leqslant \mathbb{P}\left[\langle \mathsf{u}_c, g\rangle \leqslant -2t + 3\sqrt{2\varepsilon}|V_B|^{25}\right] \\ &= \bar{\Phi}\left(2t - 3\sqrt{2\varepsilon}|V_B|^{25}\right) \\ &\leqslant \bar{\Phi}(2t) + \sqrt{\varepsilon}|V_B|^{25} && \text{(Fact 21)} \\ &\leqslant \bar{\Phi}(2t) + \frac{1}{|V_B|^{25}}. && \left(\varepsilon = \frac{1}{|V_B|^{100}}\right) \end{aligned}$$

Now, in the following step we look at the case when the assumption $\|g\| \leqslant |V_B|^{25}$ is violated.

$$\begin{aligned}\mathbb{P}\left[(\langle \mathsf{u}_a, g\rangle \geqslant t) \wedge (\langle \mathsf{u}_b, g\rangle \geqslant t) \wedge \left(\|g\| > |V_B|^{25}\right)\right] &\leqslant \mathbb{P}\left[\|g\| > |V_B|^{25}\right]\\ &\leqslant \mathbb{P}\left[\|g\|^2 > |V_B|^{50}\right]\\ &\leqslant \frac{1}{|V_B|^{49}} \qquad \left(\mathsf{E}\left[\|g\|^2\right] = |V_B| \text{ and Markov bound}\right)\end{aligned}$$

Adding up the two disjoint cases we get the required bound. ◀

▶ **Lemma 20.** *Let $\Delta \geqslant 4$ be an upper bound on the average degree of a vertex in H_B (i.e., $|E_B| \leqslant \frac{\Delta|V_B|}{3}$). Take $\alpha = \frac{1}{32}\frac{1}{\Delta^{\frac{1}{3}}(\ln\Delta)^{1/2}}$ and $\varepsilon = \frac{1}{|V_B|^{100}}$. Then, we have $\mathsf{E}\left[|S'(t)|\right] \geqslant \frac{3}{4}\alpha\,|V_B|$.*

Proof. To lower bound the expected size of $S'(t)$ we lower bound the expected size of $S(t)$ and upper bound the expected number of vertices participating in a bad edge (i.e., an edge e such that $|e \cap S(t)| \geqslant 2$) separately.

First, we lower bound the size of $|S(t)|$ as follows.

$$\mathsf{E}\left[|S(t)|\right] = \sum_{a\in V_B} \mathbb{P}\left[\langle \mathsf{u}_a, g\rangle \geqslant t\right] = \alpha\,|V_B|\,.$$

Now, to get an upper bound we note that each bad edge can contribute at most 3 vertices in the total number of vertices participating in some bad edge. Formally, we have the following.

$$\begin{aligned}\left|\bigcup_{\substack{e\in E_B\\ |e\cap S(t)|\geqslant 2}} e\right| &\leqslant \sum_{\substack{e\in E_B\\ |e\cap S(t)|\geqslant 2}} |e| && \text{(Union Bound)}\\ &\leqslant 3\left|\{e\in E_B \text{ s.t. } |e\cap S(t)|\geqslant 2\}\right| && (|e| = 3)\end{aligned}$$

If an edge $\{a,b,c\}$ is bad, i.e., we have $|\{a,b,c\}\cap S(t)| \geqslant 2$, then either $\{a,b\}\subseteq S(t)$ or $\{a,c\}\subseteq S(t)$ or $\{b,c\}\subseteq S(t)$. Therefore, by union bound $\mathsf{E}\left[|\{e\in E_B \text{ s.t. } |e\cap S(t)|\geqslant 2\}|\right]$ is at most

$$\begin{aligned}&\sum_{\{a,b,c\}\in E_B}\Big(\mathbb{P}\left[a\in S(t)\wedge b\in S(t)\right] + \mathbb{P}\left[a\in S(t)\wedge c\in S(t)\right] + \mathbb{P}\left[c\in S(t)\wedge b\in S(t)\right]\Big)\\ &\leqslant \sum_{e\in E_B} 3\cdot\left(\bar{\Phi}(2t) + \frac{2}{|V_B|^{25}}\right)\\ &= 3|E_B|\cdot\bar{\Phi}(2t) + \frac{6|E_B|}{|V_B|^{25}},\end{aligned}$$

where the second inequality follows from Lemma 19. Let us now upper bound the first term as follows.

$$\begin{aligned}3|E_B|\bar{\Phi}(2t) &\leqslant \Delta|V_B|\cdot 512\bar{\Phi}(t)^4\cdot\left(\ln(1/\bar{\Phi}(t))\right)^{3/2} && \left(|E_B|\leqslant\frac{\Delta|V_B|}{3} \text{ and Corollary 25}\right)\\ &\leqslant \Delta|V_B|\cdot 512\alpha^4\cdot(\ln(1/\alpha))^{3/2} && (\bar{\Phi}(t) = \alpha)\\ &\leqslant \frac{1}{8}\alpha|V_B|\left(\frac{\ln(1/\alpha)}{4\ln\Delta}\right)^{3/2} && (\text{Substituting } \alpha)\\ &\leqslant \frac{1}{8}\alpha|V_B| && (\Delta\geqslant 4)\end{aligned}$$

Note that in the first inequality above we could use Corollary 25 as $\Delta \geqslant 4$ implies $t \geqslant 1$. It is easy to show that $\frac{6|E_B|}{|V_B|^{25}} \leqslant \frac{1}{8}\alpha|V_B|$. Therefore, we get

$$\mathsf{E}\left[|\{e \in E_B \text{ s.t. } |e \cap S(t)| \geqslant 2\}|\right] \leqslant \frac{1}{4}\alpha|V_B|.$$

Thus, by combining the two bounds we get that

$$\mathsf{E}\left[|S'(t)|\right] = \mathsf{E}\left[|S(t)|\right] - \mathsf{E}\left[\left|\bigcup_{\substack{e \in E_B \\ |e \cap S(t)| \geqslant 2}} e\right|\right] \geqslant \left(1 - \frac{1}{4}\right)\alpha\,|V_B| \geqslant \frac{3}{4}\alpha\,|V_B|\,. \qquad \blacktriangleleft$$

Proof of Lemma 13. This follows from Lemma 20 and the proof is standard Markov bound followed by an amplification argument where you repeat Algorithm 2 polynomially many times and choose the best odd independent set among all repetitions. The probability of even the best odd independent set not being of the required size is then inverse exponential with respect number of iterations. We refer the reader to Section 13.2 of [24] for further reference. ◀

A Better SDP Rounding

Here we note that there is in fact a better way to round the SDP in the balanced case, which follows from [16] and essentially reduces the balanced case to the unbalanced case. Let $H = (V, E)$ be an ε-balanced hypergraph on n vertices. Recall that $\varepsilon \leqslant 1/n^{100}$.

As in Algorithm 2, we sample a gaussian $g \sim \mathcal{N}(0,1)^n$. For each (unit) vector u_a for $a \in V$, let $\zeta_a = \langle \mathsf{u}_a, g\rangle$. Observe that $|\zeta_a| \in [1/n^2, n^2]$ with probability $1 - O(1/n^2)$. Now for all $a \in V$, set $\zeta'_a = \zeta_a/n^2$ and set $\gamma'_a = \gamma_a + \zeta'_a$. Thus, with probability at least (roughly) $1 - O(1/n)$, for *all* vertices $a \in V$, we have

$$|\zeta'_a| \in [1/n^4, 1] \quad \text{and} \quad \gamma'_a \notin (-1/3 - 1/n^{100}, -1/3 + 1/n^{100}).$$

Since for every hyperedge $\{a, b, c\} \in E$, we have $\zeta'_a + \zeta'_b + \zeta'_c = 0$ (because $\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c = \mathbf{0}$, which implies $\langle \mathsf{u}_a, g\rangle + \langle \mathsf{u}_b, g\rangle + \langle \mathsf{u}_c, g\rangle = 0$).

Then for every hyperedge $\{a, b, c\} \in E$, we have $\gamma'_a + \gamma'_b + \gamma'_c = -1$. Thus, we can run Algorithm 1 on the inputs $\{\gamma'_a\}_{a \in V}$ and $\varepsilon = 1/n^{100}$. By Lemma 6, it will output an LO-coloring of H using at most $O(\log \frac{1}{\varepsilon})$ colors.

5 Conclusion

We have presented an improved bound on the number of colors needed to efficiently LO color a 2-LO colorable 3-uniform hypergraph, and demonstrated that SDP-based rounding methods can indeed be applied to LO coloring. A natural question is if we can do better than $O(\log n)$ colors in the balanced case; this might be a step towards improving on the bound of $O(\log n)$ colors for the general case given in [16].

References

1 Noga Alon, Pierre Kelsen, Sanjeev Mahajan, and Hariharan Ramesh. Coloring 2-colorable hypergraphs with a sublinear number of colors. *Nordic Journal of Computing*, 3:425–439, 1996.

2 Sanjeev Arora, Eden Chlamtac, and Moses Charikar. New approximation guarantee for chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 215–224, 2006.

3 Libor Barto, Diego Battistelli, and Kevin M. Berg. Symmetric promise constraint satisfaction problems: Beyond the Boolean case. In *38th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 187 of *LIPIcs*, pages 10:1–10:16, 2021. doi:10.4230/LIPICS.STACS.2021.10.

4 Libor Barto, Jakub Bulín, Andrei A. Krokhin, and Jakub Oprsal. Algebraic approach to promise constraint satisfaction. *Journal of the ACM*, 68(4):28:1–28:66, 2021. doi:10.1145/3457606.

5 Avrim Blum. New approximation algorithms for graph coloring. *Journal of the ACM*, 41(3):470–516, 1994. doi:10.1145/176584.176586.

6 Avrim Blum and David R. Karger. An $O(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Information Processing Letters*, 61(1):49–53, 1997. doi:10.1016/S0020-0190(96)00190-1.

7 Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Algebraic structure and a symmetric Boolean dichotomy. *SIAM Journal on Computing*, 50(6):1663–1700, 2021. doi:10.1137/19M128212X.

8 Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. SDPs and robust satisfiability of promise CSP. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 609–622, 2023. doi:10.1145/3564246.3585180.

9 Panagiotis Cheilaris and Géza Tóth. Graph unique-maximum and conflict-free colorings. *Journal of Discrete Algorithms*, 9(3):241–251, 2011. doi:10.1016/J.JDA.2011.03.005.

10 Hui Chen and Alan Frieze. Coloring bipartite hypergraphs. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 345–358, 1996. doi:10.1007/3-540-61310-2_26.

11 Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 344–353, 2006. doi:10.1145/1132516.1132567.

12 Irit Dinur, Oded Regev, and Clifford D. Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, 2005. doi:10.1007/S00493-005-0032-4.

13 Miron Ficak, Marcin Kozik, Miroslav Olsák, and Szymon Stankiewicz. Dichotomy for symmetric Boolean PCSPs. In *46th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 132 of *LIPIcs*, pages 57:1–57:12, 2019. doi:10.4230/LIPICS.ICALP.2019.57.

14 Marek Filakovský, Tamio-Vesa Nakajima, Jakub Opršal, Gianluca Tasinato, and Uli Wagner. Hardness of linearly ordered 4-colouring of 3-colourable 3-uniform hypergraphs. In *41st International Symposium on Theoretical Aspects of Computer Science, (STACS)*, volume 289 of *LIPIcs*, pages 34:1–34:19, 2024. doi:10.4230/LIPICS.STACS.2024.34.

15 Magnús M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appl.*, 4(1):1–16, 2000. doi:10.7155/JGAA.00020.

16 Johan Håstad, Björn Martinsson, Tamio-Vesa Nakajima, and Stanislav Živný. A logarithmic approximation of linearly-ordered colourings. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, August 28-30, 2024, London School of Economics, London, UK*, volume 317 of *LIPIcs*, pages 7:1–7:6. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.APPROX/RANDOM.2024.7.

17 David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998. doi:10.1145/274787.274791.

18 Meir Katchalski, William McCuaig, and Suzanne Seager. Ordered colourings. *Discrete Mathematics*, 1(142):141–154, 1995. doi:10.1016/0012-365X(93)E0216-Q.

19 Ken-ichi Kawarabayashi and Mikkel Thorup. Coloring 3-colorable graphs with less than $n^{1/5}$ colors. *Journal of the ACM*, 64(1):4:1–4:23, 2017. doi:10.1145/3001582.

20 Michael Krivelevich, Ram Nathaniel, and Benny Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *Journal of Algorithms*, 41(1):99–113, 2001. doi:10.1006/jagm.2001.1173.

21 Anand Louis, Alantha Newman, and Arka Ray. Improved linearly ordered colorings of hypergraphs via SDP rounding. *CoRR*, abs/2405.00427, 2024. `doi:10.48550/arXiv.2405.00427`.

22 Tamio-Vesa Nakajima and Stanislav Živný. Linearly ordered colourings of hypergraphs. *ACM Trans. Comput. Theory*, 14(3-4):1–19, 2022. `doi:10.1145/3570909`.

23 Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *Journal of the ACM*, 30(4):729–735, 1983. `doi:10.1145/2157.2158`.

24 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

25 Marcin Wrochna and Stanislav Živný. Improved hardness for H-colourings of G-colourable graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 1426–1435, 2020. `doi:10.1137/1.9781611975994.86`.

A Properties of Gaussian

Let function $\Phi : \mathsf{R} \to [0,1]$ be defined as $\Phi(t) \overset{\text{def}}{=} \mathbb{P}_{g\sim\mathcal{N}(0,1)}[g \leqslant t]$, and let function $\bar{\Phi} : \mathsf{R} \to [0,1]$ be defined as $\bar{\Phi}(t) \overset{\text{def}}{=} \mathbb{P}_{g\sim\mathcal{N}(0,1)}[g \geqslant t]$.

▶ **Fact 21.** *For any $a \leqslant b$, we have $\bar{\Phi}(b) - \bar{\Phi}(a) = \mathbb{P}_{g\sim\mathcal{N}(0,1)}[g \in [a,b]] \leqslant \frac{b-a}{\sqrt{2\pi}}$.*

Proof. The statement follows from the following computations.

$$\mathbb{P}[g \in [a,b]] = \int_a^b \frac{e^{-x^2/2}}{\sqrt{2\pi}}\,dx \leqslant \frac{1}{\sqrt{2\pi}} \int_a^b \sup_{y\in\mathbb{R}} e^{-y^2/2}\,dx = \frac{b-a}{\sqrt{2\pi}}.$$ ◀

▶ **Fact 22** (Folklore)**.** *For every $t > 0$,*

$$\frac{t}{\sqrt{2\pi}(t^2+1)} e^{-\frac{1}{2}t^2} < \bar{\Phi}(t) < \frac{1}{\sqrt{2\pi}t} e^{-\frac{1}{2}t^2}.$$

▶ **Corollary 23** (Folklore)**.** *Fix $t \geqslant 1$ and let $\beta = \bar{\Phi}(t)$. Then we have*

$$\sqrt{2\ln\frac{1}{\beta} - \ln\ln\frac{1}{\beta} - \ln 16\pi} \leqslant t \leqslant \sqrt{2\ln\frac{1}{\beta} - \ln\ln\frac{1}{\beta}} \leqslant \sqrt{2\ln\frac{1}{\beta}}.$$

In fact, $t < \sqrt{2\ln\frac{1}{\beta}}$ holds even if $t \in (0,1)$.

Proof. Let $t > 0$ (note here we allow $t \in (0,1)$) and let $\beta = \bar{\Phi}(t)$. By taking logarithm and multiplying by -2, the inequalities in Fact 22 imply

$$2\ln\left(\frac{1}{\beta}\right) > t^2 + 2\ln\left(\sqrt{2\pi}t\right), \tag{4}$$

$$2\ln\left(\frac{1}{\beta}\right) < t^2 + 2\ln\left(\sqrt{2\pi}\left(\frac{t^2+1}{t}\right)\right). \tag{5}$$

We can now use (4) to get

$$2\ln\left(\frac{1}{\beta}\right) > t^2 + 2\ln\left(\sqrt{2\pi}t\right) \geqslant t^2.$$

Hence, we have $t < \sqrt{2\ln\frac{1}{\beta}}$ for any $t > 0$. Again, by multiplying by $\frac{1}{2}$ and taking logarithms, the (4), (5) imply

$$\ln\ln\left(\frac{1}{\beta}\right) > \ln\left(t^2/2 + \ln\left(\sqrt{2\pi}t\right)\right), \tag{6}$$

$$\ln\ln\left(\frac{1}{\beta}\right) < \ln\left(t^2/2 + \ln\left(\sqrt{2\pi}\left(\frac{t^2+1}{t}\right)\right)\right). \tag{7}$$

From hereon we assume $t \geqslant 1$. (4) − (7) gives us

$$2\ln\left(\frac{1}{\beta}\right)-\ln\ln\left(\frac{1}{\beta}\right) > t^2+\ln\left(\frac{2\pi t^2}{t^2/2+\ln\left(\sqrt{2\pi}\left(\frac{t^2+1}{t}\right)\right)}\right) = t^2+\ln\left(\frac{4\pi t^2}{t^2+2\ln\left(\sqrt{2\pi}\left(\frac{t^2+1}{t}\right)\right)}\right) \quad (8)$$

▷ Claim 24. $4\pi t^2 \geqslant t^2 + 2\ln\left(\sqrt{2\pi}\left(\frac{t^2+1}{t}\right)\right)$.

Proof. Note that the above inequality is equivalent to $\left(\frac{4\pi-1}{2}\right)t^2 \geqslant \ln\sqrt{2\pi}+\ln\left(t+\frac{1}{t}\right)$. Indeed we have

$$\begin{aligned}
\ln\sqrt{2\pi}+\ln\left(t+\frac{1}{t}\right) &\leqslant \ln\sqrt{2\pi}+\ln(t+1) && (t \geqslant 1)\\
&\leqslant \ln\sqrt{2\pi}+t && (\ln(1+x)\leqslant x)\\
&\leqslant \ln\sqrt{2\pi}+t^2 && (t \geqslant 1)\\
&\leqslant \left(\frac{4\pi-3}{2}\right)+t^2 && \left(\frac{4\pi-3}{2} \geqslant \ln\sqrt{2\pi}\right)\\
&\leqslant \left(\frac{4\pi-1}{2}\right)t^2 && (t \geqslant 1)
\end{aligned}$$

◁

Using this claim and (8) we get

$$t^2 \leqslant 2\ln\frac{1}{\beta}-\ln\ln\frac{1}{\beta}.$$

Hence, we have $t \leqslant \sqrt{2\ln\frac{1}{\beta}-\ln\ln\frac{1}{\beta}}$. For the remaining inequality, we again see that (5) − (6) gives us

$$\begin{aligned}
2\ln\frac{1}{\beta}-\ln\ln\frac{1}{\beta} &< t^2+\ln\left(\frac{4\pi\left(t+\frac{1}{t}\right)}{t^2+2\ln(\sqrt{2\pi}t)}\right)\\
&\leqslant t^2+\ln 4\pi+\ln\left(\frac{(t+1)^2}{t^2}\right) && (t \geqslant 1)\\
&\leqslant t^2+\ln 4\pi+2\ln\left(1+\frac{1}{t}\right)\\
&\leqslant t^2+\ln 4\pi+2\ln 2 = t^2+\ln 16\pi && (t \geqslant 1)
\end{aligned}$$

$\sqrt{2\ln\frac{1}{\beta}-\ln\ln\frac{1}{\beta}-\ln 16\pi} \leqslant t$ follows from the above inequality. Hence, we have all the required inequalities. ◀

▶ **Corollary 25** (Folklore). *Fix $t \geqslant 1$. Then, we have*

$$\bar{\Phi}(2t) \leqslant 512\left(\ln\left(\frac{1}{\bar{\Phi}(t)}\right)\right)^{3/2}\bar{\Phi}(t)^4.$$

Proof. For any $t \geqslant 1$ and $\delta \in (0,1)$ the following holds.

$$\begin{aligned}
\bar{\Phi}(2t) &\leqslant \frac{1}{2\sqrt{2\pi}t} e^{-2t^2} && \text{(Fact 22)}\\
&\leqslant \frac{1}{2\sqrt{2\pi}t} \cdot \frac{(2\pi)^2 \left(t^2+1\right)^4}{t^4} \cdot \bar{\Phi}(t)^4 && \left(\frac{t}{\sqrt{2\pi}(t^2+1)} e^{-\frac{1}{2}t^2} \leqslant \bar{\Phi}(t) \text{ by Fact 22}\right)\\
&= (2\pi)^{3/2} \frac{1}{2t} \left(t + \frac{1}{t}\right)^4 \bar{\Phi}(t)^4 &&\\
&\leqslant (2\pi)^{3/2} (2t)^3 \bar{\Phi}(t)^4 && (t \geqslant 1)\\
&\leqslant (4\sqrt{\pi})^3 \left(\ln\left(\frac{1}{\bar{\Phi}(t)}\right)\right)^3 \cdot \bar{\Phi}(t)^4 && \text{(by Corollary 23)}\\
&\leqslant 512 \left(\ln\left(\frac{1}{\bar{\Phi}(t)}\right)\right)^3 \bar{\Phi}(t)^4 && (\sqrt{\pi} \leqslant 2).
\end{aligned}$$

◀

B Coloring of 2-LO Colorable 3-Uniform Hypergraphs

In this section, we show how to color a 2-LO colorable 3-uniform hypergraph with 2 colors. For simplicity, we define *balanced* vertices to be those with $\gamma_a = -1/3$. It is straightforward to extend to the case of ε-balanced for small but (strictly) positive ε.

Algorithm 3 2-Coloring Algorithm.

Input: A solution to SDP 2.

Two-Sided Combinatorial Rounding

1. Set $S_l := \left\{a \in V \mid \gamma_a < -\frac{1}{3}\right\}$, $S_r := \left\{a \in V \mid \gamma_a > -\frac{1}{3}\right\}$, and $S_b := \left\{a \in V \mid \gamma_a = -\frac{1}{3}\right\}$.
2. Color S_l using color 1 and S_r using color 2.

Two-Sided Hyperplane Rounding

resume Choose a uniformly random unit vector r over the sphere.

resume Set $H_r := \{a \in S_b \mid \langle r, \mathsf{u}_a\rangle \geqslant 0\}$ and $H_l := \{a \in S_b \mid \langle r, \mathsf{u}_a\rangle < 0\}$.

resume Color H_l using 1 and H_r using 2.

▶ **Lemma 26.** *Let S_l, S_r be as defined in Algorithm 3. Then, for any edge $e \in E$, we have $S_l \cap e \neq \emptyset$ if and only if $S_r \cap e \neq \emptyset$.*

Proof. Fix an edge $\{a,b,c\} = e \in E$. Observation 14 states that $\gamma_a + \gamma_b + \gamma_c = -1$ Suppose that $e \cap S_l \neq \emptyset$ and $e \subseteq S_l \cup S_b$. Then, we get $\gamma_a + \gamma_b + \gamma_c < -1$, a contradiction. Hence, $e \cap S_l \neq \emptyset$ implies $e \cap S_r \neq \emptyset$ The proof of the converse is similar. ◀

▶ **Lemma 27.** *Let r be unit vector distributed uniformly over the sphere and let H_l, and H_r be defined as in Algorithm 3. Then, for any edge $e \in E$, we have $|H_l \cap e| \leqslant 2$ and $|H_r \cap e| \leqslant 2$ (with probability 1).*

Proof. We can assume that $H_r = \{a \in S_b | \langle r, \mathsf{u}_a\rangle > 0\}$, as this is true with probability 1. Fix any $e \in E$. Assume that $\{a,b,c\} = e \subseteq S_b$; otherwise, we are done. Using Lemma 18 with $\varepsilon = 0$ we get $\|\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c\|^2 = 0$, which implies $\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c = 0$. Therefore, we have $\langle r, \mathsf{u}_a\rangle + \langle r, \mathsf{u}_b\rangle + \langle r, \mathsf{u}_c\rangle = 0$. But, if $e \subseteq H_l$, then we have $\langle r, \mathsf{u}_a\rangle + \langle r, \mathsf{u}_b\rangle + \langle r, \mathsf{u}_c\rangle < 0$, a contradiction. Similarly, if $e \subseteq H_r$, then we have a contradiction in form of $\langle r, \mathsf{u}_a\rangle + \langle r, \mathsf{u}_b\rangle + \langle r, \mathsf{u}_c\rangle > 0$. ◀

▶ **Theorem 28.** *There is a polynomial-time algorithm that, if given a 2-LO colorable 3-uniform hypergraph H with n vertices, finds an 2-coloring of H.*

Proof. Solve SDP 2 and use Algorithm 3. Consider any edge $e = \{a, b, c\}$. If e is not completely contained in S_b, then by Lemma 26 we at least one vertex each from S_l and S_r; hence, e is non-monochromatic. Otherwise, e is non-monochromatic by Lemma 27. ◀

C Omitted Proofs

Proof of Lemma 18

Before we proceed to prove Lemma 18 we need the following lemma.

▶ **Lemma 29.** *Let $\{a, b, c\} \in E$ and $\gamma_a = -1/3+\varepsilon_a$, $\gamma_b = -1/3+\varepsilon_b$, $\gamma_c = -1/3+\varepsilon_c$. Then the following hold.*

1. $\varepsilon_a + \varepsilon_b + \varepsilon_c = 0$.
2. $\langle \mathsf{v}_a, \mathsf{v}_b\rangle = -1/3+\varepsilon_c$, $\langle \mathsf{v}_b, \mathsf{v}_c\rangle = -1/3+\varepsilon_a$, *and* $\langle \mathsf{v}_c, \mathsf{v}_a\rangle = -1/3+\varepsilon_b$.

Proof. Using Observation 14 we get

$$-1/3+\varepsilon_a - 1/3+\varepsilon_b - 1/3+\varepsilon_c = -1,$$

which implies $\varepsilon_a+\varepsilon_b+\varepsilon_c = 0$. Taking inner products with $\mathsf{v}_a, \mathsf{v}_b, \mathsf{v}_c$ on both sides of constraint (1) of SDP 2 we get

$$\begin{aligned}
1 + \langle \mathsf{v}_a, \mathsf{v}_b\rangle + \langle \mathsf{v}_a, \mathsf{v}_c\rangle &= 1/3 - \varepsilon_a,\\
\langle \mathsf{v}_b, \mathsf{v}_a\rangle + 1 + \langle \mathsf{v}_b, \mathsf{v}_c\rangle &= 1/3 - \varepsilon_b,\\
\langle \mathsf{v}_c, \mathsf{v}_a\rangle + \langle \mathsf{v}_c, \mathsf{v}_b\rangle + 1 &= 1/3 - \varepsilon_c,
\end{aligned}$$

which imply

$$\langle \mathsf{v}_a, \mathsf{v}_b\rangle + \langle \mathsf{v}_a, \mathsf{v}_c\rangle = -(2/3+\varepsilon_a), \tag{9}$$
$$\langle \mathsf{v}_b, \mathsf{v}_a\rangle + \langle \mathsf{v}_b, \mathsf{v}_c\rangle = -(2/3+\varepsilon_b), \tag{10}$$
$$\langle \mathsf{v}_c, \mathsf{v}_a\rangle + \langle \mathsf{v}_c, \mathsf{v}_b\rangle = -(2/3+\varepsilon_c). \tag{11}$$

(9)+(10)−(11) gives us

$$2\langle \mathsf{v}_a, \mathsf{v}_b\rangle = -2/3 - \varepsilon_a - \varepsilon_b + \varepsilon_c.$$

Using Item 1 of this lemma and dividing by 2 we get $\langle \mathsf{v}_a, \mathsf{v}_b\rangle = -1/3+\varepsilon_c$ as needed. Similarly, we get $\langle \mathsf{v}_a, \mathsf{v}_c\rangle = -1/3+\varepsilon_b$, $\langle \mathsf{v}_c, \mathsf{v}_b\rangle_c = -1/3+\varepsilon_a$. ◀

Proof of Lemma 18. Note that

$$\begin{aligned}
\langle \mathsf{u}_a, \mathsf{u}_b\rangle &= \left\langle \frac{\mathsf{v}_a - \gamma_a \mathsf{v}_\emptyset}{\sqrt{1-\gamma_a^2}}, \frac{\mathsf{v}_b - \gamma_b \mathsf{v}_\emptyset}{\sqrt{1-\gamma_b^2}} \right\rangle = \frac{\langle \mathsf{v}_a, \mathsf{v}_b\rangle - \gamma_a \langle \mathsf{v}_\emptyset, \mathsf{v}_b\rangle - \gamma_b \langle \mathsf{v}_\emptyset, \mathsf{v}_a\rangle + \gamma_a\gamma_b \langle \mathsf{v}_\emptyset, \mathsf{v}_\emptyset\rangle}{\sqrt{(1-\gamma_a^2)(1-\gamma_b^2)}}\\
&= \frac{\langle \mathsf{v}_a, \mathsf{v}_b\rangle - \gamma_a\gamma_b}{\sqrt{(1-\gamma_a^2)(1-\gamma_b^2)}}.
\end{aligned}$$

First let us upper-bound the denominator in the above expression using $\gamma_a, \gamma_b \in [-1/3 - \epsilon, -1/3 + \epsilon]$ as follows.

$$\begin{aligned}\sqrt{(1-\gamma_a^2)(1-\gamma_b^2)} &\leqslant \sqrt{(1-(1/3-\epsilon)^2)(1-(1/3-\epsilon)^2)} \\ &= \frac{8}{9} + \frac{2\epsilon}{3} - \epsilon^2 \\ &\leqslant \frac{8}{9} + \frac{2\epsilon}{3}.\end{aligned}$$

This implies that

$$\begin{aligned}\frac{1}{\sqrt{(1-\gamma_a^2)(1-\gamma_b^2)}} &\geqslant \frac{1}{\frac{8}{9}\left(1+\frac{9\epsilon}{4}\right)} \\ &\geqslant \frac{9}{8}\left(1 - \frac{9\epsilon}{4} + \frac{\left(\frac{9\epsilon}{4}\right)^2}{\left(1+\frac{9\epsilon}{4}\right)}\right) \\ &\geqslant \frac{9}{8}\left(1 - \frac{9\epsilon}{4}\right).\end{aligned}$$

By Lemma 29, we have $\langle \mathsf{v}_a, \mathsf{v}_b \rangle \in [-1/3 - \epsilon, -1/3 + \epsilon]$. So, we can also bound the numerator in the expression for $\langle \mathsf{u}_a, \mathsf{u}_b \rangle$ by using the fact that $\langle \mathsf{v}_a, \mathsf{v}_b \rangle, \gamma_a, \gamma_b \in [-1/3 - \epsilon, -1/3 + \epsilon]$ as follows.

$$\begin{aligned}\langle \mathsf{v}_a, \mathsf{v}_b \rangle - \gamma_a \gamma_b &\leqslant -\frac{1}{3} + \epsilon - \left(\frac{1}{3} - \epsilon\right)^2 \\ &= -\frac{4}{9} + \frac{5\epsilon}{3} - \epsilon^2 \\ &\leqslant -\frac{4}{9} + \frac{5\epsilon}{3}.\end{aligned}$$

Therefore, we get

$$\begin{aligned}\langle \mathsf{u}_a, \mathsf{u}_b \rangle &\leqslant -\frac{4}{9}\left(1 - \frac{15\epsilon}{4}\right) \cdot \frac{9}{8}\left(1 - \frac{9\epsilon}{4}\right) \\ &\leqslant -\frac{1}{2}(1 - 6\epsilon).\end{aligned}$$

Finally, for the edge $\{a, b, c\}$ we get

$$\begin{aligned}\|\mathsf{u}_a + \mathsf{u}_b + \mathsf{u}_c\|^2 &= \|\mathsf{u}_a\|^2 + \|\mathsf{u}_b\|^2 + \|\mathsf{u}_c\|^2 + 2\langle \mathsf{u}_a, \mathsf{u}_b \rangle + 2\langle \mathsf{u}_b, \mathsf{u}_c \rangle + 2\langle \mathsf{u}_c, \mathsf{u}_a \rangle \\ &= 3 + 2\left(\langle \mathsf{u}_a, \mathsf{u}_b \rangle + \langle \mathsf{u}_b, \mathsf{u}_c \rangle + \langle \mathsf{u}_c, \mathsf{u}_a \rangle\right) \\ &\leqslant 18\varepsilon\end{aligned}$$

where the last inequality follows from the fact that $\langle \mathsf{u}_a, \mathsf{u}_b \rangle$, $\langle \mathsf{u}_b, \mathsf{u}_c \rangle$, and $\langle \mathsf{u}_c, \mathsf{u}_a \rangle$ are all at most $-\frac{1}{2} + 3\varepsilon$. ◀