

$(2 + \varepsilon)$ -SAT is NP-hard

Per Austrin* Venkatesan Guruswami† Johan Håstad*

* *School of Computer Science & Communication
KTH Royal Institute of Technology
Stockholm, Sweden
Email: austrin,johanh@kth.se*

† *Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
Email: guruswami@cmu.edu*

Abstract—We prove the following hardness result for a natural promise variant of the classical CNF-satisfiability problem: Given a CNF-formula where each clause has width w and the guarantee that there exists an assignment satisfying at least $g = \lceil \frac{w}{2} \rceil - 1$ literals in each clause, it is NP-hard to find a satisfying assignment to the formula (that sets at least one literal to true in each clause). On the other hand, when $g = \lceil \frac{w}{2} \rceil$, it is easy to find a satisfying assignment via simple generalizations of the algorithms for 2-SAT.

Viewing 2-SAT \in P as easiness of SAT when 1-in-2 literals are true in every clause, and NP-hardness of 3-SAT as intractability of SAT when 1-in-3 literals are true, our result shows, for any fixed $\varepsilon > 0$, the hardness of finding a satisfying assignment to instances of “ $(2 + \varepsilon)$ -SAT” where the density of satisfied literals in each clause is promised to exceed $\frac{1}{2+\varepsilon}$.

We also strengthen the results to prove that given a $(2k+1)$ -uniform hypergraph that can be 2-colored such that each edge has perfect balance (at most $k+1$ vertices of either color), it is NP-hard to find a 2-coloring that avoids a monochromatic edge. In other words, a set system with discrepancy 1 is hard to distinguish from a set system with worst possible discrepancy.

Keywords—Constraint satisfaction, complexity dichotomy, discrepancy, polymorphisms, probabilistically checkable proofs, promise problems.

I. INTRODUCTION

One of the first distinctions we learn in complexity theory is that while 2-SAT can be solved in polynomial time, 3-SAT is our favorite NP-complete problem. As there are no integers between 2 and 3 this seems to be a sharp characterization but a closer inspection shows that a more fine-grained analysis is possible. One conclusion of the current paper is that the transition from easy to hard takes place *just after two* and not just before three.

P. Austrin’s research was supported by ERC Advanced Grant 226203 and Swedish Research Council Grant 621-2012-4546.

V. Guruswami’s research was supported in part by a Packard Fellowship and NSF grant CCF-1115525.

J. Håstad’s was research supported by ERC Advanced Grant 226203, and was carried out in part while visiting the Simons institute.

Suppose we consider w -CNF formulas where each clause is of width exactly w and ask for an assignment that satisfies a literals in each clause. It follows more or less immediately from the above facts that this problem is NP-hard for $a \leq w - 2$ and solvable in polynomial time when $a \geq w - 1$. Suppose, however, that we make this into a promise problem and guarantee that there is an assignment that satisfies g literals in each clause for some $g > a$. We call the resulting problem (a, g, w) -SAT. It turns out that (a, g, w) -SAT can be solved in polynomial time if and only if $(a+1, g+1, w+1)$ -SAT can be solved in polynomial time (we give the short proof of this fact in the preliminaries). We can thus focus on the case $a = 1$.

Let us start with some easy observations. Starting with a 3-CNF formula we can turn this into a $3g$ -CNF formula by taking the union of all g -tuples of clauses. It is easy to see that if the original formula is satisfiable then we can satisfy g literals in the produced formula. From this it follows that $(1, g, w)$ -SAT is NP-hard whenever $g \leq w/3$. To the best of our knowledge, no hardness was known for any $g > w/3$.

On the algorithmic side it is not difficult to see that the probabilistic 2-SAT algorithm of Papadimitriou [1] extends to $(1, g, w)$ -SAT when $g \geq w/2$. Using a linear program we show how to construct an algorithm running in deterministic polynomial time for the same range.

It turns out that this is all that can be achieved in polynomial time. The main new result of this paper is to establish that the problem is NP-hard whenever $g < w/2$. In particular, we have the following theorem.

Theorem 1.1. *For every integer $g \geq 1$, $(1, g, 2g + 1)$ -SAT is NP-hard.*

This hardness result is the source of the above claim that the transition from easy to hard takes place at two. Once the density of satisfied literals drops strictly below one half, it

is hard to find a satisfying assignment.¹

To establish this result we give a reduction from the Label Cover problem, the usual starting point for inapproximability results (even though our results apply only to traditional constraint satisfaction problems where we want to satisfy all constraints). Given the current technology of reductions combining Label Cover with the long code, and the goal of only establishing hardness of deciding satisfiability, the technical details of the proof are not difficult. We first prove that if the clauses produced by the natural dictatorship test are all satisfied then the corresponding function only depends on a constant number of inputs. To pass the constraints that go across two long codes we show that there must be some consistency between these important variables and hence they can be used as labels to satisfy a constant fraction of the constraints in the Label Cover instance. Note that our proof method departs from the usual analytic approaches to analyze dictatorship tests (which seem ill-suited in our context) and relies on more combinatorial reasoning. The fact that our proofs are short and self-contained and yet yield a non-trivial hardness result about such a natural and easy to state variant of SAT is, in our opinion, one of the main selling points of this paper.

At first glance it might seem slightly surprising that such a strong starting point is needed for our NP-hardness result. As an indication that something non-trivial is going on we give a proof that there is no standard gadget reduction from 3-SAT to any of our problems, and in particular not to $(1, g, 2g+1)$ -SAT. This impossibility result is due to Dominik Scheder, and extends to show that there is no gadget reduction from $(1, g, 2g+1)$ -SAT to $(1, g', 2g'+1)$ -SAT for $g' > g \geq 1$.

One can also consider an approximation problem associated with $(1, g, 2g+1)$ -SAT, where where we are guaranteed that there is an assignment that strongly g -satisfies a fraction c of clauses and the goal is to find an assignment that satisfies a fraction s of the clauses. Our proof of Theorem I.1 implicitly shows that this problem is hard for $c = 1$ and some $s = s(g) < 1$, thus showing APX-hardness of $(1, g, 2g+1)$ -SAT. We observe that an application of the theorem on “uselessness of predicates” from [3] implies a strong inapproximability result for this problem (albeit only for almost satisfiable instances and assuming the Unique Games conjecture) that shows hardness for $c = 1 - \varepsilon$ and $s = 1 - 2^{-(2g+1)} + \varepsilon$ for any $\varepsilon > 0$ (see Theorem IV.8).

Hypergraph discrepancy. A problem closely related to (a, g, w) -SAT is *hypergraph discrepancy*. Here, given sets of size $2g+1$ of elements from a universe, the problem

¹This is the sense in which we implied $(2 + \varepsilon)$ -SAT is NP-hard in the paper title, but we should mention here that $(2 + \varepsilon)$ -SAT has been used previously to denote instances of satisfiability containing a mix of 2CNF and 3CNF clauses, with about ε fraction of clauses being 3CNF [2]. As the terminology $(2 + \varepsilon)$ -SAT is restricted to just the title of this paper, we hope it does not cause much confusion.

is to color the elements with two colors such that each set has a good balance of colors. We extend our methods to prove the following hardness result, showing that systems of bounded-size sets with smallest possible discrepancy are NP-hard to distinguish from set systems with worst possible discrepancy.

Theorem I.2. *For $g \geq 1$, given a $(2g+1)$ -uniform hypergraph that admits a 2-coloring under which each hyperedge is evenly balanced (g elements of one color and $g+1$ of the other), it is NP-hard to find a 2-coloring that avoids creating a monochromatic hyperedge.*

The above result implies Theorem I.1 via a simple reduction: for each hyperedge $(x_1, x_2, \dots, x_{2g+1})$ of the hypergraph, create two width $2g+1$ clauses $(x_1 \vee x_2 \vee \dots \vee x_{2g+1})$ and $(\overline{x_1} \vee \overline{x_2} \vee \dots \vee \overline{x_{2g+1}})$. But we prove Theorem I.1 first to illustrate our approach in a simpler setting that allows negated literals and repeated variables within a clause.

For the case of even-sized sets, i.e., $2g$ -uniform hypergraphs for $g \geq 2$, we can get the following statement by a simple reduction from Theorem I.2 (the first statement follows by adding a special private element to each set in the instance, and the second by taking all $2g$ -element subsets of each $(2g+1)$ -sized set in the instance).

Corollary I.3. (i) *For $g \geq 2$, given a $2g$ -uniform hypergraph that admits a 2-coloring under which each hyperedge is perfectly balanced (has g elements of each color), it is NP-hard to find a 2-coloring with discrepancy smaller than $(2g-2)$ (i.e., with more than one vertex of each color in every hyperedge).*

(ii) *For $g \geq 2$, given a $2g$ -uniform hypergraph that admits a 2-coloring with discrepancy at most 2, it is NP-hard to find a 2-coloring that avoids creating a monochromatic hyperedge.*

The above statements are best possible in the sense that if there is a perfectly balanced (discrepancy 0) coloring of a $2g$ -uniform hypergraph, then one can efficiently find a 2-coloring that avoids monochromatic hyperedges. This follows from the more general statement that $(1, g, 2g)$ -SAT can be solved in polynomial time (via the reduction mentioned after Theorem I.2).

For systems with unbounded set size, it is known that even if there is a coloring with discrepancy 0, it is NP-hard to find a coloring of discrepancy $\Omega(\sqrt{N})$ where N is the size of the universe [4].

Connection to weak polymorphisms. The crux of our proof of Theorem I.1 is to show that all “weak polymorphisms” which map assignments satisfying at least g out of $2g+1$ variables to an assignment satisfying at least one of those variables depend only on a small number of variables. (For the proof of Theorem I.2, we make do with a slightly weaker

statement that allows a small number of exceptional inputs.)

To elucidate the underlying principle governing our hardness results, we show that if the only weak polymorphisms mapping satisfying assignments of a predicate P to those of an implied predicate Q are *juntas* (i.e., depend on a fixed constant number of variables), then the associated promise CSP associated with the pair of predicates P, Q (where we are promised satisfiability according to P and the goal is to find an assignment satisfying the more lax predicate Q) is NP-hard. Details of this result can be found in the full paper [5].

Apart from the inherent interest in the given problems it is our hope that these new results will be useful as starting points for reductions to give new inapproximability results. While the inapproximability of Max-3Lin [6] is a good starting point for problems where we are counting the number of satisfied constraints, our new problems might be good starting points when it is the *worst local situation* that governs the quality of a solution. As a small step in this direction we use our result to improve the inapproximability result for hereditary discrepancy for matrices from $3/2$ to any number arbitrarily close to 2. Our Theorem I.2 was also used recently in [7] to show a tight factor 2 inapproximability for a certain scheduling problem on two machines.

Subsequent work. A natural direction left open by our work was extending our results to domain size larger than 2. This challenge is better explained in the context of hypergraph coloring. Let us define a weak ℓ -coloring of a hypergraph to be a coloring of its vertices with ℓ colors so that no hyperedge is monochromatic. One way to generalize Theorem I.2 would be to show hardness of weakly 2-coloring $(k+1)$ -uniform hypergraphs which are promised to have a coloring with k colors such that every hyperedge has vertices of *each* color². Proving such a hardness is still open. However, in a very recent development [8], the following strong inapproximability result for coloring hypergraphs of larger uniformity has been shown:

For arbitrary integers $g, k, C \geq 2$, given as input a gk -uniform hypergraph that is promised to have a k -coloring where each color appears at least $(g-1)$ times in every hyperedge, it is NP-hard to weakly C -color the hypergraph.

While this result does not imply Theorem I.2, taking $k = 2$ it implies part (ii) of Corollary I.3, and in fact a stronger form when an arbitrary constant number of colors are allowed in the soundness case (part (i) also follows by a simple reduction).

²Note that such a hypergraph surely has a balanced 2-coloring, by merging $\lfloor \frac{k}{2} \rfloor$ colors into one group and the remaining $\lceil \frac{k}{2} \rceil$ colors into another group, so such a result will indeed strengthen Theorem I.2. If the hypergraph is k -uniform (instead of $k+1$ -uniform), one can efficiently weakly 2-color it (e.g. via a random walk algorithm similar to the one in Section VII).

The proof of the above result in [8] is significantly more involved than our proofs, relying on analytic machinery such as invariance principles applied to general correlated spaces [9], [10] and reverse hypercontractivity, and requiring a smooth version of Label Cover for the reduction. The authors of [8] are also able to adapt their techniques to give an analytic proof of Theorem I.1. Establishing our main result on discrepancy (Theorem I.2), however, seems currently out of reach of these analytic methods. Besides, the main appeal of our work is the novelty of using long code based reductions in an unusual manner and the simplicity with which it yields new non-trivial statements about natural variants of SAT and NOT-ALL-EQUAL-SAT.

Organization. An outline of the paper is as follows. We start with some definitions and preliminaries in Section II. As our main contribution is on the hardness side, we discuss the hardness results first and defer the algorithmic results to Section VII towards the end of the paper. Before presenting our hardness result for $(1, g, 2g+1)$ -SAT in Section IV, we take a brief “polymorphic” detour in Section III, proving the non-existence of a simple gadget reduction from 3-SAT to $(1, g, 2g+1)$ -SAT, and also commenting on the general principle underlying our hardness result.

We present the extension of our hardness results to hypergraph discrepancy (Theorem I.2) in Section V, where we also show how to ensure that all variables appearing in a constraint are distinct. We give the application to hereditary discrepancy in Section VI, and finally we end with some concluding remarks in Section VIII.

II. PRELIMINARIES

We start with some basic definitions.

Definition II.1. A w -SAT formula is a CNF formula where each clause has *width* exactly w .

Definition II.2. A w -SAT formula Φ is *strongly g -satisfiable* if there is an assignment to the variables such that at least g literals are true in every clause of Φ .

Definition II.3. For $1 \leq a \leq g < w$, the (a, g, w) -SAT promise problem is as follows. The input is a w -SAT formula Φ and the goal is to accept instances Φ that are strongly g -satisfiable and reject instances that do not admit any assignment that strongly a -satisfies Φ .

We have defined the decision version above, and in the search version we are given a w -SAT formula Φ that is *guaranteed* to be strongly g -satisfiable and the goal is to find an assignment that strongly a -satisfies Φ .

Note that $(1, 1, w)$ -SAT is the usual w -SAT problem. Let us start with a couple of simple observations.

Observation II.4. *There is a polynomial time reduction from (a, g, w) -SAT to $(a, g, w+1)$ -SAT*

Proof: For each old clause, create two new clauses extending it by a variable and its complement. ■

Proposition II.5. *The problems (a, g, w) -SAT and $(a + 1, g + 1, w + 1)$ -SAT are interreducible to each other in polynomial time. (Thus one of them is polynomial time solvable iff the other one is.)*

Proof: We establish two easy reductions and start with the obvious one.

If we have an instance of (a, g, w) -SAT, then adding the same new variable to all clauses gives an instance of $(a + 1, g + 1, w + 1)$ -SAT of the same difficulty.

For the reduction in the other direction, take all subclauses of size w of each clause of size $w + 1$. It is readily verified that this gives a correct reduction. ■

In view of the above proposition we can focus on the case $a = 1$, i.e., the problem of finding a satisfying assignment in a w -CNF formula when we are guaranteed that there is an assignment that satisfies at least g literals in each clause.

One detail to consider is whether we allow repeated literals or two literals corresponding to the same variable in a clause. It turns out that this distinction does not change the complexity of the problem. While our algorithms apply directly to the general case, in the hardness results the requirement to use only distinct literals does create some slight technical complications. Hence we first give a cleaner proof of Theorem I.1 where we allow repeated literals, and then present a full proof of the stronger result for hypergraph discrepancy, Theorem I.2, in which we ensure that all variables in a constraint are distinct (i.e., the hypergraph is $(2g + 1)$ -uniform). We now define the discrepancy problem underlying Theorem I.2 formally.

Let S be a subset of size $2g + 1$ of some universe U . We say that $X \subseteq U$ splits S evenly if $|X \cap S| \in \{g, g + 1\}$.

Definition II.6. An instance of the g -DISCREPANCY problem consists of a collection of sets $S_1, \dots, S_m \subset U$ each of size exactly $2g + 1$, and the objective is to distinguish between

- Yes: there is an $X \subseteq U$ that splits each S_i evenly.
- No: for every $X \subseteq U$, some S_i is not split by X at all (i.e., $|X \cap S_i| \in \{0, 2g + 1\}$).

Label Cover and Long Codes. Our reductions establishing hardness results fit in the standard form for Probabilistically Checkable Proofs (PCPs), commonly used to establish inapproximability results for maximum constraint satisfaction problems. In particular our reductions start from label cover and use the long code encoding of each label.

Definition II.7. An instance $\Psi = (U, V, E, \{\pi_e : L_V \rightarrow L_U\})$ of Label Cover consists of a bipartite graph (U, V, E) , label sets L_U and L_V for U and V , and for each edge $e \in E$

a map $\pi_e : L_V \rightarrow L_U$.

A labeling ℓ is a map that assigns for each $u \in U$ a label $\ell(u) \in L_U$ and for each $v \in V$ a label $\ell(v) \in L_V$. The labeling ℓ satisfies an edge $e = (u, v)$ if $\pi_e(\ell(v)) = \ell(u)$.

The value of a labeling ℓ is the fraction of edges satisfied by ℓ , and the value $\text{Opt}(\Psi)$ of Ψ is the maximum value of any labeling.

Theorem II.8 ([11], [12]). *For every $\varepsilon > 0$, there are L_U, L_V such that given a Label Cover instance Ψ with label sets L_U and L_V it is NP-hard to distinguish between $\text{Opt}(\Psi) = 1$ and $\text{Opt}(\Psi) \leq \varepsilon$.*

The long code of $\ell \in L$ is a table $f : \{0, 1\}^L \rightarrow \{0, 1\}$ where $f(x) = x_\ell$. Whenever negation is available in the CSP for which we try to prove a lower bound we assume that tables are odd and respect negation (in standard PCP terminology “are folded”) i.e. that $f(\neg x) = \neg f(x)$ where \neg is a negation operator that works both on bits and strings (by negating each bit individually). The oddness of f is ensured by storing, for each pair $(x, \neg x)$, only the value $f(x)$ and if $f(\neg x)$ is needed then $\neg f(x)$ is used instead. We note that this is possible for our results for satisfiability but not for the hypergraph discrepancy problem which does not allow negations in the constraints.

We use some standard notation in the paper. We let $e_i \in \{0, 1\}^n$ be the unit vector with a one in position i and use \oplus to denote exclusive-or. Thus if $x \in \{0, 1\}^n$ is any assignment, x and $x \oplus e_i$ differ in exactly coordinate i .

III. A POLYMORPHIC DETOUR

Before proceeding with the actual proof that $(1, g, 2g + 1)$ -SAT is NP-hard let us gather some intuition for the problem by proving that there is no simple gadget reduction from standard 3-SAT, and also comment on the polymorphic principle governing our hardness results (whose detailed description can be found in the full paper).

Let us first consider the possibility of a gadget reduction from 3-SAT to $(1, 2, 5)$ -SAT. For each clause, say $(x_1 \vee x_2 \vee x_3)$, this hypothetical reduction introduces a number of auxiliary variables a_i (which are particular to this clause) and forms a number of constraints in the form of clauses of width 5. This reduction should satisfy:

- 1) Completeness: for each assignment to x_1, x_2 and x_3 that satisfies the original clause there is an assignment to the auxiliary variables such that at least two literals in each new clause are true, and
- 2) Soundness: if x_1, x_2, x_3 are all set to false, no assignment to the auxiliary variables satisfies all the new clauses.

We have the following observation by Dominik Scheder that such a gadget reduction does not exist.

Proposition III.1. *There is no gadget reduction from 3-SAT to $(1, 2, 5)$ -SAT.*

Proof: Consider the three cases when (x_1, x_2, x_3) takes the values $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, respectively. Consider the good assignment to the auxiliary variables in each of these three cases, satisfying at least two literals in each created clause. Define a new assignment to the auxiliary variables as the majority of these three assignments. We claim that this new assignment satisfies at least one literal in each created clause even when the x_i 's are all false.

To see this, look at a single clause of width 5 and consider the values of these 5 literals under the three assignments. In total we have at least 6 satisfied literals. Thus, one of the five literals is satisfied in two of the three assignments, and this literal is then satisfied also in the majority assignment. ■

It is not difficult to see that the proof extends to prove that there is no gadget reduction from $(1, g, 2g + 1)$ -SAT to $(1, g', 2g' + 1)$ -SAT for $g' > g$. We simply take $2g + 1$ assignments with g true literals in each such that each literal is true in g assignments. A majority of these assignments does the trick as in the above argument.

For readers familiar with the notion of polymorphisms from the CSP dichotomy theory, we note that what we have established is that majority of $2g + 1$ bits is a type of “weak polymorphism”. Namely, if each input strongly g' -satisfies a width $(2g' + 1)$ clause then the output must satisfy the clause if $g' > g$ but not for smaller values of g' . Moreover if $w \leq 2g$ then any odd majority is a polymorphism that takes assignments for width w clause that is strongly g -satisfying into a satisfying assignment. Our hardness result for $(1, g, 2g + 1)$ -SAT are based on a proof that such polymorphisms for $w > 2g$ can only depend on few variables.

One can consider the $(1, g, 2g + 1)$ -SAT problem as an instance of a large family of problems parametrized by two predicates P and Q of the same arity k , with P implying Q . One is given a large number of k -tuples of literals and the promise that there is an assignment such that all resulting k -tuples of bits satisfy P . The task now is to find an assignment such that the strings instead satisfy Q . It turns out that the lack of non-trivial polymorphisms on arbitrarily many variables that take satisfying assignments for P and map it to satisfying assignments for Q implies that this promise P vs. Q constraint satisfaction problem is NP-hard. We present a formal statement and proof of this connection between non-existence of polymorphisms and intractability in the full paper.

IV. NP-HARDNESS OF $(1, g, 2g + 1)$ -SAT

We now return to the goal of establishing that $(1, g, 2g + 1)$ -SAT is NP-hard. In what follows we write $w = 2g + 1$.

In order to simplify the presentation and cleanly convey the general idea behind the reduction without having to pay too much attention to technical details, we here allow repeated literals in each clause. Proving the result without repeated literals can be done with a little more care, using the same approach as in Section V, where we prove a stronger result for the discrepancy problem (Theorem I.2) without allowing repeated variables in a constraint.

A. A Dictatorship Gadget

First, we construct a dictatorship gadget, which is an instance defined over 2^n variables, viewed as a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which is, as discussed in the preliminaries, assumed to be folded.

The constraints on f are all clauses of the form $(f(x_1) \vee f(x_2) \vee \dots \vee f(x_w))$ where x_1, \dots, x_w are such that for each $j \in [n]$, $\sum_{i=1}^w x_{i,j} \geq g$. In other words there are at least g ones in each coordinate.

The completeness of the gadget (stated below) follows by construction.

Lemma IV.1. *If f is a dictatorship function then it strongly g -satisfies the dictatorship gadget.*

The converse of the above lemma is only true in a weaker sense.

Lemma IV.2. *Any assignment f which is odd and satisfies the dictatorship gadget depends on at most $2g - 1$ variables.*

In fact, this lemma is sharp – as noted in the previous section, the majority of $2g - 1$ variables does satisfy the gadget. The essential part of the above lemma is contained in the following claim that we establish first.

Claim IV.3. *Suppose f is odd, depends on g different variables i_1, \dots, i_g , and satisfies the dictatorship gadget. Then $f(z) = 1$ for all inputs z such that $z_{i_1} = \dots = z_{i_g} = 1$.*

Proof: Suppose for contradiction that there is an input z such that $f(z) = 0$ yet $z_{i_j} = 1$ for all $j \in [g]$. Since f depends on variables i_1, \dots, i_g , there are inputs x_1, \dots, x_g such that for each $j \in [g]$

$$f(x_j) = 1 \quad \text{and} \quad f(x_j \oplus e_{i_j}) = 0.$$

Now consider the clause

$$f(z) \vee f(\neg x_1) \vee f(x_1 \oplus e_{i_1}) \vee \dots \vee f(\neg x_g) \vee f(x_g \oplus e_{i_g}).$$

Note that this clause might contain repeated literals but this is allowed at the moment. Clearly, this clause is not satisfied by f , so if this clause appears in the gadget we have our desired contradiction. In other words, we have to show that in each coordinate $i \in [n]$ we have at least g ones.

For any coordinate $i \notin \{i_1, \dots, i_g\}$ we have that coordinate i is 1 in exactly one of $\{\neg x_j, x_j \oplus e_{i_j}\}$, for a total of

at least g ones. For the coordinate i_j , we have that for all $j' \neq j$, at least one of $\{\neg x_{j'}, x_{j'} \oplus e_{i_{j'}}\}$ has a 1 in coordinate i_j . Furthermore z has a one in coordinate j , for a total of at least g ones. ■

It is now easy to prove Lemma IV.2.

Proof of Lemma IV.2: Suppose f depends on $2g$ distinct variables $i_1, \dots, i_g, j_1, \dots, j_g$. Let z be an input such that $z_{i_1} = \dots = z_{i_g} = 1$ and $z_{j_1} = \dots = z_{j_g} = 0$. By Claim IV.3, $f(z) = 1$ and $f(\neg z) = 1$, contradicting that f is odd. ■

B. Reduction from Label Cover

Let $\Psi = (U, V, E, \{\pi_e : L_V \rightarrow L_U\})$ be a Label Cover instance. To each vertex $u \in U$ we associate a function $f_u : \{0, 1\}^{L_U} \rightarrow \{0, 1\}$ intended to be a dictator of the label ℓ_u of u , and similarly $f_v : \{0, 1\}^{L_V} \rightarrow \{0, 1\}$ for $v \in V$.

We add the following constraints:

- For each $u \in U$ (resp. $v \in V$), the dictatorship gadget on f_u (resp. f_v).
- Fix an edge $e = (u, v)$. Let $x_1, \dots, x_g \in \{0, 1\}^{L_U}$ be g inputs on the U side and $y_1, \dots, y_{g+1} \in \{0, 1\}^{L_V}$ be $g+1$ inputs on the V side. If for each $l \in L_V$ it holds that

$$\sum_{j=1}^g x_{j, \pi_e(l)} + \sum_{j=1}^{g+1} y_{j,l} \geq g$$

we add the constraint

$$f_u(x_1) \vee \dots \vee f_u(x_g) \vee f_v(y_1) \vee \dots \vee f_v(y_{g+1})$$

We also use folding to make sure that each f_u (and f_v) is odd.

Call the resulting formula Φ . The completeness is standard and follows immediately from the construction and the completeness of the dictatorship gadget.

Lemma IV.4 (Completeness). *If $\text{Opt}(\Psi) = 1$ then Φ is strongly g -satisfiable.*

We turn to the more interesting case of soundness.

Lemma IV.5 (Soundness). *If Φ is satisfiable then $\text{Opt}(\Psi) \geq 1/(2g-1)^2$.*

Proof: Fix a satisfying assignment $\{f_u\}, \{f_v\}$ to Φ . By the soundness of the dictatorship gadget (Lemma IV.2) every f_u and f_v depends on at most $2g-1$ variables.

For each variable, let $S_u \subseteq L_U$ (resp. $S_v \subseteq L_V$) be the set of variables that f_u (resp. f_v) depends on and we have the following claim.

Claim IV.6. *For every edge $e = (u, v)$ it holds that $\pi_e(S_v) \cap S_u \neq \emptyset$.*

Proof: Suppose for contradiction that $S_u \cap \pi_e(S_v) = \emptyset$. Let $x_1, \dots, x_g \in \{0, 1\}^{L_U}$ be g inputs such that $f_u(x_j) = 0$ and $x_{j,l'} = 1$ for all $l' \in L_U \setminus S_u$, and similarly let $y_1, \dots, y_{g+1} \in \{0, 1\}^{L_V}$ be $g+1$ inputs such that $f_v(y_j) = 0$ and $y_{j,l} = 1$ for all $l \in L_V \setminus S_v$.

Let us now check that

$$\sum_{j=1}^g x_{j, \pi_e(l)} + \sum_{j=1}^{g+1} y_{j,l} \geq g \quad \forall l \in L_V. \quad (1)$$

Note that this would imply that $f_u(x_1) \vee \dots \vee f_u(x_g) \vee \dots \vee f_v(y_1) \vee \dots \vee f_v(y_{g+1})$ is a clause in Φ and by construction it is not satisfied, a contradiction. For $l \notin S_v$, (1) holds because $y_{j,l} = 1$ for $j = 1, 2, \dots, g+1$. For $l \in S_v$, we have $\pi_e(l) \notin S_u$, and therefore $x_{j, \pi_e(l)} = 1$ for $j = 1, 2, \dots, g$, and (1) again holds. ■

We now finish the proof of Lemma IV.5. We construct a random labeling by picking a random label from S_u (resp. S_v) for each variable $u \in U$ (resp. $v \in V$) of Ψ . For each edge $e = (u, v)$ it follows from the claim that the probability that e is satisfied by this labeling is $\frac{1}{|S_u| \cdot |S_v|} \geq 1/(2g-1)^2$ implying the bound on $\text{Opt}(\Psi)$. ■

To finish the proof of Theorem I.1 we now simply make sure to start with a Label Cover instance with soundness at most $(2g-1)^{-2}$ and then invoke Theorem II.8.

Remark IV.7. The above proof can also be used to show that in soundness case, there is no assignment that satisfies more than $1 - \gamma(g)$ fraction of clauses, for some $\gamma(g) > 0$. The required soundness for the Label Cover instance can be achieved with $|L_U|, |L_V| \leq \text{poly}(g)$, and thus the tables f_u, f_v have sizes that only depend on g . As long as most of the tables f_u, f_v satisfy all the clauses, the constructed labeling satisfies enough Label Cover constraints.

C. Inapproximability under the Unique Games conjecture

We now note the following ‘‘approximation resistance’’ phenomenon associated with *almost-satisfiable* instances of $(1, g, 2g+1)$ -SAT.

Theorem IV.8. *Let $g \geq 1$ be an integer. Assuming the Unique Games conjecture [13], the following promise problem is hard for every $\varepsilon > 0$. Given an instance of $(2g+1)$ -SAT (with no repetitions of variables allowed within a clause), distinguish between the following two cases:*

- 1) *There is an assignment that strongly g -satisfies $(1 - \varepsilon)$ of the clauses, and*
- 2) *There is no assignment that satisfies a fraction $(1 - 2^{-2g+1} + \varepsilon)$ of the clauses.*

Note that we have a gap between the two cases both in terms of the predicate being imposed on the clauses, and the fraction of clauses satisfiable according to the respective predicates. Also, since a random assignment satisfies an

expected fraction $1 - 2^{-2g+1}$ of the clauses, the inapproximability factor is tight. The rather standard proof of Theorem IV.8 appears in the full version of the paper.

V. DISCREPANCY WITH SMALL SETS

The main result of this section is the following theorem, which is of course just an alternate statement of Theorem I.2.

Theorem V.1. *g -DISCREPANCY is NP-hard for every constant $g > 1$.*

The reduction and proof follows along the same lines as the hardness proof for $(1, g, 2g + 1)$ -SAT in the previous section, though some modifications in the constructions are needed since “true” and “false” are now treated symmetrically. Additionally, we shall now deal with the details for the case when repeated elements are not allowed.

Another distinction from $(1, g, 2g + 1)$ -SAT to g -DISCREPANCY is that we no longer have the concept of negated literals and so we can no longer assume that our long codes are folded. If we allowed repeated elements in our sets this problem can be solved very simply by adding a constraint with g copies of $f(x)$ and $g + 1$ copies of $f(\neg x)$ for any x . Since we do not allow repetitions we have to be slightly more careful.

A. Dictatorship test

Let us start with the dictatorship test for a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. This consists of all sets of $2g + 1$ (distinct) variables $\{f(x_1), \dots, f(x_{2g+1})\}$ such that for each coordinate $j \in [n]$ we have $g \leq \sum_{i=1}^{2g+1} x_{i,j} \leq g + 1$. We refer to this as the discrepancy dictatorship gadget.

The completeness of the gadget follows by construction.

Lemma V.2. *If f is a dictatorship function then it evenly splits all sets in the discrepancy dictatorship gadget.*

For the soundness, we can no longer conclude that if f does not leave any set of the gadget monochromatic then f only depends on a few variables. For instance, if we let f equal a dictator except at a few points then f depends on all variables but would split all sets in the gadget. We prove that this is the only problem, and as a first step we start with a definition.

Definition V.3. A function f is said to t -depend on variable i if there are at least t disjoint pairs $\{x_j, x_j \oplus e_i\}$, $j \in [t]$, such that $f(x_j) \neq f(x_j \oplus e_i)$.

Using this definition, we have the following analogue of Lemma IV.2.

Lemma V.4. *Any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which does not leave any set of the discrepancy dictatorship gadget monochromatic $2g$ -depends on at most $2g - 1$ variables, provided $n > 5g$.*

Towards proving this, we first note that even though we can not force f to be folded by means of negated literals, the dictatorship test forces f to be folded on all but a small number of bad inputs.

Lemma V.5. *If f does not make any set of the discrepancy dictatorship gadget monochromatic then there are at most g pairs of inputs $\{x_i, \neg x_i\}$ such that $f(x_i) = f(\neg x_i) = 0$, and at most g pairs $\{y_i, \neg y_i\}$ such that $f(y_i) = f(\neg y_i) = 1$.*

Proof: Suppose for contradiction that we have $g + 1$ distinct pairs $\{x_1, \neg x_1\}, \dots, \{x_{g+1}, \neg x_{g+1}\}$ such that $f(x_i) = f(\neg x_i) = 0$. Then f leaves $(x_1, \neg x_1, \dots, x_g, \neg x_g, x_{g+1})$ monochromatic and it is straightforward to see that this is one of allowed tests. The other claim is analogous. ■

Next we have the following analogue of Claim IV.3, saying that the value of f can be essentially fixed by some setting of g variables.

Claim V.6. *Suppose f $2g$ -depends on g different variables i_1, \dots, i_g and does not leave any set of the discrepancy dictatorship gadget monochromatic. Then there are constants $c_1, \dots, c_g \in \{0, 1\}$ such that $f(z) = 1$ for all but at most $2g$ inputs z such that $z_{i_j} = c_j$ for all $j \in [g]$.*

Proof: We proceed as in the proof of Claim IV.3. For each $j \in [g]$ there are $2g$ different inputs x satisfying $f(x) = 1$ and $f(x \oplus e_{i_j}) = 0$. By Lemma V.5, $f(\neg x) = 1$ on at most g of these.

It follows that we can choose g distinct inputs x_1, \dots, x_g such that $f(\neg x_j) = f(x_j \oplus e_{i_j}) = 0$ for each $j \in [g]$. Now let $c_j = x_{j,i_j}$. Consider any input z such that $z_{i_j} = c_j$ for each $j \in [g]$ but distinct from the $2g$ inputs $\neg x_1, x_1 \oplus e_{i_1}, \dots, \neg x_g, x_g \oplus e_{i_g}$. We claim that $\{f(z), f(\neg x_1), f(x_1 \oplus e_{i_1}), \dots, f(\neg x_g), f(x_g \oplus e_{i_g})\}$ is a set in the discrepancy dictatorship gadget and hence $f(z)$ must equal 1.

To see this, note that in coordinate i_j the inputs $\neg x_1, x_1 \oplus e_{i_1}, \dots, \neg x_g, x_g \oplus e_{i_g}$ have $g - 1$ occurrences of the value x_{j,i_j} and $g + 1$ occurrences of the value $\neg x_{j,i_j}$, and z provides one additional occurrence of the value $c_j = x_{j,i_j}$ so that this coordinate is balanced. For coordinates outside i_1, \dots, i_g the x 's provide an equal number of 0's and 1's and hence the value of z in these coordinates is irrelevant. ■

Proof of Lemma V.4: The only difference to the proof of Lemma IV.2 is that we need to make sure to avoid the up to $4g$ bad inputs where f is not folded and the at most $2g + 2g$ choices of z where Claim V.6 does not apply, but if $2^{n-2g} > 8g$ such a contradictory z exists. ■

B. Reduction from Label Cover

The reduction from Label Cover also proceeds as in Section IV-B. The main crux is to locate a set consisting of disjoint variables. To simplify the presentation, we alter

the reduction somewhat by augmenting each f_u and f_v with a set A of auxiliary variables, i.e., $f_u : \{0, 1\}^{L_U \cup A} \rightarrow \{0, 1\}$ and $f_v : \{0, 1\}^{L_V \cup A} \rightarrow \{0, 1\}$. Loosely speaking this is like creating $2^{|A|}$ copies of each variable and ensures that there is an abundance of variables allowing us to easily identify sets without repeated variables. This is technically not needed – provided the label sets L_U and L_V are sufficiently large one can find sufficiently many distinct variables in the original reduction.

In the reduction from a label cover instance Ψ , we add the discrepancy dictatorship gadget on each f_u and f_v (without any special treatment of the auxiliary variables). In the edge constraints we for an edge $e = (u, v)$ the clause

$$\{f_u(x_1), \dots, f_u(x_g), f_v(y_1), \dots, f_v(y_{g+1})\}$$

provided $g \leq \sum_{j=1}^g x_{j, \pi_e(i)} + \sum_{j=1}^{g+1} y_{j, i} \leq g + 1$ for each $i \in L_v$ and the x_i 's and y_i 's are distinct. Note that we do not have any constraint on the sum of values in the auxiliary coordinates. Calling the resulting g -DISCREPANCY instance Φ , we have the following analogue of Lemma IV.5.

Lemma V.7. *If there is an assignment to Φ which does not leave any set monochromatic then $\text{Opt}(\Psi) \geq 1/(2g - 1)^2$, provided $|A| = \Omega(\log |L_V|)$.*

Proof: As in the proof of Lemma IV.5 we take a satisfying assignment $\{f_u\}, \{f_v\}$ to Φ .

We now choose $S_u \subseteq L_U$ (resp. $S_v \subseteq L_V$) to be the set of *non-auxiliary* variables that f_u (resp. f_v) $4g$ -depends on, which by Lemma V.4 are at most $2g - 1$ variables.

Proceeding as in Lemma IV.5, we now want to re-prove Claim IV.6, showing that S_u intersects $\pi_e(S_v)$ for the new choices of S_u and S_v . Looking at the proof of Claim IV.6, the new aspects that we need to deal with is that the inputs $x_1, \dots, x_g, y_1, \dots, y_{g+1}$ picked there should be distinct, and we also have the added complication that f_u and f_v may depend on variables outside S_u and S_v (though only on a very small number of input strings).

Say that a partial assignment $z \in \{0, 1\}^{S_u \cup A}$ is good if:

- 1) $z_j = b$ for all $j \in S_u$ and some $b \in \{0, 1\}$ (i.e., z is constant on the coordinates of S_u)
- 2) f_u is odd on all inputs agreeing with z
- 3) $f_u(y) = 0$ for all $y \in \{0, 1\}^{L_U \cup A}$ such that $y|_{S_u \cup A} = z$.

Similarly we say that a partial assignment $w \in \{0, 1\}^{S_v \cup A}$ is good provided the same conditions hold with S_u and f_u replaced by S_v and f_v .

There are $2^{|A|+1}$ partial assignments satisfying the first condition, and by Lemma V.5 at most g partial assignments violating the second condition, so we have $2^{|A|+1} - g$ partial assignments satisfying the first two conditions.

Since f_u does not $4g$ -depend on any $i \notin S_u \cup A$, there are at most $8g|L_U|$ inputs $x \in \{0, 1\}^{L_U \cup A}$ for which $f(x) \neq$

$f(x \oplus e_i)$ for some $i \notin S_u \cup A$. This implies that there are $2^{|A|+1} - g - 8g|L_U|$ partial assignments that satisfy the first two conditions and f_u restricted to z is constant. Since we have excluded the points where f_u is not odd, f_u takes the value 0 on half of the remaining assignments, so there are at least $2^{|A|} - (g + 8g|L_U|)/2$ good partial assignments z .

Similarly there are at least $2^{|A|} - (g + 8g|L_V|)/2$ good partial assignments w . It follows that if $|A| = \Omega(\log |L_V|)$ we can choose $2g + 1$ distinct good partial assignments z_1, \dots, z_g , and w_1, \dots, w_{g+1} . See also Figure 1.

Any way of completing these partial assignments to inputs $x_1, \dots, x_g, y_1, \dots, y_{g+1}$ such that $\{f_u(x_1), \dots, f_u(x_g), f_v(y_1), \dots, f_v(y_{g+1})\}$ is a set appearing in Φ gives the desired contradiction since f_u (resp. f_v) is 0 on the x_j 's (resp. y_j 's) by construction.

This is easily done: we simply need to make sure that the number of ones in each column is g or $g + 1$. The perhaps only subtle point is that we need to take care with the coordinates in $j \in \pi_e^{-1}(\pi_e(S_v))$ (i.e., the j that collide with some $j' \in S_v$ under π_e). This is where we use that the partial assignments w_i are constant on S_v , as we can then use those constant values for the remaining coordinates in $\pi_e^{-1}(\pi_e(S_v))$.

This concludes the proof that S_u intersects $\pi_e(S_v)$ for each edge (u, v) , and the rest of the proof proceeds identically to Lemma IV.5. ■

VI. APPLICATION TO HEREDITARY MATRIX DISCREPANCY

Consider the following problem. Given a matrix $A = (a_{ij})$ $1 \leq i \leq n$ and $1 \leq j \leq m$ of dimension $n \times m$, the task is to find signs $x_i \in \{-1, 1\}$, to make any subset of the columns balanced. To be more precise we want for any $\alpha \subseteq [n]$, $\alpha \neq \emptyset$, the vector $b_j^\alpha = \sum_{i \in \alpha} x_i a_{ij}$ to have small L^∞ -norm. The minimum such quantity is called the *hereditary discrepancy*.

This problem was considered in [14] where it was proved to be hard to approximate within a factor $\frac{3}{2}$. It follows more or less immediately from Theorem V.1 that we can improve this inapproximability factor to $\frac{2g+1}{g+1}$ for any integer g and thus arbitrarily close to 2. Both the reduction of [14] and here actually gives a matrix such that either the discrepancy of A is large (in our case $2g + 1$) or the hereditary discrepancy is small (in our case $g + 1$).

To see this let A be the incidence matrix of the set system constructed to prove Theorem V.1. In other words $a_{ij} = 1$ if the i th element appears in set j . A setting of the x_i corresponding to a very balanced 2-coloring gives a witness that the hereditary discrepancy is at most $g + 1$ while a monochromatic set gives that the discrepancy of A is at least $2g + 1$.

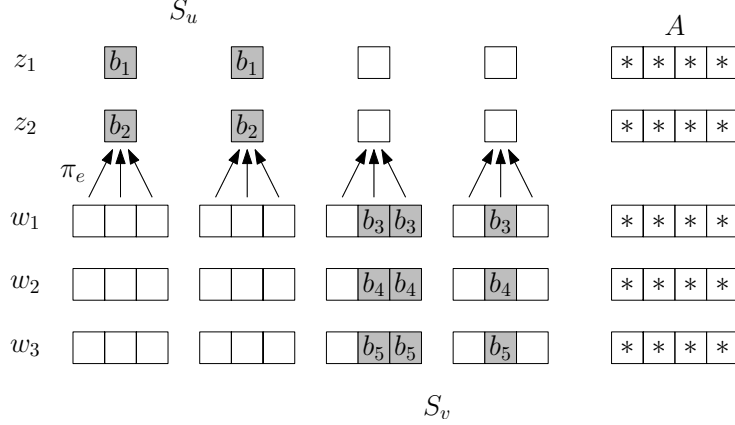


Figure 1. The good partial assignments $z_1, \dots, z_g, w_1, \dots, w_{g+1}$

VII. ALGORITHMS FOR $(1, g, 2g)$ -SAT

We now present efficient algorithms to find a satisfying assignment when at least half the literals in each clause are promised to be true under some assignment.

A. A randomized algorithm

Let us first describe a simple randomized algorithm closely following Papadimitriou's algorithm [1] for 2-SAT.

Algorithm 1: Randomized algorithm for $(1, g, w)$ -SAT.

- (1) $x \leftarrow$ arbitrary assignment
- (2) **while** x is not satisfying
- (3) Pick (arbitrarily) a falsified clause ϕ
- (4) Flip the value of a randomly chosen literal of ϕ
- (5) **return** x

The analysis of this algorithm is essentially equivalent with that of Papadimitriou.

Proposition VII.1. *If Φ is a strongly g -satisfiable width- w CNF formula and $w \leq 2g$, then Algorithm 1 finds a satisfying assignment in $O(tn^2)$ steps with probability at least $1 - 2^{-t}$.*

Proof: Let x^* be any g -satisfying assignment, and let x_i be the value of x in the i 'th iteration of Algorithm 1 and ϕ_i be the clause chosen. Define the random variable $D_i = d(x_i, x^*)$ where $d(x, y)$ is the Hamming distance between x and y . Clearly, $D_{i+1} - D_i = \pm 1$. Furthermore, since x^* satisfies g literals of ϕ_i and it contains at most $2g$ literals we have

$$\Pr[D_{i+1} = D_i - 1] \geq 1/2$$

so that $\mathbb{E}[D_{i+1} - D_i | D_i] \leq 0$. In other words D_1, D_2, \dots describes a random walk starting at some point between 0 and n where each step is unbiased or biased towards 0. Such a walk hits 0 in n^2 steps with constant probability.

The probability that it fails to hit 0 with ctn^2 steps is thus at most 2^{-t} for a suitable chosen constant c . ■

Note that this algorithm is not affected by the presence of multiple copies of the same literal within a clause. Also note that if $w < 2g$ the walk is in fact biased towards 0 and a satisfying assignment is, with high probability, found in $O(n)$ steps.

We next present a deterministic algorithm that is based on linear programming.

B. A deterministic algorithm

There is a very natural linear program connected to a w -SAT formula. Namely, relax each Boolean variable x_i to a real-valued variable y_i which takes values in $[0, 1]$. In the formula replace x_i by y_i and $\neg x_i$ by $1 - y_i$ and require that the sum over each clause is at least g . As an example in $(1, 2, 4)$ -SAT we replace the clause $(x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4)$ by the linear inequality

$$y_1 + y_2 + (1 - y_3) + (1 - y_4) \geq 2$$

This might not seem like a very useful linear program as $y_i = 1/2$ for all i satisfy all the inequalities when $w = 2g$, but forcing a single variable to take the value 0 or 1 does give useful information. Consider the procedure described in Algorithm 2 where we let $\lfloor y \rfloor$ denote the integer closest to y (we only apply this operation to numbers whose fractional part is not $1/2$ and hence this number is unique). We establish in the below proposition that the algorithm is indeed correct.

Proposition VII.2. *Given a strongly g -satisfiable w -SAT instance, where $w \leq 2g$, Algorithm 2 finds a satisfying assignment.*

Proof: Note first that if $w < 2g$ then in the LP solution any clause must contain a literal whose value is greater than $1/2$ and thus in fact the tentative assignment to x_i in line 3 is not needed.

Algorithm 2: Deterministic algorithm for $(1, g, w)$ -SAT.

- (1) **repeat**
- (2) Let x_i be some unassigned variable
- (3) Choose $b \in \{0, 1\}$ such that the basic LP with y_i forced to b is feasible
- (4) **if** the LP is infeasible for both choices of b
- (5) **return** “Not strongly g -satisfiable”
- (6) Let y_1, \dots, y_n be the LP solution when y_i is forced to b
- (7) **foreach** i such that $y_i \neq \frac{1}{2}$
- (8) Assign $x_i \leftarrow \lfloor y_i \rfloor$
- (9) Remove all satisfied clauses from the formula
- (10) **until** all variables are assigned
- (11) **return** x

When $w = 2g$ each clause that contains a literal that is not exactly $1/2$ must, in each feasible solution, contain a literal that is of value strictly greater than $1/2$. This implies that if we assign the value of some variable in a clause then in the same round we set one of its literals to true and satisfy the clause. Thus there is no risk of falsifying a clause during this process. In addition, the clauses that remain after each round consist only of unassigned variables and thus the remaining set of clauses still forms a strongly g -satisfiable instance. ■

VIII. CONCLUSIONS

We have given a sharp classification for a natural promise version of CNF-Sat. As CNF-Sat is a favorite starting point for many reductions we hope that this can give improved results quantitatively in many situations. We gave a rather modest example in Section VI but we hope there are many other possibilities.

As we show in the full paper, the *non-existence* of weak polymorphisms whose outputs satisfy a weaker predicate Q than the predicate P obeyed by its inputs implies the *hardness* of finding a Q -satisfying assignment to a P -satisfiable CSP instance. It will be extremely interesting to obtain some results in the converse direction, obtaining *algorithms* based on the *existence* of non-trivial weak polymorphisms, at least for the case of Boolean predicates P, Q . (We recall that when $P = Q$, we know in the Boolean case that the existence of non-trivial polymorphisms precisely governs the tractability of the associated CSP.)

Another interesting direction would be to extend the sort of combinatorial reasoning used in this paper to long code tests over larger domains. A concrete motivating target would be to answer the following question (a positive answer would generalize Theorem I.2): Given a $(k + 1)$ -uniform hypergraph that admits a k -coloring such that every hyperedge is polychromatic with all k colors, is it NP-hard to find a 2-coloring without monochromatic edges?

ACKNOWLEDGMENT

We are indebted to Dominik Scheder for the claim about the non-existence of gadget reductions described in Section III.

REFERENCES

- [1] C. H. Papadimitriou, “On selecting a satisfying truth assignment (extended abstract),” in *Proceedings of the 32nd annual symposium on Foundations of computer science*, ser. FOCS ’91, 1991, pp. 163–169.
- [2] D. Achlioptas, L. M. Kirousis, E. Kranakis, and D. Krizanc, “Rigorous results for random $(2+p)$ -sat,” *Theor. Comput. Sci.*, vol. 265, no. 1-2, pp. 109–129, 2001.
- [3] P. Austrin and J. Håstad, “On the usefulness of predicates,” *ACM Transactions on Computation Theory*, vol. 5, pp. 1–24, 2013.
- [4] M. Charikar, A. Newman, and A. Nikolov, “Tight hardness results for minimizing discrepancy,” in *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011, pp. 1607–1614.
- [5] P. Austrin, V. Guruswami, and J. Håstad, “ $(2 + \epsilon)$ -Sat is NP-hard,” 2013, ECCC Report TR13-159.
- [6] J. Håstad, “Some optimal inapproximability results,” *Journal of the ACM*, vol. 48, pp. 798–859, 2001.
- [7] E. Feuerstein, A. Marchetti-Spaccamela, F. Schalekamp, R. Sitters, S. van der Ster, L. Stougie, and A. van Zuylen, “Scheduling over scenarios on two machines,” *Manuscript*, 2014.
- [8] V. Guruswami and E. Lee, “Strong inapproximability results on balanced rainbow-colorable hypergraphs,” 2014, ECCC Report TR14-043.
- [9] E. Mossel, “Gaussian bounds for noise correlation of functions,” *Geometric and Functional Analysis*, vol. 19, no. 6, pp. 1713–1756, 2010.
- [10] C. Wenner, “Circumventing d -to-1 for approximation resistance of satisfiable predicates strictly containing parity of width at least four,” *Theory of Computing*, vol. 9, no. 23, pp. 703–757, 2013.
- [11] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, “Proof verification and intractability of approximation problems,” *Journal of the ACM*, vol. 45, pp. 501–555, 1998.
- [12] R. Raz, “A parallel repetition theorem,” *SIAM J. on Computing*, vol. 27, pp. 763–803, 1998.
- [13] S. Khot, “On the power of unique 2-prover 1-round games,” in *Proceedings of 34th ACM Symposium on Theory of Computing*, 2002, pp. 767–775.
- [14] A. Nikolov, K. Talwar, and L. Zhang, “The geometry of differential privacy: the sparse and approximate cases,” in *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, ser. STOC ’13. New York, NY, USA: ACM, 2013, pp. 351–360.