Improved NP-inapproximability for 2-Variable Linear Equations*

Johan Håstad[1], Sangxia Huang[1], Rajsekar Manokaran[2], Ryan O'Donnell[3], and John Wright[3]

1 **School of Computer Science and Communication**
KTH Royal Institute of Technology
Stockholm, Sweden
johanh@csc.kth.se, huang.sangxia@gmail.com

2 **Computer Science and Engineering Department**
IIT Madras
Chennai, India
rajsekar@gmail.com

3 **Department of Computer Science**
Carnegie Mellon University
Pittsburgh, PA, USA
{odonnell,jswright}@cs.cmu.edu

Abstract

An instance of the 2-Lin(2) problem is a system of equations of the form "$x_i + x_j = b \pmod 2$". Given such a system in which it's possible to satisfy all but an ϵ fraction of the equations, we show it is NP-hard to satisfy all but a $C\epsilon$ fraction of the equations, for any $C < \frac{11}{8} = 1.375$ (and any $0 < \epsilon \leq \frac{1}{8}$). The previous best result, standing for over 15 years, had $\frac{5}{4}$ in place of $\frac{11}{8}$. Our result provides the best known NP-hardness even for the Unique-Games problem, and it also holds for the special case of Max-Cut. The precise factor $\frac{11}{8}$ is unlikely to be best possible; we also give a conjecture concerning analysis of Boolean functions which, if true, would yield a larger hardness factor of $\frac{3}{2}$.

Our proof is by a modified gadget reduction from a pairwise-independent predicate. We also show an inherent limitation to this type of gadget reduction. In particular, any such reduction can never establish a hardness factor C greater than 2.54. Previously, no such limitation on gadget reductions was known.

1998 ACM Subject Classification F.2.0 Analysis of Algorithms and Problem Complexity

Keywords and phrases approximability, unique games, linear equation, gadget, linear programming

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2015.341

1 Introduction

The well known constraint satisfaction problem (CSP) 2-Lin(q) is defined as follows: Given n variables $x_1, \ldots, x_n$, as well as a system of equations (constraints) of the form "$x_i + x_j = b \pmod q$" for constants $b \in \mathbb{Z}_q$, the task is to assign values from $\mathbb{Z}_q$ to the variables so

* Abridged Version. Johan Håstad and Sangxia Huang are supported by ERC Advanced Investigator grant 226203 and Swedish Research Council. Ryan O'Donnell and John Wright are supported by NSF grants CCF-0747250 and CCF-1116594 and a grant from the MSR–CMU Center for Computational Thinking. John Wright is also supported by a Simons Fellowship in Theoretical Computer Science.

18th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'15) / 19th Int'l Workshop on Randomization and Computation (RANDOM'15).
Editors: Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim; pp. 341–360

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that there are as few unsatisfied constraints as possible. It is known [15, 17] that, from an approximability standpoint, this problem is equivalent to the notorious Unique-Games problem [14]. The special case of $q = 2$ is particularly interesting and can be equivalently stated as follows: Given a "supply graph" G and a "demand graph" H over the same set of vertices V, partition V into two parts so as to minimize the total number of cut supply edges and uncut demand edges. The further special case when the supply graph G is empty (i.e., every equation is of the form $x_i - x_j = 1 \pmod 2$) is equivalent to the Max-Cut problem.

Let's say that an algorithm guarantees an (ϵ, ϵ')-*approximation* if, given any instance in which the best solution falsifies at most an ϵ-fraction of the constraints, the algorithm finds a solution falsifying at most an ϵ'-fraction of the constraints. If an algorithm guarantees $(\epsilon, C\epsilon)$-approximation for every ϵ then we also say that it is a *factor-C approximation*.

We remark here that we are prioritizing the so-called "Min-Deletion" version of the 2-Lin(2) problem. We feel it is the more natural parameterization. For example, in the more traditional "Max-2-Lin(2)" formulation, the discrepancy between known algorithms and NP-hardness involves two quirky factors, 0.878 and 0.912. However, this disguises what we feel is the really interesting question – the same key open question that arises for the highly analogous Sparsest-Cut problem: Is there an efficient $(\epsilon, O(\epsilon))$-approximation, or even one that improves on the known $(\epsilon, O(\sqrt{\log n})\epsilon)$- and $(\epsilon, O(\sqrt{\epsilon}))$-approximations?

The relative importance of the "Min-Deletion" version is even more pronounced for the 2-Lin(q) problem. As we describe below, this version of the problem is essentially equivalent to the highly notorious Unique-Games problem. By way of contrast, the traditional maximization approximation factor measure for Unique-Games is not particularly interesting – it's known [10] that there is no constant-factor approximation for "Max-Unique-Games", but this appears to have no relevance for the Unique Games Conjecture.

1.1 History of the problem

No efficient $(\epsilon, O(\epsilon))$-approximation algorithm for 2-Lin(2) is known. The best known efficient approximation guarantee with no dependence on n dates back to the seminal work of Goemans and Williamson:

▶ **Theorem 1** ([11]). *There is a polynomial-time $(\epsilon, \frac{2}{\pi}\sqrt{\epsilon} + o(\epsilon))$-approximation algorithm for* 2-Lin(2).

Allowing the approximation to depend on n, we have the following result building on [3]:

▶ **Theorem 2** ([1]). *There is a polynomial-time factor-$O(\sqrt{\log n})$ approximation for* 2-Lin(2).

Generalizing Theorem 1 to 2-Lin(q), we have the following result of Charikar, Makarychev, and Makarychev:

▶ **Theorem 3** ([7]). *There is a polynomial time $(\epsilon, C_q\sqrt{\epsilon})$-approximation for* 2-Lin(q) *(and indeed for* Unique-Games*), for a certain $C_q = \Theta(\sqrt{\log q})$.*

The question of whether or not this theorem can be improved is known to be essentially equivalent to the influential Unique Games Conjecture of Khot [14]:

▶ **Theorem 4.** *The Unique Games Conjecture implies ([15, 17]) that for all sufficiently small $\epsilon > 0$, $(\epsilon, \frac{2}{\pi}\sqrt{\epsilon} + o(\epsilon))$-approximating* 2-Lin(2) *is* NP*-hard, and for general q, $(\epsilon, \Omega(\sqrt{\log q})\sqrt{\epsilon})$-approximating* 2-Lin($q$) *is* NP*-hard. On the other hand ([19]), if there exists $q = q(\epsilon)$ such that $(\epsilon, \omega(\sqrt{\epsilon}))$-approximating* 2-Lin($q$) *is* NP*-hard then the Unique Games Conjecture holds.*

The recent work of Arora, Barak, and Steurer has also emphasized the importance of subexponential-time algorithms in this context:

▶ **Theorem 5** ([2]). *For any $\beta \geq \frac{\log\log n}{\log n}$ there is a $2^{O(qn^\beta)}$-time algorithm for $(\epsilon, O(\beta^{-3/2})\sqrt{\epsilon})$-approximating* 2-Lin($q$). *For example, there is a constant $K < \infty$ and an $O(2^{n^{0.001}})$-time algorithm for $(\epsilon, K\sqrt{\epsilon})$-approximating* 2-Lin($q$) *for any $q = n^{o(1)}$.*

Finally, we remark that there is an *exact* algorithm for 2-Lin(2) by [21] that runs in time roughly 1.73^n.

The known NP-hardness results for 2-Lin(q) are rather far from the known algorithms. It follows easily from the PCP Theorem that for any q, there exists $C > 1$ such that factor-C approximation of 2-Lin(q) is NP-hard. However, getting an explicit value for C has been a difficult task. In 1995, Bellare, Goldreich, and Sudan [5] introduced the Long Code testing technique, which let them prove NP-hardness of approximating 2-Lin(2) to factor of roughly 1.02. Around 1997, Håstad [13] gave an optimal inapproximability result for the 3-Lin(2) problem; combining this with the "automated gadget" results of Trevisan et al. [20] allowed him to establish NP-hardness of factor-C approximation for any $C < \frac{5}{4}$. By including the "outer PCP" results of Moshkovitz and Raz [16] we may state the following more precise theorem:

▶ **Theorem 6** ([13]). *Fix any $C < \frac{5}{4}$. Then it is* NP*-hard to $(\epsilon, C\epsilon)$-approximate* 2-Lin(2) *(for any $0 < \epsilon \leq \epsilon_0 = \frac{1}{4}$). In fact ([16]), there is a reduction with quasilinear blowup; hence $(\epsilon, C\epsilon)$-approximation on size-N instances requires $2^{N^{1-o(1)}}$ time assuming the Exponential Time Hypothesis (ETH).*

Since 1997 there had been no improvement on this hardness factor of $\frac{5}{4}$, even for the (presumably much harder) 2-Lin(q) problem. We remark that Håstad [13] showed the same hardness result even for Max-Cut (albeit with a slightly smaller ϵ_0) and that O'Donnell and Wright [18] showed the same result for 2-Lin(q) (even with a slightly larger ϵ_0, namely $\epsilon_0 \to \frac{1}{2}$ as $q \to \infty$).

1.2 Our results and techniques

In this work we give the first known improvement to the factor-$\frac{5}{4}$ NP-hardness for 2-Lin(2) from [13]:

▶ **Theorem 7.** *Fix any $C < \frac{11}{8}$. Then it is* NP*-hard to $(\epsilon, C\epsilon)$-approximate* 2-Lin(2) *(for any $0 < \epsilon \leq \epsilon_0 = \frac{1}{8}$). Furthermore, the reduction takes* 3-Sat *instances of size n to* 2-Lin(2) *instances of size $n^{7+o(1)}$; hence $(\epsilon, C\epsilon)$-approximating* 2-Lin(2) *instances of size N requires at least $2^{N^{1/7-o(1)}}$ time assuming the ETH.*

▶ Remark. The power 7 in the size of the reduction comes from Chan's hardness reduction for the 7-ary Hadamard predicate [6].

We sketch the proof of this theorem in Section 3. The same theorem also holds in the special case of Max-Cut (albeit with some smaller, inexplicit value of ϵ_0). Proofs for both results can be found in the full version of the paper.

Our result is a gadget reduction from the "7-ary Hadamard predicate" CSP, for which Chan [6] recently established an optimal NP-inapproximability result. In a sense our Theorem 7 is a direct generalization of Håstad's Theorem 6, which involved an optimal gadget reduction from the "3-ary Hadamard predicate" CSP, namely 3-Lin(2). That said, we should emphasize some obstacles that prevented this result from being obtained 15 years ago.

First, we employ Chan's recent approximation-resistance result for the 7-ary Hadamard predicate. In fact, what's crucial is not its approximation-resistance, but rather the stronger fact that it's a *useless* predicate, as defined in the recent work [4]. That is, given a nearly-satisfiable instance of the CSP, it's NP-hard to assign values to the variables so that the distribution on the 7-tuples of the constraints is noticeably different from the uniform distribution.

Second, although in principle our reduction fits into the "automated gadget" framework of Trevisan et al. [20], in practice it's completely impossible to find the necessary gadget automatically, since it would involve solving a linear program with $\sim 2^{120}$ constraints. Instead we had to construct and analyze our gadget by hand. On the other hand, by also constructing an appropriate LP dual solution, we are able to show the following.

▶ **Theorem 8** (Informally stated). *Our gadget achieving factor-$\frac{11}{8}$ NP-hardness for* 2-Lin(2) *is* optimal *among gadget reductions from Chan's 7-ary Hadamard predicate hardness.*

In spite of Theorem 8, it seems extremely unlikely that factor-$\frac{11}{8}$ NP-hardness for 2-Lin(2) is the end of the line. Indeed, we view Theorem 7 as more of a "proof of concept" illustrating that the longstanding factor-$\frac{5}{4}$ barrier can be broken; we hope to see further improvements in the future. In particular, in Section 4 we present a candidate NP-hardness reduction from high-arity useless CSPs that we believe may yield NP-hardness of approximating 2-Lin(2) to any factor below $\frac{3}{2}$. The analysis of this reduction eventually depends on a certain conjecture regarding analysis of Boolean functions that we were unable to resolve; thus we leave it as an open problem.

Finally, in Section 5 we show an inherent limitation of the method of gadget reductions from pairwise-independent predicates. We prove that such reductions can never establish an NP-hardness factor better than $\frac{1}{1-e^{-1/2}} \approx 2.54$ for $(\varepsilon, C\varepsilon)$-approximation of 2-Lin(2). We believe that this highlights a serious bottleneck in obtaining hardness results matching the performance of algorithms for this problem as most optimal NP-inapproximability results involve pairwise-independent predicates.

2 Preliminaries

▶ **Definition 9.** Given $x, y \in \{-1,1\}^n$, the *Hamming distance* between x and y, denoted $d_H(x,y)$, is the number of coordinates i where x_i and y_i differ. Similarly, if $f, g : V \to \{-1,1\}$ are two functions over a variable set V, then the Hamming distance $d_H(f,g)$ between them is the number of inputs x where $f(x)$ and $g(x)$ disagree.

▶ **Definition 10.** A *predicate* on n variables is a function $\phi : \{-1,1\}^n \to \{0,1\}$. We say that $x \in \{-1,1\}^n$ *satisfies* ϕ if $\phi(x) = 1$ and otherwise that it *violates* ϕ.

▶ **Definition 11.** Given a predicate $\phi : \{-1,1\}^n \to \{0,1\}$, $\mathsf{Sat}(\phi)$ is the set of satisfying assignments.

▶ **Definition 12.** A set $S \subseteq \{-1,1\}^n$ is a *balanced pairwise-independent subgroup* if it satisfies the following properties:

1. S forms a group under bitwise multiplication.
2. If $\boldsymbol{x}$ is selected from S uniformly at random, then $\Pr[\boldsymbol{x}_i = 1] = \Pr[\boldsymbol{x}_i = -1] = \frac{1}{2}$ for any $i \in [n]$. Furthermore, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are independent for any $i \neq j$.

A predicate $\phi : \{-1,1\}^n \to \{0,1\}$ *contains a balanced pairwise-independent subgroup* if there exists a set $S \subseteq \mathsf{Sat}(\phi)$ which is a balanced pairwise-independent subgroup.

▶ **Definition 13.** For a subset $S \subseteq [n]$, the parity function $\chi_S : \{-1,1\}^n \to \{-1,1\}$ is defined as $\chi_S(x) := \prod_{i \in S} x_i$.

▶ **Definition 14.** The Had_k predicate has $2^k - 1$ input variables, one for each nonempty subset $S \subseteq [k]$. The input string $\{x_S\}_{\emptyset \neq S \subseteq [k]}$ satisfies Had_k if for each S, $x_S = \chi_S(x)$.

▶ **Fact 15.** *The Had_k predicate contains a balanced pairwise-independent subgroup. (In fact, the whole set $\mathsf{Sat}(\mathsf{Had}_k)$ is a balanced pairwise-independent subgroup.)*

Given a predicate $\phi : \{-1,1\}^n \to \{0,1\}$, an instance $\mathcal{I}$ of the Max-ϕ CSP is a variable set V and a distribution of ϕ-constraints on these variables. To sample a constraint from this distribution, we write $\mathcal{C} \sim \mathcal{I}$, where $\mathcal{C} = ((x_1, b_1), (x_2, b_2), \ldots, (x_n, b_n))$. Here the x_i's are in V and the b_i's are in $\{-1,1\}$. An assignment $A : V \to \{-1,1\}$ satisfies the constraint $\mathcal{C}$ if

$$\phi\left(b_1 \cdot A(x_1), b_2 \cdot A(x_2), \ldots, b_n \cdot A(x_n)\right) = 1.$$

We define several measures of assignments and instances.

▶ **Definition 16.** The *value* of A on $\mathcal{I}$ is just $\mathsf{val}(A;\mathcal{I}) := \Pr_{\mathcal{C} \sim \mathcal{I}}[A \text{ satisfies } \mathcal{C}]$, and the value of the instance $\mathcal{I}$ is $\mathsf{val}(\mathcal{I}) := \max_{\text{assignments } A} \mathsf{val}(A;\mathcal{I})$. We define $\mathsf{uval}(A;\mathcal{I}) := 1 - \mathsf{val}(A;\mathcal{I})$ and similarly $\mathsf{uval}(\mathcal{I})$.

▶ **Definition 17.** Let $(=) : \{-1,1\}^2 \to \{0,1\}$ be the equality predicate, i.e. $(=)(b_1, b_2) = 1$ iff $b_1 = b_2$ for all $b_1, b_2 \in \{-1,1\}$. We will refer to the Max-$(=)$ CSP as the *Max*-2-Lin(2) CSP. Any constraint $\mathcal{C} = ((x_1, b_1), (x_2, b_2))$ in a Max-2-Lin(2) instance tests "$x_1 = x_2$" if $b_1 \cdot b_2 = 1$, and otherwise tests "$x_1 \neq x_2$".

Typically, a hardness of approximation result will show that given an instance $\mathcal{I}$ of the Max-ϕ problem, it is NP-hard to tell whether $\mathsf{val}(\mathcal{I}) \geq c$ or $\mathsf{val}(\mathcal{I}) \leq s$, for some numbers $c > s$. A stronger notion of hardness is *uselessness*, first defined in [4], in which in the second case, not only is $\mathsf{val}(\mathcal{I})$ small, but any assignment to the variables A appears "uniformly random" to the constraints. To make this formal, we will require a couple of definitions.

▶ **Definition 18.** Given two probability distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ on some set S, the total variation distance d_{TV} between them is defined to be $d_{TV}(\mathcal{D}_1, \mathcal{D}_2) := \sum_{e \in S} \frac{1}{2} |\mathcal{D}_1(e) - \mathcal{D}_2(e)|$.

▶ **Definition 19.** Given a Max-ϕ instance $\mathcal{I}$ and an assignment A, denote by $\mathcal{D}(A, \mathcal{I})$ the distribution on $\{-1,1\}^n$ generated by first sampling $((x_1, b_1), \ldots, (x_n, b_n)) \sim \mathcal{I}$ and then outputting $(b_1 \cdot A(x_1), \ldots, b_n \cdot A(x_n))$.

The work of [6] showed uselessness for a wide range of predicates, including the Had_k predicate.

▶ **Theorem 20** ([6]). *Let $\phi : \{-1,1\}^n \to \{0,1\}$ contain a balanced pairwise-independent subgroup. For every $\epsilon > 0$, given an instance $\mathcal{I}$ of Max-ϕ, it is* NP*-hard to distinguish between the following two cases:*

- (Completeness): $\mathsf{val}(\mathcal{I}) \geq 1 - \epsilon$.
- (Soundness): *For every assignment A, $d_{TV}\left(\mathcal{D}(A, \mathcal{I}), \mathcal{U}_n\right) \leq \epsilon$, where $\mathcal{U}_n$ is the uniform distribution on $\{-1,1\}^n$.*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 \end{bmatrix}$$

Figure 1 The Had_3-matrix. The rows are the satisfying assignments of Had_3.

2.1 Gadgets

The work of Trevisan et al [20] gives a generic methodology for constructing gadget reductions between two predicates. In this section, we review this with an eye towards our eventual Had_k-to-2-Lin(2) gadgets.

Suppose $\phi : \{-1,1\}^n \to \{0,1\}$ is a predicate one would like to reduce to another predicate $\psi : \{-1,1\}^m \to \{0,1\}$. Set $K := |\mathsf{Sat}(\phi)|$. We begin by arranging the elements of $\mathsf{Sat}(\phi)$ as the rows of a $K \times n$ matrix, which we will call the *ϕ-matrix*. An example of this is done for the Had_3 predicate in Figure 1.

The columns of this matrix are elements of $\{-1,1\}^K$. Naming this set $V := \{-1,1\}^K$, we will think of V as the set of possible variables to be used in a gadget reduction from ϕ to ψ. One of the contributions of [20] was to show that the set V is sufficient for any such gadget reduction, and that any gadget reduction with more than 2^K variables has redundant variables which can be eliminated.

Of these variables, the n variables found as the columns of the ϕ-matrix are special; they correspond to n of the variables in the original ϕ instance and are therefore called *generic primary* variables. We will call them $v_1, v_2, \ldots, v_n$, where they are ordered by their position in the ϕ-matrix. The remaining variables are called generic *auxiliary* variables. For example, per Figure 1, $(1,1,1,1,-1,-1,-1,-1)$ and $(1,-1,-1,1,-1,1,1,-1)$ are generic primary variables in any gadget reducing from ϕ, but $(-1,-1,1,-1,1,-1,1,-1)$ is always a generic auxiliary variable.

On top of the variables V will be a distribution of ψ constraints. As a result, a gadget $\mathcal{G}$ is just an instance of the Max-ψ CSP using the variable set V. As above, we will associate $\mathcal{G}$ with the distribution of ψ constraints and write $\mathcal{C} \sim \mathcal{G}$ to sample a constraint from this distribution. Given an assignment $A : V \to \{0,1\}$, the goal is for $\mathcal{G}$ to be able to detect whether the values A assigns to the generic primary variables satisfy the ϕ predicate. For shorthand, we will say that A *satisfies* ϕ when

$$\phi\left(A(v_1), A(v_2), \ldots, A(v_n)\right) = 1.$$

On the other hand, A *fails to satisfy* ϕ when this expression evaluates to 0. Of all assignments, we are perhaps most concerned with the *dictator* assignments. The i-th dictator assignment, written $d_i : \{-1,1\}^K \to \{-1,1\}$, is defined so that $d_i(x) = x_i$ for all $x \in \{-1,1\}^K$. The following fact shows why the dictator assignments are so important:

▶ **Fact 21.** *Each dictator assignment d_i satisfies ϕ.*

Proof. The string $((v_1)_i, (v_2)_i, \ldots, (v_n)_i)$ is the i-th row of the ϕ-matrix, which, by definition, satisfies ϕ. ◀

At this point, we can now give the standard definition of a gadget. Typically, one constructs a gadget so that the dictator assignments pass with high probability, whereas every assignment which fails to satisfy ϕ passes with low probability. This is formalized in the following definition, which is essentially from [20]:

▶ **Definition 22** (Old definition)**.** A (c, s)-*generic gadget* reducing Max-ϕ to Max-ψ is a gadget $\mathcal{G}$ satisfying the following properties:

- (Completeness): For every dictator assignment d_i, $\mathsf{uval}(d_i; \mathcal{G}) \leq c$.
- (Soundness): For any assignment A which fails to satisfy ϕ, $\mathsf{uval}(A; \mathcal{G}) \geq s$.

We use uval as our focus is on the deletion version of 2-Lin(2). We include the word *generic* in this definition to distinguish it from the specific type of gadget we will use to reduce Had_k to 2-Lin(2). See Section 2.3 for details.

This style of gadget reduction is appropriate for the case when one is reducing from a predicate for which one knows an inapproximability result and nothing else. However, in our case we are reducing from predicates containing a balanced pairwise-independent subgroup, and Chan [6] has shown *uselessness* for this class of predicates (see Theorem 20). As a result, we can relax the (Soundness) condition in Definition 22; when reducing from this class of predicates, it is sufficient to show that this (Soundness) condition holds for *distributions* of assignments which *appear random on the generic primary variables*. In the following paragraph we expand on what this means.

Denote by $\mathcal{A}$ a distribution over assignments A. The value of $\mathcal{A}$ is just the average value of an assignment drawn from $\mathcal{A}$, i.e. $\mathsf{val}(\mathcal{A}; \mathcal{G}) := \mathbf{E}_{A \sim \mathcal{A}} \mathsf{val}(A; \mathcal{G})$, and similarly for $\mathsf{uval}(\mathcal{A}; \mathcal{G})$. We say that $\mathcal{A}$ is *random on the generic primary variables* if the tuple

$$(A(v_1), A(v_2), \ldots, A(v_n))$$

is, over a random $A \sim \mathcal{A}$, distributed as a *uniformly* random element of $\{-1, 1\}^n$.

▶ **Definition 23.** Denote by $\mathsf{R}^{gen}(\phi)$ the set of distributions which are *(uniformly)* random on the generic primary variables.

Our key definition is the following, which requires that our gadget only does well against distributions in $\mathsf{R}^{gen}(\phi)$.

▶ **Definition 24** (New definition)**.** A (c, s)-*generic gadget* reducing Max-ϕ to Max-ψ is a gadget $\mathcal{G}$ satisfying the following properties:

- (Completeness): For every dictator assignment d_i, $\mathsf{uval}(d_i; \mathcal{G}) \leq c$.
- (Soundness): For any $\mathcal{A} \in \mathsf{R}^{gen}(\phi)$, $\mathsf{uval}(\mathcal{A}; \mathcal{G}) \geq s$.

The following proposition is standard, and we sketch its proof for completeness.

▶ **Proposition 25.** *Suppose there exists a (c, s)-generic gadget reducing Max-ϕ to Max-ψ, where Max-ϕ is any predicate containing a balanced pairwise-independent subgroup. Then for all $\epsilon > 0$, given an instance $\mathcal{I}$ of Max-ψ, it is* NP*-hard to distinguish between the following two cases:*

- (Completeness): $\mathsf{uval}(\mathcal{I}) \leq c + \epsilon$.
- (Soundness): $\mathsf{uval}(\mathcal{I}) \geq s - \epsilon$.

Proof sketch. Let $\mathcal{I}$ be an instance of the Max-ϕ problem produced via Theorem 20. To dispense with some annoying technicalities, we will assume that every constraint $\mathcal{C}$ in the support of $\mathcal{I}$ is of the form $\mathcal{C} = ((x_1, 1), \ldots, (x_n, 1))$. Construct an instance $\mathcal{I}'$ of Max-ψ as follows: for each constraint $\mathcal{C} = ((x_1, 1), \ldots, (x_n, 1))$ in the support of $\mathcal{I}$, add in a copy of $\mathcal{G}$ – call it $\mathcal{G}_{\mathcal{C}}$ – whose total weight is scaled down so that it equals the weight of $\mathcal{C}$. Further, identify the primary variables $v_1, \ldots, v_n$ of $\mathcal{G}_{\mathcal{C}}$ with the variables $x_1, \ldots, x_n$.

Completeness: In this case, there exists an assignment A to the variables of $\mathcal{I}$ which violates at most an ϵ-fraction of the constraints. We will extend this to an assignment for all the variables of $\mathcal{I}'$ as follows: for any constraint $\mathcal{C} = ((x_1, 1), \ldots, (x_n, 1))$ which A satisfies, there is some dictator assignment to the variables of $\mathcal{G}_\mathcal{C}$ which agrees with A on the primary variables $v_1, \ldots, v_n$. Set A to also agree with this dictator assignment on the auxiliary variables in $\mathcal{G}_\mathcal{C}$. Regardless of how A is extended in the remaining $\mathcal{G}_\mathcal{C}$'s, it now labels a $(1-\epsilon)$-fraction of the $\mathcal{G}$ gadgets in $\mathcal{I}'$ with a dictator assignment, meaning that $\mathsf{uval}(A; \mathcal{I}') \leq (1-\epsilon) \cdot c + \epsilon \cdot 1 \leq c + \epsilon$.

Soundness: Let A be an assignment to the variables in $\mathcal{I}'$. Consider the distribution $\mathcal{A}$ of assignments to the gadget $\mathcal{G}$ generated as follows: sample $\mathcal{C} \sim \mathcal{I}$ and output the restriction of A to the variables of $\mathcal{G}_\mathcal{C}$. Because the distribution $(A(x_1), \ldots, A(x_n))$ is ϵ-close to uniform in total variation distance, $\mathcal{A}$ is ϵ-close in total variation distance to some distribution $\mathcal{A}' \in \mathsf{R}^{gen}(\phi)$. As a result, $\mathsf{uval}(\mathcal{A}; \mathcal{G}) \geq \mathsf{uval}(\mathcal{A}'; \mathcal{G}) - \epsilon \geq s - \epsilon$. But then $\mathsf{uval}(\mathcal{A}; \mathcal{G}) = \mathsf{uval}(A; \mathcal{I})$, which is therefore bounded below by $s - \epsilon$. ◀

2.2 Reducing into 2-Lin(2)

In this section, we consider gadgets which reduce into the 2-Lin(2) predicate. We show several convenient simplifying assumptions that can be made in this case.

▶ **Definition 26.** An assignment $A : \{-1, 1\}^K \to \{-1, 1\}$ is *folded* if $A(x) = -A(-x)$ for all $x \in \{-1, 1\}^K$. Here $-x$ is the bitwise negation of x. In addition, a distribution $\mathcal{A}$ is folded if every assignment in its support is folded.

The following proposition shows that when designing a gadget which reduces into 2-Lin(2), it suffices to ensure that its (Soundness) condition holds for folded distributions. The proof is standard.

▶ **Proposition 27.** *For some predicate ϕ, suppose $\mathcal{G}$ is a gadget reducing Max-ϕ to Max-2-*Lin*(2) which satisfies the following two conditions:*

- (Completeness): *For every dictator assignment d_i,* $\mathsf{uval}(d_i; \mathcal{G}) \leq c$.
- (Soundness): *For any* folded $\mathcal{A} \in \mathsf{R}^{gen}(\phi)$, $\mathsf{uval}(\mathcal{A}; \mathcal{G}) \geq s$.

*Then there exists a (c, s)-generic gadget reducing Max-ϕ to Max-2-*Lin*(2).*

Proof. For each pair of antipodal points x and $-x$ in $\{-1, 1\}^K$, pick one (say, x) arbitrarily, and set

$$\mathsf{canon}(x) := \mathsf{canon}(-x) := x.$$

This is the canonical variable associated to x and $-x$. The one constraint is that if either x or $-x$ is one of the generic primary variables, then it should be chosen as the canonical variable associated to x and $-x$. Now, let $\mathcal{G}'$ be the gadget whose constraints are sampled as follows:

1. Sample a constraint $A(x_1) \cdot A(x_2) = b$ from $\mathcal{G}$.
2. For $i \in \{1, 2\}$, set $b_i = 1$ if $\mathsf{canon}(x_i) = x_i$ and $b_i = -1$ otherwise.
3. Output the constraint $A(\mathsf{canon}(x_1)) \cdot A(\mathsf{canon}(x_2)) = b \cdot b_1 \cdot b_2$.

We claim that $\mathcal{G}'$ is a (c, s)-gadget reducing Max-ϕ to Max-2-Lin(2). To see this, set $\mathsf{is\text{-}canon}(x)$ to be 1 if $\mathsf{canon}(x) = x$ and (-1) otherwise. Then the probability that an assignment A fails on $\mathcal{G}'$ is the same as the probability that the assignment $A'(x) := \mathsf{is\text{-}canon}(x) \cdot A(\mathsf{canon}(x))$ fails on $\mathcal{G}$. For any dictator function d_i, $d_i(x) = \mathsf{is\text{-}canon}(x) \cdot d_i(\mathsf{canon}(x))$ for all x. Therefore,

d_i fails $\mathcal{G}'$ with probability c. Next, it is easy to see that for any assignment A, A' is folded and, due to our restriction on $\mathsf{canon}(\cdot)$, A' agrees with A on the generic primary variables. Thus, given a distribution $\mathcal{A} \in \mathsf{R}^{gen}(\phi)$, $\mathcal{A}$ fails on $\mathcal{G}'$ with the same probability that some folded distribution in $\mathsf{R}^{gen}(\phi)$ fails on $\mathcal{G}$, which is at least s. ◀

▶ **Proposition 28.** *For fixed values of c and s, let $\mathcal{G}$ be a gadget satisfying the (Completeness) and (Soundness) conditions in the statement of Proposition 27. Then there exists another gadget satisfying these conditions which only uses equality constraints.*

Proof. Let $\mathcal{G}'$ be the gadget which replaces each constraint in $\mathcal{G}$ of the form $x \neq y$ with the constraint $x = -y$. If A is a folded assignment,

$$A(x) \neq A(y) \iff A(x) = A(-y).$$

Thus, for every folded assignment A, $\mathsf{val}(A; \mathcal{G}) = \mathsf{val}(A, \mathcal{G}')$. As the (Completeness) and (Soundness) conditions in Proposition 27 only concern folded assignments, $\mathcal{G}'$ satisfies these conditions. ◀

This means that sampling from $\mathcal{G}$ can be written as $(x, y) \sim \mathcal{G}$, meaning that we have sampled the constraint "$x = y$".

2.3 The Had_k-to-2-Lin Gadget

Now we focus on our main setting, which is constructing a Had_k-to-2-Lin(2) gadget. Via Section 2.2, we need only consider how well the gadget does against folded assignments.

The Had_k predicate has $2^k - 1$ variables. In addition, it has $K := 2^k$ satisfying assignments, one for each setting of the variables $x_{\{1\}}$ through $x_{\{k\}}$. It will often be convenient to take an alternative (but equivalent) viewpoint of the variable set $V := \{-1, 1\}^K$ as the set of k-variable Boolean functions, i.e.

$$V = \left\{ f \;\middle|\; f : \{-1, 1\}^k \to \{-1, 1\} \right\}.$$

The Had_k matrix is a $2^k \times (2^k - 1)$ matrix whose rows are indexed by strings in $\{-1, 1\}^k$ and whose columns are indexed by nonempty subsets $S \subseteq [k]$. The (x, S)-entry of this matrix is $\chi_S(x)$. This can be verified by noting that for any $x \in \{-1, 1\}^k$,

$$\big(\chi_{\{1\}}(x), \chi_{\{2\}}(x), \ldots, \chi_{\{k\}}(x), \chi_{\{1,2\}}(x), \ldots, \chi_{\{1,2,\ldots,k\}}(x), \big)$$

is a satisfying assignment of the Had_k predicate. As a result, for each $S \neq \emptyset$, χ_S is a column in the Had_k matrix. Therefore, these functions are the generic primary variables. However, it will be convenient to consider a larger set of functions to be primary. For example, because we plan on using our gadget on folded assignments, χ_S and $-\chi_S$ will always have opposite values, and so the $-\chi_S$'s should also be primary variables. In addition, it is a little unnatural to have every parity function but one be a primary variable, so we will include the constant function $\chi_\emptyset$ and its negation $-\chi_\emptyset$ in the set of primary variables. In total, we have the following definition.

▶ **Definition 29.** The *primary variables* of a Had_k-to-2-Lin(2) gadget are the functions $\pm\chi_S$, for any $S \subseteq [k]$. The remaining functions are auxiliary variables.

To account for the inclusion of $\chi_\emptyset$ as a primary variable, we will have to modify some of our definitions from Section 2.1. We begin by defining a modification to the Had_k predicate.

▶ **Definition 30.** The Had_k^* predicate has 2^k input variables, one for each subset $S \subseteq [k]$. The input string $\{x_S\}_{S\subseteq[k]}$ satisfies Had_k^* if for each S, $x_S = x_\emptyset \cdot \prod_{i\in S} x_{\{i\}}$.

In other words, if $x_\emptyset = 1$, then the remaining variables should satisfy the Had_k predicate, and if $x_\emptyset = -1$, then their negations should. We will say that A *satisfies the* Had_k^* *predicate* if

$$\mathsf{Had}_k^*\left(A\left(\chi_\emptyset\right), A\left(\chi_{\{1\}}\right), \ldots, A\left(\chi_{\{k\}}\right), A\left(\chi_{\{1,2\}}\right), \ldots, A\left(\chi_{[k]}\right)\right) = 1.$$

Otherwise, A *fails to satisfy the* Had_k^* *predicate*. We say that $\mathcal{A}$ is *random on the primary variables* if the tuple

$$\left(A\left(\chi_\emptyset\right), A\left(\chi_{\{1\}}\right), \ldots, A\left(\chi_{\{k\}}\right), A\left(\chi_{\{1,2\}}\right), \ldots, A\left(\chi_{[k]}\right)\right)$$

is, over a random $A \sim \mathcal{A}$, distributed as a uniformly random element of $\{-1,1\}^K$.

▶ **Definition 31.** Denote by $\mathsf{R}(\mathsf{Had}_k)$ the set of folded distributions which are uniformly random on the primary variables.

▶ **Definition 32.** A (c,s)-*gadget* reducing Max-Had_k to Max-2-Lin(2) is a gadget $\mathcal{G}$ satisfying the following properties:

- (Completeness): For every dictator assignment d_i, $\mathsf{uval}(d_i;\mathcal{G}) \le c$.
- (Soundness): For any $\mathcal{A} \in \mathsf{R}(\mathsf{Had}_k)$, $\mathsf{uval}(\mathcal{A};\mathcal{G}) \ge s$.

▶ **Proposition 33.** *The following two statements are equivalent:*

1. *There exists a* (c,s)*-gadget reducing Max-*Had_k *to Max-*2-Lin(2)*.*
2. *There exists a* (c,s)*-generic gadget reducing Max-*Had_k *to Max-*2-Lin(2)*.*

Proof. We prove the two directions separately.

(1) ⇒ (2): Let $\mathcal{G}$ be a (c,s)-gadget reducing Max-Had_k to Max-2-Lin(2). We claim that for any folded $\mathcal{A} \in \mathsf{R}^{gen}(\mathsf{Had}_k)$, $\mathsf{uval}(\mathcal{A};\mathcal{G}) \ge s$. To see this, consider the distribution $\mathcal{A}' \in \mathsf{R}(\mathsf{Had}_k)$ which samples $A \sim \mathcal{A}$ and outputs either A or $-A$, each with half probability. Then $\mathsf{uval}(\mathcal{A}';\mathcal{G}) = \mathsf{uval}(\mathcal{A};\mathcal{G})$, and furthermore we know that $\mathsf{uval}(\mathcal{A};\mathcal{G}) \ge s$. As a result, $\mathcal{G}$ satisfies the (Completeness) and (Soundness) conditions in the statement of Proposition 27, meaning there exists a (c,s)-generic gadget reducing Max-Had_k to Max-2-Lin(2).

(2) ⇒ (1): Let $\mathcal{G}$ be a (c,s)-*generic* gadget reducing Max-Had_k to Max-2-Lin(2). Let $\mathcal{A} \in \mathsf{R}(\mathsf{Had}_k)$, and for $b \in \{-1,1\}$, write $\mathcal{A}_{(b)}$ for $\mathcal{A}$ conditioned on the variable $\chi_\emptyset$ being assigned the value b. Then $b \cdot \mathcal{A}_{(b)}$ (by which we mean the distribution where we sample $A \sim \mathcal{A}_{(b)}$ and output $b \cdot A$) is in $\mathsf{R}^{gen}(\mathsf{Had}_k)$ for both $b \in \{-1,1\}$, and so $\mathsf{uval}\left(b \cdot \mathcal{A}_{(b)};\mathcal{G}\right) \ge s$. As $\mathsf{uval}(\mathcal{A}_{(b)};\mathcal{G}) = \mathsf{uval}\left(b \cdot \mathcal{A}_{(b)};\mathcal{G}\right)$, $\mathsf{uval}(\mathcal{A};\mathcal{G}) \ge s$, and so $\mathcal{G}$ is a (c,s)-gadget reducing Max-Had_k to Max-2-Lin(2). ◀

Combining this with Proposition 25, we have the following corollary.

▶ **Corollary 34.** *Suppose there exists a* (c,s)*-gadget reducing Max-*Had_k *to Max-*2-Lin(2)*. Then for all* $\epsilon > 0$*, given an instance* $\mathcal{I}$ *of Max-*2-Lin(2)*, it is* NP*-hard to distinguish between the following two cases:*

- (Completeness): $\mathsf{uval}(\mathcal{I}) \le c + \epsilon$.
- (Soundness): $\mathsf{uval}(\mathcal{I}) \ge s - \epsilon$.

2.4 Reducing to the length-one case

When constructing good gadgets, we generally want dictators to pass with as high of probability as possible. By Proposition 28, we can assume that our gadget operates by sampling an edge (x, y) and testing equality between the two endpoints. Any such edge of Hamming distance i will be violated by $\frac{i}{K}$ of the dictator assignments. Intuitively, then, if we want to dictators to pass with high probability, we should concentrate the probability mass of our gadget $\mathcal{G}$ on edges of low Hamming distance. The following proposition shows that this is true in the extreme: so long as we are only concerned with maximizing the quantity $\frac{s}{c}$, we can always assume that $\mathcal{G}$ is entirely supported on edges of Hamming distance one.

▶ **Proposition 35.** *Suppose there exists a (c, s)-gadget $\mathcal{G}$ reducing Max-Had_k to Max-2-Lin(2). Then there exists a (c', s')-gadget reducing Max-Had_k to Max-2-Lin(2) using only length-one edges for which*

$$\frac{s'}{c'} \geq \frac{s}{c}.$$

Proof. For each $i \in \{1, \ldots, K\}$, let p_i be the probability that an edge sampled from $\mathcal{G}$ has length i, and let $\mathcal{G}_i$ denote the distribution of $\mathcal{G}$ conditioned on this event. Then sampling from $\mathcal{G}$ is equivalent to first sampling a length i with probability p_i, and then sampling an edge from $\mathcal{G}_i$.

Let $Q = 1 \cdot p_1 + 2 \cdot p_2 + \ldots + K \cdot p_K$, and for each $i \in \{1, \ldots, K\}$ define $q_i = \frac{i \cdot p_i}{Q}$. It is easy to see that the q_i's form a probability distribution. Now we may define the new gadget $\mathcal{G}'$ as follows:

1. Sample a length i with probability q_i.
2. Sample $(x, y) \sim \mathcal{G}_i$.
3. Pick an arbitrary shortest path $x = x_0, x_1, \ldots, x_i = y$ through the hypercube $\{-1, 1\}^K$.
4. Output a uniformly random edge (x_j, x_{j+1}) from this path.

Note that $\mathcal{G}'$ only uses length-one edges. Let $\mathcal{G}'_i$ denote the distribution of $\mathcal{G}'$ conditioned on i being sampled in the first step. (Note that $\mathcal{G}'_i$ is defined in a way that is different from the way $\mathcal{G}_i$ is defined.)

Let $A : \{-1, 1\}^K \to \{-1, 1\}$ be any assignment. Then

$$\mathsf{uval}(A; \mathcal{G}) = \sum_{i=1}^{K} p_i \cdot \mathsf{uval}(A; \mathcal{G}_i), \qquad \text{and} \qquad \mathsf{uval}(A; \mathcal{G}') = \sum_{i=1}^{K} q_i \cdot \mathsf{uval}(A; \mathcal{G}'_i).$$

We can relate $\mathsf{uval}(A; \mathcal{G}'_i)$ to $\mathsf{uval}(A; \mathcal{G}_i)$ as follows: if A assigns different values to the endpoints of the edge $(x, y) \sim \mathcal{G}$, then on any shortest path $x = x_0, x_1, \ldots, x_i = y$ through the hypercube $\{-1, 1\}^K$, A must assign different values to at least one of the edges (x_j, x_{j+1}). As a result, every time A errs on $\mathcal{G}_i$, it must err at least a $(1/i)$-fraction of the time on $\mathcal{G}'_i$. This means that:

$$\mathsf{uval}(A; \mathcal{G}'_i) \geq \frac{\mathsf{uval}(A; \mathcal{G}_i)}{i}. \tag{1}$$

In the case when A is a dictator function, Equation (1) becomes an equality. This is because $x = x_0, x_1, \ldots, x_i = y$ is a shortest path between x and y through the hypercube $\{-1, 1\}^K$. If A assigns the same values to x and y, then it will assign the same values to all of $x_0, x_1, \ldots, x_i$. If, on the other hand, it assigns different values to x and y, then it will assign different values to the endpoints of exactly one edge (x_j, x_{j+1}).

Now we can use this relate $\mathsf{uval}(A;\mathcal{G}')$ to $\mathsf{uval}(A;\mathcal{G})$:

$$\begin{aligned}\mathsf{uval}(A;\mathcal{G}') &= \sum_{i=1}^{K} q_i \cdot \mathsf{uval}(A;\mathcal{G}'_i) \\ &\geq \sum_{i=1}^{K} \left(\frac{i \cdot p_i}{Q}\right) \cdot \frac{\mathsf{uval}(A;\mathcal{G}_i)}{i} \\ &= \frac{1}{Q}\sum_{i=1}^{K} p_i \cdot \mathsf{uval}(A;\mathcal{G}_i) \\ &= \frac{1}{Q}\mathsf{uval}(A;\mathcal{G}). \end{aligned} \tag{2}$$

Here the inequality follows from the definition of q_i and Equation (1). As Equation (1) is an equality in the case when A is a dictator function, we have that $\mathsf{uval}(A;\mathcal{G}') = \frac{1}{Q}\mathsf{uval}(A;\mathcal{G})$ in this case.

Let $\mathcal{A} \in \mathsf{R}(\mathsf{Had}_k)$ maximize $\mathsf{val}(\mathcal{A};\mathcal{G}')$, and let d_i be any dictator function. Then

$$\frac{s'}{c'} = \frac{\mathsf{uval}(\mathcal{A};\mathcal{G}')}{\mathsf{uval}(d_i;\mathcal{G}')} \geq \frac{\frac{1}{Q}\mathsf{uval}(\mathcal{A};\mathcal{G})}{\frac{1}{Q}\mathsf{uval}(d_i;\mathcal{G})} = \frac{\mathsf{uval}(\mathcal{A};\mathcal{G})}{\mathsf{uval}(d_i;\mathcal{G})} \geq \frac{s}{c}.$$

Here the first inequality is by Equation (2) (and the fact that it is an equality for dictators), and the second inequality follows from the fact that $\mathsf{uval}(\mathcal{A},\mathcal{G}) \geq s$ and $\mathsf{uval}(d_i,\mathcal{G}) = c$. ◀

2.5 Linear programs

One of the key insights of the paper [20] is that optimal gadgets (as per Definition 22) can be computed by simply solving a linear program. Fortunately, the same holds for computing optimal gadgets as per Definition 32. In our case, the appropriate linear program (taking into account Proposition 35) is:

$$\begin{aligned} \max \quad & s \\ \text{s.t.} \quad & \mathsf{uval}(\mathcal{A};\mathcal{G}) \geq s, \quad \forall \mathcal{A} \in \mathsf{R}(\mathsf{Had}_k), \\ & \mathcal{G} \text{ is a gadget supported on edges of length one.} \end{aligned}$$

As written, this linear program has an (uncountably) infinite number of constraints, but this can fixed by suitably discretizing the set $\mathsf{R}(\mathsf{Had}_k)$. This is not so important for us, as even after performing this step, the linear program is simply too large to ever be feasible in practice. What is important for us is that we can take its dual; doing so yields the following linear program:

▶ **Definition 36.** The *dual LP* is defined as

$$\begin{aligned} \min \quad & s \\ \text{s.t.} \quad & \Pr_{A\sim\mathcal{A}}[A(x) = A(y)] \leq s, \quad \forall \text{ edges } (x,y) \text{ of length one,} && (3) \\ & \mathcal{A} \in \mathsf{R}(\mathsf{Had}_k). && (4) \end{aligned}$$

The dual linear program shows us that we can upper-bound the soundness of any gadget with the value s by exhibiting a distribution on assignments in $\mathsf{R}(\mathsf{Had}_k)$ which passes each length-one edge with probability at least s. Moreover, strong LP duality tells us that the optimum values of the two LPs are the same. Hence, we can prove a *tight* upper bound by exhibiting the right distribution. We do this in Section 3 for gadgets reducing Max-Had_3 to Max-2-Lin(2).

2.6 The Had_3 gadget

In this section, we will prove some structural results about the hypercube $\{-1,1\}^8$ which are relevant to any Had_3-to-2-Lin(2) gadget. The results of this section will be useful for Section 3.

Given a string $x \in \{-1,1\}^n$ and subset of strings $B \subseteq \{-1,1\}^n$, we define the distance of x to B as $d_H(x,B) := \min_{y \in B} d_H(x,y)$.

▶ **Proposition 37.** *The vertex set $V = \{-1,1\}^8$ can be partitioned as $V = V_0 \cup V_1 \cup V_2$, in which V_0 is the set of primary variables, and $V_i = \{x \in V \mid d_H(x,V_0) = i\}$, for $i = 1,2$.*

▶ **Proposition 38.** *$|V_0| = 16$, $|V_1| = 128$, and $|V_2| = 112$.*

▶ **Proposition 39.**

- *Each $x \in V_0$ has eight neighbors in V_1.*
- *Each $x \in V_1$ has one neighbor in V_0 and seven neighbors in V_2.*
- *Each $x \in V_2$ has eight neighbors in V_1. Furthermore, there are four elements of V_0 which are Hamming distance two away from x.*

▶ **Proposition 40.** *Let $f \in V_2$, and let g_1, g_2, g_3, and g_4 be the four elements of V_0 which are Hamming distance two away from f. Then for any $x \in \{-1,1\}^3$, three of the g_i's have the same value and one has a different value, and $f(x) = \mathsf{sign}(g_1(x) + g_2(x) + g_3(x) + g_4(x))$.*

Proof of Propositions 37, 38, 39, and 40. In this proof, we will take the viewpoint of V as the set of 3-variable Boolean functions. The primary variables are of the form $\pm\chi_S$, where $S \subseteq [3]$. There are 16 such functions, and so $|V_0| = 16$.

Let f' differ from one of the primary variables on a single input. From above, it must be at least distance 3 from any of the other primary variables. This immediately implies that f''s seven other neighbors are in V_2. There are $16 \cdot 8 = 128$ distinct ways of constructng f', and so $|V_1| = 128$.

This leaves $256 - 16 - 128 = 112$ variables in V not yet accounted for. We will now show a method for constructing 112 different elements of V_2; by the pigeonhole principle, this shows that V can be partitioned as Proposition 37 guarantees. Given three primary variables $b_1\chi_{S_1}$, $b_2\chi_{S_2}$, and $b_3\chi_{S_3}$, where $b_1, b_2, b_3 \in \{-1,1\}$, set $b_4 := -b_1 \cdot b_2 \cdot b_3$ and $S_4 := S_1 \Delta S_2 \Delta S_3$. Consider the function $f : \{-1,1\}^3 \to \{-1,1\}$ defined as

$$f(x) := \mathsf{sign}\left(b_1\chi_{S_1}(x) + b_2\chi_{S_2}(x) + b_3\chi_{S_3}(x) + b_4\chi_{S_4}(x)\right).$$

Our claim is that f is distance-2 from each of the $b_i\chi_{S_i}$'s. First, to see that this $\mathsf{sign}(\cdot)$ is well-defined, note that by definition, $\prod_{i=1}^4 b_i\chi_{S_i}(x) = -1$ for all $x \in \{-1,1\}^3$. As a result, for any x, three of the $b_i\chi_{S_i}(x)$'s have the same value, while the other one has a different value. This means that

$$\sum_{i=1}^4 b_i\chi_{S_i}(x) = 2 \cdot \mathsf{sign}\left(\sum_{i=1}^4 b_i\chi_{S_i}(x)\right).$$

for all x. Thus, the correlation of any of the $b_i\chi_{S_i}$'s with f is

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x}) \cdot b_i\chi_{S_i}] = \frac{1}{2}\mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\left(\sum_{i=1}^4 b_i\chi_{S_i}(x)\right) \cdot b_i\chi_{S_i}\right] = \frac{1}{2}.$$

In other words, $\Pr_{\boldsymbol{x}}[f(\boldsymbol{x}) = b_i\chi_{S_i}] = \frac{3}{4}$ for each $i \in \{1,\ldots,4\}$.

There are 8 neighbors of f; each $b_i\chi_{S_i}$ neighbors two of them. As a result, all of f's neighbors are in V_1. In addition, since they are neighbors to the $b_i\chi_{S_i}$'s, they can't be neighbors for any of the other primary variables. This means that the only variables in V_0 that f is distance 2 from are the $b_i\chi_{S_i}$'s.

There are $2 \cdot \binom{8}{3} = 112$ ways of selecting the $b_i\chi_{S_i}$'s. As there are only 112 variables in V which are not in either V_0 or V_1, all of the remaining variables in V must be contained in V_2, and they must all be generated in the manner above. ◀

▶ **Proposition 41.** *Let $B = \mathsf{sat}(\mathsf{Had}_3^*)$. Then*

$$\Pr_{\boldsymbol{x}}[d_H(\boldsymbol{x}, B) = 0] = \frac{1}{16}, \quad \Pr_{\boldsymbol{x}}[d_H(\boldsymbol{x}, B) = 1] = \frac{1}{2}, \text{ and } \Pr_{\boldsymbol{x}}[d_H(\boldsymbol{x}, B) = 2] = \frac{7}{16},$$

where $\boldsymbol{x}$ is a uniformly random element of $\{-1, 1\}^8$.

Proof. This can be proven using a proof similar to Proposition 38. Alternatively, we can just show a direct correspondence between the setting here and the setting in Proposition 38, as follows.

The input to Had_3^* is a set of bits $\{x_S\}_{S\subseteq[k]}$, which can also be thought of as the function $f : \mathcal{P}(\{1, 2, 3\}) \to \{-1, 1\}$ in which $f(S) := x_S$. The satisfying assignments are then any function of the form $S \to b \cdot \chi_S(x)$, where $b \in \{-1, 1\}$ and $x \in \{-1, 1\}^3$ are both fixed. For a string $x \in \{-1, 1\}^3$, let $\alpha(x)$ be the corresponding set, i.e. $\alpha(S)_i = -1$ if and only if $i \in S$. For any function $f : \mathcal{P}(\{1, 2, 3\}) \to \{-1, 1\}$, we can associate it with the function $\alpha(f) : \{-1, 1\}^3 \to \{-1, 1\}$ defined by $\alpha(f)(x) := f(\alpha(x))$ for all x. Then α maps any satisfying assignment to Had_3^* into one of the primary variables in V_0, and more generally, $d_H(f, B) = i$ if and only if $\alpha(f) \in V_i$. The proposition therefore follows by applying Proposition 38 and by noting that $\frac{16}{256} = \frac{1}{16}$, $\frac{128}{256} = \frac{1}{2}$, and $\frac{112}{256} = \frac{7}{16}$. ◀

▶ **Proposition 42.**

1. *Let $f, g \in V_0$ be a pair of distinct affine functions. Then either $d_H(f, g) = 8$, or $d_H(f, g) = 4$.*
2. *For any $x, y \in \{-1, 1\}^3$, $x \neq y$, $b_x, b_y \in \{-1, 1\}$, the number of functions $f \in V_0$ such that $f(x) = b_x$ is 8, and the number of functions $f \in V_0$ such that $f(x) = b_x$ and $f(y) = b_y$ is 4.*

Proof. Proof of (1): Let $f = b_f\chi_S$, and $g = b_g\chi_T$. Then $\mathbf{E}[fg] = b_f b_g \mathbf{E}[\chi_{S\Delta T}]$ where Δ is the symmetric difference of two sets. If $f = -g$, then clearly $d_H(f, g) = 8$. Now we assume that $f \neq \pm g$, and therefore $S \neq T$. Then $\mathbf{E}[\chi_{S\Delta T}] = 0$. This completes the proof.

Proof of (2): Consider function $f(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_3$. Construct a linear system where a_0, a_1, a_2, a_3 are the variables and $f(x) = b_x$ and $f(y) = b_y$ are the constraints. The result follows from working out the size of the solution space. ◀

2.7 Reducing to Max-Cut

▶ **Definition 43.** Let $(\neq) : \{-1, 1\}^2 \to \{0, 1\}$ be the inequality predicate, i.e. $(\neq)(b_1, b_2) = 1$ iff $b_1 \neq b_2$ for all $b_1, b_2 \in \{-1, 1\}$. The Max-Cut CSP is the special case of the Max-$(\neq)$ CSP in which every constraint $\mathcal{C} = ((x_1, b_1), (x_2, b_2))$ satisfies $b_1 = b_2 = 1$. In other words, every constraint is of the form "$x_1 \neq x_2$".

▶ **Proposition 44.** *For some predicate ϕ, suppose $\mathcal{G}$ is (c, s)-generic gadget reducing Max-ϕ to Max-*2-Lin(2)*. Then there exists a (c', s')-gadget reducing Max-ϕ to* Max-Cut *satisfying*

$$\frac{s'}{c'} = \frac{s}{c}.$$

Proof. Suppose the vertex set of $\mathcal{G}$ is $V = \{-1,1\}^K$. Let $\mathcal{G}'$ be the gadget which operates as follows:

1. With probability $1 - \frac{1}{2^{K-1}}$, pick $x \in \{-1,1\}^K$ and output the constraint "$x \neq -x$".
2. Otherwise, sample $\mathcal{C} \sim \mathcal{G}$. If $\mathcal{C}$ is of the form "$x \neq y$", output "$x \neq y$". If $\mathcal{C}'$ is of the form "$x = y$", output "$x \neq -y$".

Any folded assignment A fails $\mathcal{G}$' with probability at most $\frac{1}{2^{K-1}}$. Any assignment A which is *not* folded fails $\mathcal{G}$' with probability at least $\frac{1}{2^{K-1}}$. As a result, we can always assume that any assignment is folded.

Now, if A is folded, then for any $x, y \in \{-1,1\}^K$, $A(x) = A(y)$ if and only if $A(x) \neq A(-y)$. As a result, $\mathsf{uval}(A;\mathcal{G}') = \mathsf{uval}(A;\mathcal{G})/2^{k-1}$. Thus, $c' = c/2^{k-1}$, $s' = s/2^{k-1}$, and so $s'/c' = s/c$. ◀

3 The factor-11/8 gadget and its optimality

In this section, we prove the following main theorem.

▶ **Theorem 45.** *There is a* $\left(\frac{1}{8}, \frac{11}{64}\right)$*-gadget reducing* Had_3 *to* 2-Lin(2)*. By a simple padding argument, this implies that for any* $C < \frac{11}{8}$*, it is* NP*-hard to achieve a factor-*C *approximation for both the Max-*2-Lin(2) *and the* Max-Cut *CSPs.*

Furthermore, the value of the LP in (3) is $\frac{11}{64}$*. This means that for every* (c,s)*-gadget reducing Max-*Had_3 *to Max-*2-Lin(2)*,* $\frac{s}{c} \leq \frac{11}{8}$*. In other words, the gadget we construct is* optimal *among gadget reductions from Chan's* 7*-ary Hadamard predicate.*

The result of NP-hardness for Max-2-Lin(2) and Max-Cut follows from discussions in Section 2. We now focus on the gadget construction part. First, we give the construction of our $\left(\frac{1}{8}, \frac{11}{64}\right)$-gadget reducing Had_3 to 2-Lin(2). Recall that the set of variables in our gadget is the set of Boolean functions on 3 variables. Let $V_0 := \{\pm\chi_S\}_{S\subseteq[k]}$, V_1 be the set of 3-variable Boolean functions that are at distance 1 from some function in V_0, and V_2 be those at distance 2 from some function in V_0.

We will assign a non-negative weight to each constraint in the gadget. Our gadget will then sample each constraint with probability equal to its weight normalized by the weight of the entire gadget. As argued in Proposition 35, the gadget will only use constraints on functions at distance 1. For $f, g \in V$ with $d_H(f,g) = 1$, the weight of the edge $\{f,g\}$ is 5 if and only if either $f \in V_0$ or $g \in V_0$, and otherwise the weight is 1. The total weight of the edges in $\mathcal{G}$ is $5 \times 128 + 896 = 1536$.

To prove completeness, the fact that the dictators pass with probability $\frac{7}{8}$ follows immediately from the fact that $\mathcal{G}$ only uses edges of length one. The soundness is proved by case analysis on the effect of different partial assignments based on the structure of the gadget. Since there are no edges between variables in V_1, whenever we have a partial assignment to the variables in V_0 and V_2, we can complete it optimally by giving assignments greedily to variables in V_1. We then argue that given any folded partial assignment to variables in V_0, there is some greedy heuristics for assigning values to variables in V_2 that always achieves optimum.

To establish optimality of our gadget, we construct an optimal solution to the dual LP given in (3). Our goal is to construct $\mathcal{A} \in \mathsf{R}(\mathsf{Had}_3)$, i.e. a folded distribution of assignments which is random on the primary variables. For $i \in \{0,1,2\}$, denote by $\mathsf{R}_i(\mathsf{Had}_3)$, the set of distributions $\mathcal{A}_i$ such that over a random assignment $A \sim \mathcal{A}_i$, the string $\left(\{A(\chi_S)\}_{S\subseteq[k]}\right)$ is distributed like a uniformly random element of $\{0,1\}^8$ that is at distance i from satisfying the Had_3 predicate. We construct three separate distributions $\mathcal{A}_0$, $\mathcal{A}_1$, and $\mathcal{A}_2$ with the

property that $\mathcal{A}_i \in \mathsf{R}_i(\mathsf{Had}_3)$ for each $i \in \{0,1,2\}$. Then, we set $\mathcal{A} = \frac{1}{16}\mathcal{A}_0 + \frac{1}{2}\mathcal{A}_1 + \frac{7}{16}\mathcal{A}_2$. We study the value of $\mathcal{A}_0$, $\mathcal{A}_1$ and $\mathcal{A}_2$ on edges between V_0 and V_1, and edges between V_1 and V_2, and show that the values of $\mathcal{A}$ on both sets of edges are $\frac{53}{64}$. We complete the proof by noting that $1 - \frac{53}{64} = \frac{11}{64}$, the number guaranteed by the theorem.

A complete proof of Theorem 45 can be found in Section 3 and 4 of the full paper.

4 A candidate factor-3/2 hardness reduction

Herein we present an interesting problem concerning analysis of Boolean functions. We make a conjecture about its solution which, if true, implies NP-hardness of factor-$(\frac{3}{2} - \delta)$ approximating 2-Lin(2) for any $\delta > 0$.

▶ **Definition 46.** Let $g : \{-1,1\}^n \to \{-1,1\}$ be an odd function (i.e., $g(-x) = -g(x)$). The *Game Show*, played with *Middle Function* g, works as follows. There are two players: the *Host* and the *Contestant*. Before the game begins, the Host secretly picks a uniformly random monotone path $\boldsymbol{\pi}$ from $(1,1,\ldots,1)$ to $(-1,-1,\ldots,-1)$ in the Hamming cube. (We say that a path is monotone if at each step, a 1 is changed to a -1. Equivalently, $\boldsymbol{\pi}$ is a uniformly random permutation on $[n]$.) The Host also secretly picks $\boldsymbol{T} \sim \text{Binomial}(n, \frac{1}{2})$. We define the *secret half-path* to be the sequence of the first $\boldsymbol{T}$ edges along $\boldsymbol{\pi}$: $(x_0, x_1), (x_1, x_2), \ldots, (x_{\boldsymbol{T}-1}, x_{\boldsymbol{T}})$. Note that $x_{\boldsymbol{T}}$ is uniformly distributed on $\{-1,1\}^n$.

The Game now begins, with the current *time* being $t = 0$, the current *point* being $x_0 = (1,1,\ldots,1)$, and the current *function* being $\widetilde{g} = g$. (The current function will always be $\pm g$.) At each time step $t = 0, 1, 2, \ldots$, the Host asks whether the Contestant would like to *negate* the current function, meaning replace $\widetilde{g}$ with $-\widetilde{g}$. If the Contestant does not negate the current function there is no cost. However, if the Contestant elects to negate the current function, the Contestant must pay a cost of $w(t) \coloneqq \frac{1}{(1-t/n)^2}$. After the Contestant makes the decision, the Host reveals to the Contestant what the $(t+1)$-th point on the secret half-path is, and increases the time by 1.

As soon as time $\boldsymbol{T}$ is reached, the Game ends. At this instant, if $\widetilde{g}(x_{\boldsymbol{T}}) \neq 1$, then the Contestant incurs a further cost of $w(\boldsymbol{T})$. (It's as though the Contestant is now obliged to negate $\widetilde{g}$.) Thus one can think of the Contestant's goal throughout the Game as trying to ensure that $\widetilde{g}(x_{\boldsymbol{T}})$ will equal 1, while trying to minimize the total cost incurred by all negations.

We define $\text{cost}(g)$ to be the least expected cost that a Contestant can achieve when the Game Show is played with Middle Function g. For $g : \{-1,1\}^n \to \{-1,1\}$ and "negation pattern" $b \in \{-1,1\}^n$, we write g^{+b} to denote the function defined by $g^{+b}(x) = g(b_1 x_1, \ldots, b_n x_n)$.

Roughly speaking, our conjecture about the Game Show is that for every odd g, the average value of $\text{cost}(g^{+b})$ over all b is at least $\frac{3}{2}$. To be precise, we need to be concerned with averaging over merely pairwise-independent distributions on b.

▶ **Game Show Conjecture.** *Let $g : \{-1,1\}^n \to \{-1,1\}$ be odd and let $\mathcal{D}$ be any distribution on $\{-1,1\}^n$ which is pairwise-independent and symmetric (meaning $\Pr_{\mathcal{D}}[\boldsymbol{b}] = \Pr_{\mathcal{D}}[-\boldsymbol{b}]$). Then $\mathbf{E}_{\boldsymbol{b}\sim\mathcal{D}}[\text{cost}(g^{+\boldsymbol{b}})] \geq \frac{3}{2} - o_n(1)$.*

Our motivation for making the Game Show Conjecture is the following result:

▶ **Theorem 47.** *Suppose the Game Show Conjecture is true. Then it is* NP*-hard to approximate* 2-Lin(2) *(and hence also* Max-Cut*) to factor $\frac{3}{2} - \delta$ for any $\delta > 0$.*

The proof of the above theorem can be found in the complete version of the paper.

We remark that given a Middle Function g, in some sense it is "easy" to determine the Contestant's best strategy. It can done with a dynamic program, since the Game Show is essentially a 2-Lin(2) instance on a tree graph. Nevertheless, we have been unable to prove the conjecture. A discussion about some of our efforts in proving the conjecture can be found in the full paper.

5 Limitations of gadget reductions

In this section, we show a limitation to proving inapproximability using gadget reductions from balanced pairwise-independent predicates: that is, predicates ϕ that admit a set $S \subseteq \mathsf{sat}(\phi)$ satisfying Property 2 in Definition 12. We show that gadget reductions from ϕ to 2-Lin(2) can not prove inapproximability larger than a factor-2.54 for the deletion version. Note that this applies to the Had_k predicates and to a broader class of predicates that do not necessarily admit a natural group operation.

▶ **Theorem 48.** *Let $\mathcal{G}$ be a (c, s)-generic gadget reducing Max-ϕ to Max-2-Lin(2), where ϕ admits a balanced pairwise-independent set. Then*

$$\frac{s}{c} \leq \frac{1}{1 - e^{-1/2}} \approx 2.54.$$

Proof. As before, K is the number of satisfying assignments of ϕ. Recall that the vertex set of $\mathcal{G}$ is $V = \{-1, 1\}^K$. Further, via Propositions 27 and 28, we need only consider folded assignments to these variables, and we can assume $\mathcal{G}$ only uses (=)-constraints. Finally, via Proposition 35, we can assume that every (=)-constraint used by $\mathcal{G}$ is between two variables x and y which are Hamming distance one from each other. Let P be the set of generic primary variables, let $-P$ be their negations, and let $P^{\pm} = P \cup (-P)$ denote the union of the two. Since ϕ is balanced pairwise-independent, we have a set $S \subseteq [K]$ so that for i picked uniformly at random from S, $\Pr_i[u_i = v_i] = 1/2$ for distinct primary variables $u, v \in P$.

Define the similarity between x and y to be $\mathsf{sim}(x, y) := \Pr_i[x_i = y_i]$ and set $\mathsf{sim}(x, P^{\pm}) := \max_{y \in P^{\pm}} \mathsf{sim}(x, y)$. Pairwise-independence allows us to claim that any variable x is strongly similar (i.e. has similarity $> \frac{3}{4}$) with at most one variable $y \in P^{\pm}$; define y to be x's *closest* primary variable.

▶ **Fact 49.** *For any $x \in V$, if $\mathsf{sim}(x, y) > \frac{3}{4}$ for some $y \in P^{\pm}$, then $\mathsf{sim}(x, y') < \frac{3}{4}$ for all other $y' \in P^{\pm}$.*

Proof. If x has $\mathsf{sim}(x, y_1) > \frac{3}{4}$ and $\mathsf{sim}(x, y_2) \geq \frac{3}{4}$ for $y_1, y_2 \in P^{\pm}$, then

$$\mathsf{sim}(y_1, y_2) \geq \mathsf{sim}(y_1, x) + \mathsf{sim}(x, y_2) - 1 > \frac{1}{2},$$

contradicting the assumption on ϕ. ◀

This fact allows us to design the following "threshold-rounding" procedure to construct a distribution $\mathcal{A} \in \mathsf{R}^{gen}(\phi)$. Let $C = \frac{2}{e^2 - e^{3/2}}$, and $\mathcal{D}$ be a distribution over $[3/4, 1]$ with probability density function $\mathcal{D}(t) = C \cdot e^{2t}$, for $t \in [3/4, 1]$.

1. Pick a random assignment to the primary variables.
2. Pick a number $t \sim \mathcal{D}$. For any variable $x \in V$, call x type 1 if $\mathsf{sim}(x, P^{\pm}) > t$ and type 2 otherwise.

3. Assign all type-1 variables the value of their closest primary variable.
4. Pick a uniformly random dictator d_i and set all the type-2 variables to agree with this dictator.
5. Output the resulting assignment.

Note that the assignments are folded and are random on the primary variables. We analyse the performance of this assignment. Let (x, y) be an edge in $\{-1, 1\}^K$ of Hamming weight one. If both $\mathsf{sim}(x, P^{\pm}), \mathsf{sim}(y, P^{\pm}) \leq \frac{3}{4}$, then regardless of the value of t, x and y will both always be type-2 variables, in which case $\mathcal{A}$ violates the edge between them with the probability of a random dictator, which is $\frac{1}{K} \leq \frac{1}{1-e^{-1/2}} \cdot \frac{1}{K}$.

On the other hand, suppose WLOG that $\mathsf{sim}(x, P^{\pm}) > \mathsf{sim}(y, P^{\pm})$ and that $\mathsf{sim}(x, P^{\pm}) > \frac{3}{4}$. If we set $s := \mathsf{sim}(y, P^{\pm})$, then $\mathsf{sim}(x, P^{\pm}) = s + \frac{1}{K}$. Because y is distance one from x, $s \geq \frac{3}{4}$. Not only that, if y has a closest primary variable, then that variable is the same as x's closest primary variable (this is by Fact 49). Now, to calculate the probability that $\mathcal{A}$ violates (x, y), there are three cases:

1. If $t \in \left[\frac{3}{4}, s\right)$, then x and y are assigned the value of the same variable in $P^{\pm}$, so (x, y) is never violated in this case.
2. If $t \in \left[s, s + \frac{1}{K}\right)$, then y's value is chosen according to a uniformly random dictator assignment, meaning that it is a uniformly random ± 1-bit. independent from x's value In this case, (x, y) is violated with probabiltiy $\frac{1}{2}$.
3. If $t \in \left[s + \frac{1}{K}, 1\right]$, then both x and y are assigned values according to a random dictator, in which case (x, y) is violated with probability $\frac{1}{K}$.

In total,

$$\begin{aligned}
\Pr[\mathcal{A} \text{ violates } (x,y)] &= \frac{1}{2} \cdot \Pr_{t \sim \mathcal{D}}\left[t \in [s, s + 1/K)\right] + \frac{1}{K} \cdot \Pr_{t \sim \mathcal{D}}\left[t \in [s + 1/K, 1)\right] \\
&= \frac{1}{2} \int_s^{s+\frac{1}{K}} Ce^{2t} \mathrm{d}t + \frac{1}{K} \int_{s+\frac{1}{K}}^1 Ce^{2t} \mathrm{d}t \\
&\leq \frac{1}{2} \cdot \frac{Ce^{2s+2/K}}{K} + \frac{1}{K} \int_{s+\frac{1}{K}}^1 Ce^{2t} \mathrm{d}t \\
&= \frac{Ce^2}{2K} = \frac{1}{1-e^{-1/2}} \cdot \frac{1}{K},
\end{aligned}$$

as promised. Here the inequality follows from the fact that e^{2t} is an increasing function. As $\mathcal{G}$ only uses length-one edges, $c = \frac{1}{K}$. We have just shown that $\mathsf{uval}(\mathcal{A}; \mathcal{G}) \leq \frac{1}{1-e^{-1/2}} \cdot \frac{1}{K}$. Because $\mathcal{A} \in \mathsf{R}^{gen}(\phi)$, we conclude that $\frac{s}{c} \leq \frac{1}{1-e^{-1/2}}$. ◀

6 Conclusion

As mentioned, we view our factor-$\frac{11}{8}$ NP-hardness result more as a proof of concept, illustrating that the longstanding barrier of factor-$\frac{5}{4}$ NP-hardness for Max-Cut/2-Lin(2)/Unique-Games can be broken. There are quite a few avenues for further work:

- Derive a better NP-hardness result for 2-Lin(2) by reduction from Had_4. As one can always embed a Had_3-based gadget into a Had_4-based gadget, this will always yield a hardness of at least $\frac{11}{8}$. But presumably the optimal Had_4-based gadget will do slightly better.
- Since our analysis of the optimal Had_3 gadget is already somewhat complicated, it might be challenging to analyze the Had_4 case explicitly. A weaker but more plausible goal would be to prove (perhaps indirectly) that there *exists* a $\delta_0 > 0$ such that the optimal

Had_4 gadget achieves factor-$(\frac{11}{8}+\delta_0)$ NP-hardness. This would at least definitely establish that $\frac{11}{8}$ is not the "correct answer" either.

- Prove the Game Show Conjecture which would yield the improved NP-hardness factor of $\frac{3}{2}$. It may also be simpler to try to prove a non-optimal version of the conjecture, yielding *some* hardness factor better than $\frac{11}{8}$ but worse than $\frac{3}{2}$.
- For Max-Cut, our work establishes NP-hardness of $(\epsilon, C\epsilon)$-approximation for any $C < \frac{11}{8}$, but only for $\epsilon \leq \epsilon_0$ where ϵ_0 is some not-very-large constant (see full version for details). It would be nice to get a Max-Cut gadget yielding a larger ϵ_0, like the $\epsilon_0 = \frac{1}{8}$ we have for 2-Lin(2).
- A recent result of Gupta, Talwar, and Witmer [12] showed NP-hardness of approximating the (closely related) Non-Uniform Sparsest Cut problem to factor-$\frac{17}{16}$, by designing a gadget reduction from the old $(\frac{4}{21}, \frac{5}{21})$-approximation hardness of Håstad [13]. A natural question is whether one can use ideas from this paper to make a direct reduction from Had_2 or Had_3 to Non-Uniform Sparsest Cut, improving the NP-hardness factor of $\frac{17}{16}$.
- We are now in the situation (similar to the situation prior to [18]) wherein the best NP-hardness factor we know how to achieve for 2-Lin(q) (or Unique-Games) is achieved by taking $q = 2$. In fact, we don't know how to achieve an NP-hardness factor better than $\frac{5}{4}$ for 2-Lin(q) for any $q > 2$, even though 2-Lin(q) is presumably *harder* for larger q. Can this be remedied?
- In light of the limitations described in Section 5, it makes sense to seek alternative methodology of establishing improved NP-hardness for 2-CSPs. An example showing that this is not at all hopeless comes from the decade-old work of Chlebík and Chlebíková [8], which shows NP-hardness of approximating 2-Sat(-Deletion) to factor $8\sqrt{5} - 15 \approx 2.8885$. Their result is essentially a small tweak to the Vertex-Cover hardness of Dinur and Safra [9] and thus indeed uses a fairly radical methodology for establishing two-bit CSP-hardness, namely direct reduction from a specialized Label-Cover-type problem.

Acknowledgments. The authors would like to warmly thank Per Austrin for his assistance with computer analysis of the $\frac{11}{8}$-gadget.

References

1 Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$ approximation algorithms for Min-Uncut, Min-2CNF-Deletion, and directed cut problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 573–581, 2005.

2 Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for Unique Games and related problems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 563–572, 2010.

3 Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 222–231, 2004.

4 Per Austrin and Johan Håstad. On the usefulness of predicates. *ACM Trans. Comput. Theory*, 5(1):1:1–1:24, 2013.

5 Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and non-approximability – towards tight results. *SIAM Journal of Computing*, 27(3):804–915, 1998.

6 Siu On Chan. Approximation resistance from pairwise independent subgroups. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 447–456, 2013.

7 Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for Unique Games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 205–214, 2006.

8 Miroslav Chlebík and Janka Chlebíková. On approximation hardness of the minimum 2SAT-DELETION problem. In *Proceedings of the 29th Annual International Symposium on Mathematical Foundations of Computer Science*, pages 263–273, 2004.

9 Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

10 Uriel Feige and Daniel Reichman. On systems of linear equations with two variables per equation. In *Proceedings of the 7th Annual International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 117–127, 2004.

11 Michel Goemans and David Williamson. A 0.878 approximation algorithm for MAX-2SAT and MAX-CUT. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 422–431, 1994.

12 Anupam Gupta, Kunal Talwar, and David Witmer. Sparsest Cut on bounded treewidth graphs: algorithms and hardness results. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 281–290, 2013.

13 Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

14 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, 2002.

15 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for Max-Cut and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

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

17 Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. *Annals of Mathematics*, 171(1):295–341, 2010.

18 Ryan O'Donnell and John Wright. A new point of NP-hardness for Unique-Games. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 289–306, 2012.

19 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM Journal of Computing*, 40(6):1871–1891, 2011.

20 Luca Trevisan, Gregory Sorkin, Madhu Sudan, and David Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000.

21 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2-3):357–365, 2005.