Electronic Colloquium on Computational Complexity, Report No. 125 (2013)

Complexity of approximating CSP with Balance / Hard Constraints

Venkatesan Guruswami[*] Euiwoong Lee[†]

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213.

Abstract

We study two natural extensions of Constraint Satisfaction Problems (CSPs). *Balance*-Max-CSP requires that in any feasible assignment each element in the domain is used an equal number of times. An instance of *Hard*-Max-CSP consists of *soft constraints* and *hard constraints*, and the goal is to maximize the weight of satisfied soft constraints while satisfying *all* the hard constraints. These two extensions contain many fundamental problems not captured by CSPs, and challenge traditional theories about CSPs in a more general framework.

Max-2-SAT and Max-Horn-SAT are the only two nontrivial classes of Boolean CSPs that admit a *robust* satisfibiality algorithm, i.e., an algorithm that finds an assignment satisfying at least $(1 - g(\varepsilon))$ fraction of constraints given a $(1 - \varepsilon)$-satisfiable instance, where $g(\varepsilon) \to 0$ as $\varepsilon \to 0$, and $g(0) = 0$. We prove the inapproximability of these problems with balance or hard constraints, showing that each variant changes the nature of the problems significantly (in different ways). For instance, deciding whether an instance of 2-SAT admits a balanced assignment is NP-hard, and for Max-2-SAT with hard constraints, it is hard to find a constant-factor approximation even on $(1 - \varepsilon)$-satisfiable instances (in particular, the version with hard constraints does not admit a robust satisfiability algorithm).

We also study hardness results for a certain CSP over larger domain capturing ordering constraints: we show that hard constraints rule out constant-factor approximation algorithms. All our hardness results are almost optimal — they completely rule out algorithms with certain properties, or can be matched by simple extensions to existing algorithms.

1 Introduction

The study of the complexity of Constraint Satisfaction Problems (CSPs) has seen much progress, with beautiful well-developed theories explaining when they admit efficient satisfiability and approximation algorithms. A CSP is specified by a finite set Π of relations (relations can have different arities) over some finite domain Q. An instance of a Max-CSP(Π) consists of a set of variables $X = \{x_1, ..., x_n\}$ and a collection of constraints $C = \{C_1, ..., C_m\}$ each of which is a relation from Π applied to some tuple of variables from X. Constraints are weighted and we assume that

[*] Supported in part by a Packard Fellowship and NSF grant CCF-1115525. guruswami@cmu.edu

[†] Supported by a Samsung Fellowship, MSR-CMU Center for Computational Thinking, and NSF CCF-1115525. euiwoonl@cs.cmu.edu

ISSN 1433-8092

$\sum_i \mathsf{wt}(C_i) = 1$. For any assignment $\sigma : X \to Q$, $\mathsf{val}(\sigma)$ is the total weight of satisfied constraints by σ, and our goal is to find σ that maximizes $\mathsf{val}(\sigma)$. We consider two natural extensions of Max-CSP(Π).

Definition 1.1 (CSP with balance constraints)**.** *An instance of Balance-Max-CSP(*Π*) over domain* Q*,* $\mathcal{I} = (X, C)$ *consists of set of variables* X *and a collection of constraints* C*, as in Max-CSP(*Π*). An assignment* $\sigma : X \to Q$ *is called balanced if for each* $q \in Q$*,* $|\sigma^{-1}(q)| = \frac{n}{|Q|}$*. Define* $\mathsf{val}_B(\sigma) = \mathsf{val}(\sigma)$ *if* σ *is balanced, and* $\mathsf{val}_B(\sigma) = 0$ *otherwise. Let* $\mathsf{opt}_B(\mathcal{I}) = \max_\sigma \mathsf{val}_B(\sigma)$*. Our goal is to find* σ *that maximizes* $\mathsf{val}_B(\sigma)$*.*

The notion of Balance-Max-CSP is interesting both practically and theoretically. Partitioning a set of objects into equal-sized subsets with desired properties is a basic scheme used in Divide and Conquer algorithm. Balance-Max-Cut, also known as Max-Bisection, is one of the most well-known example of Balance-Max-CSP. Theoretically, the balance constraint is one of the most simple *non-local* constraints where the current algorithmic and hardness results on ordinary CSPs do not work.

Definition 1.2 (CSP with hard constraints)**.** *An instance of Hard-Max-CSP(*Π*)* $\mathcal{I} = (X, S, H)$ *consists of set of variables* X *and a two collections of constraints* $S = \{S_1, ..., S_{m_S}\}$ *and* $H = \{H_1, ..., H_{m_H}\}$ *(*S *stands for* soft *constraints and* H *stands for* hard *constraints, and only soft constraints are weighted). An assignment* $\sigma : X \to Q$ *is* feasible *if it satisfies all constraints in* H*. Let* $\mathsf{val}_H(\sigma)$ *be the total weight of satisfied constraints in* S *if* σ *is feasible, and* 0 *otherwise. Let* $\mathsf{opt}_H(\mathcal{I}) = \max_\sigma \mathsf{val}_H(\sigma)$*. Our goal is to find* σ *that maximizes* $\mathsf{val}_H(\sigma)$*.*

Hard-Max-CSP contains every Max-CSP by definition, and also several additional fundamental combinatorial optimization problems, such as (Hypergraph) Independent Set, Multicut, Graph-k-Coloring, and many other covering/packing problems. While every assignment is *feasible* in ordinary Max-CSP, in Hard-Max-CSP only certain assignments which satisfy all the hard constraints are considered as feasible, giving a more general framework to study combinatorial optimization problems.

By the seminal work of Schaefer [29], there are only three nontrivial classes of Boolean CSPs for which satisfiability can be checked in polynomial time: 2-SAT, Horn-SAT, and LIN-mod-2.[1] Even among them, there is a stark difference in terms of approximability. Hastad [20] showed that for Max-LIN-mod-2, it is NP-hard to find an assignment that satisfies $(\frac{1}{2} + \varepsilon)$ fraction of constraints even when there is an assignment that satisfies $(1 - \varepsilon)$ fraction of constraints, for any $\varepsilon > 0$. On the other hand, a series of works [14, 32, 7] showed that Max-2-SAT and Max-Horn-SAT admit a *robust* algorithm, which outputs an assignment satisfying at least $(1 - g(\varepsilon))$ fraction of constraints given a $(1 - \varepsilon)$-satisfiable instance, where $g(\varepsilon) \to 0$ as $\varepsilon \to 0$, and $g(0) = 0$. The exact behavior of the function $g()$ for Max-2-SAT and Max-Horn-3SAT has been pinned down under the Unique Games conjecture [23, 18]. Generalizing this, all Boolean CSPs were classified with respect to how well they can be robustly approximated (i.e., the behavior of the function $g(\cdot)$) in [9]. From our perspective, it is natural to investigate the effects of balance and hard constraints applied to the

[1] An instance of Horn-SAT is a set of Horn clauses, each with at most one unnegated literal. An instance of LIN-mod-2 is linear equations mod 2. Dual-Horn-SAT in which clauses have at most one negated literal also admits an efficient satisfability algorithm, but as it obviously has the same properties as Horn-SAT, we focus on Horn-SAT in this paper.

most tractable classes of Boolean CSPs (Max-2-SAT, Max-Horn-SAT, Max-Cut) and study how hard each variant becomes. This is the task we undertake in this paper.

Between balance and hard constraints, which one makes the original problem harder? They are not directly comparable since each variant inherits different characteristics of the original problem. For Balance-Max-SAT which includes both Balance-Max-2SAT and Balance-Max-Horn-SAT, it is easy to find σ with $\mathsf{val}_B(\sigma) \geq 0.5$ – choose an arbitrary assignment with the same number of 0's and 1's, and try it and its complement. Therefore, Balance-Max-SAT admits a constant-factor approximation algorithm, but given an instance that admits a balanced satisfying assignment, even for 2-SAT and Horn-SAT, it is not clear how to find such an assignment.

The situation is exactly opposite in the case of Hard-Max-2-SAT or Hard-Max-Horn-SAT. Given an instance $\mathcal{I}$ that admits a satisfying (both hard and soft constraints) assignment (i.e., $\mathsf{opt}_H(\mathcal{I}) = 1$), well-known algorithms for 2-SAT or Horn-SAT will find such an assignment. However, for any $\varepsilon > 0$, when $\mathsf{opt}_H(\mathcal{I}) = 1 - \varepsilon$, it is not clear how to find σ such that $\mathsf{val}_H(\sigma) \geq c$ for some absolute constant c, or even $\mathsf{val}_H(\sigma) \geq \varepsilon$. Therefore, adding hard constraints preserves the fact that satisfiability can be checked in polynomial time, but does not preserve a simple constant-factor approximation algorithm.

Though each of balance and hard constraints does not preserve one of the important characteristics of the original CSP in a simple way, there might be a hope that a more sophisticated algorithmic idea gives an efficient algorithm deciding satisfiability for Balance-Max-CSP, a constant-factor approximation algorithm for Hard-Max-CSP, or a robust algorithm for both of them.

1.1 Our Results

In this work, we prove strong hardness results for these problems, proving APX-hardness of deciding satisfiability of balanced versions and Unique Games-hardness of constant factor approximation algorithms on $(1 - \varepsilon)$-satisfiable instances of hard versions, of both Max-2-SAT and Max-Horn-SAT. The results are formally stated below.

Balanced CSP. For Balance-Max-2-SAT and Balance-Max-Horn-SAT, the Schaefer-like dichotomy due to [8] for the (decision version of) Boolean Balance-CSP, implies that we cannot efficiently decide whether the given instance is satisfiable or not. (This dichotomy was extended to all domains in [6], which is somewhat surprising given the status of the dichotomy conjecture for CSPs without any balance/cardinality constraint.)

We show the following stronger statement that rules out even a robust satisfiability algorithm for the common special case Balance-Max-Horn-2-SAT.

Theorem 1.3. *There exists an absolute constant $\delta > 0$ such that given an instance $\mathcal{I}$ of Balance-Max-Horn-2-SAT (special case of Balance-Max-2-SAT and Balance-Max-Horn-SAT), it is NP-hard to distinguish the following cases.*

- $\mathsf{opt}_B(\mathcal{I}) = 1$
- $\mathsf{opt}_B(\mathcal{I}) \leq 1 - \delta$

This result should be contrasted with the fact that a special case of Balance-Max-2-SAT, namely Balance-Max-Cut (Max-Bisection), does admit a robust algorithm [16, 28, 2]. The work [2] also shows that the guaranteed approximation ratio for Balance-Max-2-SAT (which is the worst ratio

$\frac{\mathsf{val}_B(\sigma)}{\mathsf{opt}_B(\mathcal{I})}$ over every instance $\mathcal{I}$ and σ found by the algorithm) is indeed equal to the best known approximation ratio for Max-2-SAT [25], i.e., $\alpha_{LLZ} \approx 0.9401$, so adding the balance constraint does not make the problem harder in this regard.

While several nontrivial approximation algorithms for Balance-Max-2-SAT have been studied, as far as we know, no algorithm for Balance-Max-Horn-SAT or even Balance-Max-SAT has been suggested in the literature. Other than the trivial 0.5-approximation algorithm given above, we also show that a slight modification to $\frac{3}{4}$-approximation algorithm due to Goemans and Williamson [14] gives the algorithm with the same ratio.

Theorem 1.4. *For any $\varepsilon > 0$, there is a randomized algorithm such that given an instace $\mathcal{I}$ of Balance-Max-SAT, in time poly$(size(\mathcal{I}), \frac{1}{\varepsilon})$, outputs σ with $\mathsf{val}_B(\sigma) \geq (\frac{3}{4} - \varepsilon)\mathsf{opt}_B(\mathcal{I})$ with constant probability.*

Hard CSP. In Hard-Max-2SAT, robust algorithms are ruled out in a more radical way, assuming the Unique Games Conjecture. In the sequel, we use the phrase "UG-hard" for a decision problem to mean that it is NP-hard under the Unique Games Conjecture (Conjecture 2.3).

Theorem 1.5. *For any $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-2-SAT, it is UG-hard to distinguish the following cases.*

- $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

The above result again shows a stark difference between Max-2-SAT and Max-Cut since the famous algorithm of Goemans and Williamson [15] works well with hard constraints; if vertices u and v must be separated, we require the vectors corresponding to them to be placed in antipodal positions, and any hyperplane rounding separates them. It shows that Hard-Max-Cut admits *both* a constant-factor approximation algorithm and a robust algorithm while Hard-Max-2-SAT admits *neither* of them.

For Hard-Max-Horn-2-SAT, simple algorithmic and hardness tricks show that finding σ with $\mathsf{val}_H(\sigma) \geq 1 - 2\varepsilon$ given $\mathsf{opt}_H(\mathcal{I}) = 1 - \varepsilon$ is the best possible, assuming the Unique Games Conjecture.

Observation 1.6. *There is an algorithm that for any $\varepsilon > 0$, if the given instance $\mathcal{I}$ of Max-Horn-2-SAT satisfies $\mathsf{opt}_H(\mathcal{I}) = 1 - \varepsilon$, finds σ with $\mathsf{val}_H(\sigma) \geq 1 - 2\varepsilon$. Furthermore, for any $\varepsilon, \delta > 0$, it is UG-hard to distinguish the following cases.*

- $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon - \delta$.
- $\mathsf{opt}_H(\mathcal{I}) \leq 1 - 2\varepsilon + \delta$.

The above algorithmic trick does not work for other robust algorithms, and for Hard-Max-Horn-3-SAT (or higher arities), we have the following NP-hardness result.

Theorem 1.7. *For any $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-Horn-3-SAT, it is NP-hard to distinguish the following cases.*

- $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

		Ordinary	Balance	Hard
Max-2-SAT	Ratio	0.9401 [25]	0.9401 [2]	**N/A**
	Robust	$(1-\varepsilon, 1-O(\sqrt{\varepsilon}))$ [7]	**N/A**	**N/A**
	Hardness	UG: $(1-\varepsilon, 1-\Omega(\sqrt{\varepsilon}))$ [23]	**NP:** $(1, 1-\delta)$	**UG:**$(1-\varepsilon, \varepsilon)$
Max-Cut	Ratio	0.8786 [14]	0.8776 [2]	0.8786 [14]
	Robust	$(1-\varepsilon, 1-O(\sqrt{\varepsilon}))$ [14]	Same as left [28]	Same as left [14]
	Hardness	UG: $(1-\varepsilon, 1-\Omega(\sqrt{\varepsilon}))$ [23]	Same as left [23]	Same as left [23]
Max-Horn-2-SAT	Ratio	0.9401 [25]	0.9401 [2]	**N/A**
	Robust	$(1-\varepsilon, 1-2\varepsilon)$ [18]	**N/A**	$(1-\varepsilon, 1-2\varepsilon)$ [18]
	Hardness		**NP:** $(1, 1-\delta)$	**UG:**$(1-\varepsilon, 1-2\varepsilon)$
Max-Horn-SAT	Ratio	0.7968 [4]	**0.75**	**N/A**
	Robust	$(1-\varepsilon, 1-O(\frac{\log\log(1/\varepsilon)}{\log(1/\varepsilon)}))$ [18]	**N/A**	**N/A**
	Hardness	UG: $(1-\varepsilon, 1-\Omega(\frac{1}{\log(1/\varepsilon)}))$ [32]	**NP:** $(1, 1-\delta)$	**NP:**$(1-\varepsilon, \varepsilon)$

Table 1: Summary of several Balance-Max-CSP and Hard-Max-CSP. $\varepsilon > 0$ indicates an arbitrary positive constant, while $\delta > 0$ is a fixed absolute constant. Ratio indicates the best approximation ratio. $(1-\varepsilon, 1-f(\varepsilon))$ in a row labeled Robust indicates that there is a robust algorithm that find an assignment satisfying $(1-f(\varepsilon))$ fraction of constraints given an $(1-\varepsilon)$-satisfiable instance. $(1-\varepsilon, 1-f(\varepsilon))$ in a row labeled Hardness indicates that it is NP-hard to find an assignment satisfying $(1-f(\varepsilon))$ fraction of constraints given an $(1-\varepsilon)$-satisfiable instance. NP indicates that it is an NP-hardness result; UG indicates that it is based on the Unique Games Conjecture. N/A means that a robust or constant-factor approximation algorithm are ruled out by the hardness results.

The complete algorithmic and hardness results are summarized in Table 1.1.

Ordering constraints over larger domain. If we go beyond the Boolean domain, the situation is not as clear; indeed even satisfiabilities of ordinary CSPs are not completely classified yet. We consider a simple and natural CSP of arity two over larger domains, namely Max-CSP($<$) over domain $[q] = \{1, 2, ..., q\}$ – every constraint says that one variable must be less than another. This problem can also be understood as a graph-theoretic problem; given a directed graph $G = (V, E)$, delete the minimum number of edges so that the remaining graph has no *walk* of length q (for the formal definition, see Section 4). If the domain is unbounded (indeed, $[n]$ is enough), this is exactly the well-known Max-Acyclic-Subgraph problem.

For all of these problems, a random assignment will give a constant-factor approximation algorithm. We observe that for fixed q, Max-CSP($<$) over $[q]$ also admits a robust algorithm, and the same holds also for Hard-Max-CSP($<$) over $[q]$. However, we show that adding hard constraints rules out the possibility of a constant-factor approximation algorithm, regardless of whether $[q]$ is fixed or unbounded.

Theorem 1.8. *For any $k \geq 2$ and $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-CSP($<$) over $[2k+1]$, it is UG-hard to distinguish the following cases.*

- $\mathsf{opt}_H(\mathcal{I}) \geq (1-\varepsilon)\frac{k-1}{k}$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

Over $\{1, 2, 3, 4\}$, it is also UG-hard to distinguish whether $\mathsf{opt}_H(\mathcal{I}) \geq \frac{1-\varepsilon}{2}$ or $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$.

Also, for any $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-CSP($<$) over the unbounded domain (an instance of Max-Acyclic-Subgraph with several edges that must be included), it is UG-hard to distinguish the following cases.

- $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

We also provide a constant-factor approximation algorithm for Hard-Max-CSP($<$) over $\{1, 2, 3\}$ (albeit with a worse ratio than its ordinary counterpart), showing a difference between $\{1, 2, 3\}$ and $\{1, 2, 3, 4\}$, and complete the picture of the approximability of these Ordering CSPs with hard constraints.

1.2 Discussion

Max-Cut is one of the simplest and well-studied CSPs. Its variants with balance or hard variants also inherit the desirable algorithmic properties of the original version — in particular both these variants admit robust satisfiability as well as constant-factor approximation algorithms. However, these algorithmic results do not extend even slightly more general CSPs; in Max-2-SAT, even one of balance and hard variants rule out the possibility of a robust algorithm. Furthermore, Hard-Max-2-SAT does not have a constant-factor approximation algorithm as well, and the hardness result for Balance-Max-2-SAT also holds for Balance-Max-Horn-2-SAT (which does not even capture Max-Cut).

These delicate differences even for CSPs of arity two over Boolean domain (where satisfiability is always in P) suggest that adding balance or hard constraints highlights previously unnoticed characteristics with respect to the approximability of each predicate. This work settles the approximability of almost all interesting arity two CSPs over Boolean domain, demonstrating the effect of hard and balance constraints on some classic problems.

While our techniques are relatively simple twists to known ones (with the one underlying the inapproximability of Hard-Max-2-SAT being the most significant), we think the body of results highlights subtle (and perhaps surprising) differences between CSPs in the presence of hard/balance constraints, and raises the challenge of extending our study to larger arity or domains. Attempts at such a generalization should be interesting by itself and also illuminate new aspects of CSPs.

In this regard, we record some simple observations on robust algorithms for Hard-Max-CSP(Π). In the weighted minimization version (Min-Uncut, Min-CNF-Deletion and Min-Feedback-Arc-Set are minimization versions of some problems we studied in this work), where the goal is to *minimize* the weight of *unsatisfied* constraints, hard constraints do not add to the difficulty of the original version as can be seen via the simple trick of converting all hard constraints to soft constraints and giving them large enough weight. Therefore, a constant-factor approximation algorithm for the minimization version will yield a robust satisfiability algorithm for Hard-Max-CSP(Π) as well. In general we do not have a reduction in the reverse direction as the robust algorithm might be allowed runtime exponential in ε (e.g. $n^{O(\frac{1}{\varepsilon})}$) on $(1 - \varepsilon)$-satisfiable instances. Also, when the approximation factor for the minimization version depends on the instance size (e.g. $O(\sqrt{\log n})$ for Min-CNF-Deletion [1]), we do not automatically get a robust algorithm for the associated CSP with hard constraints.

We have an example, namely Hard-Max-2-SAT, where constant-factor and robust approximations are both hard in the presence of hard constraints, and an example (Hard-Max-Horn-2-SAT) where robust approximation is possible but a constant-factor approximation is hard. It is easy to see that Hard-Max-Lin-mod-2 admits an efficient constant factor approximation but not a robust algorithm; its ordinary version (without hard constraints), however, also has the same property. This raises the following interesting question: Is there a CSP which admits an efficient robust satisfiability algorithm, and whose hard version admits a constant-factor approximation algorithm but no robust algorithm? Or must a robust algorithm also exist for the hard version under these conditions?

1.3 Organization

Section 2 proves the results about Hard-CSP; Hard-Max-2-SAT in Section 2.2 and Hard-Max-Horn-SAT in Section 2.3. This section requires some understanding of the inner workings of the reduction from Unique Games to Min-Vertex-Cover [24, 5], so we present the relevant preliminaries in Section 2.1. Section 3 proves the results about Balance-CSP; hardness results in Section 3.1 and algorithmic results in Section 3.2. Section 4 proves the results about Ordering CSP.

2 CSP with hard constraints

2.1 Preliminaries

Unique Games. Our hardness result for Max-Hard-2-SAT relies on the Unique Games Conjecture [22], which is stated below.

Definition 2.1. *An instance $\mathcal{L}(G(V,W,E),[R],\{\pi(v,w)\}_{(v,w)\in E})$ of Unique Games consists of a regular bipartite graph $G(V,W,E)$ and a set $[R]$ of labels. For each edge $(v,w)\in E$ there is a constraint specified by a permutation $\pi(v,w):[R]\to[R]$. The goal is to find a labeling $l: V\cup W\to [R]$ of the vertices such that as many edges as possible are satisfied, where an edge $e=(v,w)$ is said to be satisfied if $l(v)=\pi(v,w)(l(w))$.*

Definition 2.2. *Given a Unique Games instance $\mathcal{L}(G(V,W,E),[R],\{\pi(v,w)\}_{(v,w)\in E})$, let $\mathsf{opt}(\mathcal{L})$ denote the maximum fraction of simultaneously satisfied edges of $\mathcal{L}$ by any labeling, i.e.*

$$\mathsf{opt}(\mathcal{L}) := \frac{1}{|E|}\max_{l:V\cup W\to[R]} |\{e\in E : l \text{ satisfies } e\}|$$

Conjecture 2.3 ([22])**.** *For any constants $\varepsilon>0$, there is $R=R(\varepsilon)$ such that, for a Unique Games instance $\mathcal{L}$ with label set $[R]$, it is NP-hard to distinguish between*

- $\mathsf{opt}(\mathcal{L})\geq 1-\varepsilon$.
- $\mathsf{opt}(\mathcal{L})\leq \varepsilon$.

To show the optimal hardness result for Min-Vertex-Cover, Khot and Regev [24] introduced the following stronger conjecture, and proved that it is in fact equivalent to the original the Unique Games Conjecture.

Conjecture 2.4 ([24])**.** *For any constants $\varepsilon>0$, there is $R=R(\varepsilon)$ such that, for a Unique Games instance $\mathcal{L}$ with label set $[R]$, it is NP-hard to distinguish between*

- *There is a set $V' \subseteq V$ such that $|V'| \geq (1-\varepsilon)|V|$ and a labeling $l : V \cup W \to [R]$ that satisfies every edge (v, w) for $v \in V'$ and $w \in W$.*
- $\mathsf{opt}(\mathcal{L}) \leq \varepsilon$.

Fourier Analysis on Biased Hypercubes. Fix $p = 1/2 - \varepsilon$ for some $\varepsilon > 0$. We let $\{0,1\}^R_{(p)}$ be the R-dimensional boolean hypercube with p-biased distribution; each coordinate of $x \in \{0,1\}^R_{(p)}$ is independently set to 1 with probability p and 0 with probability $1-p$. Let $f : \{0,1\}^n_{(p)} \to \mathbb{R}$.

Definition 2.5. *The* influence *of the ith coordinate on f is given by*

$$\mathsf{Inf}_i(f) = \mathbb{E}_{(x_j)_{j \neq i}}[\mathsf{Var}_{x_i}[f(x)|(x_j)_{j \neq i}]]$$

Definition 2.6. *Let $\rho \in [-p/(1-p), 1]$. The* noisy operator *T_ρ acts on f by*

$$T_\rho f(x) = \mathbb{E}_y[f(y)] \ ,$$

where each bit y_i of y is independently sampled such that it is equal to x_i with probability ρ and sampled according to p-biased distribution with probability $(1-\rho)$. We call

$$\mathsf{Inf}_i(T_\rho f)$$

the noisy influence *of the ith coordinate.*

2.2 Max-Hard-2-SAT

At the heart of every hardness result based on the Unique Games Conjecture is an appropriate dictatorship test. The dictatorship test for Min-Vertex-Cover (and its maximization version Max-Independent-Set) is a graph $G = (V, E)$ where $V = \{0,1\}^R_{(p)}$ (vertices are weighted according to p-biased distribution). This graph must ensure that

- Completeness: For each $1 \leq i \leq R$, ith dictator function $f_i(x_1, ..., x_R) = x_i$ is the indicator function of a large independent set.
- Soundness: For every moderate-sized independent set D, its indicator function has a coordinate with large noisy influence.

The edges are constructed such that there is an edge between $(x_1, ..., x_R)$ and $(y_1, ..., y_R)$ if and only if there is no coordinate i such that $x_i = y_i = 1$. The first condition can be easily checked since $f_i(x) = f_i(y) = 1$ for some i means $x_i = y_i = 1$ and there is no edge between x and y. This construction was first used in Khot and Regev [24]. Later Austrin et al. [3] gave a different (and arguably simpler) analysis of the same test, relying on the invariance principle of Mossel et al. [26].

Theorem 2.7 (Implicit in [3])**.** *Let $G = (V, E)$ be the graph constructed as above.*

- *Completeness: For each $1 \leq i \leq R$, ith dictator function $f_i(x_1, ..., x_R) = x_i$ is the indicator function of a large independent set of weight p.*
- *Soundness: For every $\varepsilon > 0$, there exist $\tau > 0$ and $\delta > 0$ such that the following holds. If D is an independent set of weight at least ε and $f : \{0,1\}^R_{(p)} \to \{0,1\}$ is the indicator function of D, then there exists i such that $\mathsf{Inf}_i(T_{1-\delta} f) \geq \tau$.*

Combined with the standard technique converting a dictatorship test to a hardness result based on the Unique Games Conjecture, it is shown that it is UG-hard to distinguish whether the maximum independent set has weight at least $1/2 - \varepsilon$ or at most ε.

There is a simple approximation-preserving reduction from Max-Independent-Set to Hard-Max-Horn-2-SAT. Given G constructed as above,

- $X = V$; each variable corresponds to one vertex.
- For each edge $(x, y) \in E$, add a hard constraint $(\neg x \vee \neg y)$.
- For each vertex $x \in V$, add a soft constraint x.

Hard constraints ensure that two variables corresponding to neighboring vertices cannot be set to True simultaneously, and maximizing the total weight of satisfied soft constraints is equivalent to maximizing the total weight of the vertices set to True. Therefore, the same hardness result also holds for Hard-Max-2-SAT. This hardness result rules out any constant-factor approximation algorithm, but does not apply to instances that are almost satisfiable. In fact, Hard-Max-Horn-2-SAT inherits the same hardness result (as the above reduction only uses Horn clauses), but it has a robust algorithm (See Section 2.3).

Our crucial observation to *amplify* this gap to $1 - \varepsilon$ and ε is that dictatorship functions always give different values to a pair of antipodal points. Since our predicate is a disjunction of two variables (vertices), if we change soft constraints so that

$$S = \left\{ (u \vee v) : \{u, v\} \text{ is an antipodal pair in } \{0,1\}^R_{(p)} \right\}$$

Then, every dictator function f_i will satisfy all the hard and soft constraints. To ensure the soundness, we have to *perturb* the distribution a little bit, but still almost all the weight will be concentrated around antipodal points. Given the dictatorship test $G = (V, E)$ for Min-Vertex-Cover, the dictatorship test for Hard-Max-2-SAT is $\mathcal{I} = (X, S, H)$ such that

- $X = V$; each variable corresponds to one vertex.
- For each edge $(x, y) \in E$, add a hard constraint $(\neg x \vee \neg y)$.
- Sample an ordered pair (x, y) such that for each i,

  $$\mathbb{P}[x_i = y_i = 0] = 2\varepsilon \qquad \text{and} \qquad \mathbb{P}[x_i = 1, y_i = 0] = \mathbb{P}[x_i = 0, y_i = 1] = 1/2 - \varepsilon \ .$$

  Add a soft constraint $(x \vee y)$ with the weight equal to the probability of (x, y).

With this trick, we can obtain a dictatorship test for Hard-Max-2-SAT with a much larger gap.

Theorem 2.8. *Let $\mathcal{I} = (X, S, H)$ be the instance of Hard-Max-2-SAT constructed as above.*

- *Completeness: For each $1 \le i \le R$, ith dictator function $f_i(x_1, ..., x_R) = x_i$ satisfies $\mathsf{val}_H(f_i) = 1 - 2\varepsilon$.*
- *Soundness: For every $\varepsilon > 0$, there exist $\tau > 0$ and $\delta > 0$ such that the following holds. If $f : \{0,1\}^R_{(p)} \to \{0,1\}$ satisfies $\mathsf{val}_H(f) \ge 2\varepsilon$, then there exists i such that $\mathsf{Inf}_i(T_{1-\delta} f) \ge \tau$.*

Proof. Completeness: It is already shown that ith dictator function f_i satisfies all the hard constraints. The only soft constraints that f_i fail to satisfy is $(x \vee y)$ where $x_i = y_i = 0$. Since the probability of picking $x_i = y_i = 0$ is 2ε, $\mathsf{val}_H(f_i) = 1 - 2\varepsilon$.

Soundness: If f is a function that satisfies all the hard constraints, it is also the indicator function of some independent set D. We claim that the weight of D in $\{0,1\}^R_{(p)}$ with $p = 1/2 - \varepsilon$ is at least $\mathsf{val}_H(f_i)/2$. For $z \in \{0,1\}^R_{(p)}$, let $\mathsf{deg}(z)$ be the sum of the weights of soft constraints $(z \vee x)$ or $(x \vee z)$ for some x ($(z \vee z)$ contributes twice). $(x, y) \sim S$ indicates that it is sampled according to its weight defined above. Note that its marginal distribution of x (and y) is exactly the p-biased distribution on $\{0,1\}^R_{(p)}$. Therefore,

$$\begin{aligned}
\mathsf{deg}(z) &= \mathbb{P}_{(x,y)\sim S}[z = x] + \mathbb{P}_{(x,y)\sim S}[z = y] \\
&= 2\ \mathbb{P}_{(x,y)\sim S}[z = x] \\
&= 2\ \mathbb{P}_{x\sim\{0,1\}^R_{(p)}}[z = x] \\
&= 2\ \mathsf{wt}(z)\ .
\end{aligned}$$

It means that for any z, switching $f(z)$ from 0 to 1 increases $\mathsf{val}_H(f)$ by at most $2\mathsf{wt}(z)$. Therefore, $\mathsf{val}_H(f)$ is at most two times the weight of D. Thus $\mathsf{val}_H(f) \geq 2\varepsilon$ indicates that the weight of D is at least ε. We can now use Theorem 2.7 to finish the argument. □

Combined with the standard technique converting a dictatorship test to a hardness result based on the Unique Games Conjecture, the main theorem of this section is proved.

Theorem 2.9 (Restatement of Theorem 1.5)**.** *For any $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-2-SAT, it is UG-hard to distinguish the following cases.*

- $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

Proof. Given an instance of $\mathcal{L}(G(V, W, E), [R], \{\pi(v, w)\}_{(v,w)\in E})$ of Unique Games, we construct an instance $\mathcal{I} = (X, S, H)$ of Hard-Max-2-SAT.

- $X = V \times \{0,1\}^R$. The weight of (v, x) is the weight of x in $\{0,1\}^R_{(1/2-\varepsilon)}$, divided by $|V|$ (weights are only used in the analysis).
- For every pair of edges $(w, v_1), (w, v_2)$ with the same endpoint $w \in W$, and every $x, y \in \{0,1\}^R$, such that there is no i such that $x_{\pi(v_1,w)(i)} = y_{\pi(v_2,w)(i)} = 1$, add a hard constraint $(\neg(v_1, x) \vee \neg(v_2, y))$.
- For each v, sample an ordered pair (x, y) such that for each i, $\mathbb{P}[x_i = y_i = 0] = 2\varepsilon$ and $\mathbb{P}[x_i = 1, y_i = 0] = \mathbb{P}[x_i = 0, y_i = 1] = 1/2 - \varepsilon$. Add a soft constraint $((v, x) \vee (v, y))$ with the weight equal to the probability of (x, y) divided by $|V|$.

Note that the variables and hard constraints of this construction are identical to that of Min-Vertex-Cover (Max-Independent-Set) of [3]. We use the variant of the Unique Games Conjecture (see Conjecture 2.4), and prove the following:

Lemma 2.10. *Given an instance of Unique Games $\mathcal{L}$ and the instance $\mathcal{I}$ produced as above,*

- *If there is a set $V' \subseteq V$ such that $|V'| \geq (1-\varepsilon)|V|$ and a labeling $l : V \cup W \to [R]$ that satisfies every edge (v, w) for $v \in V'$ and $w \in W$, then $\mathsf{opt}_H(\mathcal{I}) \geq 1 - 3\varepsilon$.*
- *There is a function $f : \mathbb{R}^+ \to \mathbb{R}^+$ such that if $\mathsf{opt}(\mathcal{L}) \leq f(\varepsilon)$, then $\mathsf{opt}_H(\mathcal{I}) \leq 2\varepsilon$.*

Proof. We establish the completeness and soundness in turn.

Completeness: Let $V' \subseteq V$ and $l : V \cup W \to [R]$ be a subset and a labeling satisfying the above condition. For any $v \in V', x \in \{0,1\}$, set $(v, x) = x_{l(v)}$. If $v \notin V'$, $(v, x) = 0$ for all x. Note that this satisfies all the hard constraints; if there is a violated hard constraint $(\neg(v_1, x) \vee \neg(v_2, y))$, it means that $v_1, v_2 \in V'$, $x_{l(v_1)} = y_{l(v_2)} = 1$, but there exists $w \in W$ such that $\pi(v_1, w)(l(w)) = l(v_1), \pi(v_2, w)(l(w)) = l(v_2)$. It contradicts the above construction. For each v, the only soft constraints $((v, x) \vee (v, y))$ that are not satisfied by this assignment have $x_{l(v)} = y_{l(v)} = 0$, which happens with probability 2ε. Therefore, the total weight of soft constraints satisfied is at least $(1-\varepsilon)(1-2\varepsilon) \geq 1 - 3\varepsilon$.

Soundness: Suppose there is an assignment $\sigma : V \times \{0,1\}^R \to \{0,1\}$ such that it satisfies all the hard constraints and $\mathsf{val}_H(\sigma) \geq 2\varepsilon$. Let $D \subseteq V \times \{0,1\}^R$ be the support of σ. Since σ satisfies all the hard constraints, D is an independent set of the graph whose vertex set is $V \times \{0,1\}^R$ and each pair of variables in a same hard constraint forms an edge. Note that this is the same graph used in the hardness of Min-Vertex-Cover in [3].

Since the soft constraints are defined within each hypercube $v \times \{0,1\}$ for each v, we can use the same analysis from Theorem 2.8 (which says that the sum of weights of the soft constraints containing a variable in the Max-Hard-2-SAT instance is exactly two times the weight of the corresponding vertex in the Max-Independent-Set instance) to conclude that the weight of D is at least ε. We can invoke Theorem 3.1 of [3] to argue that $\mathsf{opt}(\mathcal{L}) \geq f(\varepsilon)$ for some fixed function $f : \mathbb{R}^+ \to \mathbb{R}^+$. □

Therefore, if we can distinguish whether $\mathsf{opt}_H(\mathcal{L}) \geq 1 - 3\varepsilon$ or $\mathsf{opt}_H(\mathcal{L}) \leq 2\varepsilon$ for some $\varepsilon > 0$, then we can refute Conjecture 2.4, which is equivalent to the original Unique Games Conjecture. This proves Theorem 1.5. □

2.3 Max-Hard-Horn-SAT

Suppose there is a very strong robust algorithm for Max-CSP(Π) in the sense that if the algorithm can find σ such that $\mathsf{val}_H(\sigma) \geq 1 - c\varepsilon$ whenever $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon$ for some absolute constant c (or in other words, we have a constant-factor approximation algorithm for the minimization version that seeks to minimize the weight of unsatisfied constraints). The performance of this algorithm is preserved for Hard-Max-CSP(Π) by the following trick: convert all hard constraints to soft constraints, but give them very large weight so that they will never be violated in the optimal solution. This will result in $\mathcal{I}'$ such that $\mathsf{opt}(\mathcal{I}') \geq 1 - \varepsilon'$ where $\varepsilon' \ll \varepsilon$, but the algorithm still finds σ such that $\mathsf{val}_H(\sigma) \geq 1 - c\varepsilon'$ as a solution to $\mathcal{I}'$, which also satisfies $\mathsf{val}_H(\sigma) \geq 1 - c\varepsilon$ as a solution to $\mathcal{I}$.

Now, Max-Horn-2-SAT is a problem that admits such a robust algorithm with $c = 2$ [18] ($c = 3$ was earlier shown in [21]), and therefore we can conclude that Hard-Horn-2-SAT admits a similar robust algorithm.

Furthermore, the reduction from Max-Independent-Set to Hard-Max-2-SAT introduced in Section 2.3 is an indeed a reduction to Hard-Max-Horn-2-SAT. Therefore, by previous results about Max-Independent-Set [24, 5], for any $\varepsilon > 0$, it is UG-hard to find σ with $\mathsf{val}_H(\sigma) \geq \varepsilon$ even when $\mathsf{opt}_H(\mathcal{I}) \geq 1/2 - \varepsilon$. By adding many dummy constraints that are always satisfiable, for any $\varepsilon, \delta > 0$, it is NP-hard to find σ with $\mathsf{val}_H(\sigma) \geq 1 - 2\varepsilon + \delta$ even when $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon - \delta$. These facts justfy the following observation made about Hard-Max-Horn-2-SAT.

Observation 2.11 (Restatement of Observation 1.6)**.** *There is an algorithm that for any $\varepsilon \geq 0$, if the given instance $\mathcal{I}$ of Max-Horn-2-SAT satisfies $\mathsf{opt}_H(\mathcal{I}) = 1 - \varepsilon$, finds σ with $\mathsf{val}_H(\sigma) \geq 1 - 2\varepsilon$.*

Furthermore, for any $\varepsilon, \delta > 0$, it is UG-hard to distinguish the following cases.

- $\mathsf{opt}(\mathcal{I}) \geq 1 - \varepsilon - \delta$.
- $\mathsf{opt}(\mathcal{I}) \leq 1 - 2\varepsilon + \delta$.

The above algorithmic result for Max-Horn-2-SAT does not hold for Max-Horn-SAT in general since the robust algorithm is only guaranteed to find an assignment satisfying $1 - O\left(\frac{\log\log(1/\varepsilon)}{\log(1/\varepsilon)}\right)$ fraction of clauses [32], and this exponential loss is inherent under the Unique Games conjecture [18]. In fact, even Horn-3-SAT is powerful enough to encode constraints of other hard problems with unbounded arity, which results in a very strong hardness result, stated below.[2]

Theorem 2.12 (Restatement of Theorem 1.7)**.** *For any $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-Horn-3-SAT, it is NP-hard to distinguish the following cases.*

- $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

Proof. We reduce Max-Ek-Hypergraph-Independent-Set to Max-Horn-3-SAT. An instance of Max-Ek-Hypergraph-Independent-Set is a hypergraph $G = (V, E)$ where each hyperedge $e \in E$ contains exactly k vertices. Our goal is to find a set $D \subseteq V$ with the maximum weight such that no hyperedge e is a subset of D.

Given a graph $G = (V, E)$, we construct the instance $\mathcal{I} = (X, S, H)$ of Max-Horn-3-SAT as follows.

- $X = V \cup \{y_{e,j} : e \in E, 1 \leq j \leq k\}$; there is one variable for each vertex, and k variables for each hyperedge. Each variable corresponding to a vertex indicates that the vertex is picked or not.
- For each hyperedge $e = (v_1, ..., v_k) \in E$, add hard constraints
  - $(v_1 \to \neg y_{e,1}) \equiv (\neg v_1 \vee \neg y_{e,1})$
  - $2 \leq j \leq k$: $(\neg y_{e,i-1} \wedge v_i \to \neg y_{e,i}) \equiv (y_{e,i-1} \vee \neg v_i \vee \neg y_{e,i})$
  - $y_{e,k}$
- For each vertex $v \in V$, add a soft constraint v with the same weight as in G.

[2] We learned from Andrei Krokhin that this result, with essentially the same proof, was also shown by Siavosh Bennabas. But as we are not aware of a published reference, we include the simple proof.

The above construction ensures that there is at most one unnegated literal per each clause, so this is indeed an instance of Max-Horn-3-SAT. Once $v_1, ..., v_n$ are fixed, a quick check of the second set of constraints corresponding to hyperedge e ensures that there exists $y_{e,1}, ..., y_{e,k}$ that satisfy all the constraints if and only if at least one of $v_1, ..., v_k$ is set to False. Since the weight of satisfied soft constraints is equal to the weight of vertices picked, this is an approximation-preserving reduction from Max-Ek-Hypergraph-Independent-Set to Max-Horn-3-SAT. Dinur et al [10] showed that for the former, it is NP-hard to distinguish

- There is an independent set of weight $(1 - \frac{1}{k-1} - \varepsilon)$.
- Every independent set is of weight at most ε.

By taking k large and ε small, we get the desired result for Max-Horn-3-SAT. □

3 Balance Constraints

3.1 Hardness Results

Theorem 3.1 (Restatement of Theorem 1.3)**.** *There exists an absolute constant $\delta > 0$ such that given an instance $\mathcal{I}$ of Balance-Max-Horn-2-SAT (special case of Balance-Max-2-SAT and Balance-Max-Horn-SAT), it is NP-hard to distinguish the following cases.*

- $\mathsf{opt}_B(\mathcal{I}) = 1$
- $\mathsf{opt}_B(\mathcal{I}) \leq 1 - \delta$

Proof. We reduce from Max-3-SAT(B) to Max-Independent-Set to Balance-Max-Horn-2-SAT, where in Max-3-SAT(B) each variable occurs at most B times. The following description of the reduction from Max-3-SAT(B) to Max-Independent Set is from Papadimitriou and Yannakakis [27].

> *Construct a graph with one node for every occurrence of every literal. There is an edge connecting any two occurrences of complementary literals, and also, an edge connecting literal occurrences from the same clause (thus, there is a triangle for every clause with 3 literals, and an edge for a clause with 2 literals). The size of the maximum independent set in the graph is equal to the maximum number of clauses that can be satisfied. If every variable occurs at most B times in the clauses, then the degree is at most $B + 1$.*

Note that in the above reduction, there is an independent set of size l if and only if there is an assignment that satisfies l clauses. For some constants B and $\delta_0 > 0$, it is NP-hard to find an assignment satisfying $(1 - \delta_0)$ fraction of clauses in a satisfiable instance of Max-3-SAT(B). Let m be the number of clauses, and n be the number of variables of the given Max-3-SAT(B) instance, so we have $3m$ vertices, and at most $1.5(B+1)m \leq 2Bm$ edges in the graph. Our Balance-Max-Horn-2-SAT instance $\mathcal{I}$ consists of $4m$ variables; $3m$ of them correspond to the vertices of the graph, and m of them do not participate in any constraint. For each edge (u, v) we add the constraint $(\neg u \vee \neg v)$. Finally, we have the balance constraint that exactly $2m$ of them should be 1 (1 means True in 2-SAT, and that the vertex is picked in the Independent Set problem).

If the Max-3-SAT instance is satisfiable, we have an independent set of size $2m$ consisting of m vertices from the graph and m of the dummy vertices, so the Balance-Max-Horn-2-SAT instance is also satisfiable.

Now suppose that $\mathsf{opt}_B(\mathcal{I}) > 1 - \delta$. This means that there is a balanced assignment (at least m 1's amongst the non-dummy variables) such that at most a fraction δ of the edges have both endpoints set to 1. By switching $2\delta Bm$ vertices from 1 to 0, all clauses of Balance-Max-Horn-2-SAT will be satisfied. This means that there is an independent set of size at least $(1 - 2\delta B)m$, and therefore also an assignment that satisfies $(1 - 2\delta B)m$ clauses of the Max-3-SAT(B) instance. It follows that we must have $\delta \geq \frac{\delta_0}{2B}$. □

Unlike the reduction from Max-Independent-Set to Hard-Max-Horn-2-SAT introduced in Section 2.2 (hard clauses for independence constraints, and soft clauses for the objective), we use clauses to enforce independence constraints, and the balance constraint to maximize the objective. For the soundness analysis, given a good assignment to Balance-Max-Horn-2-SAT, when viewed as a (slightly infeasible) solution to Max-Independent-Set, the balance constraint ensures that the objective is good, but there might be some adjacent vertices picked. The bounded degrees allow us to fix this solution to an independent set while still retaining many vertices.

3.2 Algorithmic Results

One of the most well-studied Balance-Max-CSP is Balance-Max-Cut (Max-Bisection). Over a long line of work [13, 19, 31, 12, 28, 2], the approximation ratio for Max-Bisection has become 0.8776 which is very close to the optimal (under the Unique Games Conjecture) approximation ratio for Max-Cut which is about 0.8786. Max-Bisection admits a robust algorithm as well [16, 28]. Note that Balance-Max-CSP(Π) is no easier to approximate than Max-CSP(Π), since any instance of Max-CSP(Π) can be reduced to an instance of Balance-Max-CSP(Π) by adding dummy variables that do not participate in any constraint.

For Balance-Max-2-SAT, the best known approximation ratio is 0.9401, matching the best known approximation ratio of Max-2-SAT [25]. This result indicates that in the approximation ratio perspective, adding the balance constraint does not make the problem harder. However, Theorem 3.1 rules out any robust algorithm for Balance-Max-2-SAT, which shows a stark difference between the balanced versions of Max-2-SAT and Max-Cut.

For Balance-Max-Horn-SAT, the same hardness result shows that we cannot hope for any robust algorithm. Therefore, the only remaining quetion is whether there is an algorithm whose approximation ratio nearly matches that of the best algorithm for ordinary Max-Horn-SAT. The best approximation ratio for Max-Horn-SAT is achieved by an algorithm for more general Max-SAT, which achieves 0.7968-approximation [4].[3]

The most recent results on Balance-Max-Cut and Balance-Max-2-SAT [28, 2] rely on Lasserre SDP hierarchies. The purpose of using a more sophisticated SDP rather than the basic SDP of Goemans and Williamson [15], is that during the rounding scheme, the vertices are not rounded independently; even if we add the balance constraint in the SDP relaxation, the final solution is not guaranteed to be approximately balanced when we do not have a guarantee about their correlations. The goal of Lasserre hierarchies is to produce an SDP solution with low global correlation so that each vertex can be rounded almost independently.

[3] The same paper gives another algorithm whose approximation ratio is 0.8434 under some conjecture.

However, in the 3/4-approximation algorithm for Max-SAT by Goemans and Williamson [14], which is based on a LP-relaxation, each variable is rounded independently. Therefore, adding the balance constraint ensures that the final solution is almost balanced by a simple application of Chernoff bound. After a simple correction phase to get perfect balance (slightly more sophisticated than [28] since the arity is not bounded), we obtain an algorithm which is not far from the best algorithm for Max-SAT in terms of approximation ratio.

Theorem 3.2 (Restatement of Theorem 1.4)**.** *For any $\varepsilon > 0$, there is a randomized algorithm such that given an instace $\mathcal{I}$ of Balance-Max-SAT, in time poly($size(\mathcal{I}), \frac{1}{\varepsilon}$), outputs σ with $\mathsf{val}_B(\sigma) \geq (\frac{3}{4} - \varepsilon)\mathsf{opt}_B(\mathcal{I})$ with constant probability.*

Proof. Let $\mathcal{I} = (X, C)$ be an instance Balance-Max-SAT, where $X = \{x_1, ..., x_n\}$ is the set of variables and $C = \{C_1, ..., C_m\}$ is the set of clauses. For each $1 \leq j \leq m$, let C_j^+ (resp. C_j^-) be the set of variables which appear in C_j *unnegated* (resp. *negated*). The following is a natural LP relaxation of Balance-Max-SAT.

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{m} \mathsf{wt}(C_j) z_j \\
\text{subject to} \quad & \forall 1 \leq j \leq m : \sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1 - y_i) \geq z_j \\
& \forall 1 \leq j \leq m : 0 \leq z_j \leq 1 \\
& \forall 1 \leq i \leq n : 0 \leq y_i \leq 1 \\
& \sum_{i=1}^{n} y_i = \frac{n}{2}
\end{aligned}$$

Given the solution (y_i, z_j) to the above LP, the rounding algorithm is simple:

1. Choose $a \in \{0, 1\}$ uniformly at random.
2. If $a = 0$, set $\sigma(x_i) = 1$ with probability y_i independently; if $a = 1$, set $\sigma(x_i) = 1$ with probability 0.5 independently.
3. If the final solution is unbalanced (there exists $b \in \{0, 1\}$ such that $|\sigma^{-1}(b)| = (0.5 + \eta)n$ for some $\eta > 0$), pick exactly ηn variables from $\sigma^{-1}(b)$, set them to $1 - b$.
4. Iterate it $O(1/\varepsilon)$ times.

Claim 3.3. *At the end of step 2, $\mathbb{E}[\mathsf{val}(\sigma)] \geq 3/4$ (note that it is $\mathsf{val}(\sigma)$, but not $\mathsf{val}_B(\sigma)$).*

Proof. For each C_j, let k be the number of variables in C_j and without loss of generality assume $C_j = x_1 \vee ... \vee x_k$ by renaming and negating variables. If $a = 0$, the probability that C_j is satisfied is

$$1 - \Pi_{i=1}^{k}(1 - y_i) \geq 1 - (\frac{\sum_{i=1}^{k}(1 - y_i)}{k})^k \geq 1 - (1 - \frac{z_i}{k})^k \geq \beta_k z_i$$

where $\beta_k = 1 - (1 - \frac{1}{k})^k$. If $a = 1$, the probability that C_j is satisfied is at least $\alpha_k = 1 - 2^{-k}$. Overall, the probability that C_j is satisfied is $\frac{\alpha_k + \beta_k}{2}$. For $k = 1, 2$, $\frac{\alpha_k + \beta_k}{2} = \frac{3}{4}$, and for $k \geq 3$, $\frac{\alpha_k + \beta_k}{2} \geq \frac{7/8 + (1 - 1/e)}{2} \geq \frac{3}{4}$. □

By a simple averaging argument, with probability at least ε, $\mathsf{val}(\sigma) \geq 3/4 - \varepsilon$. Since each variable is rounded independently once a is fixed, by Chernoff bound, $\mathbb{P}[|\sigma^{-1}(1) - \frac{n}{2}| > \varepsilon n] \leq 2exp(-\frac{2\varepsilon^2}{n}) = o_n(1)$. Therefore, at the end of step 2, the probability that $\mathsf{val}(\sigma) \geq 3/4 - \varepsilon$ and $|\sigma^{-1}(1) - \frac{n}{2}| \leq \varepsilon n$ is at least $\varepsilon/2$.

Let b be such that $|\sigma^{-1}(b)| = (1/2 + \eta)n$ for some $0 < \eta \leq \varepsilon$. In step 3, we randomly choose ηn variables from $\sigma^{-1}(b)$, and set them to $1 - b$. Fix a constraint C_j that is satisfied by σ. Since C_j is a disjunction, there is only one assignment to the variables appearing C_j that makes C_j unsatisfied. Since C_j is already satisfied, to reach the only unsatisfying assignment, at least one variable must be switched. If there is a variable that must be switched from $1 - b$ to b, then C_j is guaranteed to be satisfied even after step 3. Otherwise, there are k variables ($k \geq 1$) that must be swithced from b to $1 - b$, and the probability of switching all of them is at most the probability of switching one of them, which is $\frac{\eta n}{(1/2+\eta)n} \leq 2\eta$. Therefore, in expectation, at most 2η fraction of already satisfied clauses can be unsatisfied, and with probability half, at most $4\eta \leq 4\varepsilon$ fraction of satisfied clauses can be unsatisfied. Combining step 2 and 3, in each iteration, we get a balanced σ with $\mathsf{val}_B(\sigma) \leq 3/4 - 5\varepsilon$ with probability $\varepsilon/4$. This probability can be made to a constant by repeating $O(1/\varepsilon)$ times. □

4 Ordering with Hard Constraints

Another natural class of 2-ary CSPs over non-Boolean domain is Max-CSP(<). Max-CSP(<) over the unbounded domain (indeed $[n]$ is enough) is the famous Max-Acyclic-Subgraph, which admits a simple 2-approximation algorithm that is optimal under the Unique Games Conjecture [17]. For fixed q, Max-CSP(<) over $[q]$ also admits a simple $\frac{(q-1)}{2q}$ algorithm by taking a random assignment.

These ordering problems can be best understood in terms of choosing the maximum subgraph of a directed graph with certain desired properties. Max-CSP(<) over $[q]$ is equivalent to the following problem: given a directed graph $G = (V, E)$, find $G' = (V, E')$ with $E' \subseteq E$ such that G' does not have a walk of length q and $|E'|$ is maximized. A walk of length q is a sequence of vertices $v_0 \to \cdots \to v_q$ (vertices can be repeated) such that $(v_i, v_{i+1}) \in E$ for $0 \leq i < q$. Note that a directed cycle results in a walk of any length. Max-Acyclic-Subgraph is to find G' such that G' does not have any cycle.

Instead of choosing a subset of the edges, we can try to find a subset of the vertices so that the induced subgraph has desired properties. Let Max-Vertex-CSP(<) and Max-Vertex-Acyclic-Subgraph be the analogous problems where the goal is to choose the subset of variables (vertices) with the maximum weight so that the induced CSP is satisfiable (induced graph has the desired properties).

Despite of their similarity, Max-Vertex problems seem to be harder than their edge counterparts. There is no intuitive constant-factor approximation algorithm, like taking a random assignment in the edge problems. Svensson [30] confirmed that these problems are indeed a lot harder to approximate, assuming the Unique Games Conjecture.

Theorem 4.1 (Theorem 1 of [30]). *For any $k \geq 2$ and $\varepsilon > 0$, given an instance $\mathcal{I}$ of Max-Vertex-CSP(<) over the domain $[k-1]$, it is UG-hard to distinguish*

- $\mathsf{opt}(\mathcal{I}) \geq (1 - \varepsilon)\frac{k-1}{k}$
- $\mathsf{opt}(\mathcal{I}) \leq \varepsilon$

Also, for any $\varepsilon > 0$, given an instance $\mathcal{I}$ of Max-Acyclic-Subgraph, it is UG-hard to distinguish

- $\mathsf{opt}(\mathcal{I}) \geq 1 - \varepsilon$
- $\mathsf{opt}(\mathcal{I}) \leq \varepsilon$

We show that introducing hard constraints makes the edge versions almost as hard as the vertex counterparts. The reduction, introduced in Even et al. [11], is simple: *split* each vertex v to a soft edge $v_{in} \to v_{out}$ and add a hard edge $u_{out} \to v_{in}$ for each edge $u \to v$. This is almost enough to show the following theorem.

Theorem 4.2 (Restatement of Theorem 1.8)**.** *For any $k \geq 2$ and $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-CSP(<) over $[2k+1]$, it is UG–hard to distinguish the following cases.*

- $\mathsf{opt}_H(\mathcal{I}) \geq (1 - \varepsilon)\frac{k-1}{k}$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

Over the domain $\{1, 2, 3, 4\}$, it is also UG-hard to distinguish whether

$$\mathsf{opt}_H(\mathcal{I}) \geq \frac{1-\varepsilon}{2} \qquad \text{or} \qquad \mathsf{opt}_H(\mathcal{I}) \leq \varepsilon \ .$$

Also, for any $\varepsilon > 0$, given an instance $\mathcal{I}$ of Hard-Max-CSP(<) over the unbounded domain (an instance of Max-Acyclic-Subgraph with several edges that must be included), it is UG-hard to distinguish the following cases.

- $\mathsf{opt}_H(\mathcal{I}) \geq 1 - \varepsilon$
- $\mathsf{opt}_H(\mathcal{I}) \leq \varepsilon$

Proof. The last result about the unbounded domain (Max-Acyclic-Subgraph with hard edges) is immediate from the reduction above and the hardness of Max-Vertex-Acyclic-Subgraph.

We will now describe the first reduction from the instance $\mathcal{I}$ of Max-Vertex-CSP(<) over $[k-1]$ to the instance $\mathcal{I}'$ of Hard-Max-CSP(<) over $[2k+1]$ (in this proof, we identify each instance of the CSP with its underlying directed graph). We perform the same reduction as above, and add two special vertices s and t. Add a hard edge $s \to v_{in}$ if v_{in} has no incoming edge. Make a hard edge $v_{out} \to t$ if v_{out} has no outgoing edge.

There is a one-to-one correspondence between the vertices of $\mathcal{I}$ and the soft constraints of $\mathcal{I}'$. Let G denote a vertex-induced subgraph of $\mathcal{I}$ and G' be its corresponding edge subgraph of $\mathcal{I}'$. If there is a walk $(v_1, \dots, v_k)$ of length $k-1$ exists in G, let v_0 be any vertex which has a outgoing edge to v_1 ($v_{0,out} := s$ if v_1 has no incoming edge), and let v_{k+1} be any vertex which has a incoming edge from v_k ($v_{k+1,in} := t$ if v_k has no outgoing edge). Then $v_{0,out} \to v_{1,in} \to v_{1,out} \cdots \to v_{k,out} \to v_{k+1,in}$ is a walk of length $2k+1$ in G'. Similarly, since any walk of G' alternates soft edges and hard edges, a walk of length $2k+1$ in G' induces a walk of length $k-1$ in G. Therefore, there is a one-to-one correspondence between feasible solutions, and corresponding solutions have the same weight.

Finally, we show the hardness of Hard-Max-CSP(<) over $\{1, 2, 3, 4\}$ by reducing Max-Independent-Set to it. Given an instance (undirected graph) $\mathcal{I}$, consider it as a directed graph with edges going both directions, and apply the above reduction to get $\mathcal{I}'$. There is again one-to-one correspondence between a vertex-induced subgraph $G \subseteq \mathcal{I}$ and a edge subgraph $G' \subseteq \mathcal{I}'$. If $(u, v) \in G$,

$u_{in} \to u_{out} \to v_{in} \to v_{out} \to u_{in}$ forms a cycle of G', so it is not feasible. If there is a walk of length at least 4 in G', since it alternates soft and hard constraints, there are two neighboring vertices u and v so that $u_{in} \to u_{out}, v_{in} \to v_{out} \in G'$, so G is not an independent set. □

It is easy to get a constant-factor approximation for Hard-Max-CSP($<$) over the domain $\{1, 2\}$. Each hard constraint uniquely determine the value of the variables participating, and randomly assigning remaing variables ensure that each soft constraint, if satisfiable at all, can be satisfied by with some constant probability. A slight extension of this technique works for $\{1, 2, 3\}$ as well, in stark constrast to $\{1, 2, 3, 4\}$.

Theorem 4.3. *There is a constant-factor approximation algorithm for Hard-Max-CSP($<$) over* $\{1, 2, 3\}$.

Proof. Let $\mathcal{I}$ be an instance of Hard-Max-CSP($<$) over $\{1, 2, 3\}$. We also think $\mathcal{I}$ as a directed graph (V, E) such that for each constraint $u > v$ there is an edge $u \to v$. By virtue of certain combinations of hard constraints, namely paths of length 2, the value of some vertices becomes fixed; let F_1, F_2, F_3 be the set of vertices fixed to $1, 2, 3$ respectively. In the subgraph induced by $V \setminus (F_1 \cup F_2 \cup F_3)$, let B be the set of vertices v such that $v \to u$ is a hard edge for some u. Similarly, S be the set of vertices u such that $v \to u$ for some v. Let R be the remaining vertices. Note there is no edge from $F_1 \cup F_2$ to $B \cup S \cup R$, and no edge from $B \cup S \cup R$ to $F_2 \cup F_3$, and from S to B (e.g. if there is an edge from S to B, we have a path of length 2 and these vertices should be fixed). The algorithm is the following.

1. Choose $a \in \{0, 1\}$ uniformly at random.
2. If $a = 0$, for each vertex $v \in B$, $v = 3$; for each vertex $v \in S$, choose $v \in \{1, 2\}$ uniformly and independently.
3. If $a = 1$, for each vertex $v \in B$, choose $v \in \{2, 3\}$ uniformly and independently; for each vertex $v \in S$, choose $v = 1$.
4. Regardless of a, for each vertex $v \in R$, choose $v \in \{1, 2, 3\}$ uniformly and independently.

Simple case analyses between possible edges between the sets yield the following:

- Each edge completely inside one of B, S, R is satisfied with probability at least $1/8$.
- Each edge whose one endpoint is in R and the other in one of B, S is satisfied by with probability at least $1/12$.
- Each edge to F_1 or from F_3 is satisfied with probability at least $1/4$.

Therefore, the above simple algorithm yields a $1/12$-approximation algorithm. □

For fixed q, Max-CSP($<$) over $[q]$ admits a simple robust algorithm that finds a solution of value $(1 - q\varepsilon)$, given $(1 - \varepsilon)$-satisfiable instance: given a directed graph $G = (V, E)$, set one nonnegative variable y_e for each e, and for any walk of length q, add a constraint that the sum of the corresponding y's must be at least 1. Deleting e when $y_e \geq \frac{1}{q}$ gives a q-approximation algorithm for the minimization version. As this is a strong robust algorithm (as discussed in Section 2.3), it is preserved for Hard-Max-CSP($<$) over $[q]$.

Observation 4.4. *For fixed q, Hard-Max-CSP($<$), and consequently Max-Vertex-CSP($<$) over $[q]$ admits a robust algorithm.*

That gives the complete picture of the status these problems. The conclusion is that adding hard constraints makes it as hard as its vertex counterparts, which is much harder to approximate.

Acknowledgments

We thank Andrei Krokhin for raising the question of whether Hard-2-SAT admits a robust satisfiability algorithm, which was the impetus for this work, and also for several useful comments on the write-up. We also thank Per Austrin for useful discussions.

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 thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 573–581, New York, NY, USA, 2005. ACM.

[2] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: a 0.8776-approximation for Max Bisection. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 373–387, 2013.

[3] Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, pages 74–80, 2009.

[4] Adi Avidor, Ido Berkovitch, and Uri Zwick. Improved approximation algorithms for max naesat and max sat. In Thomas Erlebach and Giuseppe Persinao, editors, *Approximation and Online Algorithms*, volume 3879 of *Lecture Notes in Computer Science*, pages 27–40. Springer Berlin Heidelberg, 2006.

[5] Nikhil Bansal and Subhash Khot. Inapproximability of hypergraph vertex cover and applications to scheduling problems. In *Proceedings of the 37th international colloquium conference on Automata, languages and programming*, ICALP'10, pages 250–261, Berlin, Heidelberg, 2010. Springer-Verlag.

[6] Andrei A. Bulatov and Dániel Marx. Constraint satisfaction problems and global cardinality constraints. *Communications of the ACM*, 53(9):99–106, 2010.

[7] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 62–68, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[8] Nadia Creignou, Henning Schnoor, and Ilka Schnoor. Nonuniform boolean constraint satisfaction problems with cardinality constraint. *ACM Trans. Comput. Log.*, 11(4), 2010.

[9] Victor Dalmau and Andrei Krokhin. Robust satisfiability for CSPs: hardness and algorithmic results. *ACM Transactions on Computation Theory*, 2013. To appear.

[10] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM J. Comput.*, 34(5):1129–1146, 2005.

[11] Guy Even, Joseph (Seffi) Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

[12] Uriel Feige and Michael Langberg. The RPR2 rounding technique for semidefinite programs. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming,*, ICALP '01, pages 213–224, London, UK, UK, 2001. Springer-Verlag.

[13] Alan M. Frieze and Mark Jerrum. Improved approximation algorithms for Max k-Cut and Max Bisection. In *Proceedings of the 4th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 1–13, London, UK, UK, 1995. Springer-Verlag.

[14] Michel Goemans and David Williamson. New $\frac{3}{4}$-approximation algorithms for the maximum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7(4):656–666, 1994.

[15] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.

[16] Venkatesan Guruswami, Yury Makarychev, Prasad Raghavendra, David Steurer, and Yuan Zhou. Finding almost-perfect graph bisections. In *Proceedings of The Second Symposium on Innovations in Computer Science*, Innovations in Computer Science '11, pages 321–337, 2011.

[17] Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard: inapproximability of maximum acyclic subgraph. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 573–582, 2008.

[18] Venkatesan Guruswami and Yuan Zhou. Tight bounds on the approximability of almost-satisfiable horn sat and exact hitting set. *Theory of Computing*, 8(1):239–267, 2012.

[19] Eran Halperin and Uri Zwick. A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems. In *Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 210–225, London, UK, UK, 2001. Springer-Verlag.

[20] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.

[21] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000.

[22] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, STOC '02, pages 767–775, New York, NY, USA, 2002. ACM.

[23] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007.

[24] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[25] Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the Max 2-SAT and Max Di-Cut problems. In *Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 67–82, London, UK, UK, 2002. Springer-Verlag.

[26] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 21–30, 2005.

[27] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425 – 440, 1991.

[28] Prasad Raghavendra and Ning Tan. Approximating CSPs with global cardinality constraints using SDP hierarchies. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 373–387. SIAM, 2012.

[29] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. ACM.

[30] Ola Svensson. Hardness of vertex deletion and project scheduling. In Anupam Gupta, Klaus Jansen, Jos Rolim, and Rocco Servedio, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 7408 of *Lecture Notes in Computer Science*, pages 301–312. Springer Berlin Heidelberg, 2012.

[31] Yinyu Ye. A .699-approximation algorithm for Max-Bisection. *Mathematical Programming*, 90(1):101–111, 2001.

[32] Uri Zwick. Finding almost-satisfying assignments. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 551–560, New York, NY, USA, 1998. ACM.