THE UNIVERSITY OF CHICAGO

NEW RESULTS ON THE APPROXIMABILITY OF SOME CLASSICAL CONSTRAINT SATISFACTION PROBLEMS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
NENG HUANG

CHICAGO, ILLINOIS
AUGUST 2024

To my parents

"Ars longa, vita brevis"

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude for my amazing advisor Aaron Potechin for his advice and support during the past few years. Aaron's optimism, enthusiasm, and kindness will continue to be an inspiration for me.

I would like to thank my collaborators Joshua Brakensiek and Uri Zwick on these MAX CSP projects. I learned a great deal from both of them during these years of collaboration. I feel extremely fortunate to have them, as well as Aaron, as collaborators on these projects.

I would like to thank Madhur Tulsiani and Janos Simon for being on my committee. I would also like to thank Andy Drucker for being on my master's committee.

I would like to thank Aaron Potechin and Kunal Marwaha for reading the early drafts of this thesis and giving many helpful comments.

Now for projects not directly related to this thesis: I am grateful to Antares Chen and Kunal Marwaha for the good times that we spent working on the random CSP project together. I would like to thank the IDEAL institute for organizing the summer research exchange in 2023 and funding me during that summer. I would like to thank Will Perkins for his guidance during and after the IDEAL summer research exchange. I would like to thank Bill Fefferman for his guidance on quantum-related readings.

Thanks to all my good friends and colleagues in the UChicago CS / TTIC community. Because of them, this PhD journey is more enjoyable and memorable.

Finally, I want to thank my family and especially my parents. Their love and support made all this possible.

ABSTRACT

In a *constraint satisfaction problem* (CSP), we are given a set of variables and a set of constraints over these variables and the goal is to assign the variables so that as many constraints as possible are satisfied. Many problems in computer science can be phrased in this language so it is of great interest to either design efficient algorithms for this task or prove that such algorithms don't exist.

In a breakthrough result, Raghavendra proved that for all CSPs there is a canonical semi-definite programming (SDP) relaxation that is optimal under the famous Unique Games Conjecture (UGC). More specifically, any integrality gap instances for this SDP relaxation can be turned into hardness results assuming UGC (or at least that unique games is hard) and there is a polynomial time rounding algorithm that achieves the integrality gap curve. However, this result does not tell us how to explicitly construct integrality gap instances for this SDP or what the optimal approximation ratio is for a given CSP. Moreover, the rounding algorithm has a doubly exponential dependency on the error parameter, which makes it practically infeasible.

Based on Raghavendra's framework, we study the approximability for some classical CSPs, including MAX DI-CUT, MAX 2-SAT and its subproblems, and MAX NAE-SAT. In particular, assuming UGC (or at least that unique games is hard), we show that MAX DI-CUT is strictly harder to approximate than MAX CUT and MAX NAE-SAT does not have a 7/8-approximation algorithm. To do so, we construct explicit integrality gap instances for the canonical SDP relaxation for these problems. For MAX 2-SAT and its subproblems, we give tight approximability results (modulo UGC) by presenting matching approximation algorithms and unique games hardness results.

CHAPTER 1

INTRODUCTION

We often encounter the following task, both in computer science and in real-world scenarios, where we are given a set of variables and a list of constraints on these variables and we need to find an assignment to these variables that satisfy all the constraints. This is known as a constraint satisfaction problem (CSP). For example, the following problem shows up in nearly all intro CS theory classes.

Problem 1 (3-SAT)**.** Given a set of Boolean variables and a set of constraints of the form $x \vee y \vee z$, find an assignment that satisfies *all* given constraints.

In the same classes where it would show up, we would also be taught that this problem is NP-complete and there is no hope to solve it in polynomial time, unless the widely believed P $\neq$ NP conjecture should turn out to be false. This is in fact true for most CSPs that we are interested in. We then have to ask for the next best thing: satisfying as many given constraints as possible. This turns the 3-SAT problem into the following MAX 3-SAT problem.

Problem 2 (MAX 3-SAT)**.** Given a set of Boolean variables and a set of constraints of the form $x \vee y \vee z$, find an assignment that satisfies as many given constraints as possible.

MAX 3-SAT is an example of the *maximum constraint satisfaction problems* (MAX CSP). Many more natural combinatorial optimization problems fall into this category. We mention a few more examples below.

Problem 3 (MAX 3-LIN(q))**.** Given a set of linear equations over $\mathbb{Z}_q$ with exactly 3 variables in each equation, find an assignment that that satisfies as many given equations as possible.

Problem 4 (MAX CUT and MAX DI-CUT)**.** MAX CUT is the problem where given a graph $G = (V, E)$, we are asked to find a partition of the vertices into two parts L and R

such that the number of edges in E with endpoints in different parts is maximized. MAX DI-CUT is the directed version of this problem where we are given a directed graph, and we need to maximize the number of edges in E whose first endpoint is in L and second endpoint in R.

One way to find an assignment for these problems is simply choosing one uniformly at random. In the case of MAX 3-SAT, a uniformly random assignment satisfies a 7/8-fraction of the given constraints in expectation. Of course, we are unlikely to get an optimal assignment this way, but we do obtain a mathematical *guarantee* that in expectation the number of satisfied constraint by the assignment is at least a 7/8-fraction of what the optimal solution satisfies. This type of algorithms are called *approximation algorithms*, and this guaranteed fraction is called the *approximation ratio* of the algorithm.

It is then natural to ask if we can design approximation algorithms with an approximation ratio better than the uniformly random assignment, and if so, what's the best ratio that we can achieve? To answer these questions, two tasks are involved:

1. Design good approximation algorithms with provable guarantees. The difficulty here is not only coming up with the algorithm but also proving that the algorithm works for *all* instances of the given problem.

2. Prove that no approximation algorithms exist beyond a certain approximation ratio (under plausible complexity-theoretic assumptions). The difficulty here is proving hardness against *all* polynomial-time algorithms.

The best case scenario is proving *tight* approximability results, namely, for some $\alpha > 0$, showing that there exists a polynomial-time approximation algorithm that achieves an approximation ratio of α, and that it is hard to achieve an approximation ratio of $\alpha + \epsilon$ for any $\epsilon > 0$.

In addition to understanding the approximability of one problem, we are also interested in comparing the approximability of two problems. This is usually the case when one problem is a subproblem of the other problem. In the context of MAX CSPs, we may have two problems A and B where problem A allows all constraint types in problem B as well as some other constraint types that are not allowed in problem B. In this case, any instance of problem B is automatically an instance of problem A, so any approximation algorithm for problem A is in turn an algorithm for problem B. It is then natural to ask the following: is problem A strictly harder to approximate than problem B due to the additional constraint types? Or do they in fact have the same approximation ratio? To answer these questions, we need to either present an optimal algorithm for problem B that works equally well for problem A, or prove hardness for problem A that separates it from problem B.

In this thesis, we will investigate a few classical MAX CSPs and obtain some tight approximability results, as well as separation results. In particular, we will present the following results:

- In Chapter 4, we study the approximability of MAX 2-SAT and its subproblems. For MAX 2-SAT, the best hardness result is due to Austrin [Aus07], and the best algorithmic result is the LLZ algorithm due to Lewin, Livnat and Zwick [LLZ02]. However, the algorithmic result is numerical and does not give a rigorous guarantee on the approximation ratio. It is conjectured that the approximation ratio achieved by the LLZ algorithm actually matches the hardness ratio proved by Austrin. We present a plan to prove this conjecture in Chapter 4, which was carried out in more details in [BHZ24]. We also give a full characterization of the subproblems of MAX 2-SAT in terms of their approximability. The characterization is obtained by giving matching hardness constructions and algorithms for these subproblems. The results in this chapter are based on [BHZ24].

- In Chapter 5, we study the approximability of the MAX DI-CUT problem. While the approximability of MAX CUT has been well understood, the understanding for its counterpart in directed graphs lagged woefully behind. Nothing much could be said about MAX DI-CUT, other than the fact that its approximation ratio is between MAX CUT and MAX 2-AND (which as we will see is a natural generalization of MAX DI-CUT), with the possibility of attaining equality on either end. We rule out this possibility by giving a new algorithm as well as a new hardness construction for MAX DI-CUT. The results in this chapter are based on [BHPZ23].

- In Chapter 6, we study the approximability of the MAX NAE-SAT problem, in which a constraint is satisfied if and only if its inputs are not all equal. We prove that assuming UGC, it is NP-hard to approximate MAX NAE-$\{3,5\}$-SAT (the restriction of MAX NAE-SAT to clauses of lengths 3 and 5) within a factor of 0.8739. This ratio improves upon the previous hardness bound of $7/8+\epsilon$ for MAX NAE-SAT, which it inherits from MAX NAE-4-SAT. Intuitively, for clauses of length 3, a good rounding algorithm should create correlation between variables, but correlation hurts the performance of the algorithm on clauses of length 5. We formalize this intuition by analyzing the *moment functions* of the rounding schemes for MAX NAE-SAT. We believe moment functions themselves are of independent interest. The results in this chapter are based on [BHPZ21].

To describe these results, we set up the formal definitions in Chapter 2. In Chapter 3, we will give an overview on the SDP-approach to approximating MAX CSPs. In particular, we will discuss *the Basic SDP*, due to Raghavendra [Rag08], which our results rely heavily on.

In the remainder of this chapter, we will discuss some historical background for approximating MAX CSPs in Section 1.1. We conclude this chapter by surveying some adjacent areas as well as future directions in Section 1.2.

1.1 Historical Background

The satisfiability problem (SAT), one of the prototypical CSPs, is also one of the first problems shown to be NP-complete, as well as its special case 3-SAT [Coo71]. These problems are in Karp's 21 NP-complete problems [Kar72], among which the decision version of MAX CUT is also included.

For MAX CSPs, it was observed early on that many of them have constant-factor approximation algorithms. The approximation ratios are not obtained directly from, but coincides with those obtained by a uniform assignment, for example 1/2 for MAX SAT [Joh74] and MAX CUT [SG76]. It was later observed that these ratios can be obtained deterministically by derandomizing the uniform assignment using conditional expectation (see e.g. [Spe94]). Many attempts were made to improve these ratios. For MAX SAT, Yannakakis [Yan94] gave the first 3/4-approximation algorithm. Later, Goemans and Williamson [GW94] presented a simpler algorithm that achieves the same ratio. The situation for MAX CUT was a bit more embarrassing: a series of works (see e.g. [Vit81, HV91]) achieved a ratio of $1/2 + o(1)$ where the $o(1)$ term depends on various quantities in the graph, but none of them was able to obtain a ratio that's a constant strictly larger than 1/2.

A major algorithmic breakthrough came in the mid-90s, when Goemans and Williamson used semi-definite programming (SDP) to give a 0.878-approximation algorithm for MAX CUT [GW95]. This drastically improved the previous ratio of $1/2 + o(1)$. In the same paper, they showed that SDP can also be used to improve the ratios for MAX DI-CUT and MAX 2-SAT. This quickly led to an explosion of SDP-based algorithmic results for MAX CSPs, including further improvements on MAX DI-CUT and MAX 2-SAT [LLZ02], MAX 3-SAT [KZ97], MAX k-AND [MM14], MAX SAT and MAX NAE-SAT [ABZ06]. The readers are referred to [MM17] to a more detailed survey of these algorithmic results. However, due to reasons that we will explain in the next section, many of these results are numerical and do not yield a theoretical guarantee on the approximation ratio.

In terms of inapproximability results, Papadimitriou and Yannakakis, in their pioneering work [PY91], defined the complexity class of MAX SNP and proved that MAX 3-SAT and several other natural combinatorial optimization problems are complete for this class. Consequently, MAX 3-SAT has a polynomial time approximation scheme (PTAS) if and only if the whole class of MAX SNP does. A PTAS for a MAX CSP can be thought of as a family of algorithms such that for any $\epsilon > 0$, it contains an algorithm that achieves an approximation ratio of $1 - \epsilon$. This means that if we can show that MAX 3-SAT cannot be approximated beyond some constant bounded away from 1, then the same will hold for many other natural combinatorial optimization problems. This was achieved with the discovery of the PCP theorem [AS98, ALM+98]. The PCP theorem led to a flourish of inapproximability results for combinatorial optimization problems (see e.g. [Tre14] for a survey of these results). This line of work culminated in Håstad's seminal paper [Hå01], in which he showed that for any $k \geq 3$, it is NP-hard to approximate MAX k-SAT within a ratio of $(2^k - 1)/2^k + \epsilon$ for any $\epsilon > 0$. He also showed that it is NP-hard to approximation MAX k-LIN(2) within a ratio of $1/2 + \epsilon$ for any $\epsilon > 0$, again for $k \geq 3$. Note that since the constants $(2^k - 1)/2^k$ and $1/2$ are the ratios achieved by a random assignment on these problems, Håstad's results are optimal. In the same paper, Håstad also proved NP-hard ratios for several MAX 2-CSPs, including $16/17 + \epsilon$ for MAX CUT, $11/12 + \epsilon$ for MAX DI-CUT and $21/22 + \epsilon$ for MAX 2-SAT. However, these ratios do not match the best SDP-based algorithms for their respective problems.

In 2002, Khot proposed the *Unique Games Conjecture (UGC)* [Kho02], which can be thought of as a conjecture on the existence of a certain kind of PCP system. As it turned out, UGC becomes crucial in addressing the aforementioned gap for MAX 2-CSPs. Assuming UGC (or at least that unique games is hard), Khot, Kindler, Mossel and O'Donnell proved that the approximation ratio achieved by the Goemans-Williamson algorithm is in fact optimal [KKMO07]. Their proof used the Majority is Stablest theorem, proved by [MOO10]. Extending these techniques, Austrin improved the hardness results for some other MAX

2-CSPs, including MAX 2-SAT [Aus07] and MAX 2-AND [Aus10].

In a breakthrough result [Rag08], Raghavendra showed that assuming Khot's UGC (or at least that unique games is hard), the approximation ratio of any MAX CSP is given by the *integrality gap ratio* of a generic SDP relaxation for that MAX CSP. This SDP relaxation is called the Basic SDP. Furthermore, it was shown that this integrality gap ratio can be achieved (up to an arbitrarily small additive error ϵ) in polynomial time by applying a generic rounding algorithm to the Basic SDP [Rag08, RS09]. However, this rounding algorithm is obtained via brute-force techniques and takes time doubly exponential in $1/\epsilon$, which makes calculating integrality gap ratios using this algorithm practically infeasible. This means that, as powerful as Raghavendra's framework is, it does not tell us everything we want to know about MAX CSPs, in particular the explicit approximation ratio for any given MAX CSP. That said, the investigation for approximability of MAX CSPs has more or less stagnated since Raghavendra's work. We hope that this thesis will renew the interest in this area.

1.2 Future Directions and Adjacent Topics

In this section, we describe some topics that are adjacent to, but won't be explored in this thesis. Nonetheless, we hope that techniques introduced in this thesis could be useful for them. They also serve as possible directions for future research.

The MAX SAT problem: The most interesting open question in this area is arguably whether the MAX SAT problem has a 7/8-approximation algorithm. Due to the result of Håstad [Hå01], we know that it is NP-hard to approximate MAX 3-SAT within a factor of $7/8 + \epsilon$ for any $\epsilon > 0$. The question is to understand whether by allowing clauses of various lengths we can make the problem more difficult. We suspect that the answer is affirmative by the following intuitive reasoning. For clauses with length ≤ 3, it is known that hyperplane rounding achieves a ratio of 7/8 [KZ97, Zwi02], whereas for clauses with length ≥ 3, a simple uniformly random assignment will give a ratio of 7/8. However, these two rounding functions are drastically different, and it seems unlikely that there is one unified rounding function that achieves best of both worlds.

The result on MAX NAE-SAT presented in this thesis can be thought of as some evidence that there is no 7/8-algorithm for MAX SAT. However, much more work seems to be required to actually arrive at a proof for this. In general, it seems in order to obtain tight approximability results for MAX SAT and other higher-arity MAX CSPs we would need a much better understanding of extremal problems involving some generalized notion of noise stability for high-dimensional sets.

Approximation resistance and approximability: Informally, a predicate P is called *approximation resistant* if it defines a MAX CSP for which the optimal approximation ratio is achieved by the uniform random assignment, up to an $o(1)$ term, and *approximable* otherwise. If we are given a predicate, deciding whether it is approximation resistant is an easier task than finding out its approximation ratio. However, here we are interested in finding a *characterization* which tells us which predicates are approximation resistant and which are approximable. Khot, Tulsiani, and Worah [KTW13] gave such a characterization based on whether there exist certain vanishing measures over the polytope of satisfying assignments, but their characterization is not known to be decidable. It remains an open question to give a decidable characterization for approximation resistance, or to prove that one of the existing characterizations is decidable.

There has also been some interest in a special case for this problem, where the predicates are restricted to *balanced linear threshold functions* (balanced LTFs). These are predicates of the form $f(x_1, \dots, x_k) = \operatorname{sign}\left(\sum_{i=1}^{k} w_k x_k\right)$ for some arity k and $a_1, \dots, a_k \in \mathbb{R}$. It is known that for some simple cases where $w_2 = \cdots = w_k$, the predicate is approximable ([Pot18, HP20]), and it was conjectured that all balanced LTFs are approximable [ABM10]. This conjecture was recently refuted by Potechin [Pot18] who constructed a balanced LTF that is approximation resistant, assuming UGC. It is still wide open to decide which balanced LTFs are approximation resistant and which are not.

MAX CSPs with global constraints: For some MAX CSPs it is natural to consider the variant where there are additional constraints on the number of variables that are assigned true. One such example is the maximum bisection (MAX BISECTION) problem, which is the same as MAX CUT except we also require that the two parts in the partition have the same size (same number of true variables and false variables). In contrast to the constraints that are given by the predicates, the constraint on the number of true variables is of a *global* nature, since it acts on all variables at the same time. Not many tight approximability results are known in the presence of global constraints. Even for MAX BISECTION, it is not known whether we can approximate it as well as MAX CUT (the current best approximation ratio for MAX BISECTION is ≈ 0.8776 [ABG16], just shy of the Goemans-Williamson ratio which is > 0.878).

More importantly, Raghavendra's Basic SDP no longer gives a clean dichotomy in this case. It would be very interesting to develop a general theory for MAX CSPs with global constraints.

Rounding schemes based on Brownian motion: Recently, inspired by tools from the discrepancy theory, Abbasi-Zadeh, Bansal, Guruganesh, Nikolov, Schwartz and Singh proposed a new framework for rounding schemes based on *sticky Brownian motions* [AZBG$^+$22]. On a very high level, their algorithm assigns to each variable a Brownian motion, whose velocity depends on the SDP solution as well as current location in the space; they then let the Brownian motions evolve within the unit cube, and once a Brownian motion hits a boundary surface, it gets absorbed and the corresponding variable gets the integral value represented by that surface. Eldan and Naor showed that this technique can be used to achieve the same guarantee as the Goemans-Williamson algorithm [EN19]. Their proof seems readily generalizable to other 2-CSPs as well. However, when the arity of CSP is 3 or more, the analysis of such rounding schemes seems to become much more involved.

Other optimization objectives: Finally, it is also of interest to explore other optimization objectives for the approximation algorithms. For example, instead of trying to maximize the number of satisfied constraints, we can also minimize the number of unsatisfied constraints. This may seem like an equivalent objective, but note that the notion of approximation ratio is now based on the number of unsatisfied constraints: an algorithm achieves a ratio of $\alpha > 1$ if given an instance that's $(1-\epsilon)$-satisfiable, it returns a solution that satisfies a $(1-\alpha\epsilon)$-fraction of the constraints. In particular, if the input instance is perfectly satisfiable then the algorithm has to find a satisfying assignment. This makes the problem much harder than the maximizing version. For example, if we consider MIN 2-SAT-DELETION which is the minimizing version of MAX 2-SAT, then it is known that assuming UGC this problem is NP-hard to approximate within any constant factor [Kho02]. It would be interesting to develop a more general theory for Minimizing CSPs.

CHAPTER 2

PRELIMINARIES

2.1 Constraint Satisfaction Problems

Definition 2.1. Let D be some finite set with $|D| \geq 2$. A *predicate* P with *domain* D is a function $D^k \to \{0, 1\}$, where $k \in \mathbb{N}^+$ is the *arity* of P.

We say that P is a Boolean predicate when $|D| = 2$.

Definition 2.2. Let $\Gamma = (D, \mathcal{P})$, where D is a finite set called the *domain*, and $\mathcal{P}$ is a set of predicates P each with domain D. A MAX CSP(Γ) instance is given by a finite set of variables V and a finite set of constraints (sometimes also called clauses) $\mathcal{C}$, where each constraint is some predicate $P \in \mathcal{P}$ applied to a subset of variables. Given an assignment $A : V \to D$, we say that a constraint is satisfied if it evaluates to 1. The objective of the problem is to find an assignment A that satisfies as many constraints as possible. We can also consider a weighted version where the instance also specifies a weight function $w : \mathcal{C} \to \mathbb{R}^{\geq 0}$, and the objective is to find an assignment that maximizes the total weight of satisfied constraints.

In this thesis, we will only study Boolean MAX CSPs (i.e. MAX CSPs where the domain set has size 2). We will often work with the weighted version where the weight is given by a probability measure, i.e., the total weight is 1. To make the analysis easier, we will use $D = \{-1, 1\}$, where -1 represents TRUE and 1 represents FALSE. Since the domain has been fixed, we will just write MAX CSP($\mathcal{P}$).

We will refer to a variable or a negated variable as a *literal*. For Boolean MAX CSPs, it is often the case that we want to apply a predicate to negated variables, which isn't allowed in the above definition. To remedy this, for a Boolean predicate $P : \{-1, 1\}^k \to \{0, 1\}$, we define the *closure of P under negation* to be the set

$$\mathrm{cl}(P) = \left\{ P^{\oplus b} : (x_1, \ldots, x_k) \mapsto P\left(b_1 x_1, b_2 x_2, \ldots, b_k x_k\right) \mid b = (b_1, \ldots, b_k) \in \{-1, 1\}^k \right\}.$$

For a set of Boolean predicates $\mathcal{P}$, define $\mathrm{cl}(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} \mathrm{cl}(P)$. Now if we want to apply a predicate to negated variables in our CSP, we just expand the predicate set to its closure under negation.

We now introduce the predicates that we will study in the rest of this thesis.

Definition 2.3. Let $k \in \mathbb{Z}^+$. The OR predicate on k variables is defined as

$$\mathrm{OR}_k(x_1, \ldots, x_k) = \begin{cases} 0 & \text{if } x_i = 1 \text{ for all } i \in [k],^1 \\ 1 & \text{otherwise.} \end{cases}$$

Let $\mathcal{P}_{\mathrm{SAT}} = \{\mathrm{OR}_k \mid k \geq 1\}$, and for any $\ell \in \mathbb{Z}^+$, $\mathcal{P}_{\ell\text{-SAT}} = \{\mathrm{OR}_k \mid 1 \leq k \leq \ell\}$.

The MAX SAT problem is defined as MAX CSP$(\mathrm{cl}\left(\mathcal{P}_{\mathrm{SAT}}\right))$. We won't be dealing with MAX SAT in this thesis, but we will work with MAX 2-SAT in Chapter 4, defined as MAX CSP$(\mathrm{cl}\left(\mathcal{P}_{2\text{-SAT}}\right))$, as well as its subproblems, namely, MAX CSP$(\mathcal{P})$ where $\mathcal{P} \subseteq \mathrm{cl}\left(\mathcal{P}_{2\text{-SAT}}\right)$.

Definition 2.4. Let CUT : $\{-1, 1\}^2 \to \{0, 1\}$ be the predicate which is satisfied if and only if the two inputs are not equal. Let DI-CUT : $\{-1, 1\}^2 \to \{0, 1\}$ be the predicate which is satisfied if and only if the first input is 1 and the second input -1.

The MAX CUT problem is defined as MAX CSP({CUT}), the MAX DI-CUT problem is defined as MAX CSP({DI-CUT}), and the MAX 2-AND problem is defined as MAX CSP(cl({DI-CUT})). We will study these three problems in Chapter 5.

Definition 2.5. Let $k \geq 2$ be an integer. The Not-All-Equal predicate on k variables is

1. Note that 1 represents FALSE in the domain, so $x_i = 1$ means x_i is FALSE.

defined as

$$\mathrm{NAE}_k(x_1, \ldots, x_k) = \begin{cases} 0 & \text{if } x_1 = x_2 = \cdots = x_k, \\ 1 & \text{otherwise.} \end{cases}$$

Let $\mathcal{P}_{\mathrm{NAE}} = \{\mathrm{NAE}_k \mid k \geq 1\}$.

In words, an NAE predicate is satisfied if and only if the inputs are not all equal. The MAX NAE SAT problem is defined as MAX CSP$(\mathrm{cl}\,(\mathcal{P}_{\mathrm{NAE}}))$, and the monotone MAX NAE SAT problem is defined as MAX CSP$(\mathcal{P}_{\mathrm{NAE}})$, where negated literals are not allowed. These two problems will be investigated in Chapter 6.

2.2 Unique Games Conjecture

The Unique Games Conjecture (UGC), introduced by Khot [Kho02], plays a crucial role in the study of hardness of approximation of CSPs. It concerns the hardness of the following *unique games* problem, which is a CSP defined with certain permutation constraints.

Definition 2.6 (Unique Games, as stated in [BHPZ23]). In a unique games instance $I = (G, L, \Pi)$, we are given a weighted graph $G = (V(G), E(G), w)$, a set of labels $[L] = \{1, 2, \ldots, L\}$ and a set of permutations $\Pi = \{\pi_e^v : [L] \to [L] \mid e = \{v, u\} \in E(G)\}$ such that for every $e = \{u, v\} \in E(G)$, $\pi_e^v = (\pi_e^u)^{-1}$. An assignment to this instance is a function $A : V(G) \to [L]$. We say that A satisfies an edge $e = \{u, v\}$ if $\pi_e^u(A(u)) = A(v)$. The value of an assignment A is the weight of satisfied edges, i.e., $\mathrm{Val}(I, A) = \sum_{e \in E(G): A \text{ satisfies } e} w(e)$, and the value of the instance $\mathrm{Val}(I)$ is defined to be the value of the best assignment, i.e., $\mathrm{Val}(I) = \max_A \mathrm{Val}(I, A)$.

A version of the conjecture can be stated as follows.

Conjecture 1 (Unique Games Conjecture, as stated in [BHPZ23]). For any $\eta, \gamma > 0$, there exists a sufficiently large L such that the problem of determining whether a given unique games instance I with L labels has $\mathrm{Val}(I) \geq 1 - \eta$ or $\mathrm{Val}(I) \leq \gamma$ is NP-hard.

We say that a problem is UG-hard if it is NP-hard assuming the UGC.

UGC is one of the more divisive conjectures in theoretical computer science. Recently, Khot, Minzer and Safra showed that the "2-to-2" conjecture, which is a weaker version of UGC, is true [KMS23]. This seems to be the most convincing evidence for UGC being true to date. That said, even if UGC should turn out to be false, it could still be the case that solving unique games is not in polynomial time (it might be NP-intermediate for example), and this intermediate hardness would still extend to all UG-hardness results including for MAX CSPs.

All of the hardness results in this thesis are UG-hardness results.

2.3 Gaussian Density Functions

Let $\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$ be the probability density function and $\Phi(x) = \int_{-\infty}^{x} \varphi(t)\mathrm{d}t$ be the cumulative probability function of the standard normal distribution $N(0,1)$. Let

$$\varphi_\rho(x,y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right)$$

be the probability density function of a pair (X,Y) of standard normal variables with correlation $\mathbb{E}[XY] = \rho$, where $-1 < \rho < 1$. (Note that $\varphi_0(x,y) = \varphi(x)\varphi(y)$.) The cumulative distribution function of (X,Y) is then:

$$\Phi_\rho(x,y) = \Phi(x,y,\rho) = \Pr[X \le x \wedge Y \le y] = \int_{-\infty}^{x} \int_{-\infty}^{y} \varphi_\rho(t_1,t_2)\mathrm{d}t_1\mathrm{d}t_2 \ .$$

We will sometimes use the notation $X, Y \sim_\rho N(0,1)$ to denote that the pair (X,Y) is sampled from this distribution.

Lemma 2.7. *The partial derivatives of $\Phi(x, y, \rho) = \Phi_\rho(x, y)$ are:*

$$\frac{\partial \Phi(x,y,\rho)}{\partial x} = \varphi(x)\Phi\left(\frac{y-\rho x}{\sqrt{1-\rho^2}}\right) ,$$
$$\frac{\partial \Phi(x,y,\rho)}{\partial y} = \varphi(y)\Phi\left(\frac{x-\rho y}{\sqrt{1-\rho^2}}\right) ,$$
$$\frac{\partial \Phi(x,y,\rho)}{\partial \rho} = \frac{1}{2\pi\sqrt{1-\rho^2}}\exp\left(-\frac{x^2-2\rho xy+y^2}{2(1-\rho^2)}\right) .$$

The first two partial derivatives can be easily derived from definition and the last equation is Equation 4 from [DW90].

The following proposition is taken from [Aus06] where a simple proof can also be found.

Proposition 2.8. *For all $x, y \in \mathbb{R}$, $\rho \in [-1, 1]$, we have*

$$\Phi_\rho(x,y) - \Phi_\rho(-x,-y) = \Phi(x) + \Phi(y) - 1 .$$

Definition 2.9. Let $t_1, t_2 \in \mathbb{R}$. Define

$$\mathbb{T}_\rho(t_1,t_2) = \mathop{\mathbb{E}}_{X,Y\sim_\rho N(0,1)}[T_{t_1}(X)T_{t_2}(Y)]$$

where $T_t(x) = \mathbf{1}\{x \le t\} - \mathbf{1}\{x \ge t\}$.

Proposition 2.10. *For any $t_1, t_2 \in \mathbb{R}$,*

$$\mathbb{T}_\rho(t_1,t_2) = 4\Phi_\rho(t_1,t_2) - 2(\Phi(t_1) + \Phi(t_2)) + 1.$$

Proof. By definition we have

$$
\begin{aligned}
\mathbb{T}_\rho(t_1, t_2) = &\Pr_{X,Y\sim_\rho N(0,1)}[X \le t_1 \wedge Y \le t_2] + \Pr_{X,Y\sim_\rho N(0,1)}[X \ge t_1 \wedge Y \ge t_2] \\
&- \Pr_{X,Y\sim_\rho N(0,1)}[X \le t_1 \wedge Y \ge t_2] - \Pr_{X,Y\sim_\rho N(0,1)}[X \ge t_1 \wedge Y \le t_2] \\
= &\Phi_\rho(t_1, t_2) + \big(\Phi_\rho(t_1, t_2) - \Phi(t_1) - \Phi(t_2) + 1\big) \\
&- \big(\Phi(t_1) - \Phi_\rho(t_1, t_2)\big) - \big(\Phi(t_2) - \Phi_\rho(t_1, t_2)\big) \\
= &4\Phi_\rho(t_1, t_2) - 2(\Phi(t_1) + \Phi(t_2)) + 1.
\end{aligned}
$$

Here in the second equality we used Proposition 2.8. □

2.4 Fourier Analysis of Boolean Functions

We recall some definitions and basic facts from the analysis of Boolean functions. The most important fact that we need is the following *Fourier expansion theorem* which allows us to express any Boolean function as a multilinear polynomial.

Theorem 2.11 (See e.g. Theorem 1.1 in [O'D14])**.** *Every function $f : \{-1, 1\}^n \to \mathbb{R}$ can be uniquely expressed in the form*

$$f(x_1, \dots, x_n) = \sum_{S\subseteq[n]} \hat{f}_S \cdot \prod_{i\in S} x_i. \tag{2.1}$$

This expression is called the Fourier expansion *of f, and the coefficients $\hat{f}_S$ are called* Fourier coefficients*.*

One way to prove the Fourier expansion theorem is to think of the monomials as an orthonormal basis of the space of functions $\{-1, 1\}^n \to \mathbb{R}$, equipped with the expectation norm induced by the uniform distribution over $\{-1, 1\}^n$. This can be generalized to the situation where instead of the uniform distribution, we have independent but biased coin flips for each coordinate. More specifically, for some $q \in (0,1)$, let B_q^n be the probability space over $\{-1,1\}^n$ where each bit is independently set to -1 with probability q and to 1 with probability $1-q$. Let $U_q(1) = \sqrt{\frac{q}{1-q}}$ and $U_q(-1) = -\sqrt{\frac{1-q}{q}}$, and for any $S \subseteq [n]$, let $U_q^S(x_1, \ldots, x_n) = \prod_{i \in S} U_q(x_i)$. Then it is easy to verify (c.f., Proposition 2.7 of [Aus10]) that

$$\left\{U_q^S : B_q^n \to \mathbb{R} \mid S \subseteq [n]\right\}$$

is an orthonormal basis for real-valued functions on B_q^n with respect to the inner product defined via expectation. We can again define the Fourier coefficients as $\hat{f}_S = \mathop{\mathbb{E}}_{\mathbf{x} \sim B_q^n}[f(\mathbf{x}) U_q^S(\mathbf{x})]$ (note that q is implicit in the domain of f), which gives us the following decomposition:

$$f = \sum_{S \subseteq [n]} \hat{f}_S U_q^S. \tag{2.2}$$

The decomposition (2.2) generalizes (2.1), which is a special case where we have $q = 1/2$.

In our application, we are also interested in computing correlation of two functions with different biases.

Definition 2.12. Let $f : B_{q_1}^n \to \mathbb{R}$ and $g : B_{q_2}^n \to \mathbb{R}$. The ρ-correlation between f and g is defined as

$$\mathbb{S}_\rho(f, g) := \mathbb{E}[f(\mathbf{x}) g(\mathbf{y})],$$

where $\mathbf{x} \sim B_{q_1}^n$, $\mathbf{y} \sim B_{q_2}^n$, and furthermore the i-th coordinate of $\mathbf{x}$ and the i-th coordinate of $\mathbf{y}$ has correlation ρ, i.e., $\frac{\mathbb{E}[x_i y_i] - \mathbb{E}[x_i]\,\mathbb{E}[y_i]}{\sqrt{(1-\mathbb{E}[x_i]^2)(1-\mathbb{E}[y_i]^2)}} = \rho$.

Definition 2.13. Let $f : B_q^n \to \mathbb{R}$ and $k \in [n]$. The k-low-degree influence of coordinate i on f is defined as

$$\mathrm{Inf}_i^{\leq k}[f] := \sum_{S : i \in S \subseteq [n], |S| \leq k} \hat{f}_S^2.$$

It is straightforward from the definition that $\mathrm{Inf}_i^{\leq k}$ is convex.

Proposition 2.14. *Let $f : B_q^n \to [-1,1]$. For any $\eta > 0$ and $k \in [n]$, we have*

$$\left|\left\{i \in [n] \mid \mathrm{Inf}_i^{\leq k}[f] > \eta\right\}\right| \leq \frac{k}{\eta}.$$

Proof. We have

$$\sum_{i=1}^{n} \mathrm{Inf}_i^{\leq k}[f] = \sum_{i=1}^{n} \sum_{\substack{S: i \in S \subseteq [n], \\ |S| \leq k}} \hat{f}_S^2 = \sum_{|S| \leq k} |S| \hat{f}_S^2 \leq k \cdot \sum_{|S| \leq k} \hat{f}_S^2 \leq k.$$

The proposition follows immediately. □

It turns out that for functions with small low-degree influences, the extremal behavior of their ρ-correlations is characterized by threshold functions in Gaussian space.

Theorem 2.15 (Corollary 2.19, [Aus10])**.** *For any $\epsilon > 0$ and $\rho \in (-1,1)$, there exist $k \in \mathbb{N}$ and $\eta > 0$ such that for all $f : B_{q_1}^n \to \mathbb{R}$ and $g : B_{q_2}^n \to \mathbb{R}$ satisfying $\min(\mathrm{Inf}_i^{\leq k}[f], \mathrm{Inf}_i^{\leq k}[g]) \leq \eta$ for every $i \in [n]$, we have*

$$4\Phi_{-|\rho|}(t_1,t_2) - \epsilon \leq \mathbb{S}_\rho(f,g) - \mathbb{E}[f] - \mathbb{E}[g] + 1 \leq 4\Phi_{|\rho|}(t_1,t_2) + \epsilon,$$

where $t_1 = \Phi^{-1}\left(\frac{1-\mathbb{E}[f]}{2}\right)$ and $t_2 = \Phi^{-1}\left(\frac{1-\mathbb{E}[g]}{2}\right)$.

We will use a restatement of the above theorem in terms of $\mathbb{T}_\rho(t_1,t_2)$.

Corollary 2.16. *For any $\epsilon > 0$, there exist $k \in \mathbb{N}$ and $\eta > 0$ such that for all $f : B_{q_1}^n \to \mathbb{R}$ and $g : B_{q_2}^n \to \mathbb{R}$ satisfying $\min(\mathrm{Inf}_i^{\leq k}[f], \mathrm{Inf}_i^{\leq k}[g]) \leq \eta$ for every $i \in [n]$, we have*

$$-\mathbb{T}_{|\rho|}(t_1,t_2) - \epsilon \leq \mathbb{S}_\rho(f,g) \leq \mathbb{T}_{|\rho|}(t_1,t_2) + \epsilon,$$

where $t_1 = \Phi^{-1}\left(\frac{1-\mathbb{E}[f]}{2}\right)$ and $t_2 = \Phi^{-1}\left(\frac{1-\mathbb{E}[g]}{2}\right)$.

Proof. Follows directly from Proposition 2.10 and Theorem 2.15. □

In the above corollary we can also take $t_1 = \Phi^{-1}\left(\frac{1+\mathbb{E}[f]}{2}\right) = -\Phi^{-1}\left(\frac{1-\mathbb{E}[f]}{2}\right)$ and $t_2 = \Phi^{-1}\left(\frac{1+\mathbb{E}[g]}{2}\right) = -\Phi^{-1}\left(\frac{1-\mathbb{E}[g]}{2}\right)$, since $\mathbb{T}_\rho(x,y) = \mathbb{T}_\rho(-x,-y)$ for any ρ, x, y.

2.5 Approximation Algorithms and Approximation Ratios

An approximation algorithm for an optimization problem is an efficient algorithm which produces a solution that is provably close to the optimal solution. In the study of MAX CSPs, "closeness" can be formalized using the notion of approximation ratios.

Definition 2.17. We say that an (possibly randomized) algorithm $\mathcal{A}$ achieves an *approximation ratio* of α for MAX CSP($\mathcal{P}$), if for every instance I of MAX CSP($\mathcal{P}$), we have

$$\mathbb{E}\big[\mathrm{Val}(I, \mathcal{A}(I))\big] \geq \alpha \cdot \mathrm{OPT}(I).$$

Here, $\mathcal{A}(I)$ is the (possibly random) assignment that $\mathcal{A}$ produces given I.

We will also talk about the approximation ratio of some MAX CSP($\mathcal{P}$), by which we mean the best approximation ratio achieved by any polynomial time algorithm for this problem. We will sometimes denote this ratio by $\alpha_{\mathcal{P}}$. To compare the approximation ratios between two problems, we can use an approximation ratio preserving reduction.

Definition 2.18 (See e.g. [Vaz03])**.** An *approximation ratio preserving reduction* from MAX CSP($\mathcal{P}$) to MAX CSP($\mathcal{Q}$) is a pair of functions (f, g) with the following properties.

- Both f and g can be computed in polynomial time.
- f maps instances of MAX CSP($\mathcal{P}$) to instances of MAX CSP($\mathcal{Q}$), and for every instance I of MAX CSP($\mathcal{P}$), $\mathrm{OPT}(I) \leq \mathrm{OPT}(f(I))$.
- Given any instance I of MAX CSP($\mathcal{P}$), and an assignment A to $f(I)$ (which is an instance of MAX CSP($\mathcal{Q}$), g produces an assignment $g(A)$ to I such that $\mathrm{Val}(I, g(A)) \geq \mathrm{Val}(f(I), A)$.

Lemma 2.19. *If there exists an approximation ratio preserving reduction from MAX CSP*($\mathcal{P}$) *to MAX CSP*($\mathcal{Q}$)*, then* $\alpha_{\mathcal{P}} \geq \alpha_{\mathcal{Q}}$.

Proof. Let (f, g) be an approximation ratio preserving reduction from MAX CSP($\mathcal{P}$) to MAX CSP($\mathcal{Q}$). Let $\mathcal{A}$ be an approximation algorithm for MAX CSP($\mathcal{Q}$) that achieves achieves an approximation ratio of $\alpha_{\mathcal{Q}}$. Consider the following algorithm for MAX CSP($\mathcal{P}$): given any instance I of MAX CSP($\mathcal{P}$), we apply f on I to obtain $f(I)$, solve $f(I)$ using $\mathcal{A}$, and apply g on $\mathcal{A}(f(I))$ to obtain an assignment for I. We have

$$\begin{aligned}\mathbb{E}[\mathrm{Val}(I, g(\mathcal{A}(f(I))))] &\geq \mathbb{E}[\mathrm{Val}(f(I), \mathcal{A}(f(I)))] \\ &\geq \alpha_{\mathcal{Q}} \cdot \mathrm{OPT}(f(I)) \\ &\geq \alpha_{\mathcal{Q}} \cdot \mathrm{OPT}(I).\end{aligned}$$

It follows that this algorithm achieves an approximation ratio of $\alpha_{\mathcal{Q}}$ on MAX CSP($\mathcal{P}$), so $\alpha_{\mathcal{P}} \geq \alpha_{\mathcal{Q}}$. □

2.6 Interval Arithmetic

A few results discussed in this thesis are proved with the help of computer assistance using a technique called *interval arithmetic*. In interval arithmetic, instead of doing arithmetic with numbers, we apply arithmetic operations to intervals of numbers. More specifically, let *op* be a k-ary operation, and $I_1, I_2, \ldots, I_k$ intervals, then the interval arithmetic for $op(I_1, I_2, \ldots, I_k)$ produces an interval I_{op} with the following guarantee:

$$\forall (g_1, g_2, \ldots, g_k) \in I_1 \times I_2 \times \cdots \times I_k, \quad op(g_1, g_2, \ldots, g_k) \in I_{op}.$$

This can be used to certify inequalities of the form $f \geq 0$ where f is a function defined on some direct product of intervals. To do this, we simply implement f using interval arithmetic, and if the output interval is contained in $[0, +\infty)$, this will give us a computer-assisted, *rigorous* proof that $f \geq 0$.

However, since the implementation of arithmetic operations are usually not exact when floating-point numbers are involved, to maintain correctness, the interval I_{op} usually also contains elements that are not in the range of the operation. One way to deal with this is using a *divide-and-conquer* approach. The idea is simple: if the intervals $I_1, \dots, I_k$ are all small, then the output interval I_{op} will also be small and the absolute error will be reduced. This works well when we are certifying inequalities of the form $f \geq 0$ but f is actually sufficiently bounded away from 0. Note that this does come with a price: the number of arithmetic operations increase exponentially as the desired accuracy increases.

In the case that f actually achieves 0, we cannot hope to certify $f \geq 0$ using interval arithmetic alone, unless we have some very special condition on f that makes exact evaluation possible. In this scenario, we may need to check the partial derivatives of f. This may also be combined with analytical proofs for properties of f.

The interval arithmetic proofs discussed in this thesis are implemented using the Arb library [Joh17].

CHAPTER 3

THE BASIC SDP AND ROUNDING SCHEMES

In this chapter, we describe the SDP framework for approximating CSPs. We describe a generic SDP relaxation called the Basic SDP, formulated by Raghavendra [Rag08, Rag09], and its special case for 2-CSPs [Aus10]. We also describe optimal or conjectured optimal rounding algorithms for these SDP relaxations, which provide the framework for the approximability and inapproximability results in later chapters.

3.1 SDP-Based Approximation Algorithms: Framework and Challenges

Semi-definite programming (SDP) plays a central role in the design of approximation algorithms for MAX CSP, as well as for many other combinatorial optimization problems. In a semi-definite program, we have a set of vector-valued variables, and both the constraints and the objective function are expressed in terms of inner products of these vector-valued variables. For MAX CSP, a typical SDP-based approximation algorithm consists of the following two steps:

- **Relaxation**. We write the given instance of MAX CSP as an integer program, where each variable takes a value in $\{-1, 1\}$. This can usually be done in a straightforward manner. The integer program describes the original problem in an exact way, such that if we could solve the integer program, we would be able to recover a solution to the original problem. However, solving integer programs in general is NP-hard. We therefore *relax* the integer program to a semi-definite program by replacing each integer-valued variable with a vector-valued variable.

- **Rounding** After solving the semi-definite program, the algorithm then uses a *rounding scheme* which is an algorithm that produces integer values based on the SDP vectors.

This rounding scheme will usually be a randomized algorithm, and we will analyze its performance in expectation.

As an illustrating example, let us consider the MAX CUT problem. Given an instance $G = (V, E)$, we can formulate MAX CUT using the following integer program.

$$\begin{aligned} \text{maximize} \quad & \sum_{\{i,j\} \in E} \frac{1 - x_i x_j}{2} \\ \text{subject to} \quad & x_i^2 = 1, \quad \forall i \in V \end{aligned}$$

Note that x_i is restricted to being either -1 or 1, which naturally induces a cut. The expression $\frac{1-x_i x_j}{2}$ is evaluated to 1 if x_i and x_j have different signs, and 0 if they have the same sign, so the objective is indeed maximizing the number of edges across the cut. This integer program is NP-hard to solve, so we relax it to a *semi-definite program*, where each variable is now vector-valued:

$$\begin{aligned} \text{maximize} \quad & \sum_{\{i,j\} \in E} \frac{1 - \mathbf{v}_i \cdot \mathbf{v}_j}{2} \\ \text{subject to} \quad & \mathbf{v}_i \cdot \mathbf{v}_i = 1, \quad \forall i \in V \end{aligned}$$

This is the famous Goemans-Williamson SDP [GW95]. A semi-definite program can be alternatively formulated using the *moment matrix* $B = (b_{i,j})_{1 \le i,j \le n}$ where $b_{i,j} = b_{j,i} = \mathbf{v}_i \cdot \mathbf{v}_j$. In this alternative formulation, the objective function, as well as the constraints, will be expressed as linear functions on the entries of B, and B, being a Gram matrix, has to be positive semi-definite (hence the name of semi-definite programming). On the other hand, as long as B is a positive semi-definite matrix, we can find vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ such that $\mathbf{v}_i \cdot \mathbf{v}_j = b_{ij}$, so this is indeed an equivalent formulation. From this formulation it is straightforward to see that semi-definite programming is a convex optimization problem, and therefore it can be solved up to any fixed additive error in polynomial time (see e.g. [VB96]).

Solving the Goemans-Williamson SDP produces a set of unit vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Note that since the semi-definite program is a relaxation of the integer program, we have $\mathrm{SDP}(G) \geq \mathrm{OPT}(G)$, where $\mathrm{SDP}(G)$ is the value of the semi-definite program and $\mathrm{OPT}(G)$ is the value of the integer program (i.e., the value of the maximum cut in G). We now have the task of *rounding* these vectors to Boolean values 1 or -1. To do this, Goemans and Williamson proposed the following *hyperplane rounding* algorithm.

Algorithm 1 Hyperplane rounding algorithm [GW95]

Input: $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ unit vectors obtained by solving the Goemans-Williamson SDP

Output: $x_1, \ldots, x_n \in \{-1, 1\}$ rounded Boolean assignment to the variables

$\mathbf{r} \leftarrow N(0, I_n)$

for $i \leftarrow 1$ to n **do**

$t_i \leftarrow \mathbf{r} \cdot \mathbf{v}_i$

$x_i \leftarrow 1$ if $t_i \geq 0$, and $x_i \leftarrow -1$ otherwise

We will see that hyperplane rounding algorithm lies in the RPR^2 rounding family ([FL06]), which we will discuss in Chapter 6. The algorithm is called hyperplane rounding because we can think of $\mathbf{r}$ as the normal vector of some random hyperplane, and variables on one side of the hyperplane are rounded to 1, while those on the other side are rounded to -1. Given any edge $\{i, j\}$, the probability that it is cut by the hyperplane rounding is equal to the probability that $\mathbf{v}_i$ and $\mathbf{v}_j$ lie on different sides of the hyperplane, which happens with probability $\frac{1}{\pi} \arccos(\mathbf{v}_i \cdot \mathbf{v}_j)$. It then follows that

$$\sum_{\{i,j\} \in E} \mathbb{E}\left[\frac{1 - x_i x_j}{2}\right] = \sum_{\{i,j\} \in E} \frac{1}{\pi} \arccos(\mathbf{v}_i \cdot \mathbf{v}_j) \geq \sum_{\{i,j\} \in E} \alpha \cdot \frac{1 - \mathbf{v}_i \cdot \mathbf{v}_j}{2} = \alpha \cdot \mathrm{SDP}(G),$$

where $\alpha = \min_{b \in (-1,1)} \frac{2 \arccos(b)}{\pi(1-b)} \geq 0.87856$. Since $\mathrm{SDP}(G) \geq \mathrm{OPT}(G)$, this produces the guarantee that the output of the hyperplane rounding achieves an approximation ratio of α

in expectation.

The crucial observation here is that the analysis for the algorithm is *local* in nature: a lower bound on the ratio achieved by the algorithm on all individual constraint will give us a lower bound on the overall approximation ratio of this algorithm. We remark that this is true for SDP-based algorithms for MAX CSPs in general.

As it turns out, there is in fact a generic SDP relaxation, called the *Basic SDP* for all MAX CSPs. Raghavendra [Rag08] showed that if UGC is true, then an algorithm optimal up to any additive error can be obtained from rounding this SDP. We will describe this relaxation in the next section. Raghavendra's result takes away the "creativity" part of the relaxation step, so to design an SDP-based approximation algorithm, it is sufficient to focus on the rounding step. The challenge for the rounding step is two-fold:

- Find a good rounding scheme. This is a highly non-trivial task since there is a huge space of potential rounding schemes. In the hyperplane rounding algorithm, only one Gaussian vector $\mathbf{r}$ is used, and we only looked at the sign of the product $\mathbf{r} \cdot \mathbf{v}_i$, but in general we can consider any number of Gaussian vectors and the rounding scheme can depend arbitrarily on the numerical values of the inner products obtained from the Gaussian vectors.

- Given a candidate good rounding scheme, *certify* the approximation ratio that it achieves. This is also difficult because we need to show that the rounding scheme works on all instances, which is again an optimization problem over a very large domain. In many cases, computer assisted proofs are necessary for such certification.

3.2 Formulation of the Basic SDP

Suppose we have an MAX CSP instance with variable set $V = \{x_1, \ldots, x_n\}$, constraint set $\mathcal{C}$ and weight function $w : \mathcal{C} \to \mathbb{R}^{\geq 0}$, then the basic SDP for this instance is formulated as follows [Rag08, Rag09][1]:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{C \in \mathcal{C}} w(C) \left(\sum_{\alpha \in \mathcal{A}(C)} p_C(\alpha) C(\alpha) \right) \\
\text{subject to} \quad & \mathbf{v}_i \cdot \mathbf{v}_i = 1, \quad \forall i \in \{0, 1, 2, \ldots, n\} \\
& \mathbf{v}_i \cdot \mathbf{v}_j = \sum_{\alpha \in \mathcal{A}(C)} \alpha\left(x_i\right) \alpha\left(x_j\right) p_C(\alpha), \quad \forall C \in \mathcal{C}, \forall z_i, z_j \in C \\
& \mathbf{v}_i \cdot \mathbf{v}_0 = \sum_{\alpha \in \mathcal{A}(C)} \alpha\left(x_i\right) p_C(\alpha), \quad \forall C \in \mathcal{C}, \forall z_i \in C \\
& p_C(\alpha) \geq 0, \quad \forall C \in \mathcal{C}, \forall \alpha \in \mathcal{A}(C).
\end{aligned}
$$

The SDP variables are the vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$ and non-negative real numbers $\{p_C(\alpha) \mid C \in \mathcal{C}, \alpha \in \mathcal{A}(C)\}$. We can think of $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$ as "global variables" and $\{p_C(\alpha) \mid C \in \mathcal{C}, \alpha \in \mathcal{A}(C)\}$ as "local variables". In particular, $p_C(\alpha)$ represents the probability of choosing the local assignment α for variables that are involved in constraint C. Note that $p_C(\alpha)$ can be easily expressed using vector products, by introducing some new vector-valued variables $\mathbf{v}_{C,\alpha}$ and have $p_C(\alpha) = \mathbf{v}_{C,\alpha} \cdot \mathbf{v}_{C,\alpha}$, but for conceptual clarity we will keep them as scalars instead.

For any constraint C, the SDP looks at the set $\mathcal{A}(C)$ which contains all local assignments to the variables which appear in C, and chooses $\{p_C(\alpha) \mid \alpha \in \mathcal{A}(C)\}$ which describes a distribution over $\mathcal{A}(C)$ (the first two constraints imply that $\sum_\alpha p_C(\alpha) = 1$). For a local assignment α to C, $C(\alpha)$ denotes the value (satisfied/unsatisfied) of C under this assignment, and $\alpha(x_i)$ denotes the value that the local assignment α assigns to the variable x_i. We use z_i to denote a literal that is either x_i or the negation of x_i.

As for the "global variables" $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$, they correspond to the variables in the original MAX CSP instance ($\mathbf{v}_0$ is used as a special vector intended to be the truth value "False"),

1. In [Rag08, Rag09], the SDP is formulated for MAX CSP over general finite domains. The formulation here is a special case for Boolean MAX CSPs.

and are a relaxation of the one-dimensional case in which they would be assigned Boolean values in $\{-1, 1\}$. Using these variables the SDP enforces upon all local constraints the consistency requirement that the first and second moments of the local distributions match. Namely, if we define *biases* $b_i := \mathbf{v}_0 \cdot \mathbf{v}_i$ and *pairwise biases* $b_{i,j} := \mathbf{v}_i \cdot \mathbf{v}_j$, then the second and third constraints in the SDP require that $b_i = \mathbb{E}_{p_C}[x_i], b_{i,j} = \mathbb{E}_{p_C}[x_i x_j]$ for every C. Under this consistency requirement, the SDP searches for a distribution p_C of local assignments for every clause $C \in \mathcal{C}$ that maximizes the probability that C is satisfied.

We remark that in order to be able to solve this SDP in polynomial time, the number of constraints should be a polynomial in the size of given MAX CSP. This requires that the arity of any predicate in the given MAX CSP should be uniformly bounded by a constant. This condition is satisfied by all MAX CSPs analyzed in this thesis.

It is natural to ask if we can strengthen the SDP by adding even more consistency constraints. For example, what happens if we insist that the third moments also match? This can be captured by the SOS hierarchy, also known as the Lasserre hierarchy [Las01], which is a family of more and more powerful semi-definite programs. Raghevendra showed that if UGC is true, then adding (polynomially many) more constraints does not help in the worst case. To formally state his results, we introduce the notion of *approximability curve.*

Definition 3.1. For any MAX CSP instance Φ, let SDP(Φ) be its objective value in the Basic SDP[2]. The approximability curve of MAX CSP(Γ) is a function $s_\Gamma : [0, 1] \to [0, 1]$ defined as

$$s_\Gamma(c) = \inf\{\mathrm{OPT}(\Phi) \mid \Phi \in \mathrm{MAXCSP}(\Gamma), \mathrm{SDP}(\Phi) = c\}.$$

Theorem 3.2 ([Rag08, Rag09, RS09])**.** *Fix any $\epsilon > 0$. Assuming UGC, then for any c it is NP-hard to distinguish between instances of MAX CSP(Γ) with SDP value at least c and those with OPT value at most $s_\Gamma(c + \epsilon) + \epsilon$. Moreover, there exists a polynomial-time algorithm that, given an instance Φ of MAX CSP(Γ) with SDP value c, produces an assignment to Φ*

2. This is sometimes also called the *completeness* of Φ.

with value at least $s_\Gamma(c-\epsilon)-\epsilon$.

We briefly discuss the proof ideas for Theorem 3.2. To show the hardness result, Raghavendra's construction takes any MAX CSP(Γ) instance with SDP value at least c and OPT value at most $s_\Gamma(c)$ (such instances are called *integrality gap instances* since there is a gap between the optimal integral value and the SDP value) and converts them into *dictatorship tests* with corresponding completeness and soundness values (with a loss of ϵ). This then gives UG-hardness results via standard machinery. We shall give a proof of the special case of this for 2-CSPs in the next section.

As for the algorithmic result, [RS09] shows that we can project the SDP vectors into some d-dimensional space, such that if d is chosen to be a large enough constant, then most of the SDP constraints will be approximately preserved. They then use an ϵ-net to group those variables together whose corresponding projected vectors are close enough and identify the variables in the same group. This will give as an instance with constantly many variables, which can then be solved using brute force.

Theorem 3.2 is a remarkable breakthrough result. It would appear that Theorem 3.2 completely resolves the problem of approximating MAX CSPs assuming UGC. However, this is not the case. There are two main drawbacks.

- Theorem 3.2 does not tell us where to find the hardest instance. That is, for each c, we don't know how to quickly find Φ which achieves or nearly achieves the infimum in the definition of $s_\Gamma(c)$. It merely converts any given integrality gap instance into a UG-hardness result. Furthermore, the rounding algorithm in Theorem 3.2 is a brute-force algorithm and has running time that is doubly exponential in $1/\epsilon$. While theoretically for fixed $\epsilon > 0$ this is a polynomial time algorithm, but in practice, as soon as ϵ is moderately small, this algorithm quickly becomes infeasible.

- Given two MAX CSPs, this algorithm is unable to certify that the two problems have the same approximation ratio. Indeed, if the approximation ratios are different, then by choosing ϵ smaller than the difference the algorithm would be able to certify that they have different approximation ratios. However, if the two MAX CSPs have the same approximation ratio, then no longer how small ϵ we choose, the algorithm won't be able to return a definitive answer.

That said, Theorem 3.2 gives us a powerful framework for studying the approximability of MAX CSPs. On one hand, to design approximation algorithms for MAX CSPs, it suffices to construct clever rounding schemes for the Basic SDP. On the other hand, to obtain inapproximability results, it is sufficient to prove limitations on rounding algorithms for the Basic SDP.

3.3 Austrin's Formulation for 2-CSPs

In this section, we look at the special case of the Basic SDP where all predicates in the given MAX CSP have arity at most 2. It can be formulated as follows [Aus10].

$$\begin{aligned}
\text{Maximize} \quad & \sum_{C(x_i,x_j)\in\mathcal{C}} w(C)\cdot\left(\hat{C}_\varnothing+\hat{C}_1\mathbf{v}_0\cdot\mathbf{v}_i+\hat{C}_2\mathbf{v}_0\cdot\mathbf{v}_j+\hat{C}_{1,2}\mathbf{v}_i\cdot\mathbf{v}_j\right)\\
\text{subject to} \quad & \mathbf{v}_i\cdot\mathbf{v}_i=1,\quad \forall i\in\{0,1,2,\ldots,n\}\\
& \|\mathbf{v}_i-\mathbf{v}_j\|^2+\|\mathbf{v}_j-\mathbf{v}_k\|^2\geq\|\mathbf{v}_i-\mathbf{v}_k\|^2,\quad \forall i,j,k\in\{0,1,2,\ldots,n\}
\end{aligned}$$

The most obvious difference here is that all the local variables $p_C(\alpha)$ and constraints involving them are gone. Instead, we have the new inequalities $\|\mathbf{v}_i-\mathbf{v}_j\|^2+\|\mathbf{v}_j-\mathbf{v}_k\|^2\geq\|\mathbf{v}_i-\mathbf{v}_k\|^2$, which are called *triangle inequalities*. Note that these are not typical triangle inequalities in the sense of the L_2 norm, but it can be easily verified that any one dimensional solution satisfies them. As for the objective, instead of having to express the value using the local distribution, we can now take the Fourier expansion of the predicate and replace the linear and quadratic terms with inner products between corresponding SDP vectors.

It is clear that the global variables $\mathbf{v}_0, \ldots, \mathbf{v}_n$ in any solution to the Basic SDP satisfy the triangle inequality. To show that this formulation is actually equivalent to the Basic SDP in the case of 2-CSPs, it suffices to show that any vectors $\mathbf{v}_0, \ldots, \mathbf{v}_n$ satisfying the triangle inequalities induce a local distribution matching the biases and pairwise biases on any constraint involving at most 2 variables.

Proposition 3.3. *Let $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ be three unit vectors such that*

$$\|\mathbf{v}_i - \mathbf{v}_j\|^2 + \|\mathbf{v}_j - \mathbf{v}_k\|^2 \geq \|\mathbf{v}_i - \mathbf{v}_k\|^2, \quad \forall i, j, k \in \{0, 1, 2\}.$$

Then there exists a distribution over $(x_1, x_2) \in \{-1, 1\}^2$ such that $\mathbb{E}[x_1] = \mathbf{v}_0 \cdot \mathbf{v}_1, \mathbb{E}[x_2] = \mathbf{v}_0 \cdot \mathbf{v}_2, \mathbb{E}[x_1 x_2] = \mathbf{v}_1 \cdot \mathbf{v}_2$.

Proof. We can take the following probabilities:

$$\begin{aligned}
\Pr[x_1 = -1, x_2 = -1] &= \frac{1 - \mathbf{v}_0 \cdot \mathbf{v}_1 - \mathbf{v}_0 \cdot \mathbf{v}_2 + \mathbf{v}_1 \cdot \mathbf{v}_2}{4}, \\
\Pr[x_1 = -1, x_2 = +1] &= \frac{1 - \mathbf{v}_0 \cdot \mathbf{v}_1 + \mathbf{v}_0 \cdot \mathbf{v}_2 - \mathbf{v}_1 \cdot \mathbf{v}_2}{4}, \\
\Pr[x_1 = +1, x_2 = -1] &= \frac{1 + \mathbf{v}_0 \cdot \mathbf{v}_1 - \mathbf{v}_0 \cdot \mathbf{v}_2 - \mathbf{v}_1 \cdot \mathbf{v}_2}{4}, \\
\Pr[x_1 = +1, x_2 = +1] &= \frac{1 + \mathbf{v}_0 \cdot \mathbf{v}_1 + \mathbf{v}_0 \cdot \mathbf{v}_2 + \mathbf{v}_1 \cdot \mathbf{v}_2}{4}.
\end{aligned}$$

By the triangle inequalities, these values are all non-negative. Since they also sum up to 1, we obtain a valid distribution, and we indeed have $\mathbb{E}[x_1] = \mathbf{v}_0 \cdot \mathbf{v}_1, \mathbb{E}[x_2] = \mathbf{v}_0 \cdot \mathbf{v}_2, \mathbb{E}[x_1 x_2] = \mathbf{v}_1 \cdot \mathbf{v}_2$. □

Proposition 3.3, together with the previous discussion, shows that the two SDP formulations are indeed equivalent. It follows from Theorem 3.2 that it suffices to analyze the simpler form with the triangle inequalities. It turns out that not only does the SDP become simpler in this case, Austrin [Aus10] gave evidence that the optimal rounding scheme might also be simpler. To state his results, we first define the notion of *configurations*.

Definition 3.4. A configuration consists of biases and pairwise biases that appear in the same constraint, as well as a predicate type which is the predicate that is used to define this constraint. It can be represented using a tuple $\theta = ((b_{i_j})_{j=1}^k, (b_{i_j,i_\ell})_{1\le j<\ell\le k}, P)$ where P is some k-ary constraint containing variables $i_1, \dots, i_k$. To emphasize the arity, we will sometimes call this a k-configuration. We will also write $(b_i, b_j, b_{i,j}, P)$ in the case where $k = 2$ and (b_i, P) where $k = 1$, for the sake of simplicity. We use Θ to denote a distribution over configurations.

Definition 3.5. We say that a configuration is feasible, if its biases and pairwise biases can be obtained from a local distribution. In particular, a 2-configuration is feasible if and only if it satisfies the triangle inequalities.

Definition 3.6. For any configuration $\theta = ((b_i)_{i=1}^k, (b_{i,j})_{1\le i<j\le k}, P)$, let $\mathrm{SDP}(\theta)$ be the SDP value achieved by this configuration in the Basic SDP. For 2-configurations, this can be written as $\mathrm{SDP}(\theta) = \hat{P}_\varnothing + \hat{P}_1 \mathbf{v}_0 \cdot \mathbf{v}_i + \hat{P}_2 \mathbf{v}_0 \cdot \mathbf{v}_j + \hat{P}_{1,2} \mathbf{v}_i \cdot \mathbf{v}_j$. Let $\mathrm{SDP}(\Theta) = \mathbb{E}_{\theta\sim\Theta}[\mathrm{SDP}(\theta)]$.

For 2-CSPs, we will pay special attention to the following family of rounding schemes.

Algorithm 2 $\mathcal{THRESH}^-$ rounding scheme with threshold function $f : [-1, 1] \to [-1, 1]$

Input: $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^n$ unit vectors obtained by solving the Basic SDP
Output: $x_1, \dots, x_n \in \{-1, 1\}$ rounded Boolean assignment to the variables
$\mathbf{r} \leftarrow N(0, I_n)$
for $i \leftarrow 1$ to n **do**
  if $|b_i| \neq 1$ **then**
    $\mathbf{v}_i^\perp \leftarrow \frac{\mathbf{v}_i - b_i \cdot \mathbf{v}_0}{\sqrt{1-b_i^2}}$
  else
    $\mathbf{v}_i^\perp \leftarrow$ some unit vector that's orthogonal to all other vectors seen in this algorithm
  if $\mathbf{v}_i^\perp \cdot \mathbf{r} \ge \Phi^{-1}(\frac{1+f(b_i)}{2})$ **then**
    $x_i \leftarrow -1$
  else
    $x_i \leftarrow 1$

$\mathcal{THRESH}^-$, introduced and named by Lewin, Livnat and Zwick [LLZ02], is a small but powerful class of rounding algorithms. They also proposed using a distribution of

$\mathcal{THRESH}^-$ rounding schemes, and they called this larger family $\mathcal{THRESH}$. Intuitively, in a $\mathcal{THRESH}^-$ rounding scheme with threshold function f, the function f tells us how much confidence we have in the biases produced by the SDP (recall that b_i is intended to be $\mathbb{E}[x_i]$), so that the expected value of any output x_i is equal to $\mathbb{E}[f(b_i)]$. It also tries to maximize the correlation between the output values using a threshold rounding scheme.

Definition 3.7. For any configuration $\theta = ((b_i)_{i=1}^k, (b_{i,j})_{1\le i<j\le k}, P)$, let $\mathsf{Prob}(\theta, f)$ be the probability that the $\mathcal{THRESH}^-$ rounding scheme with threshold function f satisfies the constraint represented by θ. Let $\mathsf{Prob}(\Theta, f) = \mathbb{E}_{\theta\sim\Theta}[\mathsf{Prob}(\theta, f)]$.

Definition 3.8. Let $\theta = (b_i, b_j, b_{i,j}, P)$ be a 2-configuration. We define its relative pairwise bias to be $\rho(\theta) = \frac{b_{i,j}-b_i b_j}{\sqrt{(1-b_i^2)(1-b_j^2)}}$, if $(1-b_i^2)(1-b_j^2) \neq 0$, and $\rho(\theta) = 0$ otherwise. We say that θ is a positive configuration if $\hat{P}_{i,j} \cdot \rho(\theta) \ge 0$.

We are now ready to state Austrin's hardness result for 2-CSPs.

Theorem 3.9. *Let MAX CSP*(Γ) *be such that any predicate in* Γ *has arity at most 2. Let* Θ *be a distribution of configurations for MAX CSP*(Γ) *such that any 2-configuration in the support of* Θ *is a positive configuration. Let* $c = \mathrm{SDP}(\Theta), s = \sup_f \mathsf{Prob}(\Theta, f)$*, then it is NP-hard to approximate MAX CSP*(Γ) *within a factor of* $s/c + \epsilon$ *for any* $\epsilon > 0$*, assuming UGC.*

Theorem 3.9 is a slight extension of Theorem 5.1 in [Aus10]. Austrin in his work only considered the case where the MAX CSP is defined by one single predicate with arity 2, and negating variables is allowed in the instances. Here we state the result for general MAX CSPs with arity at most 2. For completeness, we include the proof, although the proof is only a minor modification from that in [Aus10]. Intuitively, when all configurations are positive, the rounding scheme should try to maximize the correlation between variables, and this is achieved by $\mathcal{THRESH}^-$ due to Theorem 2.15.

All existing hardness constructions for MAX 2-CSPs use only positive configurations. In fact, Austrin made the following conjecture:

Conjecture 2 (Positivity Conjecture, [Aus10])**.** For all 2-CSPs, the hardest distribution of configurations consists of only positive configurations.

It would be very interesting to resolve this conjecture, since a proof for it would imply the optimality of $\mathcal{THRESH}$. We note that in the search for hardest distributions it is easy to rule out distributions consisting of only negative configurations, since for them a uniform random assignment will do very well. The tricky part of the conjecture is to rule out the case where we have a distribution that contains both positive and negative configurations. That said, we do not need to rely on this conjecture to establish hardness result for a given 2-CSP.

3.3.1 Proof of Theorem 3.9

The following proof is taken and slightly modified from Appendix A in [BHPZ23], which is in turn a small modification from that in [Aus10].

For any permutation $\pi : [L] \to [L]$ and vector $\mathbf{x} = (x_1, \ldots, x_L) \in \mathbb{R}^L$, let $\pi\mathbf{x}$ be the vector $(x_{\pi(1)}, \ldots, x_{\pi(L)})$. Given a distribution of configurations Θ for MAX CSP(Γ), consider the PCP protocol $\mathsf{Verifier}_\Theta(I, F)$ shown in Algorithm 3 (c.f., Algorithm 1 in [Aus10]).

Lemma 3.10 (Completeness, c.f., Lemma 5.2 of [Aus10])**.** *If* $\mathrm{Val}(I) \geq 1 - \eta$*, then there exists* F *such that* $\mathsf{Verifier}_\Theta(I, F)$ *accepts with probability at least* $(1 - 2\eta) \cdot \mathrm{SDP}(\Theta)$*.*

Proof. Since $\mathrm{Val}(I) \geq 1-\eta$, there exists an assignment A such that $\mathrm{Val}(I, A) \geq 1-\eta$. For any $v \in V(G)$, let $f_v : \{-1, 1\}^L \to \{-1, 1\}$ be the dictatorship function $(x_1, x_2, \ldots, x_L) \mapsto x_{A(v)}$, and let $F = \{f_v \mid v \in V(G)\}$. If θ has arity 1, then we have

$$\mu = f_u(\pi_e^u\mathbf{x}) = (\pi_e^u\mathbf{x})_{A(u)} = x_{\pi_e^u(A(u)))} = x_{A(v)},$$

Algorithm 3 PCP protocol $\mathsf{Verifier}_\Theta(I, F)$

Input: A Unique Games instance $I = (G, L, \Pi)$, and a set of functions $F = \{f_v : \{-1,1\}^L \to \{-1,1\} \mid v \in V(G)\}$
Output: Accept or Reject
Sample $v \sim V(G)$ ▷ probability proportional to the weights in G
Sample $\theta \sim \Theta$.
if θ has arity 2 **then**
  $(b_1, b_2, b_{12}, P) \leftarrow \theta$
  Sample two edges $e_1 = \{v, u_1\}$, $e_2 = \{v, u_2\}$ incident to v
  for $i \leftarrow 1$ to L **do** ▷ Independently for each i
    Sample $x_i^{(1)}, x_i^{(2)} \sim \{-1,1\}$ such that $\mathbb{E}[x_i^{(1)}] = b_1, \mathbb{E}[x_i^{(2)}] = b_2, \mathbb{E}[x_i^{(1)} x_i^{(2)}] = \rho(\theta)$
  $\mathbf{x}^{(1)} \leftarrow (x_1^{(1)}, x_2^{(1)}, \ldots, x_L^{(1)})$, $\mathbf{x}^{(2)} \leftarrow (x_1^{(2)}, x_2^{(2)}, \ldots, x_L^{(2)})$
  $\mu_1 \leftarrow f_{u_1}(\pi_{e_1}^{u_1} \mathbf{x}^{(1)})$, $\mu_2 \leftarrow f_{u_2}(\pi_{e_2}^{u_2} \mathbf{x}^{(2)})$
  Accept if $P(\mu_1, \mu_2)$, and reject otherwise.
else ▷ θ has arity 1
  $(b, P) \leftarrow \theta$
  Sample an edge $e = \{v, u\}$ incident to v
  for $i \leftarrow 1$ to L **do** ▷ Independently for each i
    Sample $x_i \sim \{-1,1\}$ such that $\mathbb{E}[x_i] = b$
  $\mathbf{x} \leftarrow (x_1, x_2, \ldots, x_L)$, $\mu \leftarrow f_u(\pi_e^u \mathbf{x})$
  Accept if $P(\mu)$, and reject otherwise.

and it follows that

$$\begin{aligned}
&\Pr[\mathsf{Verifier}_\Theta(I, F) \text{ accepts} \mid \theta \text{ has arity } 1] \\
\geq\ & \Pr[A \text{ satisfies } e] \cdot \Pr[\mathsf{Verifier}_\Theta(I, F) \text{ accepts} \mid A \text{ satisfies } e, \text{ and } \theta \text{ has arity } 1] \\
\geq\ & (1 - \eta) \cdot \mathop{\mathbb{E}}_{\theta \sim \Theta} \left[P\left(x_{A(v)} \right) \mid \theta \text{ has arity } 1 \right] \\
\geq\ & (1 - \eta) \cdot \mathop{\mathbb{E}}_{\theta \sim \Theta} \left[P\left(b \right) \mid \theta \text{ has arity } 1 \right] \\
=\ & (1 - \eta) \cdot \mathop{\mathbb{E}}_{\theta \sim \Theta} \left[\mathrm{SDP}(\theta) \mid \theta \text{ has arity } 1 \right].
\end{aligned}$$

Here we used the fact that $\Pr[A \text{ satisfies } e] \geq 1 - \eta$ since $\mathrm{Val}(I, A) \geq 1 - \eta$.

If θ has arity 2, then by similar computations we have $\mu_1 = x^{(1)}_{A(v)}$ and $\mu_2 = x^{(2)}_{A(v)}$, and

$$
\begin{aligned}
&\Pr[\mathsf{Verifier}_\Theta(I,F) \text{ accepts} \mid \theta \text{ has arity } 2] \\
\geq\; &\Pr[A \text{ satisfies } e_1, e_2] \cdot \Pr[\mathsf{Verifier}_\Theta(I,F) \text{ accepts} \mid A \text{ satisfies } e_1, e_2, \text{ and } \theta \text{ has arity } 2] \\
\geq\; &(1-2\eta) \cdot \mathop{\mathbb{E}}_{\theta\sim\Theta} \left[P\left(x^{(1)}_{A(v)}, x^{(2)}_{A(v)} \right) \mid \theta \text{ has arity } 2 \right] \\
\geq\; &(1-2\eta) \cdot \mathop{\mathbb{E}}_{\theta\sim\Theta} \left[P\left(b_1, b_2 \right) \mid \theta \text{ has arity } 2 \right] \\
=\; &(1-2\eta) \cdot \mathop{\mathbb{E}}_{\theta\sim\Theta} \left[\mathrm{SDP}(\theta) \mid \theta \text{ has arity } 2 \right].
\end{aligned}
$$

The lemma now follows from the Law of Total Expectation. $\square$

Lemma 3.11 (Soundness, c.f., Lemma 5.3 of [Aus10])**.** *For any $\epsilon > 0$ there exists $\gamma > 0$ such that, if $\mathrm{Val}(I) \leq \gamma$, then for any F, $\mathsf{Verifier}_\Theta(I,F)$ accepts with probability at most $\sup_h \mathsf{Prob}(\Theta, h) + \epsilon$.*

Proof. Fix some $\epsilon > 0$. We need to find some $\gamma > 0$ with the following property: if there exists $F = \{f_v \mid v \in V(G)\}$ such that $\mathsf{Verifier}_\Theta(I,F)$ accepts with probability greater than $\sup_h \mathsf{Prob}(\Theta, h) + \epsilon$, then $\mathrm{Val}(I) > \gamma$. Assume the existence of such F, it suffices to show that $\mathrm{Val}(I)$ is lower-bounded by some constant only depending on ϵ.

For $v \in V(G)$ and $b \in (-1,1)$, we define $g_v^b : B^L_{(1-b)/2} \to [-1,1]$ as

$$
g_v^b(\mathbf{x}) = \mathop{\mathbb{E}}_{e=\{v,u\}\in E(G)} [f_u(\pi_e^u \mathbf{x})].
$$

Notice that the family of functions $\{g_v^b\}$ naturally lead to the family of thresholds $h_v(b) := \Phi^{-1}\left(\frac{1+\mathbb{E}_{\mathbf{x}}[g_v^b(\mathbf{x})]}{2}\right)$, under which a variable with bias b has expected value $\mathbb{E}_{\mathbf{x}}[g_v^b(\mathbf{x})]$ after rounding. For each v, this gives us a $\mathcal{THRESH}^-$ rounding scheme. We have

$$\begin{aligned}\sup_h \mathsf{Prob}(\Theta, h) + \epsilon &\geq \mathop{\mathbb{E}}_v[\mathsf{Prob}(\Theta, h_v)] + \epsilon \\ &= \mathop{\mathbb{E}}_{\theta\sim\Theta,v}[\mathsf{Prob}(\theta, h_v)] + \epsilon.\end{aligned}$$

On the other hand, we can express the accepting probability of the verifier as $\mathbb{E}_{\theta\sim\Theta,v,\boldsymbol{\mu}}[P(\boldsymbol{\mu})]$ where P is the predicate type of θ and we used $\boldsymbol{\mu}$ to denote that there may be 1 or 2 inputs to P. By our assumption, we have

$$\mathop{\mathbb{E}}_{\theta\sim\Theta,v,\boldsymbol{\mu}}[P(\boldsymbol{\mu})] \geq \mathop{\mathbb{E}}_{\theta\sim\Theta,v}[\mathsf{Prob}(\theta, h_v)] + \epsilon.$$

It follows that there exists θ_0 with predicate type Q in the support of Θ such that

$$\mathop{\mathbb{E}}_{v,\boldsymbol{\mu}}[Q(\boldsymbol{\mu})] \geq \mathop{\mathbb{E}}_v[\mathsf{Prob}(\theta_0, h_v)] + \epsilon.$$

Note that for any θ with arity 1, we have

$$\mathop{\mathbb{E}}_{v,\mu}[P(\mu)] = \mathop{\mathbb{E}}_{v,u,\mathbf{x}}[P(f_u(\pi_e^u\mathbf{x}))] = \mathop{\mathbb{E}}_{v,\mathbf{x}}\left[\hat{P}_\varnothing + \hat{P}_1 \cdot g_v^b(\mathbf{x})\right] = \mathop{\mathbb{E}}_v[\mathsf{Prob}(\theta, h_v)],$$

so θ_0 must be some 2-configuration $(b_1, b_2, b_{1,2}, Q)$. We then have, by a similar computation,

$$\begin{aligned}\mathop{\mathbb{E}}_{v,\mu_1,\mu_2}[Q(\mu_1,\mu_2)] &= \mathop{\mathbb{E}}_{v,\mathbf{x}_1,\mathbf{x}_2}\left[\hat{Q}_\varnothing + \hat{Q}_1 \cdot g_v^{b_1}(\mathbf{x}_1) + \hat{Q}_2 \cdot g_v^{b_2}(\mathbf{x}_2) + \hat{Q}_{1,2} \cdot g_v^{b_1}(\mathbf{x}_1) g_v^{b_2}(\mathbf{x}_2)\right] \\ &= \mathop{\mathbb{E}}_v\left[\hat{Q}_\varnothing + \hat{Q}_1 \cdot \mathop{\mathbb{E}}_{\mathbf{x}_1}[g_v^{b_1}(\mathbf{x}_1)] + \hat{Q}_2 \cdot \mathop{\mathbb{E}}_{\mathbf{x}_2}[g_v^{b_2}(\mathbf{x}_2)] + \hat{Q}_{1,2} \cdot \mathbb{S}_{\rho(\theta)}(g_v^{b_1}, g_v^{b_2})\right].\end{aligned}$$

We also have

$$\mathop{\mathbb{E}}_v[\mathsf{Prob}(\theta_0, h_v)] = \mathop{\mathbb{E}}_v\left[\hat{Q}_\varnothing + \hat{Q}_1 \cdot \mathop{\mathbb{E}}_{\mathbf{x}_1}[g_v^{b_1}(\mathbf{x}_1)] + \hat{Q}_2 \cdot \mathop{\mathbb{E}}_{\mathbf{x}_2}[g_v^{b_2}(\mathbf{x}_2)] + \hat{Q}_{1,2} \cdot \mathbb{T}_{\rho(\theta)}(h_v(b_1), h_v(b_2))\right].$$

This and the previous equation together imply that

$$\mathop{\mathbb{E}}_{v}\left[\hat{Q}_{1,2}\cdot\mathbb{S}_{\rho(\theta)}(g_v^{b_1},g_v^{b_2})\right]\geq\mathop{\mathbb{E}}_{v}\left[\hat{Q}_{1,2}\cdot\mathbb{T}_{\rho(\theta)}(h_v(b_1),h_v(b_2))\right]+\epsilon.$$

Since both $\mathbb{S}_{\rho(\theta)}(g_v^{b_1},g_v^{b_2})$ and $\mathbb{T}_{\rho(\theta)}(h_v(b_1),h_v(b_2))$ are bounded by some absolute constant, we can find $C>0$ such that for at least an ϵ fraction of $v\in V(G)$, we have

$$\hat{Q}_{1,2}\cdot\mathbb{S}_{\rho(\theta)}(g_v^{b_1},g_v^{b_2})-\hat{Q}_{1,2}\cdot\mathbb{T}_{\rho(\theta)}(h_v(b_1),h_v(b_2))\geq\frac{\epsilon}{C}.$$

Let V_0 be the set of $v\in V(G)$ that satisfy the above inequality. Since the 2-configurations in Θ are all positive, we have $\hat{Q}_{1,2}\cdot\rho(\theta)\geq 0$. Since $\hat{Q}_{1,2}\neq 0$, we have either $\hat{Q}_{1,2}>0$ and

$$\mathbb{S}_{\rho(\theta)}(g_v^{b_1},g_v^{b_2})\geq\mathbb{T}_{|\rho(\theta)|}(h_v(b_1),h_v(b_2))+\frac{\epsilon}{C\cdot\hat{Q}_{1,2}}$$

or $\hat{Q}_{1,2}<0$ and

$$\mathbb{S}_{\rho(\theta)}(g_v^{b_1},g_v^{b_2})\leq-\mathbb{T}_{|\rho(\theta)|}(h_v(b_1),h_v(b_2))-\frac{\epsilon}{C\cdot|\hat{Q}_{1,2}|}.$$

In either case, by Corollary 2.16, there exist $\eta>0$ and $k\in\mathbb{N}$ such that, for every $v\in V_0$ there is some $i\in[n]$ with

$$\mathrm{Inf}_i^{\leq k}[g_v^{b_1}]\geq\min(\mathrm{Inf}_i^{\leq k}[g_v^{b_1}],\mathrm{Inf}_i^{\leq k}[g_v^{b_2}])\geq\eta.$$

Since $\mathrm{Inf}_i^{\leq k}$ is convex, we also have

$$\begin{aligned}\eta\leq\mathrm{Inf}_i^{\leq k}[g_v^{b_1}]&=\mathrm{Inf}_i^{\leq k}\left[\mathop{\mathbb{E}}_{e=\{v,u\}\in E(G)}[f_u\circ\pi_e^u]\right]\\&\leq\mathop{\mathbb{E}}_{e=\{v,u\}\in E(G)}\left[\mathrm{Inf}_i^{\leq k}[f_u\circ\pi_e^u]\right].\end{aligned}$$

Since $\mathrm{Inf}_i^{\le k}$ takes value in $[0,1]$, there is an $\eta/2$ fraction of $u \sim v$ such that

$$\mathrm{Inf}_i^{\le k}[f_u \circ \pi_e^u] = \mathrm{Inf}_{(\pi_e^u)^{-1}(i)}^{\le k}[f_u] \ge \eta/2.$$

Now let $L_1(v) = \{i \in [n] \mid \mathrm{Inf}_i^{\le k}[g_v^{b_1}] \ge \eta\}$ and $L_2(v) = \{i \in [n] \mid \mathrm{Inf}_i^{\le k}[f_v^{b_1}] \ge \eta/2\}$. By Proposition 2.14, we have $|L_1(v)| \le \frac{k}{\eta}$ and $|L_2(v)| \le \frac{2k}{\eta}$, and by union bound $|L_1(v) \cup L_2(v)| \le \frac{3k}{\eta}$.

Now consider the following labeling strategy for I: for every $v \in V(G)$, if $L_1(v) \cup L_2(v)$ is non-empty, then choose a label $A(v) \in L_1(v) \cup L_2(v)$ uniformly at random, otherwise choose $A(v) \in [R]$ uniformly at random. By our analysis above, if we choose an edge $e = (u, v)$ with $v \in V_0$, then there is at least $\epsilon \cdot \eta/2$ probability such that there is some $i \in L_1(v)$ with $\pi_e^v(i) \in L_2(u)$, which our strategy will then find with probability at least $1/(3k/\eta)^2$, so $\mathrm{Val}(I, A)$ is at least $\epsilon \cdot \eta/2 \cdot 1/(3k/\eta)^2$, which is a constant only depending on ϵ, and the lemma is proven. $\square$

CHAPTER 4

TIGHT INAPPROXIMABILITY RESULTS FOR MAX 2-SAT AND ITS SUBPROBLEMS

In this chapter, we study the approximability of MAX 2-SAT and its subproblems. We will a classification of these subproblems in terms of their approximability by giving tight approximability/inapproximability results for each of them. This chapter is based on [BHZ24], which appeared in SODA'24.

4.1 Overview

We summarize the results in the following table, which is taken from [BHZ24].

Name	$x \vee y$	$\bar{x} \vee y$	$\bar{x} \vee \bar{y}$	x	$\bar{x}$	Approximation Ratio
MAX 2-SAT	✓	❍	✓	❍	❍	≈ 0.94016567
MAX HORN-2-SAT	✓	✓	✗	❍	✓	≈ 0.94615981
MAX CSP($\{x \vee y, x, \bar{x}\}$)	✓	✗	✗	❍	✓	≈ 0.95397990
MAX CSP($\{\bar{x} \vee y, x, \bar{x}\}$)	✗	❍	✗	❍	❍	1
MAX CSP($\{x \vee y, \bar{x} \vee y, x\}$)	❍	❍	✗	❍	✗	1

Table 4.1: The approximation ratios of MAX 2-SAT and its subproblems, assuming UGC. Table taken from [BHZ24].

In this table, ✓ indicates that the type of constraints is allowed, ✗ indicates that the type of constraints is not allowed, and ❍ indicates that whether allowing the type of constraints or not does not change the approximation ratio of the problem since it does not appear in the hardness construction.

The first subproblem is MAX 2-SAT itself. The hardness result for the MAX 2-SAT ratio provided in the table was proved by Austrin [Aus07]. Austrin's proof only uses constraint types $x \vee y$ and $\bar{x} \vee \bar{y}$, so any restriction that contains these two types is automatically as hard as MAX 2-SAT itself. On the other hand, Lewin, Livnat and Zwick gave an algorithm for MAX 2-SAT that has a conjectured performance matching this ratio [LLZ02]. In [BHZ24], we proved that the LLZ algorithm is indeed tight, thereby establishing tight approximability result for MAX 2-SAT.

Moving on to other subproblems, we disallow the constraint type $\bar{x} \vee \bar{y}$ so that Austrin's hardness proof does not apply. The next two problems, MAX HORN-2-SAT and MAX CSP($\{x \vee y, x, \bar{x}\}$), cover the case where we do not allow $\bar{x} \vee \bar{y}$, but allow $x \vee y$ and $\bar{x}$ (as it turns out, for these two cases constraints of type x do not affect the approximation ratio). Their approximability hasn't been studied previously in the literature. We prove tight approximability results for them by giving matching algorithmic and hardness results, showing that the approximation ratios slightly improve from that of MAX 2-SAT. Roughly speaking, our strategy is first giving a conjectured optimal algorithm for the problem, and then using the hardest configurations for the algorithm to construct a distribution that is hard for any $\mathcal{THRESH}$ rounding scheme. This combined with Theorem 3.9 establishes UG-hardness and confirms the optimality of the algorithm that we started with.

The last two of these subproblems, where we disallow the constraint type $\bar{x} \vee \bar{y}$ and in addition disallow either $x \vee y$ or $\bar{x}$, can be solved exactly in polynomial time. For MAX CSP($\{x \vee y, \bar{x} \vee y, x\}$), we can assign true to every variable, and this satisfies all constraints since there is at least one positive literal in every constraint. MAX CSP($\{\bar{x} \vee y, x, \bar{x}\}$) can be solved exactly using a reduction to the Minimum s-t Cut problem. Indeed, given an instance Φ of MAX CSP($\{\bar{x} \vee y, x, \bar{x}\}$), we can construct a directed graph G whose vertex set is the variable set of Φ plus two new vertices s, t, and add edges (s, x) for every constraint x, (x, t) for every constraint $\bar{x}$, and (x, y) for every constraint $\bar{x} \vee y$. If we think of variables connected to s as those assigned true, and variables connected to t as false, then it is easy to see that any truth assignment is in 1-1 correspondence to s-t cuts in G, and the constraints violated by an assignment are exactly the edges cut by the corresponding s-t cut.

4.2 MAX 2-SAT and Simplicity Conjecture

We first take a look at MAX 2-SAT itself. Lewin, Livnat and Zwick proposed using a linear threshold function f in the $\mathcal{THRESH}^-$ rounding scheme (see Algorithm 2) for MAX 2-SAT [LLZ02], namely, for some parameter $\beta \in [0,1]$, we have $f : b \mapsto \beta b$. To make the parameter explicit, we will denote this function by f_β. Given a 2-configuration $\theta = (b_i, b_j, b_{i,j})$ whose predicate type is $x \vee y$, we have

$$\mathrm{SDP}(\theta) = \mathrm{SDP}(b_i, b_j, b_{i,j}) = \frac{3 - b_i - b_j - b_{i,j}}{4},$$

and since the probability that both variables in this configuration are set to false $(+1)$ is equal to $\Phi_{\rho(\theta)}\left(\Phi^{-1}\left(\frac{1+\beta b_i}{2}\right), \Phi^{-1}\left(\frac{1+\beta b_j}{2}\right)\right)$,

$$\mathsf{Prob}(\theta, f_\beta) = 1 - \Phi_{\rho(\theta)}\left(\Phi^{-1}\left(\frac{1+\beta b_i}{2}\right), \Phi^{-1}\left(\frac{1+\beta b_j}{2}\right)\right).$$

Note that here the quadratic coefficient in the Fourier expansion is equal to $-1/4$, and therefore θ is a positive configuration if and only if $\rho(\theta) \le 0$. The approximation ratio achieved by $\mathcal{THRESH}^-$ with f_β is then equal to

$$\inf_{\theta:\mathrm{SDP}(\theta)\neq 0} \frac{\mathsf{Prob}(\theta, f_\beta)}{\mathrm{SDP}(\theta)}. \tag{4.1}$$

We wish to find β that maximizes this expression. However, this is a highly nontrivial task because finding the minimizing θ for (4.1) which depends on three parameters is not easy. Austrin proposed that we reduce the number of parameters by focusing on *simple configurations* instead.

Definition 4.1. A 2-configuration $(b_i, b_j, b_{i,j})$ is called simple, if $b_i = b_j$ and $b_{i,j} = -1+2|b_i|$.

Note that the condition $b_{i,j} = -1 + 2|b_i|$ ensures that a simple configuration in on the boundary of one of the triangle inequalities. It can be easily checked that for a simple configuration $\theta = (b, b, -1+2|b|)$, we have $\rho(\theta) = \frac{-(1-|b|)^2}{1-b^2} \leq 0$, so simple configurations are all positive configurations for 2-SAT. Austrin showed the following result.

Theorem 4.2 ([Aus07]). *There exists $\beta^-_{LLZ} \approx 0.94016567$ such that*

$$\inf_{\theta=(b,b,-1+2|b|)} \frac{\mathsf{Prob}(\theta, f_{\beta^-_{LLZ}})}{\mathrm{SDP}(\theta)} = \max_{\beta} \inf_{\theta=(b,b,-1+2|b|)} \frac{\mathsf{Prob}(\theta, f_\beta)}{\mathrm{SDP}(\theta)} = \beta^-_{LLZ}.$$

This theorem shows that, surprisingly, the best ratio that can be achieved on simple configurations by $\mathcal{THRESH}^-$ with f_β is achieved when β is chosen to be this ratio itself! Austrin proved Theorem 4.2 by looking at a distribution over two simple configurations $\theta_1 = (-b_0, -b_0, -1+2b_0)$ and $\theta_2 = (b_0, b_0, -1+2b_0)$, where $b_0 \approx 0.169$. He showed that on these two configurations, the optimal choice for β is equal to β^-_{LLZ}. More specifically, for all odd f,

$$\min\left\{\frac{\mathsf{Prob}(\theta_1, f)}{\mathrm{SDP}(\theta_1)}, \frac{\mathsf{Prob}(\theta_2, f)}{\mathrm{SDP}(\theta_2)}\right\} \leq \beta^-_{LLZ},$$

where the equality is achieved by taking $f = f_{\beta^-_{LLZ}}$. He then showed that among all simple configurations, θ_1 and θ_2 are the two minimizers for the ratio $\mathsf{Prob}(\theta, f_{\beta^-_{LLZ}})/\mathrm{SDP}(\theta)$. Since simple configurations are all positive configurations for MAX 2-SAT, by Theorem 3.9, Austrin's argument immediately implies the following theorem.

Theorem 4.3 ([Aus07]). *Assuming UGC, for any $\epsilon > 0$, it is NP-hard to approximate MAX 2-SAT with an approximation ratio of $\beta^-_{LLZ} + \epsilon$.*

So far there remains the possibility that for $\mathcal{THRESH}^-$ with $f_{\beta^-_{LLZ}}$, there exist even harder configurations that are not simple, in which case we can hope to obtain a hardness result with an even lower ratio. However, numerical experiments in [LLZ02] suggest that this is not the case, and that θ_1 and θ_2 are very likely the hardest configurations for $f_{\beta^-_{LLZ}}$[1]. Based on this, Austrin conjectured that $\beta^-_{LLZ} = \beta_{LLZ}$, where β_{LLZ} is the ratio obtained by the optimal $\mathcal{THRESH}^-$ algorithm on *all*, not necessarily simple, configurations, and hence on all instances of MAX 2-SAT (this conjecture is also implicit in [LLZ02]).

Our main result for MAX 2-SAT is a proof for this conjecture. More specifically, we show the following theorem.

Theorem 4.4 ([BHZ24])**.** *Let* $g_\beta(b_i, b_j, b_{i,j}) = \mathsf{Prob}((b_i, b_j, b_{i,j}), f_\beta) - \beta \cdot \mathrm{SDP}(b_i, b_j, b_{i,j})$. *We have* $\min_{(b_i,b_j,b_{i,j})} g_{\beta^-_{LLZ}}(b_i, b_j, b_{i,j}) = 0$*, where* $(b_i, b_j, b_{i,j})$ *ranges over all feasible configurations.*

In words, this confirms the numerical evidence that θ_1 and θ_2 are indeed the hardest configurations for $f_{\beta^-_{LLZ}}$. The proof of Theorem 4.4 uses both analytical tools and computer-assisted tools. Here we give a brief outline of how this theorem is obtained, and refer interested readers to [BHZ24] for more detail. Our proof consists of the following steps.

- We first show that any minimizer of $g_{\beta^-_{LLZ}}(b_i, b_j, b_{i,j})$ is of the form $(b_i, b_j, -1 + |b_i + b_j|)$. This means that any minimizer must be on the boundary of one of the triangle inequalities. This step is obtained by certifying that any point in the interior of the feasible region either has $g_{\beta^-_{LLZ}} > 0$ or a nonzero gradient for $g_{\beta^-_{LLZ}}$ using interval arithmetic.

- We then restrict our attention to the boundary of triangle inequalities. We show that if any feasible configuration is on the boundary, but still far away from the two configurations θ_1 and θ_2, then it cannot be a minimizer. This step is similarly obtained using interval arithmetic.

1. We remark here that in [LLZ02] a slightly different parametrization for the $\mathcal{THRESH}^-$ rounding family was used, and therefore it wasn't observed that the optimal ratio is also the optimal paramter for β.

- Finally, we show that for any bias b near b_0 or $-b_0$, the function

$$h_{b,\beta^-_{LLZ}}(t) := \mathsf{Prob}_{\beta^-_{LLZ}}(b+t, b-t, -1+2|b|)$$

  achieves its minimum at $t = 0$ in a large enough neighborhood of 0. This means that the minimizer must be a simple configuration.

The first two steps were obtained using interval arithmetic, whereas the last step was proven analytically.

Theorem 4.3 and Theorem 4.4 together settle the approximation ratio for MAX 2-SAT, modulo UGC. We can summarize the strategy as following:

- Identify the optimal rounding scheme. For MAX 2-SAT, this is the LLZ algorithm: $\mathcal{THRESH}^-$ scheme with $f_{\beta^-_{LLZ}}$. We don't need a proof of optimality for this step, as it will be confirmed later.

- Find the hardest configurations for the optimal rounding scheme. We DO require a proof that the configuration are hardest in this step. In other words, we need to certify the approximation ratio (but not optimality) for the candidate optimal rounding scheme. The corresponds to proving Theorem 4.4 for MAX 2-SAT.

- Using the configurations from the last step to construct a distribution that is hard against all $\mathcal{THRESH}^-$ schemes. The corresponds to proving Theorem 4.2 for MAX 2-SAT. This gives a matching hardness result to the ratio certified in the previous step, thereby establishing optimality of the candidate rounding scheme and pinpoint the approximation ratio for the problem.

In the following sections, we will carry out this plan for other subproblems of MAX 2-SAT.

4.3 MAX CSP($\{x \vee y, x, \bar{x}\}$)

This section is mostly taken from Section 4 in [BHZ24].

4.3.1 The Rounding Algorithm

Let us consider the $\mathcal{THRESH}^-$ rounding scheme with $f : b \mapsto -1 + \gamma(1 + b)$ for some parameter $\gamma \in [0, 1]$. For any $\bar{x}$ constraint with bias b, we have that its SDP value is equal to $\frac{1+x}{2}$, and $b \mapsto -1 + \gamma(1 + b)$ satisfies it (assigns false to x) with probability $\Phi\left(\Phi^{-1}\left(\frac{1-1+\gamma(1+b)}{2}\right)\right) = \frac{\gamma(1+b)}{2}$. For any x constraint with bias b, its SDP value is $\frac{1-x}{2}$ and $b \mapsto -1 + \gamma(1 + b)$ satisfies it with probability

$$1 - \Phi\left(\Phi^{-1}\left(\frac{1-1+\gamma(1+b)}{2}\right)\right) = 1 - \frac{\gamma(1+b)}{2} \geq \frac{\gamma(1-b)}{2},$$

where the last inequality is because $\gamma \in [0, 1]$. This shows that the parameter γ is also the approximation ratio that the rounding scheme achieves on unary constraints. Note that the function $-1+\gamma(1+b)$ is increasing in γ for every b, which means we are less likely to satisfy any $x \vee y$ constraints if we increase γ. To optimize γ, it is then sufficient to find $\gamma = \gamma^*$ such that $b \mapsto -1 + \gamma^*(1 + b)$ achieves an approximation ratio of also γ^* on 2-configurations.

Similar to g_β, we define $h_\gamma(b_i, b_j, b_{i,j}) = \left(1 - \Phi_\rho\left(\Phi^{-1}\left(\frac{\gamma(1+b_i)}{2}\right), \Phi^{-1}\left(\frac{\gamma(1+b_j)}{2}\right)\right)\right) - \gamma \cdot \mathrm{SDP}(b_i, b_j, b_{i,j})$ where $\rho = \rho(b_i, b_j, b_{i,j})$.

Proposition 4.5. *For every feasible configuration* $(b_i, b_j, b_{i,j})$, $h_\gamma(b_i, b_j, b_{i,j})$ *monotonically decreases with* γ. *In particular* $\min_{(b_i,b_j,b_{i,j})} h_\gamma(b_i, b_j, b_{i,j})$ *decreases with* γ, *where the minimum is taken over all feasible configurations. Furthermore* $\min_{(b_i,b_j,b_{i,j})} h_{\gamma^*}(b_i, b_j, b_{i,j}) = 0$.

Proof. $h_\gamma(b_i, b_j, b_{i,j})$ monotonically decreases since $\Phi_\rho\left(\Phi^{-1}\left(\frac{\gamma(1+b_i)}{2}\right), \Phi^{-1}\left(\frac{\gamma(1+b_j)}{2}\right)\right)$ and $\gamma \cdot \mathrm{SDP}(b_i, b_j, b_{i,j})$ are both increasing in γ. Note that $\min h_\gamma(b_i, b_j, b_{i,j}) \geq 0$ implies that $-1 + \gamma(1 + b)$ achieves an approximation ratio of at least γ, and therefore for the optimal $\gamma = \gamma*$ the equality must be achieved. $\square$

The following theorem is proved with computer assistance, with a plan similar to the Simplicity Conjecture that we discussed earlier.

Theorem 4.6. *We have* $\gamma^* \in [0.9539798, 0.95398]$. *Furthermore, the minimum in the expression* $\min_{\theta=(b_i,b_j,b_{i,j})} h_{\gamma^*}(b_i, b_j, b_{i,j})$ *is achieved at some point* $(b, b, -1-2b)$ *for some* $b \in [b_0 - \epsilon, b_0 + \epsilon]$ *where* $b_0 = -0.1824167935$ *and* $\epsilon = 10^{-6}$.

4.3.2 Matching Hardness

Let $b^* = b(\gamma^*) \approx -0.1824$ be the hardest bias for γ^* from Theorem 4.6 and $p_1, p_2 \in [0, 1]$ be some parameters to be chosen later. Consider the following distribution Θ_1 of configurations.

Configuration	Probability	Predicate type
$\theta_1 = (b^*, b^*, -1-2b^*)$	p_1	$x \vee y$
$\theta_2 = (b^*)$	p_2	$\bar{x}$

Table 4.2: The hardest distribution of configurations for MAX CSP$(\{x \vee y, x, \bar{x}\})$

We will prove that Θ_1 is hard against all $\mathcal{THRESH}^-$ rounding schemes $b \mapsto f(b)$. Since there is only one bias involved, it is sufficient to consider the threshold for that bias. Let $\rho = \rho(\theta_1) = -\frac{1+b^*}{1-b^*}$. Recall that SDP$(\Theta_1)$ denotes the SDP value of this distribution, and slightly abusing the notation, let $\mathsf{Prob}(\Theta_1, t)$ be the probability of satisfying a configuration sampled from Θ_1 if $f(b^*) = 2\Phi(t) - 1$. With this parametrization, we have that the $\mathcal{THRESH}^-$ rounding scheme with f sets any variable with bias b^* to false with probability $\frac{1+f(b^*)}{2} = \Phi(t)$ and to true with probability $(1 - \Phi(t))$.

Proposition 4.7. *We have* $\mathrm{SDP}(\Theta_1) = p_1 + p_2 \cdot \frac{1+b^*}{2}$ *and* $\mathsf{Prob}(\Theta_1, t) = p_1 \cdot (1 - \Phi_\rho(t, t)) + p_2 \cdot \Phi(t)$.

Proof. For the first configuration in Θ_1, we have that its SDP value is 1, and its satisfied by

f unless both variables are set to false, which happens with probability $\Phi_\rho(t,t)$. The second configuration has SDP value $\frac{1+b^*}{2}$ and is satisfied with probability $\Phi(t)$. $\square$

It is straightforward to find the best threshold t using calculus. We have

Proposition 4.8. *Let $t^* = \sqrt{\frac{1+\rho}{1-\rho}} \cdot \Phi^{-1}\left(\frac{p_2}{p_1}\right)$. For every $t \in \mathbb{R} \cup \{\pm\infty\}$ we have*

$$\mathsf{Prob}(\Theta_1, t) \leq \mathsf{Prob}(\Theta_1, t^*) \ .$$

Proof. Using Lemma 2.7, we have

$$\begin{aligned}\frac{\partial}{\partial t}\mathsf{Prob}(\Theta_1, t) &= -p_1 \cdot \varphi(t) \cdot \Phi\left(\sqrt{\frac{1-\rho}{1+\rho}} \cdot t\right) + p_2 \cdot \varphi(t) \\ &= \varphi(t) \cdot \left(-p_1 \cdot \Phi\left(\sqrt{\frac{1-\rho}{1+\rho}} \cdot t\right) + p_2\right) \ .\end{aligned}$$

The proposition follows since $\frac{\partial}{\partial t}\mathsf{Prob}(\Theta_1, t) > 0$ when $t \leq t_*$ and $\frac{\partial}{\partial t}\mathsf{Prob}(\Theta_1, t) < 0$ when $t \geq t_*$. $\square$

Theorem 4.9. *For every $\epsilon > 0$, it is UG-hard to approximate MAX CSP($\{x \vee y, x, \bar{x}\}$ within a ratio of $\gamma^* + \epsilon$. Moreover, there exists a $\mathcal{THRESH}^-$ rounding scheme that achieves an approximation ratio of γ^* for MAX CSP($\{x \vee y, x, \bar{x}\}$.*

Proof. Let us take $\frac{p_2}{p_1} = \Phi\left(\sqrt{\frac{1-\rho}{1+\rho}} \cdot \Phi^{-1}\left(\frac{\gamma^*(1+b^*)}{2}\right)\right)$. Then the value of t^* in Proposition 4.8 will be

$$\sqrt{\frac{1+\rho}{1-\rho}} \cdot \Phi^{-1}\left(\frac{p_2}{p_1}\right) = \Phi^{-1}\left(\frac{\gamma^*(1+b^*)}{2}\right),$$

and this coincides with the value given by $b \mapsto -1 + \gamma^*(1+b)$ at $b = b^*$. This shows that for such p_1 and p_2, $b \mapsto -1 + \gamma^*(1+b)$ is an optimal $\mathcal{THRESH}^-$ scheme on Θ_1. Since we already know that $b \mapsto -1 + \gamma^*(1+b)$ has approximation ratio γ^* on both configurations in this distribution, it follows that any $\mathcal{THRESH}^-$ scheme has approximation ratio at most γ^*, and Theorem 3.9 immediately implies the UG-hardness. The algorithmic result is essentially a restatement of Theorem 4.6. □

4.4 MAX HORN-2-SAT

This section is mostly taken from Section 5 in [BHZ24].

4.4.1 The Rounding Algorithm

We consider the following $\mathcal{THRESH}$ rounding scheme F_α:

$$\begin{aligned} b &\mapsto b && \text{with probability } \alpha, \\ b &\mapsto -1 && \text{with probability } 1-\alpha. \end{aligned}$$

In other words, with some probability α we round with the odd threshold function $b \mapsto b$, and with the remaining probability $1-\alpha$ we set every variable to true. We need to find α that maximizes the approximation ratio of F_α. We have the following property for the optimal α.

Proposition 4.10. *Assume that the approximation ratio of F_α is maximized when $\alpha = \alpha^*$. Then the approximation ratio of F_{α^*} is also equal to α^*.*

Proof. Observe that any 1-configuration with bias b and predicate type $\bar{x}$ has SDP value $\frac{1+b}{2}$, while the function $b \mapsto b$ satisfies it with probability $\frac{1+b}{2}$ as well. This implies that $b \mapsto b$ achieves an approximation ratio of 1 on all $\bar{x}$ constraints. On the other hand, if we set every variable to true, then we never satisfy any $\bar{x}$ constraint. Therefore, on the 1-configurations, F_α has an overall approximation ratio α for every α.

Now, note that by setting every variable to true we satisfy all constraints of the forms $x \vee y$ and $\bar{x} \vee y$, so by decreasing α we increase the approximation ratio on the 2-configurations and vice versa. This means that for the optimal α, F_α must achieve the same approximation ratio on both 1-configurations and 2-configurations, otherwise we can adjust α to increase the approximation ratio. $\square$

For $\theta = (b_i, b_j, b_{i,j})$, we define

$$\mathsf{Prob}(\theta) = \mathsf{Prob}(b_i, b_j, b_{i,j}) = 1 - \Phi_{\rho(\theta)}\left(\Phi^{-1}\left(\frac{1+b_i}{2}\right), \Phi^{-1}\left(\frac{1+b_j}{2}\right)\right)$$

and $g(b_i, b_j, b_{i,j}) = \mathsf{Prob}(b_i, b_j, b_{i,j}) - \mathrm{SDP}(b_i, b_j, b_{i,j})$. The following lemma gives an expression for α^*.

Lemma 4.11. *α^* satisfies the following equality:*

$$1 - \frac{1}{\alpha^*} = \min_{\theta=(b_i,b_j,b_{i,j})} g(b_i, b_j, b_{i,j}),$$

where θ ranges over all feasible 2-configurations.

Proof. The probability that F_{α^*} satisfies any 2-configuration $(b_i, b_j, b_{i,j})$ is given by

$$1 - \alpha^* + \alpha^* \cdot \mathsf{Prob}(b_i, b_j, b_{i,j}) \ ,$$

where $1 - \alpha^*$ is contributed by the function $b \mapsto -1$ and $\alpha^* \cdot \mathsf{Prob}(b_i, b_j, b_{i,j})$ is contributed by $b \mapsto b$. Since F_{α^*} achieves an approximation ratio of α^*, we have

$$1 - \alpha^* + \alpha^* \cdot \mathsf{Prob}_1(b_i, b_j, b_{i,j}) \geq \alpha^* \cdot \mathrm{SDP}(b_i, b_j, b_{i,j}) \ .$$

Rearranging, we obtain that

$$1 - \frac{1}{\alpha^*} \leq \mathsf{Prob}(b_i, b_j, b_{i,j}) - \mathrm{SDP}(b_1, b_2, b_{12}) \ .$$

Since this is true for every feasible configuration and equality is achieved on some configuration, we obtain that

$$1 - \frac{1}{\alpha^*} = \min_{\theta=(b_i,b_j,b_{i,j})} \mathsf{Prob}(b_i, b_j, b_{i,j}) - \mathrm{SDP}(b_i, b_j, b_{i,j}). \qquad \square$$

We use interval arithmetic to prove the following theorem.

Theorem 4.12. *The minimum in the expression* $\min_{\theta=(b_1,b_2,b_{12})} f(b_1, b_2, b_{12})$ *is achieved at some point* $(b^*, b^*, -1 + 2|b^*|)$ *for some* b^* *with* $|b^*| \in [b_0 - \epsilon, b_0 + \epsilon]$ *where* $b_0 = 0.1489442$ *and* $\epsilon = 10^{-6}$.

4.4.2 Matching Hardness

Let $b^* > 0$ be a hardest bias on which f achieves its minimum as in Theorem 4.12. Using b^*, we construct the following distribution Θ_2.

Configuration	Probability	Predicate type
$\theta_1 = (-b^*, -b^*, -1 + 2b^*)$	p_1	$x \vee y$
$\theta_2 = (b^*, b^*, -1 + 2b^*)$	p_2	$x \vee y$
$\theta_3 = (-b^*, b^*, 1 - 2b^*)$	p_3	$\bar{x} \vee y$
$\theta_4 = (b^*, -b^*, 1 - 2b^*)$	p_4	$\bar{x} \vee y$
$\theta_5 = (-b^*)$	p_5	$\bar{x}$
$\theta_6 = (b^*)$	p_6	$\bar{x}$

Table 4.3: The hardest distribution of configurations for MAX HORN-2-SAT

Let $\rho = \rho(b^*) = -\frac{1-b^*}{1+b^*}$ be the relative pairwise bias of the configuration $(-b^*, -b^*, -1 + 2b^*)$.

Similar to the definitions in Section 4.3.2, let $\mathrm{SDP}(\Theta_2)$ be the SDP value of this distribution and $\mathsf{Prob}(\Theta_2, t_1, t_2)$ be the probability of a $\mathcal{THRESH}^-$ scheme f satisfying a configuration sampled from Θ_2 if $f(-b^*) = 2\Phi(t_1) - 1$ and $f(b^*) = 2\Phi(t_2) - 1$.

Proposition 4.13. *We have*

$$\mathrm{SDP}(\Theta_2) = (p_1 + p_4) + (p_2 + p_3) \cdot (1 - b^*) + p_5 \cdot \frac{1 - b^*}{2} + p_6 \cdot \frac{1 + b^*}{2}$$

and

$$\begin{aligned}\mathsf{Prob}(\Theta_2, t_1, t_2) = p_1 \cdot (1 - \Phi_\rho(t_1, t_1)) + p_2 \cdot (1 - \Phi_\rho(t_2, t_2)) + p_3 \cdot (1 - \Phi_\rho(-t_1, t_2)) \\ + p_4 \cdot (1 - \Phi_\rho(-t_2, t_1)) + p_5 \cdot \Phi(t_2) + p_6 \cdot \Phi(t_1).\end{aligned}$$

We also have the following partial derivatives for $\mathsf{Prob}(\Theta_2, t_1, t_2)$.

Proposition 4.14. *We have*

$$\begin{aligned}&\frac{\partial}{\partial t_1}\mathsf{Prob}(\Theta_2, t_1, t_2) \\ &= -2p_1\varphi(t_1)\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_1\right) + p_3\varphi(t_1)\Phi\left(\frac{t_2 + \rho t_1}{\sqrt{1-\rho^2}}\right) - p_4\varphi(t_1)\Phi\left(\frac{-t_2 - \rho t_1}{\sqrt{1-\rho^2}}\right) + p_6\varphi(t_1) \\ &= \varphi(t_1) \cdot \left(-2p_1\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_1\right) + p_3\Phi\left(\frac{t_2 + \rho t_1}{\sqrt{1-\rho^2}}\right) - p_4\Phi\left(\frac{-t_2 - \rho t_1}{\sqrt{1-\rho^2}}\right) + p_6\right)\end{aligned}$$

and

$$\begin{aligned}&\frac{\partial}{\partial t_2}\mathsf{Prob}(\Theta_2, t_1, t_2) \\ &= -2p_2\varphi(t_2)\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_2\right) - p_3\varphi(t_2)\Phi\left(\frac{-t_1 - \rho t_2}{\sqrt{1-\rho^2}}\right) + p_4\varphi(t_2)\Phi\left(\frac{t_1 + \rho t_2}{\sqrt{1-\rho^2}}\right) + p_5\varphi(t_2) \\ &= \varphi(t_2) \cdot \left(-2p_2\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_2\right) - p_3\Phi\left(\frac{-t_1 - \rho t_2}{\sqrt{1-\rho^2}}\right) + p_4\Phi\left(\frac{t_1 + \rho t_2}{\sqrt{1-\rho^2}}\right) + p_5\right)\end{aligned}$$

We have the following second derivatives for $\mathsf{Prob}(\Theta_2, t_1, t_2)$.

Proposition 4.15. *We have*

$$\frac{\partial^2}{\partial t_1^2}\mathsf{Prob}(\Theta_2, t_1, t_2)$$
$$= -t_1\varphi(t_1)\cdot\left(-2p_1\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_1\right)+p_3\Phi\left(\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}}\right)-p_4\Phi\left(\frac{-t_2-\rho t_1}{\sqrt{1-\rho^2}}\right)+p_6\right)$$
$$+\varphi(t_1)\cdot\left(-2p_1\sqrt{\frac{1-\rho}{1+\rho}}\cdot\varphi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_1\right)+(p_3+p_4)\cdot\frac{\rho}{\sqrt{1-\rho^2}}\cdot\varphi\left(\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}}\right)\right),$$

and

$$\frac{\partial^2}{\partial t_2^2}\mathsf{Prob}(\Theta_2, t_1, t_2)$$
$$= -t_2\varphi(t_2)\cdot\left(-2p_2\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_2\right)-p_3\Phi\left(\frac{-t_1-\rho t_2}{\sqrt{1-\rho^2}}\right)+p_4\Phi\left(\frac{t_1+\rho t_2}{\sqrt{1-\rho^2}}\right)+p_5\right)$$
$$+\varphi(t_2)\cdot\left(-2p_2\sqrt{\frac{1-\rho}{1+\rho}}\cdot\varphi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_2\right)+(p_3+p_4)\cdot\frac{\rho}{\sqrt{1-\rho^2}}\cdot\varphi\left(\frac{t_1+\rho t_2}{\sqrt{1-\rho^2}}\right)\right),$$

and

$$\begin{aligned}\frac{\partial^2}{\partial t_1\partial t_2}\mathsf{Prob}(\Theta_2, t_1, t_2) &= (p_3+p_4)\cdot\varphi_\rho(t_1,-t_2)\\ &= (p_3+p_4)\cdot\frac{1}{\sqrt{1-\rho^2}}\cdot\varphi(t_1)\varphi\left(\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}}\right).\end{aligned}$$

We would like to find the probabilities $p_1,\dots,p_6$ that minimizes the maximum ratio achieved by any $\mathcal{THRESH}^-$ scheme $\max_{t_1,t_2}\mathsf{Prob}(\Theta_2,t_1,t_2)/\mathrm{SDP}(\Theta_2)$. To do this, we will first heuristically derive a set of probabilities assuming $t_1=-t_2$, and then verify that for these probabilities $\mathsf{Prob}(\Theta_2,t_1,t_2)$ is indeed maximized at a point where $t_1=-t_2$.

For $\mathsf{Prob}(\Theta_2,t,-t)$, we have

$$\mathsf{Prob}(\Theta_2,t,-t) = (p_1+p_4)\cdot(1-\Phi_\rho(t,t))+(p_2+p_3)\cdot(1-\Phi_\rho(-t,-t))+p_5\Phi(-t)+p_6\Phi(t).$$

We will choose $p_5 = p_6 = p$, which intuitively makes sense as F_{α^*} achieves the same ratio α^* on all 1-configurations. Under this assumption we have

$$\mathsf{Prob}(\Theta_2, t, -t) = (p_1 + p_4) \cdot (1 - \Phi_\rho(t,t)) + (p_2 + p_3) \cdot (1 - \Phi_\rho(-t,-t)) + p,$$

and

$$\frac{\partial}{\partial t}\mathsf{Prob}(\Theta_2, t, -t) = (p_1+p_4)\cdot\left(-2\varphi(t)\cdot\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t\right)\right)+(p_2+p_3)\cdot\left(2\varphi(t)\cdot\Phi\left(-\sqrt{\frac{1-\rho}{1+\rho}}t\right)\right)$$

Following the same strategy we used in Section 4.3.2, we want the above to attain 0 at $t = t^* = \Phi^{-1}((1-b^*)/2)$. This implies that

$$\frac{p_2 + p_3}{p_1 + p_2 + p_3 + p_4} = \Phi\left(\sqrt{\frac{1-\rho}{1+\rho}} \cdot t^*\right) := r.$$

This gives us the ratio between the probabilities of the 2-configurations, we can then choose p so that the two $\mathcal{THRESH}^-$ schemes $b \mapsto b$ and $b \mapsto 0$ in F_{α^*} achieves the same satisfying probability on Θ_2, i.e.,

$$\begin{aligned}\mathsf{Prob}(\Theta_2, t^*, -t^*) &= (p_1 + p_4) \cdot (1 - \Phi_\rho(t^*, t^*)) + (p_2 + p_3) \cdot (1 - \Phi_\rho(-t^*, -t^*)) + p \\ &= p_1 + p_2 + p_3 + p_4.\end{aligned}$$

This implies that

$$\frac{p}{p_1 + p_2 + p_3 + p_4} = 1 - (1-r) \cdot (1 - \Phi_\rho(t^*, t^*)) - r \cdot (1 - \Phi_\rho(-t^*, -t^*)) := r'.$$

Since we also have $p_1 + p_2 + p_3 + p_4 + 2p = 1$, the above gives

$$p = \frac{2r'}{1+2r'}, \quad p_1 + p_4 = (1-r) \cdot (1-2p), \quad p_2 + p_3 = r \cdot (1-2p)\ .$$

Finally, by setting the partial derivatives

$$\frac{\partial}{\partial t_1}\mathsf{Prob}(\Theta_2, t^*, -t^*) = \frac{\partial}{\partial t_2}\mathsf{Prob}(\Theta_2, t^*, -t^*) = 0,$$

we obtain that $p_1 = p_2 = p$ as well. In summary, we obtain the following probabilities:

$$p_1 = p_2 = p_5 = p_6 = p = \frac{2r'}{1+2r'}, \quad p_3 = r \cdot (1-2p) - p, \quad p_4 = (1-r)\cdot(1-2p) - p.$$

The numeric values for these probabilities are listed as follows:

$$p_1 = p_2 = p_5 = p_6 \approx 0.0858, \quad p_3 \approx 0.1737, \quad p_4 \approx 0.4831.$$

We remark that since we have chosen the hardest bias b^*, the approximation ratio achieved by F_{α^*} on this distribution is exactly α^*. In fact, by design, both functions in F_{α^*} achieve exactly $\alpha^* = \frac{\mathsf{Prob}(\Theta_2,-\infty,-\infty)}{\mathrm{SDP}(\Theta_2)} = \frac{\mathsf{Prob}(\Theta_2,t^*,-t^*)}{\mathrm{SDP}(\Theta_2)}$.

Now we prove that with the probabilities computed in the previous section, $(t^*, -t^*) \in \mathbb{R}^2$ is indeed a global maximum for $\mathsf{Prob}(\Theta_2, t_1, t_2)$. To give a better sense of the function that we are working with, we give a contour plot of $\mathsf{Prob}(\Theta_2, t_1, t_2)$ in Figure 4.1.

It can be seen that aside from $(t^*, -t^*)$, there are two other critical points which are saddle points. This creates complications for an analytic proof. We will circumvent this difficulty by employing interval arithmetic. We first prove the following statement with interval arithmetic.

Lemma 4.16 (Interval arithmetic)**.** *For every $t_1, t_2 \in \mathbb{R}$, at least one of the following is true:*

- $\mathsf{Prob}(\Theta_2, t_1, t_2) < \mathsf{Prob}(\Theta_2, t^*, -t^*)$.
- $t_1, t_2 \le \Phi^{-1}(0.0001)$ *or* $t_1, t_2 \ge \Phi^{-1}(0.9999)$.

- $|t_1 - t^*|, |t_2 + t^*| < 0.01$ *and the Hessian matrix for* $\mathsf{Prob}(\Theta_2, t_1, t_2)$ *is negative definite.*

- $\frac{\partial}{\partial t_1}\mathsf{Prob}(\Theta_2, t_1, t_2) \neq 0$ *or* $\frac{\partial}{\partial t_2}\mathsf{Prob}(\Theta_2, t_1, t_2) \neq 0$

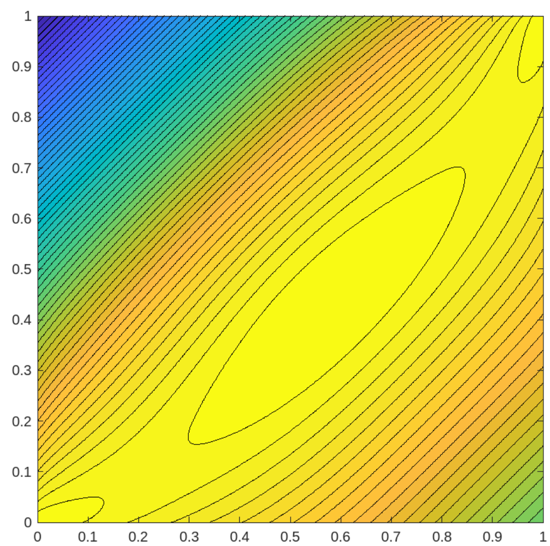

Figure 4.1: A contour plot of $\mathsf{Prob}(\Theta_2, t_1, t_2)$, where the x-axis is $\Phi(t_1)$ and the y-axis is $\Phi(t_2)$

Since the gradient of $\mathsf{Prob}(\Theta_2, t_1, t_2)$ vanishes at $(t^*, -t^*)$, the third item shows that $\mathsf{Prob}(\Theta_2, t_1, t_2) \leq \mathsf{Prob}(\Theta_2, t^*, -t^*)$ for every $t_1 \in [t^* - 0.01, t^* + 0.01], t_2 \in [-t^* - 0.01, -t^* + 0.01]$.

The following proposition deals with the boundary situation that our interval arithmetic does not certify directly.

Proposition 4.17. *Let* $t_1, t_2 \in \mathbb{R}^2$ *be such that* $t_1, t_2 \leq \Phi^{-1}(0.0001)$ *or* $t_1, t_2 \geq \Phi^{-1}(0.9999)$, *then we have* $\frac{\partial}{\partial t_1}\mathsf{Prob}(\Theta_2, t_1, t_2) \neq 0$ *or* $\frac{\partial}{\partial t_2}\mathsf{Prob}(\Theta_2, t_1, t_2) \neq 0$.

Proof. Assume that $t_1, t_2 \leq \Phi^{-1}(0.0001)$.

Since $\varphi(t_1), \varphi(t_2) > 0$, the partial derivatives being 0 is equivalent to

$$-2p\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_1\right) + p_3\Phi\left(\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}}\right) - p_4\Phi\left(\frac{-t_2-\rho t_1}{\sqrt{1-\rho^2}}\right) + p = 0,$$
$$-2p\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_2\right) - p_3\Phi\left(\frac{-t_1-\rho t_2}{\sqrt{1-\rho^2}}\right) + p_4\Phi\left(\frac{t_1+\rho t_2}{\sqrt{1-\rho^2}}\right) + p = 0.$$

Using the fact that $\Phi(x) = 1 - \Phi(-x)$, we can rewrite the above as

$$-2p\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_1\right) + (p_3+p_4)\Phi\left(\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}}\right) - p_4 + p = 0,$$
$$-2p\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_2\right) + (p_3+p_4)\Phi\left(\frac{t_1+\rho t_2}{\sqrt{1-\rho^2}}\right) - p_3 + p = 0.$$

Since

$$\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}} + \frac{t_1+\rho t_2}{\sqrt{1-\rho^2}} = \frac{(t_1+t_2)\sqrt{1+\rho}}{\sqrt{1-\rho}},$$

we have either

$$\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}} \leq \frac{(t_1+t_2)\sqrt{1+\rho}}{2\sqrt{1-\rho}}$$

or

$$\frac{t_1+\rho t_2}{\sqrt{1-\rho^2}} \leq \frac{(t_1+t_2)\sqrt{1+\rho}}{2\sqrt{1-\rho}}.$$

A simple estimation shows that we would then have either

$$\Phi\left(\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}}\right) \leq \Phi\left(\frac{(t_1+t_2)\sqrt{1+\rho}}{2\sqrt{1-\rho}}\right) < 0.12$$

or

$$\Phi\left(\frac{t_1+\rho t_2}{\sqrt{1-\rho^2}}\right) \leq \Phi\left(\frac{(t_1+t_2)\sqrt{1+\rho}}{2\sqrt{1-\rho}}\right) < 0.12.$$

But in either case we would violate at least one of the equations, since we'd have either

$$-2p\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_1\right)+(p_3+p_4)\Phi\left(\frac{t_2+\rho t_1}{\sqrt{1-\rho^2}}\right)-p_4+p<(p_3+p_4)\cdot 0.12-p_4+p<0$$

or

$$-2p\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t_2\right)+(p_3+p_4)\Phi\left(\frac{t_1+\rho t_2}{\sqrt{1-\rho^2}}\right)-p_3+p<(p_3+p_4)\cdot 0.12-p_3+p<0.$$

This shows that at least one of the partial derivatives is non-zero. The case where $t_1, t_2 \geq \Phi^{-1}(0.9999)$ can be dealt with similarly. □

Proposition 4.18. *For every $t_1, t_2 \in \mathbb{R} \cup \{\pm\infty\}$, we have*

$$\mathsf{Prob}(\Theta_2, t_1, t_2) \leq \mathsf{Prob}(\Theta_2, t^*, -t^*).$$

Proof. We first consider the infinity cases. If $t_1 = -\infty$, then we have

$$\begin{aligned}
\mathsf{Prob}(\Theta_2, t_1, t_2) &= p_1\cdot(1-\Phi_\rho(t_1,t_1)) + p_2\cdot(1-\Phi_\rho(t_2,t_2)) + p_3\cdot(1-\Phi_\rho(-t_1,t_2)) \\
&\quad + p_4\cdot(1-\Phi_\rho(-t_2,t_1)) + p_5\cdot\Phi(t_2) + p_6\cdot\Phi(t_1) \\
&= p_1 + p_2\cdot(1-\Phi_\rho(t_2,t_2)) + p_3\cdot(1-\Phi(t_2)) + p_4 + p_5\cdot\Phi(t_2).
\end{aligned}$$

Since $p_3 > p_5$, the $\mathsf{Prob}(\Theta_2, -\infty, t_2)$ is monotonically decreasing in t_2, so we should choose $t_2 = -\infty$ as well. A similar analysis shows that if $t_1 = +\infty$, then we should also set $t_2 = +\infty$, and furthermore $\mathsf{Prob}(\Theta_2, -\infty, -\infty) = \mathsf{Prob}(\Theta_2, \infty, \infty)$.

Now assume that there is a global maximum (t_1, t_2) with

$$\mathsf{Prob}(\Theta_2, t_1, t_2) > \mathsf{Prob}(\Theta_2, t^*, -t^*) = \mathsf{Prob}(\Theta_2, -\infty, -\infty).$$

Since $\mathsf{Prob}(\Theta_2, t_1, t_2)$ is a smooth function, the gradient vanishes at the global maximum, so by Lemma 4.16 and Proposition 4.17 we must have $|t_1 - t^*|, |t_2 + t^*| < 0.001$. However, the negative definiteness of the Hessian matrix in that neighborhood would then imply that $\mathsf{Prob}(\Theta_2, t_1, t_2) \leq \mathsf{Prob}(\Theta_2, t^*, -t^*)$. This contradiction shows that there is no global maximum strictly larger than $\mathsf{Prob}(\Theta_2, t^*, -t^*)$, and therefore $(t^*, -t^*)$ itself must be a global maximum. $\square$

The above analysis combined with Theorem 3.9 immediately implies the following theorem.

Theorem 4.19. *For every $\epsilon > 0$, it is UG-hard to approximate MAX HORN-2-SAT within a factor of $\alpha^* + \epsilon$.*

CHAPTER 5

SEPARATING MAX DI-CUT FROM MAX CUT AND MAX 2-AND

In this chapter, we study the MAX DI-CUT problem, which can be thought of a variant of MAX CUT on directed graphs. While the approximability of MAX CUT is well understood, the same cannot be said for MAX DI-CUT. Via a simple reduction, it can be shown that MAX DI-CUT is at least as hard to approximate as MAX CUT, but strict separation between the two problems was not known. Moreover, MAX 2-AND, being a generalization of MAX DI-CUT for reasons we will see in a moment, is at least as hard as MAX DI-CUT, but here strict separation was also unknown. The main result of this chapter is the following theorem, which shows that there is strict separation in both cases if we assume UGC.

Theorem 5.1. *Let $\alpha_{\textit{2-AND}}, \alpha_{\textit{DI-CUT}}, \alpha_{\textit{CUT}}$ be the approximation ratios of MAX 2-AND, MAX DI-CUT and MAX CUT respectively. Assuming UGC, we have*

$$\alpha_{\textit{2-AND}} < \alpha_{\textit{DI-CUT}} < \alpha_{\textit{CUT}}.$$

This chapter is based on [BHPZ23] which appeared in FOCS'23.

5.1 Three Problems: MAX CUT, MAX DI-CUT and MAX 2-AND

Recall the following definitions from Section 2.1. We have predicates $\mathrm{CUT}, \mathrm{DI\text{-}CUT} : \{-1, 1\}^2 \to \{0, 1\}$ where CUT is satisfied if and only if the two inputs are not equal and DI-CUT is satisfied if and only if the first input is 1 and the second input -1. MAX CUT is defined by MAX CSP($\{$CUT$\}$), MAX DI-CUT is defined by MAX CSP($\{$DI-CUT$\}$). Note that if we think of 1 as false and -1 and true, then the DI-CUT predicate can be expressed as $\mathrm{DI\text{-}CUT}(x, y) = \bar{x} \wedge y$. This means that by allowing variable negations we can obtain 2-AND constraints, and therefore MAX 2-AND can be defined as MAX CSP(cl({DI-CUT})). We first show the following folklore proposition which explains why we are comparing them.

Proposition 5.2. *Let $\alpha_{2\text{-}AND}, \alpha_{DI\text{-}CUT}, \alpha_{CUT}$ be the approximation ratios of MAX 2-AND, MAX DI-CUT and MAX CUT respectively. We have*

$$\alpha_{2\text{-}AND} \leq \alpha_{DI\text{-}CUT} \leq \alpha_{CUT}.$$

Proof. It is clear that $\alpha_{\text{2-AND}} \leq \alpha_{\text{DI-CUT}}$ since MAX DI-CUT is a subproblem of MAX 2-AND. To show that $\alpha_{\text{DI-CUT}} \leq \alpha_{\text{CUT}}$, we exhibit an approximation ratio preserving reduction (f, g) from MAX CUT to MAX DI-CUT as defined in Definition 2.18. Given a MAX CUT instance I, which is an undirected graph, f simply replaces each undirected edge with two opposing directed edges, each having the same weight as the undirected edge. g is the identity function, that is, given a solution A to $f(I)$ which is a cut, $g(A) = A$ is the same cut on the original graph. It is easy to see that $\text{Val}(I, A) = \text{Val}(f(I), A)$ for any assignment A, so all conditions for an approximation ratio preserving reduction are satisfied. By Lemma 2.19, we have $\alpha_{\text{DI-CUT}} \leq \alpha_{\text{CUT}}$. □

Given Proposition 5.2, it is then natural to ask whether these two inequalities are strict. We do know that at least one of them is strict assuming UGC: for MAX CUT, we have $\alpha_{\text{CUT}} \geq 0.878$ due to Goemans and Williamson [GW95] as mentioned earlier, and for MAX 2-AND, Austrin showed that we have $\alpha_{\text{2-AND}} \leq 0.87435$ assuming UGC [Aus10]. This puts $\alpha_{\text{DI-CUT}}$ in a narrow interval $[0.87435, 0.878]$. However, we do not know towards which end of the interval $\alpha_{\text{DI-CUT}}$ lies. The best hardness result for MAX DI-CUT is inherited from MAX CUT, and the best algorithm for MAX DI-CUT, due to Lewin, Livnat and Zwick [LLZ02], achieves an approximation ratio[1] of $\approx 0.874 < 0.87435$. This means that prior to our work it could still be the case that equality might be achieved in one of the inequalities in

1. This result is numerical and the claimed ratio is not rigorous.

Proposition 5.2.

In the following two sections, we will present both new algorithm and hardness results for MAX DI-CUT. Before we proceed, let us record the following Fourier expansion of DI-CUT which will be heavily used throughout this chapter.

Proposition 5.3. DI-CUT$(x, y) = \frac{1+x-y-xy}{4}$.

Since the quadratic coefficient in the Fourier expansion is $-1/4$, a DI-CUT configuration is positive if and only if its relative pairwise bias is not positive.

We also note that we have the following symmetry for the MAX DI-CUT problem: given a directed graph, if we flip the direction of every edge in the graph, then an optimal solution to this new instance can be obtained by flipping all the signs in an optimal solution to the original instance. For configurations, this symmetry corresponds to swapping the two biases and then changing the signs.

Definition 5.4 (Flipping a configuration)**.** Let $\theta = (b_i, b_j, b_{ij})$ be a DI-CUT configuration. We define its *flip* to be $\text{flip}(\theta) = (-b_j, -b_i, b_{ij})$.

The following proposition can be easily verified.

Proposition 5.5. *Let* $\theta = (b_i, b_j, b_{ij})$ *be a* DI-CUT *configuration. We have*

1. $\rho(\theta) = \rho(\text{flip}(\theta))$.

2. $\text{SDP}(\theta) = \text{SDP}(\text{flip}(\theta))$.

5.2 Separating MAX DI-CUT and MAX CUT

In this section, we prove the following theorem, which improves the upper bound on the approximation ratio of MAX DI-CUT and thereby separates MAX DI-CUT from MAX CUT. The proofs presented in this section are taken from Section 3 in [BHPZ23].

Theorem 5.6. *Assuming the Unique Games Conjecture, it is NP-hard to approximate MAX DI-CUT within a factor of* 0.87461.

To prove Theorem 5.6 we construct a distribution of positive configurations Θ, compute its completeness, and show that no $\mathcal{THRESH}^-$ rounding scheme can achieve an approximation ratio of 0.87461 on it. The UG-hardness result then follows from Theorem 3.9.

The distribution Θ used to obtain the upper bound is extremely simple. Let p_1, p_2, b, c be some parameters to be chosen later. We will choose them so that $b, p_1, p_2 \in (0, 1)$, $c \in (-1, -b^2)$, and $2p_1 + p_2 = 1$. Consider the following distribution of configurations Θ supported on $\{\theta_1, \theta_2, \theta_3\}$:

Configuration	Probability
$\theta_1 = (-b, -b, -1 + 2b)$	p_1
$\theta_2 = (\ \ b, -b, \quad c \quad)$	p_2
$\theta_3 = (\ \ b, \quad b, -1 + 2b)$	p_1

Table 5.1: A hard distribution of configurations for MAX DI-CUT

Note that in the θ_1 and θ_3 one of the triangle inequalities is tight, while in θ_2 none of the triangle inequalities are tight. This is in stark contrast to MAX 2-SAT and its subproblems: in Chapter 4, we have seen that for them the hardest configurations lie on the boundary of the triangle inequalities. We remark that the current best hardness construction for MAX 2-AND, given by Austrin [Aus10], is also supported on boundary inequalities.

We remark that this distribution is symmetric with respect to flip, since $\text{flip}(\theta_1) = \theta_3$ and $\text{flip}(\theta_2) = \theta_2$.

We first verify that Θ satisfies the positivity condition.

Proposition 5.7. Θ *is a distribution of positive configurations.*

Proof. In θ_1 and θ_3, the relative pairwise bias is equal to $\rho_1 = \frac{-1+2b-b^2}{1-b^2} = -\frac{1-b}{1+b} < 0$. In θ_2, the relative pairwise bias is equal to $\rho_2 = \frac{c+b^2}{1-b^2} < 0$ since we choose $c < -b^2$. $\square$

The completeness of this instance can be easily computed.

Proposition 5.8. $\mathrm{SDP}(\Theta) = p_1 \cdot (1-b) + p_2 \cdot \dfrac{1+2b-c}{4}.$

Proof. We have

$$\begin{aligned}
&\mathrm{SDP}(\Theta) \\
&= p_1 \cdot \frac{1+(-b)-(-b)-(-1+2b)}{4} + p_2 \cdot \frac{1+b-(-b)-c}{4} + p_1 \cdot \frac{1+b-b-(-1+2b)}{4} \\
&= p_1 \cdot \frac{2-2b}{4} + p_2 \cdot \frac{1+2b-c}{4} + p_1 \cdot \frac{2-2b}{4} \\
&= p_1 \cdot (1-b) + p_2 \cdot \frac{1+2b-c}{4}.
\end{aligned}$$

$\square$

We now give an upper bound on the performance of any $\mathcal{THRESH}^-$ rounding scheme on this distribution. Let t_1, t_2 be the thresholds for $-b, b$ respectively. Let $s(t_1, t_2)$ be the soundness of this rounding scheme on Θ. By definition of $\mathcal{THRESH}^-$, we have

$$s(t_1, t_2) = p_1 \cdot \Phi_{-\rho_1}(t_1, -t_1) + p_2 \cdot \Phi_{-\rho_2}(t_2, -t_1) + p_1 \cdot \Phi_{-\rho_1}(t_2, -t_2) \ ,$$

where $\rho_1, \rho_2 < 0$ are computed in Proposition 5.7. We first look at the case where $-\infty < t_1, t_2 < \infty$. The case where $t_1 = \pm\infty$ or $t_2 = \pm\infty$, which corresponds to always setting one or both variables to 1 or -1, can be dealt with separately via a simple case analysis.

As we discussed earlier, a $\mathcal{THRESH}^-$ rounding scheme for MAX DI-CUT is not necessarily odd, but as the following lemma shows, the simple and symmetric structure of our construction ensures that any finite critical point of s is necessarily symmetric around the origin.

Lemma 5.9. *Let $x, y \in \mathbb{R}$. If (x, y) is a critical point of $s(t_1, t_2)$, then $y = -x = |x|$.*

Proof. Recall that by Lemma 2.7 we have

$$\frac{\partial}{\partial t_1}\Phi_\rho(t_1,t_2) = \varphi(t_1)\cdot\Phi\left(\frac{t_2-\rho t_1}{\sqrt{1-\rho^2}}\right).$$

The partial derivatives of $s(t_1,t_2)$ are

$$\frac{\partial s}{\partial t_1} = p_1\left(\varphi(t_1) - 2\varphi(t_1)\cdot\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}t_1\right)\right) + p_2\left(-\varphi(t_1)\cdot\Phi\left(\frac{t_2-\rho_2 t_1}{\sqrt{1-\rho_2^2}}\right)\right),$$

$$\frac{\partial s}{\partial t_2} = p_1\left(\varphi(t_2) - 2\varphi(t_2)\cdot\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}t_2\right)\right) + p_2\left(\varphi(t_2) - \varphi(t_2)\cdot\Phi\left(\frac{t_1-\rho_2 t_2}{\sqrt{1-\rho_2^2}}\right)\right).$$

In the above computation, we used Lemma 2.7 and the chain rule. Since (x,y) is a critical point of s and φ is strictly positive, we have

$$p_1\left(1-2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}x\right)\right) + p_2\left(-\Phi\left(\frac{y-\rho_2 x}{\sqrt{1-\rho_2^2}}\right)\right) = 0\,,$$

$$p_1\left(1-2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}y\right)\right) + p_2\left(1-\Phi\left(\frac{x-\rho_2 y}{\sqrt{1-\rho_2^2}}\right)\right) = 0\,.$$

The first equation can be rewritten as

$$p_1\left(1-2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}x\right)\right) = p_2\cdot\Phi\left(\frac{y-\rho_2 x}{\sqrt{1-\rho_2^2}}\right)\,. \tag{5.1}$$

Since Φ is a positive function, the right hand side of (5.1) is positive and therefore we have $1-2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}x\right) > 0$, which implies that $x < 0$.

Since $1-\Phi(t) = \Phi(-t)$, the second equation can be rewritten as

$$p_1\left(1-2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}y\right)\right) = -p_2\cdot\Phi\left(\frac{-x+\rho_2 y}{\sqrt{1-\rho_2^2}}\right)\,. \tag{5.2}$$

By similar logic we can deduce that $y > 0$. We now show that we must have $|x| = |y|$. Assume for the sake of contradiction that $|x| \neq |y|$. We have two cases:

- $|x| > |y|$. It follows that

$$\begin{aligned} p_1 \cdot \left|1 - 2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}x\right)\right| &= p_1 \cdot \left|\Phi\left(-\sqrt{\frac{1-\rho_1}{1+\rho_1}}x\right) - \Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}x\right)\right| \\ &> p_1 \cdot \left|\Phi\left(-\sqrt{\frac{1-\rho_1}{1+\rho_1}}y\right) - \Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}y\right)\right| \\ &= p_1 \cdot \left|1 - 2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}}y\right)\right| . \end{aligned}$$

  Note that here we again used $1 - \Phi(t) = \Phi(-t)$, as well as the fact that $|\Phi(t) - \Phi(-t)|$ is an increasing function in $|t|$. On the other hand, by (5.1) and (5.2) this implies that

$$\left|p_2 \cdot \Phi\left(\frac{y - \rho_2 x}{\sqrt{1-\rho_2^2}}\right)\right| > \left|-p_2 \cdot \Phi\left(\frac{-x + \rho_2 y}{\sqrt{1-\rho_2^2}}\right)\right| .$$

  Since Φ is a positive and monotone function, this implies that

$$\frac{y - \rho_2 x}{\sqrt{1-\rho_2^2}} > \frac{-x + \rho_2 y}{\sqrt{1-\rho_2^2}} ,$$

  Rearranging the terms, we obtain

$$(1 - \rho_2)y > (1 - \rho_2) \cdot (-x) .$$

  But this would imply that $|y| > |x|$, which contradicts our assumption.

- $|y| > |x|$. This can be dealt with in a similar manner.

We conclude that we must have $y = -x = |x|$. □

Lemma 5.10. *If $p_1 > p_2$, then $s(t_1, t_2)$ has a unique critical point.*

Proof. Assume (x, y) is a critical point. In the previous lemma, we established that $x = -y < 0$, so we can now plug $y = -x$ into (5.1) and get

$$p_1 \left(1 - 2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot x\right)\right) = p_2 \cdot \Phi\left(\frac{-1-\rho_2}{\sqrt{1-\rho_2^2}} \cdot x\right) = p_2 \cdot \Phi\left(-\sqrt{\frac{1+\rho_2}{1-\rho_2}} \cdot x\right) .$$

We need to show the equation above has only one solution when $p_1 > p_2$. To this end, define

$$g(t) = p_1 \left(1 - 2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot t\right)\right) - p_2 \cdot \Phi\left(-\sqrt{\frac{1+\rho_2}{1-\rho_2}} \cdot t\right) , \qquad t \leq 0 .$$

We have $g(0) = -p_2/2 < 0$ and $\lim_{t \to -\infty} g(t) = p_1 - p_2 > 0$, so $g(t) = 0$ has at least one solution in $(-\infty, 0)$ by Intermediate Value Theorem. To show that the solution is unique, we compute the derivative of g:

$$g'(t) = p_1 \left(-2\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot \varphi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot t\right)\right) + p_2 \cdot \sqrt{\frac{1+\rho_2}{1-\rho_2}} \cdot \varphi\left(-\sqrt{\frac{1+\rho_2}{1-\rho_2}} \cdot t\right) .$$

By setting $g'(t) = 0$, we obtain

$$2p_1\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot \varphi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot t\right) = p_2 \cdot \sqrt{\frac{1+\rho_2}{1-\rho_2}} \cdot \varphi\left(-\sqrt{\frac{1+\rho_2}{1-\rho_2}} \cdot t\right)$$

Plugging in the definition of φ, we get

$$2p_1\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1-\rho_1}{1+\rho_1} \cdot \frac{t^2}{2}\right) = p_2 \cdot \sqrt{\frac{1+\rho_2}{1-\rho_2}} \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1+\rho_2}{1-\rho_2} \cdot \frac{t^2}{2}\right),$$

which is equivalent to

$$2p_1\sqrt{\frac{1-\rho_1}{1+\rho_1}} \cdot \exp\left(-\left(\frac{1-\rho_1}{1+\rho_1} - \frac{1+\rho_2}{1-\rho_2}\right) \cdot \frac{t^2}{2}\right) = p_2 \cdot \sqrt{\frac{1+\rho_2}{1-\rho_2}} \, .$$

Since $\rho_1, \rho_2 < 0$ and exp is monotone, this equation has exactly one solution $t^* \in (-\infty, 0)$. Furthermore, $g'(t) > 0$ for $t \in (-\infty, t^*)$ and $g'(t) < 0$ for $t \in (t^*, 0)$. It follows that g has no root in $(-\infty, t^*)$ and has a unique root in $(t^*, 0)$. □

We now deal with the boundary cases. Since our distribution is symmetric with respect to flip, it is sufficient to look at the case where $t_1 = \pm\infty$.

Lemma 5.11. *We have $s(+\infty, +\infty) = s(-\infty, -\infty) = s(+\infty, -\infty) = 0$, $s(-\infty, +\infty) = p_2$. For $t_2 \in \mathbb{R}$, we have $s(-\infty, t_2) > s(+\infty, t_2)$. Furthermore, if $p_1 > p_2$, then $s(-\infty, t_2)$ is maximized when $t_2 = t^* = \sqrt{\frac{1+\rho_1}{1-\rho_1}} \cdot \Phi^{-1}(\frac{p_1+p_2}{2p_1})$.*

Proof. Setting a threshold to $+\infty$ corresponds to always setting a variable to false, and $-\infty$ corresponds to always true. When $(t_1, t_2) \in \{(+\infty, +\infty), (-\infty, -\infty), (+\infty, -\infty)\}$, none of the configurations are satisfied, giving a soundness of 0. When $(t_1, t_2) = (-\infty, +\infty)$, only the second configuration is satisfied and this gives a soundness of p_2. For the second claim, we have

$$s(-\infty, t_2) = p_2 \cdot \Phi(t_2) + p_1 \cdot \Phi_{-\rho_1}(t_2, -t_2) > p_1 \cdot \Phi_{-\rho_1}(t_2, -t_2) = s(+\infty, t_2) \, ,$$

and

$$\frac{\partial s(-\infty, t_2)}{\partial t_2} = \varphi(t_2)\left(p_2 + p_1\left(1 - 2\Phi\left(\sqrt{\frac{1-\rho_1}{1+\rho_1}} t_2\right)\right)\right) \, .$$

When $p_1 > p_2$, we have $\frac{\partial s(-\infty, t_2)}{\partial t_2} > 0$ on $(-\infty, t^*)$ and $\frac{\partial s(-\infty, t_2)}{\partial t_2} < 0$ on (t^*, ∞). □

With Lemma 5.10 and Lemma 5.11, it becomes very easy to determine the maximum of s by simply computing the unique critical point and comparing it with the boundary cases. It turns out that when $b = 0.1757079776$, $c = -0.6876930116$, $p_1 = 0.3770580295$, the unique critical point of $s(t_1, t_2)$ is at $(-t_0, t_0)$ where $t_0 \approx 0.1887837358$, which is also a global maximum whose value is about 0.8746024732. It follows that with these parameters, any $\mathcal{THRESH}^-$ rounding scheme achieves a ratio of at most 0.87461.

We remark that by adding more biases to Θ, we seem to obtain an even harder distribution. However, analyzing this new distribution by hand seems impossible so we only have a numerical bound.

Configuration	Probability
$(b_2, b_1, -1 + b_1 + b_2)$	0.1907744673
$(-b_1, -b_2, -1 + b_1 + b_2)$	0.1907744673
$(b_2, b_2, -1 + 2b_2)$	0.1858539509
$(-b_2, -b_2, -1 + 2b_2)$	0.1858539509
$(b_1, -b_1, -0.6874089540)$	0.2371153723
$(b_1, -b_2, -0.6876719134)$	0.0048138957
$(b_2, -b_1, -0.6876719134)$	0.0048138957

Table 5.2: A distribution that uses two pairs of biases that seems to yield an upper bound $\alpha_{\text{DI-CUT}} \leq 0.8745896786$, where $b_1 = 0.1644279457$ and $b_2 = 0.1797733117$. (Not verified rigorously.) Table taken from [BHPZ23].

From the numerical experiments it seems that the hardness ratio of the distribution continues to improve if we add even more biases. Based on this observation we make the following conjecture.

Conjecture 3. The hardest distribution of configurations for MAX DI-CUT is supported on infinitely many configurations.

Austrin made a similar observation for MAX 2-AND in [Aus10]. It would be very interesting to understand why MAX DI-CUT and MAX 2-AND behave so differently from other

MAX 2-CSPs like MAX CUT and MAX 2-SAT, for which the hardest distributions take on very simple forms.

5.3 Separating MAX DI-CUT and MAX 2-AND

In this section, we present the following theorem. The materials in the section are taken from Section 4 in [BHPZ23].

Theorem 5.12. $\alpha_{DI\text{-}CUT} \geq 0.87447$.

Since $\alpha_{\text{2-AND}} \leq 0.87435$ (assuming UGC), this separates MAX DI-CUT from MAX 2-AND (again assuming UGC) and completes the other part of Theorem 5.1. Let us quickly discuss some intuition for why this separation can be expected. To prove that $\alpha_{\text{2-AND}} \leq 0.87435$, Austrin used a family of hard distributions, the simplest (and easiest) of which is of the following form:

Configuration	Probability
$(0, -b, b-1)$	0.64612
$(0, b, b-1)$	0.35388

Table 5.3: Austrin's hard distribution of configurations for MAX 2-AND [Aus10], $b = 0.33633$

Harder distributions in Austrin's construction can be obtained by adding more non-zero biases, but the bias 0 always appears in these distributions. This is due to an important restriction in MAX 2-AND: the threshold function used by any optimal $\mathcal{THRESH}$ rounding scheme for MAX 2-AND has to be odd, since the variables can be freely negated when we construct a MAX 2-AND instance. This restriction is absent from MAX DI-CUT, for which negated variables are not allowed. Indeed, for the distribution in Table 5.3, we can take a $\mathcal{THRESH}$ rounding scheme in which variables with bias 0 are always rounded to some fixed truth value, and this will give an approximation ratio better than 1!

	g_1	g_2	g_3	g_4	g_5	g_6	g_7
prob	0.996902	0.000956	0.000956	0.000393	0.000393	0.000200	0.000200
−1.000000	−1.601709	−2.000000	−2.000000	−0.034381	−0.430994	−2.000000	2.000000
−0.700000	−0.853605	−2.000000	−2.000000	−0.034381	−0.430994	−2.000000	2.000000
−0.450000	−0.517014	−2.000000	−0.629564	−0.440988	−0.896878	−2.000000	2.000000
−0.300000	−0.333109	−1.520523	1.711824	−1.406591	1.643936	−2.070000	1.970000
−0.250000	−0.274589	−0.687582	2.019266	−0.622399	−0.127984	−1.629055	2.070000
−0.179515	−0.192926	−0.195474	−0.229007	−0.268471	−0.339566	−0.544957	−0.103307
−0.164720	−0.175942	−0.381789	−0.649998	−0.116530	−0.073069	−0.361234	−0.575047
−0.100000	−0.105428	−0.026636	−1.175439	0.066139	−0.123693	2.070000	−1.351740
0.000000	0.000000	2.046025	−2.046025	1.728858	−1.728858	2.050000	−2.050000
0.100000	0.105428	1.175439	0.026636	0.123693	−0.066139	1.351740	−2.070000
0.164720	0.175942	0.649998	0.381789	0.073069	0.116530	0.575047	0.361234
0.179515	0.192926	0.229007	0.195474	0.339566	0.268471	0.103307	0.544957
0.250000	0.274589	−2.019266	0.687582	0.127984	0.622399	−2.070000	1.629055
0.300000	0.333109	−1.711824	1.520523	−1.643936	1.406591	−1.970000	2.070000
0.450000	0.517014	0.629564	2.000000	0.896878	0.440988	−2.000000	2.000000
0.700000	0.853605	2.000000	2.000000	0.430994	0.034381	−2.000000	2.000000
1.000000	1.601709	2.000000	2.000000	0.430994	0.034381	−2.000000	2.000000

Table 5.4: A $\mathcal{THRESH}$ rounding scheme $\mathcal{F}$ that gives a rigorously verified approximation ratio of at least 0.87447 for MAX DI-CUT. (The actual ratio is probably about 0.874502.) This table is taken from Table 1 in [BHPZ23].

To prove Theorem 5.12, we give a new algorithm for MAX DI-CUT from the $\mathcal{THRESH}$ family that achieves the claimed ratio. The specifications for this algorithm can be found in Table 5.3. The table describes a distribution over 7 piecewise-linear functions $g_1, g_2, \ldots, g_7$ defined on 17 control points. The first column contains the control points, and the following columns describe the value of each function at the corresponding control point. The function g_1 is odd and is very close to the function used by [LLZ02] in their MAX DI-CUT algorithm. The other six functions come in pairs. The two functions in each pair are flips of each other.

Note that here the threshold function is parameterized in a slightly different way than in Algorithm 2. For each $i \in [7]$, if we define $f_i : [-1,1] \to [-1,1], b \mapsto 2\Phi(g_i(b)) - 1$, then each f_i gives a $\mathcal{THRESH}^-$ rounding scheme as defined in Algorithm 2. This distribution $\mathcal{F}$ is obtained via a computer search, the details of which can be found in Section 4 of [BHPZ23].

It should be noted that this algorithm is not optimal: by adding more functions to the distribution, we can get a new $\mathcal{THRESH}$ scheme with marginally better approximation ratio. In fact, we make the following conjecture that is similar in spirit to Conjecture 3.

Conjecture 4. The optimal $\mathcal{THRESH}$ rounding scheme for MAX DI-CUT is supported on infinitely many $\mathcal{THRESH}^-$ schemes.

Conjectures 3 and 4 are in some sense dual to each other. We believe understanding one of them will also help us understand the other.

Verifying the approximation ratio of such a complicated algorithm is an impossible task by hand. We will use a computer-assisted proof, deploying the technique of interval arithmetic which we discussed earlier in Section 2.6. We need to prove the following inequality

$$\left(\mathrm{SDP}(\theta) \neq 0 \implies \frac{\mathbb{E}_{f\sim\mathcal{F}}[\mathsf{Prob}(\theta, f)]}{\mathrm{SDP}(\theta)} \geq \alpha\right),$$

for every configuration θ, or equivalently

$$\mathop{\mathbb{E}}_{f\sim\mathcal{F}}[\mathsf{Prob}(\theta, f)] - \alpha \cdot \mathrm{SDP}(\theta) \geq 0 \; .$$

The pseudocode of the algorithm is presented in Algorithm 4. The CHECKVALIDITY function checks if there exists a valid configuration in $I_1 \times I_2 \times I_{1,2}$ (recall that a DI-CUT configuration is represented by a triple), i.e., a configuration that satisfies all triangle inequalities, and returns true if it does. If CHECKVALIDITY returns false, then the algorithm returns true, since in this case the region consists entirely of invalid configurations and there is nothing to check. Otherwise, the algorithm continues to compute an interval I, using the

IntervalArithmeticEvaluate subroutine, such that

$$\forall \theta \in I_1 \times I_2 \times I_{1,2}, \quad \mathop{\mathbb{E}}_{f \sim \mathcal{F}}[\mathsf{Prob}(\theta, f)] - \alpha \cdot \mathrm{SDP}(\theta) \in I.$$

The algorithm then checks if I is entirely non-negative or entirely negative, in which cases we can decide that either the ratio is achieved over the entire region, or there exists a valid configuration that violates the ratio, and exit the algorithm accordingly. Otherwise, I consists of both positive and negative values, but the negative values may come from evaluation of invalid configurations, or more intrinsically the error produced by interval arithmetic itself. In this case, we subdivide the longest interval into two equal-length subintervals and recursively apply the algorithm.

We implemented this verification algorithm in C using the interval arithmetic library Arb [Joh17]. To speed up the computation, we split the various tasks between cores using GNU Parallel [O11].

Remark 5.13. To further speed up the computation, we also computed partial derivatives of $\mathbb{E}_{f \sim \mathcal{F}}[\mathsf{Prob}(\theta, f)] - \alpha \cdot \mathrm{SDP}(\theta)$ with respect to the biases and pairwise bias, and skip a region entirely if we find that one of the partial derivatives is strictly positive or strictly negative. This is justified by the fact that the minimizer of $\mathbb{E}_{f \sim \mathcal{F}}[\mathsf{Prob}(\theta, f)] - \alpha \cdot \mathrm{SDP}(\theta)$ will never be in the interior of such a region. We stress that we only perform this optimization in regions that are within the interior of the valid region defined by the triangle inequalities, since the minimum could appear on the boundary.

Algorithm 4 Interval arithmetic verification algorithm

procedure CheckRatio(I_1, I_2, $I_{1,2}$) ▷ Returns a Boolean value

  if CheckValidity($I_1, I_2, I_{1,2}$) = FALSE **then**

    return TRUE

  $I \leftarrow$ IntervalArithmeticEvaluate($I_1, I_2, I_{1,2}$).

  if $I \subseteq [0, \infty)$ **then**

    return TRUE

  else if $I \subseteq (\infty, 0)$ **then**

    return FALSE

  else

    if $|I_1| = \max(|I_1|, |I_2|, |I_{1,2}|)$ **then**

      Split I_1 into two equal-length sub-intervals $I_1 = I_1^l \cup I_1^r$.

      return CheckRatio($I_1^l, I_2, I_{1,2}$) $\wedge$ CheckRatio($I_1^r, I_2, I_{1,2}$)

    else if $|I_2| = \max(|I_1|, |I_2|, |I_{1,2}|)$ **then**

      Split I_2 into two equal-length sub-intervals $I_2 = I_2^l \cup I_2^r$.

      return CheckRatio($I_1, I_2^l, I_{1,2}$) $\wedge$ CheckRatio($I_1, I_2^r, I_{1,2}$)

    else

      Split $I_{1,2}$ into two equal-length sub-intervals $I_{1,2} = I_{1,2}^l \cup I_{1,2}^r$.

      return CheckRatio($I_1, I_2, I_{1,2}^l$) $\wedge$ CheckRatio($I_1, I_2, I_{1,2}^r$)

We obtain the following lemma.

Lemma 5.14. *$\mathcal{F}$ achieves an approximation ratio of* 0.87448 *on all DI-CUT configurations with completeness at least* 10^{-4}.

Note that in the lemma above there is a requirement on completeness in the next subsection. As we discussed, interval arithmetic in general cannot certify nonnegativity of a function which attains 0. Unfortunately, the function that we care about, $\mathbb{E}_{f\sim\mathcal{F}}[\mathsf{Prob}(\theta, f)] - \alpha \cdot$

SDP(θ), does attain 0, regardless of the choice of r, as the following proposition shows.

Proposition 5.15. *Let $\theta = (b_i, b_j, b_{ij})$ be a DI-CUT configuration with $b_i = b_j = b$ and $\rho(\theta) = 1$. Then for any f,*

$$\mathsf{Prob}(\theta, f) = \mathrm{SDP}(\theta) = 0 \ .$$

Proof. Since $\rho(\theta) = 1$, we have $b_{ij} = b_i b_j + \rho\sqrt{1-b_i^2}\sqrt{1-b_j^2} = b^2 + (1-b^2) = 1$ and

$$\mathrm{SDP}(\theta) = \frac{1 + b_i - b_j - b_{ij}}{4} = \frac{1 + b - b - 1}{4} = 0 \ .$$

For soundness, we have $\mathsf{Prob}(\theta, f) = \Phi_{-\rho}(f(b_i), -f(b_j)) = \Phi_{-\rho}(f(b), -f(b))$. Since $\rho = 1$, this is equal to $\Pr_{X\sim N(0,1)}[X \le f(b) \wedge -X \le -f(b)] = \Pr_{X\sim N(0,1)}[X = f(b)] = 0$. $\square$

Luckily, on configurations with small completeness, it is known that independent rounding, which assigns true to each variable independently with probability 1/2, does very well. Indeed, this rounding scheme achieves satisfies the DI-CUT constraint with probability 1/4 on every configuration. This implies that $\mathcal{F}$ combined with the independent rounding will achieve a good approximation ratio over all DI-CUT configurations.

Proof of Theorem 5.12. Consider the rounding algorithm where we use the $\mathcal{THRESH}$ rounding scheme $\mathcal{F}$ with probability $(1 - 10^{-5})$ and independent rounding with probability 10^{-5}. We show that this algorithm achieves a ratio of 0.87447 on all configurations of DI-CUT.

Let θ be a DI-CUT configuration. If $\mathrm{SDP}(\theta) \ge 10^{-6}$, then by Lemma 5.14, we achieve a ratio of at least $0.87448 \times (1 - 10^{-5}) > 0.87447$. If $\mathrm{SDP}(\theta) < 10^{-6}$, then independent rounding contributes a soundness of $0.25 \times 10^{-5} = 2.5 \times 10^{-6} > 0.87447 \cdot \mathrm{SDP}(\theta)$. $\square$

Remark 5.16. In a subsequent version of [BHPZ23], the use of independent rounding in the proof of Theorem 5.12 is replaced with a more careful analysis of the behavior of the algorithm on the small-completeness configurations, and consequently we no longer lose 10^{-5} in the verified ratio.

CHAPTER 6

7/8-HARDNESS FOR MONOTONE MAX NAE-SAT

In this chapter, we study the MAX NAE-SAT problem, which can be thought of as a symmetrized and harder version of the MAX SAT problem. We prove that obtaining a 7/8-approximation for MAX NAE-SAT is UG-hard. We then extend this result to its monotone version, where variable negations are not allowed. This chapter is based on [BHPZ21] which appeared in SODA'21 and its extended version.

6.1 The MAX NAE-SAT Problem

MAX NAE-SAT can be thought of as a symmetrized and harder version of MAX SAT. For any MAX SAT constraint, the only non-satisfying assignment is the all-false assignment, whereas for a MAX NAE-SAT constraint, there are two non-satisfying assignments: the all-true assignment and the all-false assignment. This makes the two truth values symmetric for MAX NAE-SAT. More concretely, we have the following well-known proposition.

Proposition 6.1. *There exists an approximation-preserving reduction from MAX $\{k\}$-SAT to MAX NAE-$\{k+1\}$-SAT.*

Proof. Let Φ be a MAX $\{k\}$-SAT instance, with variable set $V = \{x_1, \ldots, x_n\}$ and constraint set $\mathcal{C} = \{C_1, \ldots, C_m\}$. We construct a MAX NAE-$\{k+1\}$-SAT instance Ψ as follows. The variable set of Ψ will be $V' = V \cup \{x_0\}$. For any clause $C_i = \mathrm{OR}_k(\ell_{i,1}, \ldots, \ell_{i,k})$, we create a clause $C_i' = \mathrm{NAE}_{k+1}(x_0, \ell_{i,1}, \ldots, \ell_{i,k})$. We then take the constraint set of Ψ to be $\mathcal{C}' = \{C_1', \ldots, C_m'\}$. Given any solution to Ψ, we may assume $x_0 = 1$ (false) since NAE_{k+1} is an even predicate, and with this assumption we have that

$$\begin{aligned} C_i \text{ is satisfied} &\iff \text{There exists a true literal in } C_i \\ &\iff C_i' \text{ is satisfied.} \end{aligned}$$

It follows that this reduction is approximation-preserving. $\square$

By allowing clauses of arbitrary length, the same reduction in the above proof also implies the following proposition.

Proposition 6.2. *There exists an approximation-preserving reduction from MAX SAT to MAX NAE-SAT.*

For this reason, if there is a 7/8-approximation algorithm for MAX NAE-SAT, then it would also imply the existence of a 7/8-approximation algorithm for MAX SAT. It is therefore very natural to ask whether such an algorithm exists for MAX NAE-SAT.

$$\boxed{\textit{Question: is there a 7/8-approximation algorithm for MAX NAE-SAT?}}$$

One glimmer of hope is that tight approximation algorithms are known for MAX NAE-$\{k\}$-SAT, for every $k \geq 2$, under the assumption of Unique Games Conjecture and/or P is not equal to NP, and all of them have approximation ratios at least 7/8:

- When $k = 2$, it is known that there is an algorithm achieving an approximation ratio of $\alpha_{GW} \approx 0.8786$ [GW95], but achieving $\alpha_{GW} + \epsilon$ for any $\epsilon > 0$ is UG-hard [KKMO07].
- When $k = 3$, it is known that there is an algorithm achieving an approximation ratio of $\alpha_{\mathrm{NAE}_3} \approx 0.9089$, but achieving $\alpha_{\mathrm{NAE}_3} + \epsilon$ for any $\epsilon > 0$ is UG-hard [BHPZ21].
- When $k \geq 4$, taking a uniformly random assignment gives an approximation ratio of $1/2^{k-1}$, and achieving $1/2^{k-1} + \epsilon$ for any $\epsilon > 0$ is NP-hard [Hå01].

However, it turns out that assuming UGC (or at least that unique games is hard), there is no 7/8-approximation algorithm for MAX NAE-SAT.

Theorem 6.3. *It is UG-hard to approximate monotone MAX NAE-$\{3,5\}$-SAT (i.e., the version of MAX NAE-SAT with clauses of length $\in \{3,5\}$ and no negated literals) with a ratio better than $\frac{3(\sqrt{21}-4)}{2} \approx 0.8739$.*

This chapter will be organized as follows. In Section 6.2, we describe the RPR^2 rounding family which contains the optimal rounding algorithm for MAX NAE-SAT and monotone MAX NAE-SAT. In Section 6.3, we introduce *moment functions* of RPR^2 rounding schemes and prove several useful properties about them. As a warm-up, we then prove the hardness result for MAX NAE-SAT (with negated literals) in Section 6.4. Finally, we prove the result for monotone MAX NAE-SAT in Section 6.5. The proofs in Section 6.4 and Section 6.5 are obtained via showing limitations of RPR^2 rounding schemes. We will also give a construction for a explicit family of gap instances that achieve the same result in Section 6.6.

Before we proceed further, we recall the following Fourier expansion for not-all-equal predicates.

Proposition 6.4. *The Fourier expansion of* $\mathrm{NAE}_k : \{-1,1\}^k \to \{0,1\}$ *is given by*

$$\mathrm{NAE}_k(x_1,\dots,x_k) = \sum_{S\subseteq[k]} \widehat{\mathrm{NAE}}_k(S)\prod_{i\in S} x_i,$$

where $\widehat{\mathrm{NAE}}_k(\varnothing) = 1 - \frac{1}{2^{k-1}}$, $\widehat{\mathrm{NAE}}_k(S) = 0$ *if* $|S|$ *is odd, and* $\widehat{\mathrm{NAE}}_k(S) = -\frac{1}{2^{k-1}}$ *if* $|S|$ *is even and at least 2.*

We explicitly write down the Fourier expansions when $k = 3$ and $k = 5$, which will be needed in the following sections.

- $\mathrm{NAE}_3(x_1,x_2,x_3) = \frac{3-x_1x_2-x_1x_3-x_2x_3}{4}$,
- $\mathrm{NAE}_5(x_1,x_2,\dots,x_5) = \frac{15-\sum_{1\le i<j\le 5} x_ix_j-\sum_{1\le i<j<k<l\le 5} x_ix_jx_kx_l}{16}$.

6.2 Rounding Schemes for MAX NAE-SAT

In this section we introduce RPR^2 rounding schemes and argue that for MAX NAE-SAT it is sufficient to consider these rounding schemes only. These rounding schemes were first formulated and named (RPR^2 is short for "random projection, randomized rounding") by

Feige and Langberg [FL06]. A RPR^2 rounding scheme chooses a function $f : \mathbb{R} \to [-1, 1]$, which is referred to as a *rounding function*, and performs the actions described in Algorithm 5.

Algorithm 5 RPR^2 rounding scheme with rounding function $f : \mathbb{R} \to [-1, 1]$

Input: $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ unit vectors obtained by solving the Basic SDP

Output: $x_1, \ldots, x_n \in \{-1, 1\}$ rounded Boolean assignment to the variables

$\mathbf{r} \leftarrow N(0, I_n)$

for $i \leftarrow 1$ to n **do**

  $t_i \leftarrow \mathbf{r} \cdot \mathbf{v}_i$

  $x_i \leftarrow 1$ with probability $\frac{1+f(t_i)}{2}$, and $x_i \leftarrow -1$ with probability $\frac{1-f(t_i)}{2}$.

It can be easily seen that RPR^2 generalizes hyperplane rounding, which is simply RPR^2 using the sign function as the rounding function. In fact, we can easily generalize RPR^2 even further, by using a higher dimensional rounding function.

Algorithm 6 RPR^2 rounding scheme with rounding function $f : \mathbb{R}^d \to [-1, 1]$

Input: $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ unit vectors obtained by solving the Basic SDP

Output: $x_1, \ldots, x_n \in \{-1, 1\}$ rounded Boolean assignment to the variables

for $j \leftarrow 1$ to d **do**

  $\mathbf{r}_j \leftarrow N(0, I_n)$

for $i \leftarrow 1$ to n **do**

  for $j \leftarrow 1$ to d **do**

    $t_{i,j} \leftarrow \mathbf{v}_i \cdot \mathbf{r}_j$

  $x_i \leftarrow 1$ with probability $\frac{1+f(t_{i,1},\cdots,t_{i,d})}{2}$, and $x_i \leftarrow -1$ with probability $\frac{1-f(t_{i,1},\cdots,t_{i,d})}{2}$.

We now compare Algorithm 6 with the brute-force rounding scheme described in Chapter 3. The brute-force rounding scheme there can be thought of as a variation on Algorithm 6, where we are allowed to choose the best f after seeing the projections $t_{i,j}$ (the dimension

d has to be chosen beforehand). This gives the rounding algorithm extra power. Indeed, Algorithm 6 as written does not have access to $\mathbf{v}_0$, which is crucial for the rounding of many problems including MAX SAT. However, if we are allowed to choose f after projecting the vectors, then we may choose some f that has the "built-in" knowledge of $\mathbf{v}_0 \cdot \mathbf{r}_j$ as well, which, if d is sufficiently large, essentially give us all information on $\mathbf{v}_0$. However, for MAX NAE-SAT, since the predicates are even, the Basic SDP can be assumed not to be biased toward either 1 or -1, and $\mathbf{v}_0$ does not play a role in the rounding algorithm at all. For this reason, Algorithm 6 does capture the full power of the brute-force rounding scheme in the case of MAX NAE-SAT. We formally prove it in the following lemma.

Lemma 6.5. *Let $c > s > 0$ and Φ be a MAX NAE-SAT instance with the following properties:*

- $\mathrm{SDP}(\Phi) \geq c$*, and*
- *for all f, Algorithm 6 with rounding function f satisfies at most s fraction of the constraints in expectation,*

then it is UG-hard to approximate MAX NAE-SAT within a ratio of $s/c + \epsilon$ for any $\epsilon > 0$.

Proof. Fix $\epsilon > 0$. Let $d > 0$ be a large integer to be chosen later, and $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ be an SDP solution to Φ that achieves the completeness c. Let B_ϵ be an ϵ-net on the unit sphere in $\mathbb{R}^d$, i.e., a partition of the unit sphere such that any two vectors that belong to the same piece of B_ϵ have ℓ_2 distance at most ϵ. Let us now construct another MAX NAE-SAT Φ' instance. The variables of Φ' will be the pieces in B_ϵ. The constraints of Φ' will be constructed in the following way:

- sample a constraint $P(x_{i_1}, \ldots, x_{i_k}) \sim \Phi$,
- sample i.i.d. standard Gaussian vectors $\mathbf{r}_1, \ldots, \mathbf{r}_d \in \mathbb{R}^n$

- for $j \in [k]$, compute $\mathbf{u}_j = \frac{1}{\sqrt{d}}(\mathbf{v}_{i_j} \cdot \mathbf{r}_1, \ldots, \mathbf{v}_{i_j} \cdot \mathbf{r}_d) \in \mathbb{R}^d$, and let p_j be the piece in B_ϵ that is closest to $\mathbf{u}_j$.

- if $|\mathbf{u}_j \cdot \mathbf{u}_\ell - \mathbf{v}_{i_j} \cdot \mathbf{v}_{i_\ell}| \le \epsilon$ for every $j, \ell \in [k]$, then output the constraint $P(p_1, \ldots, p_k)$, otherwise we output some dummy clause.

Note that although Φ' is described using probabilistic terms, it is fully deterministic since the probabilities translate to the weights on the clauses. Using the same arguments in [RS09], we can show that $|\mathrm{SDP}(\Phi') - \mathrm{SDP}(\Phi)| \le \epsilon$ if d is sufficiently large. On the other hand, any good assignment to Φ' automatically induces a rounding function f with which Algorithm 6 achieves the same satisfying fraction (within $o(1)$) on Φ. □

In the following, we will refer to Algorithm 6 with function f as RPR^2 scheme with function f. The following lemma shows that when negations are allowed, we only need to consider RPR^2 schemes with odd f.

Lemma 6.6. *Let $f : \mathbb{R}^d \to [-1, 1]$ and $f^{\mathrm{odd}}(x) = (f(x) - f(-x))/2$ be its odd part. For any MAX CSP, the worst case performance of the* RPR^2 *scheme with f' is at least as good as the* RPR^2 *scheme with f^{odd} .*

Proof. Consider an arbitrary MAX CSP and let Φ be a worst-case instance for $Round_{f'}$. Observe that the $Round_{f'}$ procedure is equivalent to the following: independently for every variable x_i, with probability $1/2$ apply the rounding function f on $\mathbf{v}_i$, and with probability $1/2$ apply the rounding function f on $-\mathbf{v}_i$ and flip the result. Observe further that by replacing $\mathbf{v}_i$ with $-\mathbf{v}_i$ and flipping the outcome, we are essentially applying f to a new instance with $-x_i$ in place of x_i in Φ. This implies that the value of $Round_{f'}$ on Φ is an average of 2^n values (n is the number of variables in Φ) where each value is $Round_f$ evaluated on an instance obtained by flipping some variables in Φ. It follows that in some of these instances $Round_f$ has a value as bad as the value of $Round_{f'}$ on Φ. □

6.3 Moment Functions of RPR^2 Rounding Schemes

In this section, we define *moment functions* of RPR^2 Rounding Schemes, which will be an indispensable tool for us to analyze the performances of these rounding algorithms. We then state and prove a few lemmas about the moment functions that are needed in our analysis.

Definition 6.7. For any $k \geq 2$, we say that $(b_{i,j})_{1 \leq i < j \leq k}$ is a tuple of *valid* pairwise biases if there exist unit vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}$ such that $b_{i,j} = \mathbf{u}^{(i)} \cdot \mathbf{u}^{(j)}$ for every $1 \leq i < j \leq k$.

Definition 6.8. Let $f : \mathbb{R}^d \to [-1, 1], \rho \in [0, 1]$, $k \in \mathbb{N}^+$. Let $(b_{i,j})_{1 \leq i < j \leq k}$ be a tuple of valid pairwise biases. We define

$$F_k[f](b_{1,2}, b_{1,3}, \ldots, b_{k-1,k}) = \mathop{\mathbb{E}}_{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}} \left[f\left(\mathbf{x}^{(1)}\right) \cdots f\left(\mathbf{x}^{(k)}\right) \right].$$

Here $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$ are d-dimensional standard Gaussian vectors such that $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are $b_{i,j}$-correlated[1]. When $b_{1,2} = b_{1,3} = \ldots = b_{k-1,k} = \rho$, we will write $F_k[f](\rho)$ as a short hand for $F_k[f](b_{1,2}, b_{1,3}, \ldots, b_{k-1,k})$. When $k = 1$, the moment function has no input and we have $F_1[f] = \mathbb{E}_{\mathbf{x} \sim N(0, I_d)}[f(\mathbf{x})]$. We will write $\mathbb{E}_{\mathbf{x} \sim N(0, I_d)}[f(\mathbf{x})]$ as $\mathbb{E}[f]$ if the context is clear.

Note that one way to generate the vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$ in the above definition is by taking $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}$ from the definition of valid pairwise biases and d independent standard Gaussian vectors $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(d)}$ and letting $\mathbf{x}^{(i)} = (\mathbf{r}^{(1)} \cdot \mathbf{u}^{(i)}, \ldots, \mathbf{r}^{(d)} \cdot \mathbf{u}^{(i)})$. These are exactly the vectors generated by the RPR^2 rounding scheme with function f. In other words, moment functions describe the expected values of monomials where each variable in the monomial is the rounded value output by the given RPR^2 rounding scheme.

The first lemma we need is the following simple result concerning the case in which there is one vector that's orthogonal to all other vectors.

---

1. We say that two d-dimensional standard Gaussian vectors $\mathbf{x}$ and $\mathbf{y}$ are ρ-correlated if the correlation matrix of $(\mathbf{x}_1, \ldots, \mathbf{x}_d, \mathbf{y}_1, \ldots, \mathbf{y}_d)$ is given by $\begin{pmatrix} I_d & \rho I_d \\ \rho I_d & I_d \end{pmatrix}$.

Lemma 6.9. *Let $f : \mathbb{R}^d \to [-1,1], \rho \in [0,1],\ k \in \mathbb{N}^+$. Let $(b_{i,j})_{1 \le i < j \le k}$ be a tuple of valid pairwise biases. If $b_{i,k} = 0$ for every $1 \le i \le k-1$, then*

$$F_k[f](b_{1,2}, b_{1,3}, \ldots, b_{k-1,k}) = F_{k-1}[f](b_{1,2}, b_{1,3}, \ldots, b_{k-2,k-1}) \cdot \mathbb{E}[f].$$

Proof. By definition we have

$$F_k[f](b_{1,2}, b_{1,3}, \ldots, b_{k-1,k}) = \mathop{\mathbb{E}}_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(k)}} \left[f\left(\mathbf{x}^{(1)}\right) \cdots f\left(\mathbf{x}^{(k)}\right) \right].$$

where $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$ are d-dimensional standard Gaussian vectors such that $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are $b_{i,j}$-correlated. If $b_{i,k} = 0$ for every $1 \le i \le k-1$, then $\mathbf{x}^{(k)}$ is independent from all other vectors, and it follows that

$$\begin{aligned} \mathop{\mathbb{E}}_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(k)}} \left[f\left(\mathbf{x}^{(1)}\right) \cdots f\left(\mathbf{x}^{(k)}\right) \right] &= \mathop{\mathbb{E}}_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(k-1)}} \left[f\left(\mathbf{x}^{(1)}\right) \cdots f\left(\mathbf{x}^{(k-1)}\right) \right] \mathop{\mathbb{E}}_{\mathbf{x}^{(k)}} \left[f\left(\mathbf{x}^{(k)}\right) \right] \\ &= F_{k-1}[f](b_{1,2}, b_{1,3}, \ldots, b_{k-2,k-1}) \cdot \mathbb{E}[f]. \end{aligned}$$ □

The following corollary is immediate.

Corollary 6.10. *Let $f : \mathbb{R}^d \to [-1,1], \rho \in [0,1],\ k \in \mathbb{N}^+$. Let $(b_{i,j})_{1 \le i < j \le k}$ be a tuple of valid pairwise biases. If f is an odd function and $b_{i,k} = 0$ for every $1 \le i \le k-1$, then*

$$F_k[f](b_{1,2}, b_{1,3}, \ldots, b_{k-1,k}) = 0.$$

Proof. f being an odd function implies that $\mathbb{E}[f] = 0$. The corollary then follows directly from Lemma 6.9. □

The next lemma gives us an alternative expression for the moment function when the input biases are all equal and non-negative.

Lemma 6.11. *Let $f : \mathbb{R}^d \to [-1,1], \rho \in [0,1]$, $k \in \mathbb{N}^+$. We have*

$$F_k[f](\rho) = \mathop{\mathbb{E}}_{\mathbf{x} \sim N(0,I_d)} \left[\left(\mathrm{U}_{\sqrt{\rho}} f(\mathbf{x}) \right)^k \right].$$

Proof. By definition,

$$F_k[f](\rho) = \mathop{\mathbb{E}}_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(k)}} \left[f\left(\mathbf{x}^{(1)}\right) \cdots f\left(\mathbf{x}^{(k)}\right) \right]$$

where $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$ are d-dimensional Gaussian vectors such that each $\mathbf{x}^{(i)} \sim N(0, I_d)$, and $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are ρ-correlated for $1 \le i < j \le k$. One way to generate such a distribution is by having $k+1$ independent d-dimensional standard Gaussian vectors $\mathbf{u}$, $\boldsymbol{\epsilon}^{(1)}, \ldots, \boldsymbol{\epsilon}^{(k)}$, and setting $\mathbf{x}^{(i)} = \sqrt{\rho} \cdot \mathbf{u} + \sqrt{1-\rho} \cdot \boldsymbol{\epsilon}^{(i)}$ for $1 \le i \le k$. It follows that

$$\begin{aligned}
F_k[f](\rho) &= \mathop{\mathbb{E}}_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(k)}} \left[f\left(\mathbf{x}^{(1)}\right) \cdots f\left(\mathbf{x}^{(k)}\right) \right] \\
&= \mathop{\mathbb{E}}_{\mathbf{u},\boldsymbol{\epsilon}^{(1)},\ldots,\boldsymbol{\epsilon}^{(k)}} \left[\prod_{i=1}^{k} f\left(\sqrt{\rho} \cdot \mathbf{u} + \sqrt{1-\rho} \cdot \boldsymbol{\epsilon}^{(i)} \right) \right] \\
&= \mathop{\mathbb{E}}_{\mathbf{u}} \left[\prod_{i=1}^{k} \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}^{(i)}} \left[f\left(\sqrt{\rho} \cdot \mathbf{u} + \sqrt{1-\rho} \cdot \boldsymbol{\epsilon}^{(i)} \right) \right] \right] \\
&= \mathop{\mathbb{E}}_{\mathbf{u}} \left[\left(\mathrm{U}_{\sqrt{\rho}} f(\mathbf{u}) \right)^k \right].
\end{aligned}$$

□

We will write $\mathbb{E}_{\mathbf{x} \sim N(0,I_d)} \left[\left(\mathrm{U}_{\sqrt{\rho}} f(\mathbf{x}) \right)^k \right]$ as $\mathbb{E}\left[\left(\mathrm{U}_{\sqrt{\rho}} f \right)^k \right]$ if the context is clear.

Corollary 6.12. *Let $f : \mathbb{R}^d \to [-1,1], \rho \in [0,1]$. We have*

$$F_4[f](\rho) \ge F_2[f](\rho)^2.$$

Proof. By Lemma 6.11, we have $F_4[f, f, \ldots, f](\rho) = \mathbb{E}\left[\left(\mathrm{U}_{\sqrt{\rho}} f \right)^4 \right]$ and $F_2[f, f, \ldots, f](\rho) =$

$\mathbb{E}\left[\left(\mathrm{U}_{\sqrt{\rho}}f\right)^2\right]$. The corollary then follows from Jensen's inequality. $\square$

Given a function $f : \mathbb{R}^k \to [-1,1]$, we will use f^{odd} to denote its odd part, defined by $\mathbf{v} \mapsto \frac{f(\mathbf{v})-f(-\mathbf{v})}{2}$, and f^{even} its even part, defined by $\mathbf{v} \mapsto \frac{f(\mathbf{v})+f(-\mathbf{v})}{2}$. Clearly, $f^{\mathrm{even}}, f^{\mathrm{odd}} : \mathbb{R}^k \to [-1,1]$ and $f = f^{\mathrm{even}} + f^{\mathrm{odd}}$.

Lemma 6.13. *Let $f : \mathbb{R}^d \to [-1,1], \eta \in [0,1]$, $k \in \mathbb{N}^+$. We have*

$$\mathbb{E}\left[(\mathrm{U}_\eta f)^k\right] = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{2i} \cdot \mathbb{E}\left[\left(\mathrm{U}_\eta f^{\mathrm{odd}}\right)^{2i} \left(\mathrm{U}_\eta f^{\mathrm{even}}\right)^{k-2i}\right].$$

Proof. It can be easily seen from the definition of U_η that it is a linear operator, so $\mathrm{U}_\eta f = \mathrm{U}_\eta f^{\mathrm{odd}} + \mathrm{U}_\eta f^{\mathrm{even}}$. Lemma 6.13 is then a simple consequence of the binomial theorem, plus the fact that any term that contains an odd power of $\mathrm{U}_\eta f^{\mathrm{odd}}$ will have zero expectation. $\square$

To end this section, we remark that F_2 coincides with the notion of *noise stability*, which has been studied previously and is well understood. Indeed, we have the following characterization:

Lemma 6.14 (c.f. [OW08])**.** *Let $\rho \in [-1,1]$, $f : \mathbb{R}^k \to [-1,1]$. Then there exist constants $c_0(f), c_1(f), \ldots$ such that $F_2[f](\rho) = \sum_{i=0}^\infty c_i(f)^2 \cdot \rho^i$.*

Corollary 6.15. *Let $\rho \in [-1,1]$, $f : \mathbb{R}^k \to [-1,1]$. If f is even, then $F_2[f](\rho) \geq 0$.*

Corollary 6.16. *Let $\rho \in [-1,1]$, $f : \mathbb{R}^k \to [-1,1]$. Then*

$$F_2[f](\rho) = F_2[f^{\mathrm{even}}](|\rho|) + \mathrm{sign}(\rho) F_2[f^{\mathrm{odd}}](|\rho|).$$

Proof. By Lemma 6.14, we have $F_2[f](\rho) = \sum_{i=0}^\infty c_i(f)^2 \cdot \rho^i$, $F_2[f^{\mathrm{even}}](\rho) = \sum_{i \text{ even}} c_i(f)^2 \cdot \rho^i$, $F_2[f^{\mathrm{odd}}](\rho) = \sum_{i \text{ odd}} c_i(f)^2 \cdot \rho^i$. The proposition follows by observing that $F_2[f^{\mathrm{even}}](\rho) = F_2[f^{\mathrm{even}}](|\rho|)$ and $F_2[f^{\mathrm{odd}}](\rho) = \mathrm{sign}(\rho) F_2[f^{\mathrm{odd}}](|\rho|)$. $\square$

6.4 7/8-Hardness for Non-Monotone MAX NAE-SAT

In this section, we present a construction of hard-to-round SDP solutions. As a warm-up, we will first show in Theorem 6.17 that this construction is hard to round as MAX NAE-SAT configurations, i.e., when the rounding schemes have to choose an odd f. We will show in the next section that actually this construction is hard to round even as monotone MAX NAE-SAT configurations, in which case f can be arbitrary.

Theorem 6.17. *It is UG-hard to approximate MAX NAE-$\{3,5\}$-SAT (i.e., the version of MAX NAE-SAT with clauses of length $\in \{3,5\}$) with a ratio better than* $\frac{3(\sqrt{21}-4)}{2} \approx 0.8739$.

We will use the following pairwise biases in the SDP solution: $\left(-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}\right)$ for NAE_3 and $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0\right)$ for NAE_5. We show that these pairwise biases "fool" the Basic SDP (has completeness 1) but are in fact very difficult to round.

Lemma 6.18. *Let Φ be a* MAX NAE-$\{3,5\}$-SAT *instance whose* 3*-clauses all have pairwise biases* $(b_{1,2}, b_{1,3}, b_{2,3}) = (-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3})$ *and* 5*-clauses all have pairwise biases* $(b_{1,2}, b_{1,3}, \ldots, b_{4,5}) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0)$, *then Φ has completeness* 1.

Proof. It suffices to show that for every clause, there exists a distribution of satisfying assignments that agrees with the pairwise biases.

- 3-clauses. The uniform distribution on $\{(1, 1, -1), (1, -1, 1), (-1, 1, 1)\}$ has the same pairwise biases.

Probability	x_1	x_2	x_3
1/3	1	-1	-1
1/3	-1	1	-1
1/3	-1	-1	1

- 5-clauses. The following distribution on satisfying assignments has the same pairwise biases.

Probability	x_1	x_2	x_3	x_4	x_5
1/6	-1	1	1	1	1
1/6	1	-1	1	1	1
1/6	1	1	-1	1	1
1/6	1	1	1	-1	1
1/3	1	1	1	1	-1

Note that although these distribution don't have zero biases, we can easily transform them into ones that do by taking the negations of the assignments. $\square$

Lemma 6.19. *Let Φ be a MAX NAE-$\{3,5\}$-SAT instance with completeness 1, and $\mathcal{A}$ a distribution of assignments to Φ. If the following conditions hold for some $F_2, F_4 \in [0,1]$,*

1. *The expected fraction of 3-clauses satisfied by $\mathcal{A}$ is at most $\frac{3+3F_2}{4}$,*
2. *The expected fraction of 5-clauses satisfied by $\mathcal{A}$ is at most $\frac{15-6F_2-F_4}{16}$,*
3. *$F_4 \geq F_2^2$,*

then by possibly re-weighting the clauses in Φ we can obtain another instance Φ' with completeness 1 such that the expected fraction of clauses satisfied by $\mathcal{A}$ on Φ' is at most $\frac{3(\sqrt{21}-4)}{2} <$ 0.8739.

Proof. Let $p \in [0,1]$ be some parameter to be chosen later. We construct Φ' by taking the distribution where we choose a random 3-clause from Φ with probability $1-p$ and choose a random 5-clause from Φ with probability p (here we think of weights on the clauses as probability weights). Then the expected fraction of clauses in Φ' satisfied by $\mathcal{A}$ is at most

$$(1-p)\frac{3+3F_2}{4} + p\frac{15-6F_2-F_4}{16} = \frac{12+3p+(12-18p)F_2-pF_4}{16} .$$

Since $F_4 \geq F_2^2$, this is at most

$$\frac{12+3p+(12-18p)F_2-pF_2^2}{16} = \frac{12+3p+\frac{(6-9p)^2}{p}-p\left(F_2-\frac{(6-9p)}{p}\right)^2}{16}$$
$$\leq \frac{84p+\frac{36}{p}}{16}-6\ .$$

Taking the derivative with respect to p, we get that this expression is minimized when $\frac{1}{16}(84-\frac{36}{p^2})=0$, which happens when $p=\frac{3}{\sqrt{21}}$. When $p=\frac{3}{\sqrt{21}}$,

$$\frac{84p+\frac{36}{p}}{16}-6=\frac{12\sqrt{21}+12\sqrt{21}}{16}-6=\frac{3(\sqrt{21}-4)}{2}\ ,$$

and this completes the proof. □

Theorem 6.20. *A* 0.8739*-approximation for* MAX NAE-$\{3,5\}$-SAT *(clauses of size* 3 *and* 5*) is UG-hard, even when the instance has completeness* $1-\epsilon$*, for* $\epsilon > 0$ *arbitrarily small.*

Proof. Let Φ be an instance that satisfies the conditions in Lemma 6.18. Note that such an instance always exists, since we can take one 3-clause and one 5-clause on disjoint variables. We first analyze how an RPR^2 scheme with odd f performs on the SDP solution described in Lemma 6.18. Recall that we have the Fourier expansions:

$$\mathrm{NAE}_3(x_1,x_2,x_3)=\frac{3-x_1x_2-x_1x_3-x_2x_3}{4}\ ,$$
$$\mathrm{NAE}_5(x_1,x_2,x_3,x_4,x_5)=\frac{15-\sum_{1\leq i<j\leq 5}x_ix_j-\sum_{1\leq i<j<k<l\leq 5}x_ix_jx_kx_l}{16}\ .$$

Using this, we make the following observations:

1. If we have a 3-clause $\mathrm{NAE}_3(x_1,x_2,x_3)$ where $b_{1,2}=b_{1,3}=b_{2,3}=-\frac{1}{3}$ then

$$\mathbb{E}[\mathrm{NAE}_3(x_1,x_2,x_3)]=\frac{3-3F_2[f](-1/3)}{4}=\frac{3+3F_2[f](1/3)}{4}.$$

In the second equality we used the fact that f is odd and Corollary 6.16.

2. If we have a 5-clause $\mathrm{NAE}_5(x_1, x_2, x_3, x_4, x_5)$ where $b_{1,2} = b_{1,3} = b_{1,4} = b_{2,3} = b_{2,4} = b_{3,4} = \frac{1}{3}$ and $b_{1,5} = b_{1,5} = b_{2,5} = b_{3,5} = b_{4,5} = 0$ then

$$\mathbb{E}[\mathrm{NAE}_5(x_1, x_2, x_3, x_4, x_5)] = \frac{15 - 6F_2[f](1/3) - F_4[f](1/3)}{16}.$$

Here, all moments that contain the 5th variable evaluate to 0 due to Lemma 6.9.

We can now apply Lemma 6.19, with $F_2 = F_2[f](1/3), F_4 = F_4[f](1/3)$, $\mathcal{A}$ being the distribution of assignments induced by the RPR^2 scheme, to obtain another instance Φ' such that the expected satisfied fraction by the RPR^2 scheme on Φ' is at most $\frac{3(\sqrt{21}-4)}{2} < 0.8739$. The theorem now follows from Lemma 6.5 and Lemma 6.6. □

6.5 7/8-Hardness for Monotone MAX NAE-SAT

In this section, we extend the analysis in the previous section to monotone MAX NAE-SAT, i.e., when negated literals do not appear in the instance. The only difference here is that now the RPR^2 rounding scheme doesn't need to use an odd f.

Let $\rho = \frac{1}{3}$. For the soundness of the 3-clause, we have

$$\frac{3 - 3F_2[f]\,(-\rho)}{4}. \tag{6.1}$$

For the soundness of the 5-clause, we now have

$$\frac{15 - 6F_2[f]\,(\rho) - 4F_2[f](0) - F_4[f](\rho) - 4F_4[f](\rho, \rho, \rho, 0, 0, 0)}{16} \tag{6.2}$$

We will show that both (6.1) and (6.2) increase if we replace f with its odd part f^{odd}. This is spelled out in the following lemma.

Lemma 6.21. *Let $f : \mathbb{R}^k \to [-1,1]$ and $\rho \in [0,1]$. Then the following inequalities hold:*

(a) $\frac{3-3F_2[f](-\rho)}{4} \le \frac{3-3F_2[f^{\text{odd}}](-\rho)}{4}$.

(b) $\frac{15-6F_2[f](\rho)-4F_2[f](0)-F_4[f](\rho)-4F_4[f](\rho,\rho,\rho,0,0,0)}{16} \le \frac{15-6F_2[f^{\text{odd}}](\rho)-F_4[f^{\text{odd}}](\rho)}{16}$.

Note that this lemma immediately implies that the analysis in the previous section works for monotone MAX NAE-SAT as well.

The remainder of the section will be denoted to the proof of Lemma 6.21. It will be seen that part (a) of Lemma 6.21 is straightforward to prove, but part (b) is trickier, mostly because of the term $F_4[f](\rho,\rho,\rho,0,0,0)$ which can be either positive or negative. Our strategy, roughly speaking, is to show that the gain in other terms more than compensates for what's potentially lost in $F_4[f](\rho,\rho,\rho,0,0,0)$. To implement this, we first write down explicitly how these terms change if we replace f with its odd part.

Proposition 6.22. *Let $f : \mathbb{R}^k \to [-1,1]$ and $\rho \in [0,1]$. We have the following equalities:*

(a) $F_4[f](\rho,\rho,\rho,\rho,\rho,\rho) = \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^4\right] + \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^4\right] + 6\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^2(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right]$.

(b) $6F_2[f]\,(\rho) + 4F_2[f](0) = 6\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^2\right] + 6\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right] + 4\,\mathbb{E}[f]^2$.

(c) $F_4[f](\rho,\rho,\rho,0,0,0) = \mathbb{E}[f]\cdot\left(\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x}))^3\right] + 3\,\mathbb{E}\left[\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x}) \cdot (\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}}(\mathbf{x}))^2\right]\right)$.

Proof. For (a), by Lemma 6.11, $F_4[f](\rho,\rho,\rho,\rho,\rho,\rho) = \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f)^4\right]$. By Lemma 6.13,

$$\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f)^4\right] = \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^4\right] + \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^4\right] + \binom{4}{2} \cdot \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^2(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right].$$

Part (b) follows directly from Lemma 6.11 and Corollary 6.16.

For part (c), by Lemma 6.9, we have

$$F_4[f](\rho,\rho,\rho,0,0,0) = F_3[f](\rho)\cdot\mathbb{E}[f].$$

By Lemma 6.11, we have $F_3[f](\rho) = \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f(\mathbf{x}))^3\right]$. Again, by Lemma 6.13,

$$\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f(\mathbf{x}))^3\right] = \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}}(\mathbf{x}))^3\right] + 3\,\mathbb{E}\left[\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}}(\mathbf{x}) \cdot (\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}}(\mathbf{x}))^2\right]. \qquad \square$$

We now show that

Proposition 6.23. *Let* $f : \mathbb{R}^k \to [-1,1]$ *and* $\rho \in [0,1]$. *We have*

(a) $\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2 (\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}})^2\right] + \mathbb{E}[f]^2 \geq \left|2\,\mathbb{E}[f] \cdot \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2 \cdot \mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}}\right]\right|$.

(b) $\mathbb{E}[f]^2 + F_2[f^{\mathrm{even}}](\rho) \geq \left|2\,\mathbb{E}[f] \cdot \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}})^3\right]\right|$.

Proof. For part (a), note that for any random variables X and Y we have $\mathbb{E}\left[X^2(Y - \mathbb{E}[Y])^2\right] = \mathbb{E}[X^2Y^2] + \mathbb{E}[X^2]\,\mathbb{E}[Y]^2 - 2\,\mathbb{E}\left[X^2 Y\right]\mathbb{E}[Y] \geq 0$, and by letting $X = \mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}}$ and $Y = \mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}}$ we have

$$\begin{aligned}
\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2 (\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}})^2\right] + \mathbb{E}[f]^2 &\geq \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2 (\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}})^2\right] + \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2\right]\mathbb{E}[f]^2 \\
&\geq 2\,\mathbb{E}[f] \cdot \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2 \cdot \mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}}\right].
\end{aligned}$$

By considering $\mathbb{E}\left[X^2(Y + \mathbb{E}[Y])^2\right]$, a similar argument shows that

$$\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2 (\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}})^2\right] + \mathbb{E}[f]^2 \geq -2\,\mathbb{E}[f] \cdot \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{odd}})^2 \cdot \mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}}\right].$$

Thus, part (a) follows. For part (b), we have

$$\begin{aligned}
\left|\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}})^3\right]\right| &\leq \mathbb{E}\left[\left|\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}}\right|^3\right] \\
&\leq \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\mathrm{even}})^2\right] \leq \sqrt{F_2[f^{\mathrm{even}}](\rho)}.
\end{aligned}$$

Here we have used the fact that $|\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}| \le 1$. It follows that

$$\left|2\,\mathbb{E}[f]\cdot \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^3\right]\right| \le \left|2\,\mathbb{E}[f]\sqrt{F_2[f^{\text{even}}](\rho)}\right| \le \mathbb{E}[f]^2 + F_2[f^{\text{even}}](\rho). \qquad \square$$

We are now finally ready to prove Lemma 6.21.

Proof of Lemma 6.21. The first part follows from Corollary 6.15 and Corollary 6.16:

$$\frac{3-3F_2[f]\,(-\rho)}{4} = \frac{3-3F_2[f^{\text{even}}]\,(\rho)+3F_2[f^{\text{odd}}]\,(\rho)}{4} \le \frac{3+3F_2[f^{\text{odd}}]\,(\rho)}{4}.$$

For the second part, by Proposition 6.22 we have

$$\begin{aligned}
&\frac{15-6F_2[f^{\text{odd}}]\,(\rho)-F_4[f^{\text{odd}}](\rho)}{16} \\
&\quad - \frac{15-6F_2[f]\,(\rho)-4F_2[f](0)-F_4[f](\rho)-4F_4[f](\rho,\rho,\rho,0,0,0)}{16} \\
&= \frac{6\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right] + 4\,\mathbb{E}[f]^2 + \mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^4\right] + 6\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^2(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right]}{16} \\
&\quad + \frac{4\,\mathbb{E}[f]\cdot\left(\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x}))^3\right] + 3\,\mathbb{E}\left[\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x})\cdot(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}}(\mathbf{x}))^2\right]\right)}{16} \\
&\ge \frac{2\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right] + 8\,\mathbb{E}[f]^2 + 6\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^2(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right]}{16} \\
&\quad + \frac{4\,\mathbb{E}[f]\cdot\left(\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x}))^3\right] + 3\,\mathbb{E}\left[\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x})\cdot(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}}(\mathbf{x}))^2\right]\right)}{16} \\
&= \frac{2\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right] + 2\,\mathbb{E}[f]^2 + 4\,\mathbb{E}[f]\cdot\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x}))^3\right]}{16} \\
&\quad + \frac{6\,\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}})^2(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right] + 12\,\mathbb{E}[f]\cdot\mathbb{E}\left[\mathrm{U}_{\sqrt{\rho}} f^{\text{even}}(\mathbf{x})\cdot(\mathrm{U}_{\sqrt{\rho}} f^{\text{odd}}(\mathbf{x}))^2\right] + 6\,\mathbb{E}[f]^2}{16} \\
&\ge 0.
\end{aligned}$$

Here in the first inequality we used $\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^2\right] \ge \mathbb{E}[f]^2$ and $\mathbb{E}\left[(\mathrm{U}_{\sqrt{\rho}} f^{\text{even}})^4\right] \ge 0$, while in the second inequality we used Proposition 6.23. $\square$

6.6 Constructing an Explicit Gap Instance

In this section, we explicitly construct a family of gap instances whose integrality ratio tends to $\frac{3(\sqrt{21}-4)}{2} \approx 0.8739$. Let $\{\mathbf{e}_i \mid i \in [n]\}$ be the canonical basis of $\mathbb{R}^n$. Consider the subset of $\mathbb{R}^n$ containing vectors that have exactly three nonzero coordinates, each being $1/\sqrt{3}$ or $-1/\sqrt{3}$, namely

$$V_n = \left\{ \frac{b_1\mathbf{e}_i + b_2\mathbf{e}_j + b_3\mathbf{e}_k}{\sqrt{3}} \;\middle|\; b_1, b_2, b_3 \in \{-1, 1\}, 1 \le i < j < k \le n \right\} .$$

Note that V_n is closed under vector negation. For every $\mathbf{v} \in V_n$, we assign a Boolean variable $x_{\mathbf{v}} \in \{-1, 1\}$. In the following we will assume $x_{\mathbf{v}} = -x_{-\mathbf{v}}$ and prove the result for MAX NAE-SAT, but this assumption can be removed and the result strengthened to monotone MAX NAE-SAT, as explained in Remark 6.34. We will define a MAX NAE-SAT instance Φ with variables $x_{\mathbf{v}}$ such that assigning $\mathbf{v}$ to $x_{\mathbf{v}}$ is an SDP solution with perfect completeness, while any integral solution has value at most $\frac{3(\sqrt{21}-4)}{2}$ as n tends to infinity.

Definition 6.24. We define $\mathcal{C}_3$ to be the set of 3-clauses of the form $\mathrm{NAE}(x_{\mathbf{v}_1}, x_{\mathbf{v}_2}, x_{\mathbf{v}_3})$ where

1. $\mathbf{v}_1 = \frac{1}{\sqrt{3}}(s_1\mathbf{e}_{i_1} - s_2\mathbf{e}_{i_2} + s_4\mathbf{e}_{i_4})$
2. $\mathbf{v}_2 = \frac{1}{\sqrt{3}}(s_2\mathbf{e}_{i_2} - s_3\mathbf{e}_{i_3} + s_5\mathbf{e}_{i_5})$
3. $\mathbf{v}_3 = \frac{1}{\sqrt{3}}(s_3\mathbf{e}_{i_3} - s_1\mathbf{e}_{i_1} + s_6\mathbf{e}_{i_6})$

for some distinct indices $i_1, \ldots, i_6 \in [n]$ and signs $s_1, \ldots, s_6 \in \{-1, 1\}$.

Definition 6.25. We define $\mathcal{C}_5$ to be the set of 5-clauses of the form $\mathrm{NAE}(x_{\mathbf{v}_1}, x_{\mathbf{v}_2}, x_{\mathbf{v}_3}, x_{\mathbf{v}_4}, x_{\mathbf{v}_5})$ where

1. For all $j \in \{1, 2, 3, 4\}$, $\mathbf{v}_j = \frac{1}{\sqrt{3}}(s_1\mathbf{e}_{i_1} + s_{2j}\mathbf{e}_{i_{2j}} + s_{2j+1}\mathbf{e}_{i_{2j+1}})$

2. $\mathbf{v}_5 = \frac{1}{\sqrt{3}}(s_{10}\mathbf{e}_{i_{10}} + s_{11}\mathbf{e}_{i_{11}} + s_{12}\mathbf{e}_{i_{12}})$

for some distinct indices $i_1, \ldots, i_{12} \in [n]$ and signs $s_1, \ldots, s_{12} \in \{-1, 1\}$.

Remark 6.26. These sets of clauses are designed so that the pairwise biases for the 3-clauses are $(-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3})$ and the pairwise biases for the 5-clauses are $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0)$.

Definition 6.27. Let Φ be the MAX NAE-$\{3,5\}$-instance with variable set $\{X_{\mathbf{v}} \mid \mathbf{v} \in V_n\}$ and clause set $\mathcal{C}_3 \cup \mathcal{C}_5$, where every clause in $\mathcal{C}_3$ has weight $\frac{1-\frac{3}{\sqrt{21}}}{|\mathcal{C}_3|}$ and every clause in $\mathcal{C}_5$ has weight $\frac{3}{\sqrt{21}|\mathcal{C}_5|}$.

Theorem 6.28. *For any integral solution to Φ, the weight of the satisfied clauses is at most $\frac{3(\sqrt{21}-4)}{2} + O(\frac{1}{n})$.*

Proof. To analyze the weight of the satisfied constraints for a given solution, we consider the following distributions.

Definition 6.29. For every $k < n/2$, we define $\mathcal{D}_k$ to be the following distribution over V_n^k:

1. Sample $2k+1$ distinct indices $i_1, i_2, \ldots, i_{2k+1} \in [n]$ uniformly at random.

2. Sample $2k+1$ independent random coin flips $b_1, \ldots, b_{2k+1} \in \{-1, 1\}$.

3. For every $j \in [k]$, let $\mathbf{v}_j = \frac{1}{\sqrt{3}}(b_1\mathbf{e}_{i_1} + b_{2j}\mathbf{e}_{i_{2j}} + b_{2j+1}\mathbf{e}_{i_{2j+1}})$. Return the k-tuple $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k)$.

Informally speaking, this distribution samples k vectors from V_n of "sunflower shape" in the sense that all of them share exactly one index on which they are nonzero.

Definition 6.30. Given an assignment, we let

$$F_2 = \mathop{\mathbb{E}}_{(\mathbf{v}_1,\mathbf{v}_2)\sim\mathcal{D}_2}[x_{\mathbf{v}_1}x_{\mathbf{v}_2}] \quad , \quad F_4 = \mathop{\mathbb{E}}_{(\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3,\mathbf{v}_4)\sim\mathcal{D}_4}[x_{\mathbf{v}_1}x_{\mathbf{v}_2}x_{\mathbf{v}_3}x_{\mathbf{v}_4}] \, .$$

Remark 6.31. Here F_2 and F_4 come from an actual assignment rather than a rounding scheme, but they play the same role in the argument.

Proposition 6.32. *Given an assignment, the proportion of* 3*-clauses which are satisfied is* $\frac{3+3F_2}{4}$ *and the proportion of* 5*-clauses which are satisfied is* $\frac{15-6F_2-F_4}{16}$.

Proof sketch. This can be shown by expanding out each constraint as a polynomial. □

By Lemma 6.19, if we had that $F_4 \geq F_2^2$ then we would have that the total weight of the satisfied clauses is at most $\frac{3(\sqrt{21}-4)}{2}$. Instead, we show that $F_4 \geq F_2^2 - O(\frac{1}{n})$. Adapting the argument in Lemma 6.19 accordingly, this implies that the total weight of the satisfied clauses is at most $\frac{3(\sqrt{21}-4)}{2} + O(\frac{1}{n})$.

Lemma 6.33. *For any assignment,*

$$F_4 \geq F_2^2 - O\Big(\frac{1}{n}\Big) .$$

Proof. Let $k = \lfloor n/2 \rfloor - 1 < n/2$. Sample $(\mathbf{v}_1, \ldots, \mathbf{v}_k) \sim \mathcal{D}_k$. Note that the marginal distribution of any pair of these vectors is exactly $\mathcal{D}_2$ and any 4 vectors exactly $\mathcal{D}_4$. Now let $X = \sum_{i=1}^{k} x_{\mathbf{v}_i}$. By Jensen's inequality

$$\mathbb{E}[X^4] - \Big(\mathbb{E}[X^2]\Big)^2 \geq 0.$$

We have that

$$\mathbb{E}\left[X^2\right] = \mathbb{E}\left[\left(\sum_{i=1}^{k} X_{\mathbf{v}_i}\right)^2\right] = \sum_{i=1}^{k} \mathbb{E}[X_{\mathbf{v}_i}^2] + \sum_{i \neq j} \mathbb{E}[X_{\mathbf{v}_i} X_{\mathbf{v}_j}] = k + k(k-1)F_2 .$$

Here we used the fact that $X_{\mathbf{v}_i} \in \{-1, 1\}$ and $X_{\mathbf{v}_i}^2 = 1$. Similarly we can compute

$$\mathbb{E}\left[X^4\right] = \mathbb{E}\left[\left(\sum_{i=1}^{k} X_{\mathbf{v}_i}\right)^4\right] = 3k^2 - 2k + k(k-1)(6k-8)F_2 + k(k-1)(k-2)(k-3)F_4 .$$

Plugging in these two expressions to the inequality above, we get

$$3k^2 - 2k + k(k-1)(6k-8)F_2 + k(k-1)(k-2)(k-3)F_4 - (k + k(k-1)F_2)^2 \geq 0.$$

Our lemma follows by shifting terms. dividing both sides by $k(k-1)(k-2)(k-3)$, and using the fact that $k = \Theta(n)$. $\square$

$\square$

Remark 6.34. As mentioned earlier, we can remove the constraint $x_{\mathbf{v}} = -x_{-\mathbf{v}}$ and treat the instance as a monotone MAX NAE-SAT instance. Then, the proofs in Section 6.5 will go through with $\frac{1}{k}\sum_{i=1}^{k} x_{\mathbf{v}_i}$ playing the role of $\mathrm{U}_{\sqrt{\rho}} f$, where $(\mathbf{v}_1, \ldots, \mathbf{v}_k) \sim \mathcal{D}_k$ as in the previous lemma. The computation will stay essentially the same, with an error term that goes to 0 as n goes to infinity.

A natural question is whether there exists an assignment such that the weight of the satisfied constraints is at least $\frac{3(\sqrt{21}-4)}{2}$. If no such assignment exists, then it would be possible to further improve the upper bound. However, we show that for this set of constraints, our analysis is tight and there exists an assignment such that the weight of the satisfied constraints is at least $\frac{3(\sqrt{21}-4)}{2}$. That said, there may be another set of constraints which gives a better upper bound.

Theorem 6.35. *There is an assignment that satisfies a* $\frac{3(\sqrt{21}-4)}{2}$*-fraction of the clauses in* Φ*.*

Proof. It suffices to show there exists a probability distribution that satisfies $\frac{3(\sqrt{21}-4)}{2}$-fraction of the clauses in Φ in expectation. From the proof of Lemma 6.19 it can be seen that in order for a solution to achieve $\frac{3(\sqrt{21}-4)}{2}$ it suffices to have $F_2 = 2\sqrt{21} - 9$ and $F_4 = F_2^2$. We verify this as follows: given that $F_2 = 2\sqrt{21} - 9$ and $F_4 = F_2^2$, on 3-clauses we achieve a ratio of

$$\frac{3+3F_2}{4} = \frac{3+3\cdot(2\sqrt{21}-9)}{4} = \frac{3\left(\sqrt{21}-4\right)}{2} ,$$

and on 5-clauses we achieve a ratio of

$$\begin{aligned}\frac{15-6F_2-F_4}{16} &= \frac{15-6(2\sqrt{21}-9)-(2\sqrt{21}-9)^2}{16} \\ &= \frac{15-6(2\sqrt{21}-9)-(165-36\sqrt{21})}{16} \\ &= \frac{3\left(\sqrt{21}-4\right)}{2} .\end{aligned}$$

Let us now consider the following rounding algorithm: if v has 3 positive coordinates, round $X_{\mathbf{v}}$ to 1 with probability p_1 and -1 with probability $1-p_1$; if $\mathbf{v}$ has exactly 2 positive coordinates, round $X_{\mathbf{v}}$ to 1 with probability p_2 and -1 with probability $1-p_2$; if $\mathbf{v}$ has less than 2 positive coordinates, we let $X_{\mathbf{v}} = -X_{-\mathbf{v}}$. We need to analyze the quantity $\mathbb{E}_{\mathbf{v}_1,\ldots,\mathbf{v}_k\sim\mathcal{D}_k}[X_{\mathbf{v}_1}\cdots X_{\mathbf{v}_k}]$ for $k=2$ and $k=4$.

Let $(\mathbf{v}_1,\ldots,\mathbf{v}_k)\sim\mathcal{D}_k$. Recall that these vectors have a "sunflower" shape: they all share a common non-zero coordinate and each vector has a "petal" of two non-zero coordinates. Without loss of generality, assume that their common coordinate is positive. Then, every vector independently has 3 positive coordinates with probability 1/4, 2 positive coordinates with probability 1/2 and 1 positive coordinate with probability 1/4. So every variable $X_{\mathbf{v}_i}$ is rounding to 1 independently with probability

$$\frac{1}{4}\cdot p_1 + \frac{1}{2}\cdot p_2 + \frac{1}{4}\cdot(1-p_2) = \frac{p_1-p_2+1}{4} .$$

So we have

$$\mathbb{E}[X_{\mathbf{v}_i}] = \frac{p_1-p_2+1}{4} - \left(1-\frac{p_1-p_2+1}{4}\right) = \frac{p_1-p_2-1}{2}$$

and

$$F_2 = \left(\frac{p_1 - p_2 - 1}{2}\right)^2, \quad F_4 = \left(\frac{p_1 - p_2 - 1}{2}\right)^4 = F_2^2 .$$

Note that the range of F_2 is $[0, 0.25]$ and $2\sqrt{21} - 9 \approx 0.165$, so by setting p_1 and p_2 appropriately we can also have $F_2 = 2\sqrt{21} - 9$. This completes the proof. $\square$

REFERENCES

[ABG16] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better Balance by Being Biased: A 0.8776-Approximation for Max Bisection. *ACM Transactions on Algorithms*, 13(1):2:1–2:27, October 2016.

[ABM10] Per Austrin, Siavosh Benabbas, and Avner Magen. On Quadratic Threshold CSPs. In Alejandro López-Ortiz, editor, *LATIN 2010: Theoretical Informatics*, Lecture Notes in Computer Science, pages 332–343. Springer Berlin Heidelberg, 2010.

[ABZ06] Adi Avidor, Ido Berkovitch, and Uri Zwick. Improved Approximation Algorithms for MAX NAE-SAT and MAX SAT. In Thomas Erlebach and Giuseppe Persinao, editors, *Approximation and Online Algorithms*, pages 27–40, Berlin, Heidelberg, 2006. Springer.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

[Aus06] Per Austrin. Balanced Max 2-Sat might not be the hardest. Technical Report TR06-088, Electronic Colloquium on Computational Complexity (ECCC), July 2006. ISSN: 1433-8092.

[Aus07] Per Austrin. Balanced max 2-sat might not be the hardest. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 189–197, New York, NY, USA, June 2007. Association for Computing Machinery.

[Aus10] Per Austrin. Towards Sharp Inapproximability for Any 2-CSP. *SIAM Journal on Computing*, 39(6):2430–2463, January 2010. Publisher: Society for Industrial and Applied Mathematics.

[AZBG+22] Sepehr Abbasi-Zadeh, Nikhil Bansal, Guru Guruganesh, Aleksandar Nikolov, Roy Schwartz, and Mohit Singh. Sticky Brownian Rounding and its Applications to Constraint Satisfaction Problems. *ACM Transactions on Algorithms*, 18(4):33:1–33:50, October 2022.

[BHPZ21] Joshua Brakensiek, Neng Huang, Aaron Potechin, and Uri Zwick. On the Mysteries of MAX NAE-SAT. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 484–503. Society for Industrial and Applied Mathematics, January 2021.

[BHPZ23] Joshua Brakensiek, Neng Huang, Aaron Potechin, and Uri Zwick. Separating MAX 2-AND, MAX DI-CUT and MAX CUT. In *64th IEEE Symposium on Foundations of Computer Science (FOCS) 2023*, Santa Cruz, CA, USA, 2023.

[BHZ24] Joshua Brakensiek, Neng Huang, and Uri Zwick. Tight approximability of MAX 2-SAT and relatives, under UGC. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1328–1344. Society for Industrial and Applied Mathematics, January 2024.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, pages 151–158, New York, NY, USA, May 1971. Association for Computing Machinery.

[DW90] Zvi Drezner and G. O. Wesolowsky. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35(1-2):101–107, March 1990. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00949659008811236.

[EN19] Ronen Eldan and Assaf Naor. Krivine diffusions attain the Goemans–Williamson approximation ratio, June 2019. arXiv:1906.10615 [cs].

[FL06] Uriel Feige and Michael Langberg. The RPR2 rounding technique for semidefinite programs. *Journal of Algorithms*, 60(1):1–23, July 2006.

[GW94] Michel X. Goemans and David P. Williamson. New $\frac{3}{4}$-Approximation Algorithms for the Maximum Satisfiability Problem. *SIAM Journal on Discrete Mathematics*, 7(4):656–666, November 1994. Publisher: Society for Industrial and Applied Mathematics.

[GW95] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[HP20] Neng Huang and Aaron Potechin. On the Approximability of Presidential Type Predicates. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:20, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISSN: 1868-8969.

[HV91] D.J. Haglin and S.M. Venkatesan. Approximation and intractability results for the maximum cut problem and its variants. *IEEE Transactions on Computers*, 40(1):110–113, January 1991. Conference Name: IEEE Transactions on Computers.

[Hå01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, July 2001.

[Joh74] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, December 1974.

[Joh17] Fredrik Johansson. Arb: Efficient Arbitrary-Precision Midpoint-Radius Interval Arithmetic. *IEEE Transactions on Computers*, 66(8):1281–1292, August 2017.

[Kar72] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pages 85–103. Springer US, Boston, MA, 1972.

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 25–, May 2002. ISSN: 1093-0159.

[KKMO07] 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, January 2007. Publisher: Society for Industrial and Applied Mathematics.

[KMS23] Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in Grassmann graph have near-perfect expansion. *Annals of Mathematics*, 198(1):1–92, July 2023. Publisher: Department of Mathematics of Princeton University.

[KTW13] Subhash Khot, Madhur Tulsiani, and Pratik Worah. A Characterization of Approximation Resistance. *arXiv:1305.5500 [cs]*, May 2013. arXiv: 1305.5500.

[KZ97] Howard Karloff and Uri Zwick. A 7/8-Approximation Algorithm for MAX 3SAT? In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, FOCS '97, page 406, USA, October 1997. IEEE Computer Society.

[Las01] Jean B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, January 2001. Publisher: Society for Industrial and Applied Mathematics.

[LLZ02] Michael Lewin, Dror Livnat, and Uri Zwick. Improved Rounding Techniques for the MAX 2-SAT and MAX DI-CUT Problems. In William J. Cook and Andreas S. Schulz, editors, *Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, pages 67–82, Berlin, Heidelberg, 2002. Springer.

[MM14] Konstantin Makarychev and Yury Makarychev. Approximation Algorithm for Non-Boolean Max-k-CSP. *Theory of Computing*, 10:341–358, October 2014. Number: 13 Publisher: Theory of Computing.

[MM17] Konstantin Makarychev and Yury Makarychev. Approximation Algorithms for CSPs. *DROPS-IDN/v2/document/10.4230/DFU.Vol7.15301.287*, 2017. Publisher: Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[MOO10] 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. Publisher: Annals of Mathematics.

[O11] Tange O. Gnu parallel-the command-line power tool. *USENIX Mag*, 36(1):42, 2011.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, June 2014. Google-Books-ID: 5xlvAwAAQBAJ.

[OW08] Ryan O'Donnell and Yi Wu. An optimal sdp algorithm for max-cut, and equally optimal long code tests. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, STOC '08, pages 335–344, New York, NY, USA, May 2008. Association for Computing Machinery.

[Pot18] Aaron Potechin. On the Approximation Resistance of Balanced Linear Threshold Functions. *arXiv:1807.04421 [cs]*, July 2018. arXiv: 1807.04421.

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, December 1991.

[Rag08] Prasad Raghavendra. Optimal Algorithms and Inapproximability Results for Every CSP? In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 245–254, New York, NY, USA, 2008. ACM.

[Rag09] Prasad Raghavendra. *Approximating NP-hard Problems: Efficient Algorithms and their Limits*. PhD thesis, University of Washington, 2009.

[RS09] Prasad Raghavendra and David Steurer. How to Round Any CSP. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 586–594, October 2009. ISSN: 0272-5428.

[SG76] Sartaj Sahni and Teofilo Gonzalez. P-Complete Approximation Problems. *Journal of the ACM*, 23(3):555–565, July 1976.

[Spe94] Joel Spencer. *Ten Lectures on the Probabilistic Method*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, January 1994.

[Tre14] Luca Trevisan. Inapproximability of Combinatorial Optimization Problems. In *Paradigms of Combinatorial Optimization*, pages 381–434. John Wiley & Sons, Ltd, 2014. Section: 13 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119005353.ch13.

[Vaz03] Vijay V. Vazirani. *Approximation Algorithms*. Springer, Berlin, Heidelberg, 2003.

[VB96] Lieven Vandenberghe and Stephen Boyd. Semidefinite Programming. *SIAM Review*, 38(1):49–95, March 1996. Publisher: Society for Industrial and Applied Mathematics.

[Vit81] Paul M. B. Vitányi. How well can a graph be n-colored? *Discrete Mathematics*, 34(1):69–80, January 1981.

[Yan94] M. Yannakakis. On the Approximation of Maximum Satisfiability. *Journal of Algorithms*, 17(3):475–502, November 1994.

[Zwi02] Uri Zwick. Computer assisted proof of optimal approximability results. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '02, pages 496–505, USA, January 2002. Society for Industrial and Applied Mathematics.