Balanced Max 2-Sat Might Not be the Hardest

Per Austrin*
KTH – Royal Institute of Technology
Stockholm, Sweden
austrin@kth.se

ABSTRACT

We show that, assuming the Unique Games Conjecture, it is NP-hard to approximate MAX 2-SAT within $\alpha^-_{LLZ}+\epsilon$, where $0.9401 < \alpha^-_{LLZ} < 0.9402$ is the believed approximation ratio of the algorithm of Lewin, Livnat and Zwick [28].

This result is surprising considering the fact that balanced instances of MAX 2-SAT, i.e., instances where each variable occurs positively and negatively equally often, can be approximated within 0.9439. In particular, instances in which roughly 68% of the literals are unnegated variables and 32% are negated appear less amenable to approximation than instances where the ratio is 50%-50%.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Theory

Keywords

Max 2-Sat, Unique Games Conjecture, Inapproximability

1. INTRODUCTION

In their break-through paper [16], Goemans and Williamson used semidefinite programming techniques to construct 0.8785-approximation algorithms for MAX CUT and MAX 2-SAT, as well as a 0.7960-approximation algorithm for MAX DI-CUT. Since then, improved approximation algorithms based on semidefinite programming have been constructed for many other important NP-hard problems, including coloring of k-colorable graphs [22, 6, 17, 2], fairly general versions of integer quadratic programming on the hypercube [9] and MAX k-CSP [18, 7].

*Research funded by Swedish Research Council Project Number 50394001.

STOC'07, June 11–13, 2007, San Diego, California, USA.

Meanwhile, the study of *inapproximability* has seen a perhaps even bigger revolution, starting with the discovery of the PCP Theorem [4, 3]. It has led to inapproximability results for many NP-hard problems, several of them tight in the sense that they match the best known algorithmic results up to lower order terms (for instance SET COVER [13], CHROMATIC NUMBER [15], MAX CLIQUE [19], and MAX 3-SAT [20]).

However, for constraint satisfaction problems in which each constraint acts on two variables, tight results have been more elusive. In recent years, the so-called Unique Games Conjecture (UGC) has proved to be a possible means for obtaining such results. The UGC, which asserts the existence of a very powerful two-prover system with some specific properties, was introduced by Khot, who used it to show superconstant hardness for MIN 2-SAT-DELETION [23]. Since then, the UGC has been shown to imply hardness for several other problems, including $2-\epsilon$ hardness for VERTEX COVER [26], superconstant hardness for SPARSEST CUT [10, 27] and MULTICUT [10], coloring 3-colorable graphs with as few colors as possible [12], approximating MAX INDEPENDENT SET within $d/\text{poly}\log d$ in degree-d graphs [32], and approximation resistance for random predicates [21].

Recently, thanks to both improved algorithms and improved hardness results, we have seen several cases where the performance of the best algorithm known, based on semidefinite programming, exactly matches (up to lower order terms) the best hardness results based on the UGC. Examples include $\alpha_{GW}+\epsilon$ hardness for MAX CUT [24] (where $\alpha_{GW}\approx 0.8785$ is the approximation ratio of the Goemans-Williamson algorithm), MAX CUT-GAIN [9, 25], Unique Games themselves [8, 24], and $\Theta(k2^{-k})$ approximation of MAX k-CSP [7, 32]. For some of these results, there is no apparent connection between the best hardness result and the best algorithm, apart from the fact that they yield matching approximation ratios. But in some cases, most notably Khot et al.'s hardness for MAX CUT [24] and Khot and O'Donnell's hardness for MAX CUT-GAIN [25], the Long Code tests which are the core components of the hardness results arise in a natural way when studying the corresponding SDP relaxation.

In other words, there appears to be a very strong connection between the power of the semidefinite programming paradigm for designing approximation algorithms, and the power of UGC-based hardness of approximation results. However, this connection is not yet very well understood, and this is still a very active topic of research. The key question is of course whether the UGC is true, indicating that the semidefinite programming paradigm captures the power of polynomial-time computations (assuming $P\neq NP$), or whether the UGC is false, indicating that we might be able to improve upon existing algorithms, but that we would have to come up with some completely new techniques in order to do so. However, resolving this question appears to be well outside the reach of our current techniques.

In this paper, we continue to explore this tight connection between semidefinite programming relaxations and the UGC, by showing hardness of MAX 2-SAT that matches the approximation ratio of the best algorithm known. As with the hardness results for MAX CUT and MAX CUT-GAIN, the parameters for our hardness result arise in the study of worst case configurations for a certain rounding method for the semidefinite relaxation of MAX 2-SAT. This rounding method is significantly more complicated than the rounding method for MAX CUT, and it is interesting that it should yield an apparently optimal approximation ratio.

For MAX 2-SAT and MAX DI-CUT, Goemans and Williamson's algorithms were improved first by Feige and Goemans [14], subsequently by Matuura and Matsui [29, 30], and then by Lewin, Livnat and Zwick [28] who obtained a 0.9401-approximation algorithm for MAX 2-SAT, and a 0.8740-approximation algorithm for MAX DI-CUT. These stand as the current best results for both problems. It should be pointed out that these two ratios arise as the solutions of complex numeric optimization problems. As far as we are aware of, it has not yet been proved formally that these are the actual optima, though there seems to be little doubt that this is indeed the case.

For both problems, better approximation algorithms are known for the special case of so-called *balanced instances*. For MAX 2-SAT this corresponds to the case when every variable occurs negated and unnegated equally often, and for MAX DI-CUT this corresponds to each vertex having the same indegree as outdegree. The approximation ratios achieved are ≈ 0.9439 and α_{GW} respectively, and they match the best known inapproximability ratios under the UGC [24].[1] The best current unconditional hardness results are $21/22+\epsilon \approx 0.9546$ for MAX 2-SAT and $11/12+\epsilon \approx 0.9167$ for MAX DI-CUT [20].

It is natural to conjecture, especially considering these results, that balanced instances should be the hardest (and indeed, Khot et al. [24] do that), i.e., that we should always be able to use the presence of bias as "hints" of how to set the variables. However, as the main result of our paper shows, this might actually not be the case:

THEOREM 1.1. *Assuming the Unique Games Conjecture, for any $\epsilon > 0$ it is NP-hard to approximate* MAX 2-SAT *within $\alpha^-_{LLZ}+\epsilon$, where $\alpha^-_{LLZ} \approx 0.94017$.*

Here, α^-_{LLZ} is the *believed* approximation ratio of Lewin et al.'s MAX 2-SAT algorithm mentioned above. In other words, assuming that their analysis of the algorithm is correct, Theorem 1.1 is tight. The (in our opinion very remote) possibility that their analysis is not correct, i.e., that the approximation ratio of their algorithm is smaller than α^-_{LLZ}, does not affect Theorem 1.1, it would just indicate that it might not be tight, i.e., that MAX 2-SAT might be even harder to approximate than indicated by our result. The reason that the tightness of the result relies on the analysis of Lewin et al. being correct is that our PCP reduction is controlled by a parameter corresponding to a worst-case vector configuration for Lewin et al.'s algorithm. However, the reduction requires this vector configuration to be of a specific form. Fortunately, the (apparently) worst configurations for Lewin et al.'s algorithm are of this form.

A quite surprising part of this result is the "amount" of imbalance: in our hard instances, every variable occurs positively more than twice as often as negatively (the ratio is roughly 68-32).

[1] This is not very surprising, since the balanced versions of both problems are equivalent to the MAX CUT problem with a linear transformation on the scoring function.

The proof relies on a careful analysis of the algorithm of Lewin, Livnat and Zwick. This analysis provides the optimal parameters for a PCP reduction which is very similar to (but more involved than) Khot et al.'s reduction for MAX CUT.

The paper is organized as follows. In Section 2 we set up notation and give some necessary background, including the MAX 2-SAT problem, Fourier analysis, and the Unique Games Conjecture. In Section 3, we discuss Lewin et al.'s MAX 2-SAT algorithm and its approximation ratio. In Section 4 we reduce UNIQUE LABEL COVER to MAX 2-SAT, establishing Theorem 1.1. In Section 5, we conclude and discuss some related open problems.

A full version of this paper is available as [5].

2. PRELIMINARIES

We associate the boolean values true and false with -1 and 1, respectively. Thus, $-x$ denotes "not x", and a disjunction $x \vee y$ is false iff $x = y = 1$.

We denote by $\Phi^{-1} : [0,1] \to \mathbb{R}$ the inverse of the normal distribution function. Furthermore, for $\rho, \mu_1, \mu_2 \in [-1,1]$, we define the function

$$\Gamma_\rho(\mu_1,\mu_2) = \Pr[X_1 \le t_1 \wedge X_2 \le t_2], \tag{1}$$

where $t_i = \Phi^{-1}\left(\frac{1-\mu_i}{2}\right)$ and $X_1, X_2 \in N(0,1)$ with covariance ρ. In other words, Γ_ρ is the bivariate normal distribution function with a transformation on the input. For convenience, we also define $\Gamma_\rho(\mu) = \Gamma_\rho(\mu,\mu)$. The following nice property of Γ_ρ will be very useful to us.

PROPOSITION 2.1. *For all $\rho, \mu_1, \mu_2 \in [-1,1]$, we have*

$$\Gamma_\rho(-\mu_1,-\mu_2) = \Gamma_\rho(\mu_1,\mu_2) + \mu_1/2 + \mu_2/2 \tag{2}$$

A proof can be found in the full version of the paper [5].

2.1 Max 2-Sat

A MAX 2-SAT instance Ψ on a set of n variables consists of a set of clauses, where each clause $\psi \in \Psi$ is a disjunction $l_1 \vee l_2$ on two literals, where each literal is either a variable or a negated variable, i.e., of the form $b \cdot x_i$ for $b \in \{-1,1\}$ and some variable x_i. Additionally, each clause ψ has a nonnegative weight $\mathrm{wt}(\psi)$ (by [11], weighted and unweighted MAX 2-SAT are equally hard to approximate, up to lower order terms). The MAX 2-SAT problem is to find an assignment $x \in \{-1,1\}^n$ of the variables such that the sum of the weights of the satisfied clauses is maximized. MAX 2-SAT can be viewed as an integer programming problem by arithmetizing each clause $(b_1x_i \vee b_2x_j)$ as $\frac{3-b_1x_i-b_2x_j-b_1b_2x_ix_j}{4}$. Note that the latter expression is 1 if the clause is satisfied, and 0 otherwise. The value of an assignment $x \in \{-1,1\}^n$ to Ψ is then

$$\mathrm{Val}_\Psi(x) = \sum_{\psi=(b_1x_i \vee b_2x_j)\in\Psi} \mathrm{wt}(\psi)\cdot\frac{3-b_1x_i-b_2x_j-b_1b_2x_ix_j}{4},$$

and we can write a MAX 2-SAT instance Ψ as the (quadratic) integer program

$$\begin{array}{l}\text{Maximize } \mathrm{Val}_\Psi(x)\\ \text{Subject to } x_i \in \{-1,1\} \quad \forall i\end{array} \tag{3}$$

In this paper, we will be especially interested in the family of MAX 2-SAT instances consisting of the following two clauses for every pair of variables x_i, x_j: the clause $(x_i \vee x_j)$ with weight $\mathrm{wt}_{ij}\cdot\frac{1+\Delta}{2}$, and the clause $(-x_i \vee -x_j)$ with weight $\mathrm{wt}_{ij}\cdot\frac{1-\Delta}{2}$, where the nonnegative weight wt_{ij} controls the "importance" of the pair x_i, x_j (we allow $\mathrm{wt}_{ij} = 0$), and $\Delta \in [-1, 1]$ is a constant controlling the "imbalance" of the instance. Note that if $\Delta = \pm 1$ every variable occurs only positively/negatively, and the instance is trivially satisfiable, whereas if $\Delta = 0$ the instance is balanced and can be approximated within 0.9439. For our hard instances, we will use a carefully chosen Δ which will be approximately 0.3673 (in other words, the relative weight on the positive clauses will be roughly $\frac{1+0.3673}{2} \approx 68\%$).

We will use the terminology Δ-*mixed clause (of weight* wt*)* for a pair of clauses $(x_i \vee x_j)$ with weight $\mathrm{wt} \cdot \frac{1+\Delta}{2}$ and $(-x_i \vee -x_j)$ with weight $\mathrm{wt} \cdot \frac{1-\Delta}{2}$. For a MAX 2-SAT instance Ψ of the above form (i.e. an instance that can be viewed as a set of Δ-mixed clauses), $\mathrm{Val}_\Psi(x)$ can be rewritten as

$$\mathrm{Val}_\Psi(x) = \sum_{i<j} \mathrm{wt}_{ij} \frac{3 - \Delta x_i - \Delta x_j - x_i x_j}{4}. \tag{4}$$

Note that the effect of Δ on the integer program simply constitutes a dampening of the linear terms.

2.2 Fourier analysis and Majority is Stablest

In this section, we give an informal overview of some concepts from Fourier analysis that are used to prove the soundness of the PCP verifier, including the Majority is Stablest Theorem (or rather, an extension of it). Formal definitions and statements can be found in Appendix A.

The study of properties of the Fourier expansion of Boolean functions has yielded many important results, both in computer science and other fields. In particular, Fourier analysis has been a key technique when analyzing the soundness of PCP verifiers.

The *long code* of an integer $i \in [n]$ is a bit string $b_1 b_2 \dots b_N$ of length $N = 2^n$. Identifying an index $I \in [N]$ with a subset $S(I) \subseteq [n]$ bijectively in some canonical way, the Ith bit of the long code of i is -1 if $i \in S(I)$, and 1 otherwise. Put another way, we can view it as the table of a function f on n bits such that $f(x_1, \dots, x_n) = x_i$, i.e., a dictator function.

The *low-degree influence* of a variable x_i on a boolean function f on n variables is, in a *very* loose sense, a measure of how close the function f is to depending only on the variable x_i, i.e., how close f is to being the long code of $i \in [n]$.

The *noise stability* of a Boolean function f is a measure of how much the function tends to change when flipping a certain fraction of the input bits.

For proving hardness of MAX CUT, Khot et al. [24] made a conjecture called Majority Is Stablest, essentially stating that any Boolean function with noise stability significantly higher than the majority function must have a variable with high low-degree influence (and thus in a sense be close to a long code). Majority Is Stablest was subsequently proved by Mossel et al. [31], using a very powerful invariance principle which, essentially, allows for considering the corresponding problem over Gaussian space instead. For our result, we use a generalization of Majority is Stablest due to Dinur et al. [12].

2.3 The Unique Games Conjecture (UGC)

The Unique Games Conjecture was introduced by Khot [23] as a possible means to obtain new strong inapproximability results. As is common, we will formulate it in terms of a Label Cover problem.

DEFINITION 2.2. *An instance*

$$X = (V, E, \mathrm{wt}, [L], \{\sigma_e^v, \sigma_e^w\}_{e=\{v,w\}\in E})$$

of UNIQUE LABEL COVER *is defined as follows: given is a weighted graph* $G = (V, E)$ *(which may have multiple edges) with weight function* $\mathrm{wt} : E \to [0, 1]$*, a set* $[L]$ *of allowed labels, and for each edge* $e = \{v, w\} \in E$ *two permutations* $\sigma_e^v, \sigma_e^w \in \mathfrak{S}_L$ *such that* $\sigma_e^w = (\sigma_e^v)^{-1}$*, i.e., they are each other's inverse. We say that a function* $\ell : V \to [L]$ *(called a labelling of the vertices) satisfies an edge* $e = \{v, w\}$ *if* $\sigma_e^v(\ell(v)) = \ell(w)$*, or equivalently, if* $\sigma_e^w(\ell(w)) = \ell(v)$*. The value of* ℓ *is the total weight of edges satisfied by it, i.e.,*

$$\mathrm{Val}_X(\ell) = \sum_{\substack{e \\ \ell \text{ satisfies } e}} \mathrm{wt}(e) \tag{5}$$

The value of X *is the maximum fraction of satisfied edges for any labelling, i.e.,*

$$\mathrm{Val}(X) = \max_{\ell} \mathrm{Val}_X(\ell). \tag{6}$$

Without loss of generality, we will always assume that $\sum_e \mathrm{wt}(e) = 1$, i.e., that wt is in fact a probability distribution over the edges of X. We denote by $E(v)$ the subset of edges adjacent to v, i.e., $E(v) = \{\, e \,|\, v \in e \,\}$. The probability distribution wt induces a natural probability distribution on the vertices of X where the probability of choosing v is $\frac{1}{2}\sum_{e\in E(v)} \mathrm{wt}(e)$, and wt also induces a natural distribution on the edges of $E(v)$ where the probability of choosing $e \in E(v)$ is $\frac{\mathrm{wt}(e)}{\sum_{e\in E(v)} \mathrm{wt}(e)}$.

Whenever we speak of choosing a random element of V, E or $E(v)$, it will be according to these probability distributions, but to simplify the presentation, we will simply refer to it as a random element. For the same reason we will refer to a fraction c of the elements of V, E or $E(V)$ when in fact we mean a set of vertices/edges with probability mass c.

A UNIQUE LABEL COVER problem where G is bipartite can be viewed as a two-prover (one-round) game in which the acceptance predicate of the verifier is such that given the answer for one of the provers, there is always a unique answer from the other prover such that the verifier accepts. The probability that the verifier accepts assuming that the provers use an optimal strategy is then $\mathrm{Val}(X)$. Hence the terminology "Unique Games". We will be interested in the gap version of the UNIQUE LABEL COVER problem, which we define as follows.

DEFINITION 2.3. GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ *is the problem, given a* UNIQUE LABEL COVER *instance* X *with label set* $[L]$*, to determine whether* $\mathrm{Val}(X) \geq 1 - \eta$ *or* $\mathrm{Val}(X) \leq \gamma$.

Khot's Unique Games Conjecture (UGC) then asserts that the gap version is hard to solve for arbitrarily small η and γ, provided we take a sufficiently large label set.

CONJECTURE 2.4 (UNIQUE GAMES CONJECTURE [23]). *For every* $\eta > 0$, $\gamma > 0$*, there is a constant* $L > 0$ *such that* GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ *is NP-hard.*

Note that even if the UGC turns out to be false, it might still be the case that GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ is hard in the sense of not being solvable in polynomial time, and such a (weaker) hardness would also apply to MAX 2-SAT and (as far as we are aware, all) other problems for which hardness has been shown under the UGC.

3. APPROXIMATING MAX 2-SAT

To approximate MAX 2-SAT, the common approach is to relax the integer program Equation (3) to a semidefinite program by relaxing each variable x_i to a vector $v_i \in \mathbb{R}^{n+1}$. In addition, we introduce the variable $v_0 \in \mathbb{R}^{n+1}$, which is supposed to encode the value "false". The constraint $x_i \in \{-1, 1\} = S^0$ translates to the constraint that $v_i \in S^n$, i.e., that each vector v_i should be a unit vector. The value of an assignment $v = (v_0, \ldots, v_n) \in (S^n)^{n+1}$ to the relaxation is then

$$\text{SDP-Val}_\Psi(v) = \sum_{\substack{\psi=(b_1x_i \vee b_2x_j)\\ \psi\in\Psi}} \text{wt}(\psi) \cdot \frac{3 - b_1 v_i \cdot v_0 - b_2 v_j \cdot v_0 - b_1 b_2 v_i \cdot v_j}{4},$$

where $v_i \cdot v_j$ is the standard inner product on vectors in $\mathbb{R}^n$.

This semidefinite relaxation was studied by Goemans and Williamson [16]. For their improved approximation algorithm, Feige and Goemans [14] considered a strengthening of this semidefinite program, by adding, for each triple $\{v_i, v_j, v_k\} \subseteq \{v_0, \ldots, v_n\}$ the triangle inequalities

$$\begin{aligned} v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k &\geq -1 \\ -v_i \cdot v_j + v_i \cdot v_k - v_j \cdot v_k &\geq -1 \\ v_i \cdot v_j - v_i \cdot v_k - v_j \cdot v_k &\geq -1 \\ -v_i \cdot v_j - v_i \cdot v_k + v_j \cdot v_k &\geq -1. \end{aligned}$$

These are equivalent to inequalities of the form $||v_i - v_j||^2 + ||v_j - v_k||^2 \geq ||v_i - v_k||^2$, which clearly holds for the case that all vectors lie in a one-dimensional subspace of S^n (so this is still a relaxation of the original integer program), but may not necessarily be true otherwise.

In general, we cannot find the exact optimum of a semidefinite program. It is however possible to find the optimum to within an additive relative error of ϵ in time polynomial in $\log 1/\epsilon$ [1]. Since this error is small enough for our purposes, we will ignore this small point for notational convenience and assume that we can solve the semidefinite program exactly.

Given solution vectors $(v_0, \ldots, v_n)$ maximizing $\text{SDP-Val}_\Psi(v)$, we will produce a solution $(x_1, \ldots x_n) \in \{-1, 1\}^n$ using some rounding method, which will typically be randomized. For consistency, we require that this rounding method always rounds v_i and $-v_i$ to opposite values. To determine the approximation ratio of the algorithm, we analyze the worst possible approximation ratio on the clause $(x_i \vee x_j)$ for any vector configuration.[2] This gives a lower bound on the approximation ratio:

$$\min_{v\in(S^n)^{n+1}} \frac{\mathbb{E}[3 - x_i - x_j - x_i x_j]}{3 - v_0 \cdot v_i - v_0 \cdot v_j - v_i \cdot v_j}, \tag{7}$$

where the minimum is over all *feasible* vector solutions to the SDP, and the expected value is over the randomness of the rounding method. Typically, the rounding of the vector v_i will only depend on v_0 and v_i, and so the minimum in Equation (7) only needs to be taken over the three vectors v_0, v_i and v_j.

3.1 The LLZ algorithm

The best approximation algorithm known for MAX 2-SAT is due to Lewin, Livnat and Zwick [28] (hereafter referred to as the LLZ algorithm). It uses the SDP relaxation described above, including the triangle inequalities. In order to describe the rounding method, it is convenient to introduce some notation. Given a solution $(v_0, \ldots, v_n)$ to the SDP, we define $\xi_i = v_0 \cdot v_i$ and $v_i = \xi_i v_0 + \sqrt{1 - \xi_i^2}\tilde{v}_i$, i.e., $\tilde{v}_i$ is the part of v_i orthogonal to v_0, normalized to a unit vector.

Lewin et al. consider the following general class of rounding methods, which they call $\mathcal{THRESH}^-$: First, a standard normal random vector r is chosen in the n-dimensional subspace of $\mathbb{R}^{n+1}$ orthogonal to v_0. Then, the variable x_i is set to true iff $\tilde{v}_i \cdot r \leq T(\xi_i)$, where the threshold function $T(\cdot)$ is (almost) arbitrary, and it is convenient for us to have it on the form

$$T(x) = \Phi^{-1}\left(\frac{1 - a(x)}{2}\right), \tag{8}$$

where $a : [-1, 1] \rightarrow [-1, 1]$ is an (almost) arbitrary function.[3] The consistency requirement on the rounding method translates to requiring that T is an odd function (or equivalently, that a is an odd function).

The reason that it is natural to formulate T in terms of the function a becomes evident when we analyze the performance ratio of the algorithm. Note that $\tilde{v}_i \cdot r$ is a standard $N(0, 1)$ variable, implying that x_i is set to true with probability $\frac{1-a(\xi_i)}{2}$. In other words, the expected value of x_i is simply $\mathbb{E}[x_i] = a(\xi_i)$, and thus, we can think of the function a as controlling exactly how much we lose on the linear terms when we round the solution to the semidefinite program.

In order to evaluate the performance of the algorithm, we also need to analyze performance on the quadratic terms, which we do by analyzing the probability that two variables x_i and x_j are rounded to the same value. Let $\rho := v_i \cdot v_j$ and $\tilde{\rho} := \tilde{v}_i \cdot \tilde{v}_j = \frac{\rho - \xi_i\xi_j}{\sqrt{(1-\xi_i^2)(1-\xi_j^2)}}$. It is readily verified that the scalar products $\tilde{v}_i \cdot r$ and $\tilde{v}_j \cdot r$ are standard $N(0, 1)$ variables with covariance $\tilde{\rho}$, and thus, the probability that $\tilde{v}_i \cdot r \leq T(\xi_i)$ and $\tilde{v}_j \cdot r \leq T(\xi_j)$ is simply $\Gamma_{\tilde{\rho}}(a(\xi_i), a(\xi_j))$ (see Section 2 for the definition of Γ). By symmetry, the probability that both x_i and x_j are set to false is $\Gamma_{\tilde{\rho}}(-a(\xi_i), -a(\xi_j))$. Using Proposition 2.1, we get that the expected value of the term $x_i x_j$ is

$$2\Pr[x_i = x_j] - 1 = 4\Gamma_{\tilde{\rho}}(a(\xi_i), a(\xi_j)) + a(\xi_i) + a(\xi_j) - 1,$$

and the expected value of the clause $x_i \vee x_j$ becomes

$$\frac{3 - \mathbb{E}[x_i] - \mathbb{E}[x_j] - \mathbb{E}[x_i x_j]}{4} = \frac{4 - 2a(\xi_i) - 2a(\xi_j) - 4\Gamma_{\tilde{\rho}}(a(\xi_i), a(\xi_j))}{4}$$

It turns out that to get the best approximation ratio, we should choose $a(x) := \beta \cdot x$ to be a linear function, where $\beta \approx 0.94016567$, the apparent approximation ratio [33]. This is not quite the same choice as originally described by Lewin et al., but is more natural and achieves a marginally better approximation ratio. See Appendix C for details on the difference between the two choices rounding functions. Next, define

$$\alpha_\beta(\xi_i, \xi_j, \rho) = \frac{4 - 2\beta(\xi_i + \xi_j) - 4\Gamma_{\tilde{\rho}}(\beta\xi_i, \beta\xi_j)}{3 - \xi_i - \xi_j - \rho}, \tag{9}$$

i.e., the expected approximation ratio of the configuration (ξ_i, ξ_j, ρ), using a specific choice of β. Let

$$\alpha(\beta) = \min_{\xi_i,\xi_j,\rho} \alpha_\beta(\xi_i, \xi_j, \rho), \tag{10}$$

i.e., a lower bound on the approximation ratio achieved by this algorithm for a specific β, where (ξ_i, ξ_j, ρ) ranges over all configu-

[2] Note that because of the consistency requirement, the approximation ratio on, e.g., the clause $(-x_i \vee x_j)$ for some vector configuration $(v_0, \ldots, v_n)$ equals the approximation ratio on the clause $(x_i \vee x_j)$ with v_i negated, and similarly for other clauses with negated variables.

[3] In the notation of [28], this corresponds to setting $S(x) = T(x)\sqrt{1 - x^2}$, or $a(x) = 1 - 2\Phi\left(S(x)/\sqrt{1 - x^2}\right)$ (we may, without loss of generality, assume that $\xi_i \neq \pm 1$ for all i).

rations satisfying the triangle inequalities. Finally, let

$$\alpha_{LLZ} = \max_{\beta \in [-1,1]} \alpha(\beta), \tag{11}$$

i.e., a lower bound on the best possible approximation ratio when letting a be any linear function.

3.2 Simple configurations

We represent a vector configuration for the SDP by the three scalar products (ξ_i, ξ_j, ρ), where $\rho = v_i \cdot v_j$. When showing hardness of MAX 2-SAT, we will reduce UNIQUE LABEL COVER to MAX 2-SAT. The reduction is parametrized by a configuration (ξ_i, ξ_j, ρ) of the SDP, yielding a hardness result matching the performance of the LLZ algorithm on this configuration. However, the reduction needs this configuration to be of a specific form.

First, it needs the configuration to satisfy $\xi_i = \xi_j$, in other words, that both v_i and v_j have the same angle to v_0. This restriction is not entirely artificial; considering the symmetry of the linear terms in the quadratic program, it seems intuitive that the weight on the two linear terms should be distributed fifty-fifty for a worst case configuration, i.e., that $\xi_i = \xi_j$.

Second, the reduction needs the configuration to satisfy $-2|\xi_i| + \rho = -1$, in other words, that we have equality in one of the triangle inequalities. This restriction is quite natural; the triangle inequalities cut away a part of the configuration space in which there are extremely bad configurations, and sticking as close as possible to this part of the configuration space would intuitively seem like a good approach for finding bad configurations.

We will refer to a configuration satisfying these two criterions, i.e., a configuration of the form $(\xi, \xi, -1 + 2|\xi|)$ for some $\xi \in [-1, 1]$, as a *simple configuration* ξ. Extensive numerical computations, both our own and those of Lewin et al., indicate that the worst case configurations for the LLZ algorithm are indeed simple.

Motivated by this restriction to simple configurations, we define

$$\alpha_\beta^-(\xi) = \alpha_\beta(\xi, \xi, -1 + 2|\xi|) = \frac{2 - 2\beta\xi - 2\Gamma_{\tilde{\rho}}(\beta\xi)}{2 - \xi - |\xi|} \tag{12}$$

to be the expected approximation ratio on a specific simple configuration ξ, where $\tilde{\rho} = \frac{-1+2|\xi|-\xi^2}{1-\xi^2} = \frac{|\xi|-1}{|\xi|+1}$ is the value of $\tilde{\rho}$ for the simple configuration ξ. Analogously to $\alpha(\beta)$ and α_{LLZ}, let

$$\alpha^-(\beta) = \min_{\xi \in [-1,1]} \alpha_\beta^-(\xi) \tag{13}$$

$$\alpha_{LLZ}^- = \max_{\beta \in [-1,1]} \alpha^-(\beta), \tag{14}$$

i.e., lower bounds on the approximation ratio for a specific choice of β and the best approximation ratio for any choice of β, when only considering simple configurations. Clearly, we have $\alpha_{LLZ} \leq \alpha_{LLZ}^-$, and unless Lewin et al.'s analysis is wrong, we have equality. In Appendix B, we briefly discuss the actual numeric value of $\alpha_{LLZ}^- \approx 0.94017$.

It is possible to show that the right hand side of Equation (14) is indeed maximized by setting $\beta = \alpha_{LLZ}^-$ (a proof is given in the full version of this paper [5]), and in fact, this will be needed in order to obtain an expression for α_{LLZ}^- that exactly matches the inapproximability yielded by the reduction from UNIQUE LABEL COVER.

4. REDUCTION FROM UNIQUE LABEL COVER

In this section, we reduce UNIQUE LABEL COVER to MAX 2-SAT. Let $\epsilon > 0$. We will show hardness of approximating MAX 2-SAT within $\alpha_{LLZ}^- + \mathcal{O}(\epsilon)$. Let $\eta > 0$ and $\gamma > 0$ be parameters which will be chosen sufficiently small and let L be the corresponding label size given by the UGC. We will reduce GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ to the problem of approximating MAX 2-SAT via a PCP verifier whose queries correspond to checking a Δ-mixed MAX 2-SAT clause. The reduction is controlled by a parameter $\xi \in (-1, 0)$ and an imbalance parameter $\Delta \in (-1, 1)$, the values of which will be chosen later.

ith bit		Probability
x_1	x_2	
1	1	$(\lvert\xi\rvert + \xi)/2 = 0$
-1	1	$(1 - \lvert\xi\rvert)/2 = (1 + \xi)/2$
1	-1	$(1 - \lvert\xi\rvert)/2 = (1 + \xi)/2$
-1	-1	$(\lvert\xi\rvert - \xi)/2 = -\xi$

Table 1: Distribution of the ith bit of x_1 and x_2 (recall that $\xi < 0$).

Given is a UNIQUE LABEL COVER instance

$$X = (V, E, [L], \{\sigma_e^v\}_{e=\{v,w\}\in E}).$$

A proof Σ that X is $(1 - \eta)$-satisfiable will consist of supposed long codes of the labels of all $v \in V$. Denote by $f_v : \{-1, 1\}^L \to \{-1, 1\}$ the purported long code of the label of vertex v. For a permutation $\sigma \in \mathfrak{S}_L$ and $x = x_1 \dots x_L \in \{-1, 1\}^L$, we let $\sigma x = x_{\sigma(1)} \dots x_{\sigma(L)}$. The PCP verifier $\mathcal{V}$ is described in Algorithm 1.

Algorithm 1: The verifier $\mathcal{V}$

$\mathcal{V}(X, \Sigma = \{f_v\}_{v \in V})$

(1) Pick a random $v \in V$.

(2) Pick $e_1 = \{v, w_1\}$ and $e_2 = \{v, w_2\}$ randomly from $E(v)$.

(3) Pick $x_1, x_2 \in \{-1, 1\}^L$ such that each bit of x_j is picked independently with expected value ξ and that the ith bits of x_1 and x_2 are $(-1 + 2|\xi|)$-correlated (see Table 1).

(4) For $i = 1$, 2, let $b_i = f_{w_i}(\sigma_{e_i}^v x_i)$.

(5) With probability $\frac{1+\Delta}{2}$, accept iff $b_1 \vee b_2$.

(6) Otherwise, i.e., with probability $\frac{1-\Delta}{2}$, accept iff $-b_1 \vee -b_2$.

The completeness and soundness of $\mathcal{V}$ are as follows.

LEMMA 4.1 (COMPLETENESS). *If* $\mathrm{Val}(X) \geq 1 - \eta$, *then there is a proof* Σ *that makes* $\mathcal{V}$ *accept with probability at least*

$$(1 - 2\eta)\frac{2 - \Delta\xi - |\xi|}{2} \tag{15}$$

LEMMA 4.2 (SOUNDNESS). *For any* $\epsilon > 0$, $\xi \in (-1, 0)$ *and* $\Delta \in (-1, 1)$ *there exists a* $\gamma > 0$, *such that if* $\mathrm{Val}(X) \leq \gamma$, *then for any proof* Σ, *the probability that* $\mathcal{V}$ *accepts is at most*

$$\max_{\mu \in [-1,1]} \frac{2 - (1 + \Delta)\mu - 2\Gamma_{\tilde{\rho}}(\mu)}{2} + \epsilon, \tag{16}$$

where $\tilde{\rho} = \frac{|\xi|-1}{|\xi|+1}$.

Proofs of Lemmas 4.1 and 4.2 can be found in Appendix D. Combining the Lemmas and picking η small enough, we get that, assuming the UGC, it is NP-hard to approximate MAX 2-SAT within a factor

$$\max_{\mu \in [-1,1]} \frac{2 - (1 + \Delta)\mu - 2\Gamma_{\tilde{\rho}}(\mu)}{2 - \Delta\xi - |\xi|} + \mathcal{O}(\epsilon)\,. \tag{17}$$

As a final step, we show that, choosing the right ξ and Δ, the first term is exactly α^-_{LLZ}.

PROPOSITION 4.3. *There are $\xi \in (-1, 0)$ and $\Delta \in (-1, 1)$ such that*

$$\alpha^-_{LLZ} = \max_{\mu \in [-1,1]} \frac{2 - (1+\Delta)\mu - 2\Gamma_{\tilde{\rho}}(\mu)}{2 - \Delta\xi - |\xi|}, \qquad (18)$$

where $\tilde{\rho} = \frac{|\xi|-1}{|\xi|+1}$.

A proof is given in the full version of this paper [5]. Applying Proposition 4.3 to Equation (17), we obtain Theorem 1.1.

The values of ξ and Δ given by Proposition 4.3 are roughly $\xi \approx -0.1625$, $\Delta \approx 0.3673$. The large value of Δ in particular is interesting, since the weights on positive and negative occurences of variables are $\frac{1+\Delta}{2}$ and $\frac{1-\Delta}{2}$, which is roughly 68% vs. 32%. We find it remarkable that so greatly imbalanced instances should be the hardest to approximate. We remark that the choice of sign for ξ is arbitrary (it corresponds to the choice of whether most of the variable occurences in our hard MAX 2-SAT instance should be positive or negative), the proposition holds for $\xi \in (0, 1)$ as well.

Also, note the strong connection between the LLZ algorithm and the PCP reduction. On a high level, the PCP verifier chooses some configuration of vectors, and in the soundness case, a good strategy for the prover is essentially just a rounding method (from the class of rounding methods considered by Lewin et al.) which has a good performance on the SDP configurations chosen by the verifier. Chosing a configuration of vectors which is particularly difficult to round, we get a good verifier.

5. CONCLUDING REMARKS

We have shown that it is hard to approximate MAX 2-SAT within $\alpha^-_{LLZ} + \epsilon$. The constant $\alpha^-_{LLZ} \approx 0.94017$ is the guaranteed performance ratio of the LLZ algorithm on vector configurations which are of a certain form which we call simple configurations. Furthermore, all numerical evidence (both that of Lewin et al., and our own computations), heavily indicates that the worst possible configurations for the LLZ algorithm are simple – in other words that the approximation ratio of the LLZ algorithm is α^-_{LLZ}, and that our result is tight.

5.1 Open problems and further work

Beside the obvious importance of resolving the Unique Games Conjecture, there are a few other, quite possibly easier, questions that would be nice to settle.

- Given the result in this paper and previous works on integrality gap for e.g. MAX CUT [27], it seems likely that we should be able to show a matching integrality gap for the SDP relaxation of MAX 2-SAT (since otherwise, the UGC would be false, and it seems unlikely that a careful analysis of the MAX 2-SAT SDP should be enough to disprove the conjecture). So far, however, our attempts at showing this has been elusive.
- It would be nice to have a proof that there are worst configurations for the LLZ algorithm that are simple, i.e., that the performance ratio is indeed α^-_{LLZ}.
- It would be interesting to determine how the hardness of approximating MAX 2-SAT depends on the imbalance of the instances considered (for a suitable definition of imbalance for general instances and not just instances consisting only of Δ-mixed clauses). For instance, how large can we make the imbalance and still have instances that are hard to approximate within, say, 0.95?

5.2 Acknowledgements

I would like to thank Johan Håstad for suggesting that I work with the MAX 2-SAT problem in the first place, as well as for his patience and many insightful comments along the way. I would also like to thank Uri Zwick for discussing the LLZ algorithm with me.

6. REFERENCES

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[2] S. Arora, E. Chlamtac, and M. Charikar. New Approximation Guarantee for Chromatic Number. In *STOC 2006*, pages 205–214, 2006.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and the Hardness of Approximation Problems. *Journal of the ACM*, 45(3):501–555, 1998.

[4] S. Arora and S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[5] P. Austrin. Balanced Max 2-Sat might not be the hardest. Technical report, Electronic Colloquium on Computational Complexity Report TR06-088, 2006.

[6] A. Blum and D. Karger. An $\tilde{\mathcal{O}}(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Information Processing Letters*, 61(1):49–53, 1997.

[7] M. Charikar, K. Makarychev, and Y. Makarychev. Approximation Algorithm for the Max k-CSP Problem, 2006.

[8] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for unique games. In *STOC 2006*, pages 205–214, 2006.

[9] M. Charikar and A. Wirth. Maximizing Quadratic Programs: Extending Grothendieck's Inequality. In *FOCS 2004*, pages 54–60, 2004.

[10] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *20th Annual IEEE Conference on Computational Complexity*, pages 144–153, June 2005.

[11] P. Crescenzi, R. Silvestri, and L. Trevisan. On Weighted vs Unweighted Versions of Combinatorial Optimization Problems. *Information and Computation*, 167(1):10–26, 2001.

[12] I. Dinur, E. Mossel, and O. Regev. Conditional hardness for approximate coloring. In *STOC 2006*, pages 344–353, 2006.

[13] U. Feige. A threshold of ln *n* for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[14] U. Feige and M. Goemans. Aproximating the Value of Two Prover Proof Systems, With Applications to MAX 2SAT and MAX DICUT. In *ISTCS 1995*, pages 182–189, 1995.

[15] U. Feige and J. Kilian. Zero Knowledge and the Chromatic Number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998.

[16] M. X. Goemans and D. P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, 42:1115–1145, 1995.

[17] E. Halperin, R. Nathaniel, and U. Zwick. Coloring

k-colorable graphs using smaller palettes. In *SODA 2001*, pages 319–326, 2001.

[18] G. Hast. Approximating Max kCSP – Outperforming a Random Assignment with Almost a Linear Factor. In *ICALP 2005*, pages 956–968, 2005.

[19] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 48(4):105–142, 1999.

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

[21] J. Håstad. On the approximation resistance of a random predicate. Manuscript, 2006.

[22] D. R. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.

[23] S. Khot. On the power of unique 2-prover 1-round games. In *STOC 2002*, pages 767–775, 2002.

[24] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal Inapproximability Results for Max-Cut and Other 2-Variable CSPs? In *FOCS 2004*, pages 146–154. IEEE Computer Society, 2004.

[25] S. Khot and R. O'Donnell. SDP gaps and UGC-hardness for MAXCUTGAIN. In *FOCS*, pages 217–226. IEEE Computer Society, 2006.

[26] S. Khot and O. Regev. Vertex Cover Might be Hard to Approximate to within $2-\epsilon$. In *IEEE Conference on Computational Complexity*, pages 379–. IEEE Computer Society, 2003.

[27] S. Khot and N. K. Vishnoi. The Unique Games Conjecture, Integrality Gap for Cut Problems and Embeddability of Negative Type Metrics into l_1. In *FOCS 2005*, pages 53–62, 2005.

[28] M. Lewin, D. Livnat, and U. Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *IPCO 2002*, volume 2337 of *Lecture Notes in Computer Science*, pages 67–82, 2002.

[29] S. Matuura and T. Matsui. 0.863-approximation algorithm for max dicut. In *RANDOM-APPROX*, pages 138–146, 2001.

[30] S. Matuura and T. Matsui. 0.935-Approximation Randomized Algorithm for MAX 2SAT and Its Derandomization. Technical Report METR 2001-03, Department of Mathematical Engineering and Information Physics, the University of Tokyo, Japan, 2001.

[31] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. Preprint, 2005.

[32] A. Samorodnitsky and L. Trevisan. Gowers uniformity, influence of variables, and PCPs. In *STOC 2006*, pages 11–20, 2006.

[33] U. Zwick. Personal communication, 2005.

APPENDIX

A. FOURIER ANALYSIS AND MAJORITY IS STABLEST

Fourier analysis (of Boolean functions) is a crucial tool in most strong inapproximability results. Since we need to work with biased distributions rather than the standard uniform ones, we will review some important concepts. The facts in this section are well-known, and proofs can be found in e.g. [5]. We denote by μ_q^n the probability distribution on $\{-1,1\}^n$ where each bit is set to -1 with probability q, independently, and we let B_q^n be the probability space $\left(\{-1,1\}^n, \mu_q^n\right)$. We define a scalar product on the space of functions from B_q^n to $\mathbb{R}$ by

$$\langle f, g\rangle = \mathop{\mathbb{E}}_{x\in B_q^n}[f(x)g(x)], \tag{19}$$

and for each $S \subseteq [n]$ the function $U_q^S : B_q^n \to \mathbb{R}$ by $U_q^S(x) = \prod_{i\in S} U_q(x_i)$ where

$$U_q(x_i) = \begin{cases} -\sqrt{\frac{1-q}{q}} & \text{if } x_i = -1 \\ \sqrt{\frac{q}{1-q}} & \text{if } x_i = 1 \end{cases}.$$

It is a well known fact that the set of functions $\{U_q^S\}_{S\subseteq[n]}$ forms an orthonormal basis w.r.t. the scalar product $\langle\cdot,\cdot\rangle$, and thus, any function $f : B_q^n \to \mathbb{R}$ can be written as

$$f(x) = \sum_{S\subseteq[n]} \hat{f}_S U_q^S(x).$$

The coefficients $\hat{f}_S = \left\langle f, U_q^S\right\rangle$ are the Fourier coefficients of the function f. A concept that is very important in PCP applications is that of low-degree influence.

DEFINITION A.1. *For $k \in \mathbb{N}$, the* low-degree influence *of the variable i on the function $f : B_q^n \to \mathbb{R}$ is*

$$\mathrm{Inf}_i^{\le k}(f) = \sum_{\substack{S\subseteq[n]\\ i\in S\\ |S|\le k}} \hat{f}_S^2. \tag{20}$$

A nice property of the low-degree influence is the fact that for $f : B_q^n \to \{-1,1\}$, we have $\sum_i \mathrm{Inf}_i^{\le k}(f) \le k$, implying that the number of variables having low-degree influence more than τ must be small (think of k and τ as constants not depending on the number of variables n). Informally, one can think of the low-degree influence as a measure of how close the function f is to depending *only* on the variable i, i.e., for the case of boolean-valued functions, how close f is to being the long code of i (or its negation).

Next, we define the Beckner operator T_ρ on a function $f : B_q^n \to \mathbb{R}$. For the unbiased distribution $q = 1/2$, $T_\rho f(x)$ is simply the expectation of $f(y)$ over a random variable y that is ρ-correlated with x. For biased distributions, the definition is a bit more complicated.

DEFINITION A.2. *Given $\rho \in [-1,1]$ satisfying $\rho \ge -\frac{q}{1-q}$ and $\rho \ge -\frac{1-q}{q}$, the Beckner operator T_ρ on a function $f : B_q^n \to \mathbb{R}$ is defined by*

$$T_\rho f(x) = \mathop{\mathbb{E}}_{y}[f(y)]. \tag{21}$$

where the expectation is over an n-bit string y in which each bit y_i is picked independently as follows: if $x_i = 1$ then $y_i = -x_i$ with probability $q(1-\rho)$, and if $x_i = -1$ then $y_i = -x_i$ with probability $(1-q)(1-\rho)$ (see Table 2).

x_i	b	$\Pr[y_i = b \mid x_i]$
1	1	$1-q(1-\rho)$
1	-1	$q(1-\rho)$
-1	1	$(1-q)(1-\rho)$
-1	-1	$1-(1-q)(1-\rho)$

Table 2: Distribution of y_i depending on x_i.

Note that the lower bound on ρ is needed to make this a valid probability distribution. For $\rho \ge 0$, the probability distribution of y_i

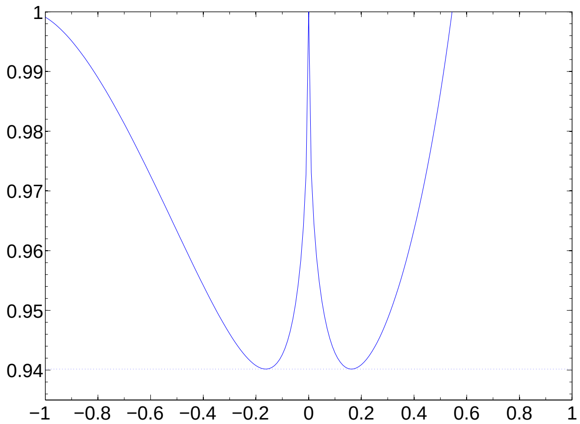

Figure 1: $\alpha^-_{0.94016567248}(\xi)$

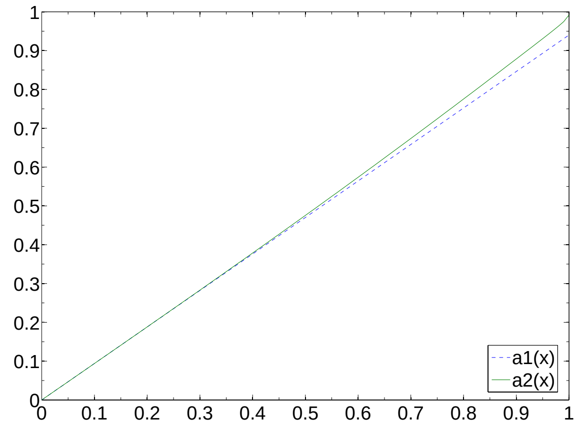

Figure 2: $a_1(x)$ **vs.** $a_2(x)$

can be formulated as follows: with probability ρ, we let $y_i = x_i$, and with probability $1-\rho$, we pick y_i from B^1_q.

The effect of T_ρ on f can also be expressed using the Fourier representation of f as follows:

$$T_\rho f(x) = \sum_{S\subseteq[n]} \rho^{|S|} \hat{f}_S U^S_q(x). \tag{22}$$

DEFINITION A.3. *The* noise stability *of* $f : B^n_q \to \mathbb{R}$ *is*

$$\mathbb{S}_\rho(f) = \langle f, T_\rho f\rangle = \sum_{S\subseteq[n]} \rho^{|S|} \hat{f}^2_S \tag{23}$$

Finally, we state a simplified version of Dinur et al.'s generalization of the Majority is Stablest theorem [12].

THEOREM A.4. *Let* $\epsilon > 0$, $q \in (0,1)$ *and* $\rho \in (-1,0)$. *Then there is a* $\tau > 0$ *and a* $k \in \mathbb{N}$ *such that for every function* $f : B^n_q \to [-1,1]$ *satisfying* $\mathbb{E}[f] = \mu$ *and* $\mathrm{Inf}^{\le k}_i(f) \le \tau$ *for all* i, *we have*

$$\mathbb{S}_\rho(f) \ge 4\Gamma_\rho(\mu) + 2\mu - 1 - \epsilon. \tag{24}$$

B. THE NUMERIC VALUE OF α^-_{LLZ}

In this section we will (very briefly) discuss the actual numeric value of α^-_{LLZ}. Let $b = 0.9401656724$. To give a feel for $\alpha^-_b(\xi)$, Figure 1 gives a plot of this function in the interval $\xi \in [-1,1]$, along with the line $y = b$ (dashed). The one-dimensional optimization problem

$$\min_\xi \alpha_b(\xi) \tag{25}$$

can be solved numerically to a high level of precision. This gives a lower bound $\alpha^-_{LLZ} \ge 0.9401656724$. The two minima seen in Figure 1 turn out to be roughly $\xi_1 = -0.1624783294$ and $\xi_2 = 0.1624783251$. In order to obtain an upper bound on α^-_{LLZ}, we can then solve the one-dimensional optimization problem

$$\max_\beta \min\left(\alpha^-_\beta(\xi_1), \alpha^-_\beta(\xi_2)\right) \tag{26}$$

numerically to a high level of precision. This results in an upper bound of $\alpha^-_{LLZ} \le 0.9401656725$. In conclusion, we have $|\alpha^-_{LLZ} - 0.94016567245| \le 5\cdot 10^{-11}$.

C. THE TALE OF THE TWO ROUNDING FUNCTIONS

The rounding function of the LLZ algorithm that is used in this paper is due to Zwick [33], and differs from the rounding function originally used by Lewin et al. [28]. The rounding function we use is $a_1(x) = \beta\cdot x$, where $\beta = \alpha^-_{LLZ} \approx 0.94016567$ (see Section 3.1 for further details). The rounding function used in [28] is $a_2(x) = 1-2\Phi(S(x)/\sqrt{1-x^2})$. Here, $S(x) = -2\cot(f(\arccos x))\sqrt{1-x^2}$ where f is a linear function given by

$$f(\theta) \approx 0.588314558\theta + 0.64667394. \tag{27}$$

$a_2(x)$ can be simplified to

$$\begin{aligned} a_2(x) &= 1 - 2\Phi(-2\cot(f(\arccos x))) \\ &= 2\Phi(2\cot(f(\arccos x))) - 1. \end{aligned} \tag{28}$$

Figure 2 gives plots of the functions $a_1(x)$ and $a_2(x)$ for the interval $x \in [0,1]$ (both functions are odd, so we restrict our attention to positive x). As can be seen, the functions are fairly close to each other. Most importantly, the functions behave almost the same in the critical interval $x \in [0.1, 0.2]$. Nevertheless, there is a small difference between the functions in this interval as well, and this causes the worst simple configurations $\xi \approx \pm 0.1625$ when using $a_1(x)$ to be slightly different from the worst simple configurations $\xi \approx \pm 0.169$ when using $a_2(x)$. This small difference results in a marginally better approximation ratio when using $a_1(x)$ than when using $a_2(x)$, but the improvement is very small.

For large x, the functions $a_1(x)$ and $a_2(x)$ differ noticeably, but here the particular choice of rounding function is not crucial since these are configurations that are in some sense easy to round, and any function with a reasonable behaviour suffices to get a sufficiently good approximation ratio.

D. PROOFS OF COMPLETENESS AND SOUNDNESS FOR THE VERIFIER

In this section, we prove Lemmas 4.1 and 4.2, providing the completeness and soundness of the PCP verifier constructed in Section 4.

Arithmetizing the acceptance predicate of $\mathcal{V}$, we get that the probability that $\mathcal{V}$ accepts a proof is

$$\mathop{\mathbb{E}}_{v,e_1,e_2,x_1,x_2}\left[\frac{3-\Delta(b_1+b_2)-b_1b_2}{4}\right], \tag{29}$$

where $b_i = f_{w_i}(\sigma^v_{e_i} x_i)$ and v, e_1, e_2, x_1, x_2 are picked with the same distribution as by the verifier.

PROOF OF LEMMA 4.1 (COMPLETENESS). Suppose there is an assignment of labels to the vertices of X such that the fraction of satisfied edges is at least $1-\eta$. Fix such a labelling, and let $f_v : \{-1,1\}^L \to \{-1,1\}$ be the long code of the label of v. Note that for a satisfied edge $e = \{v, w\}$, $f_w(\sigma^v_e x_i)$ equals the value of the l_v:th bit of x_i (where l_v is the label of vertex v)

By the union bound, the probability that any of the two edges e_1 and e_2 are not satisfied is at most 2η. For a choice of edges e_1, e_2 that *are* satisfied, the expected value of $f_{w_i}(\sigma^v_{e_i} x_i)$ is simply the expected value of the l_v:th bit in x_i, i.e. ξ, and the expected value of $f_{w_1}(\sigma^v_{e_1} x_1) f_{w_2}(\sigma^v_{e_2} x_2)$ is the expected value of the l_v:th bit of $x_1 x_2$, i.e. $-1 + 2|\xi|$. Thus, for such a choice of edges, the acceptance probability becomes

$$\frac{3 - 2\Delta\xi - (-1 + 2|\xi|)}{4} = \frac{2 - \Delta\xi - |\xi|}{2}, \qquad (30)$$

and we are done. □

PROOF OF LEMMA 4.2 (SOUNDNESS). As is common, the proof is by contradiction. Assume that the value of X is at most $\mathrm{Val}(X) \le \gamma$. Take any proof $\Sigma = \{f_v\}_{v \in V}$. Define

$$g_v(x) := \mathop{\mathbb{E}}_{e=\{v,w\} \in E(v)} [f_w(\sigma^v_e x)], \qquad (31)$$

and $\mu_v := \mathbb{E}_x[g_v(x)]$. Assume that the probability that the verifier accepts this proof is at least

$$\Pr[\mathcal{V} \text{ accepts } \Sigma] \ge \mathop{\mathbb{E}}_v \left[\frac{2 - (1+\Delta)\mu_v - 2\Gamma_{\tilde\rho}(\mu_v)}{2} + \epsilon \right]. \qquad (32)$$

We will show that in that case, it is possible to satisfy a constant (that depends only on ξ and ϵ) fraction of the edges of X. Setting γ smaller than this constant will yield the desired result.

Note that the probability distribution of x_1, x_2 is the same as that induced by first picking x_1 at random in B^n_q and then constructing x_2 from x_1 in the same way y is constructed from x in the Beckner operator $T_{\tilde\rho}$, for $q = \frac{1-\xi}{2}$ and $\tilde\rho = -\frac{1-q}{q} = \frac{|\xi|-1}{|\xi|+1}$. Thus, the expected value of $g_v(x_1) g_v(x_2)$ equals $\mathbb{S}_{\tilde\rho}(g_v)$. So by the definition of g_v and μ_v, we can rewrite the probability that the verifier accepts as

$$\begin{aligned}
&\Pr[\mathcal{V} \text{ accepts } \Sigma] \\
&= \mathop{\mathbb{E}}_{v,x_1,x_2} \left[\frac{3 - \Delta(g_v(x_1) + g_v(x_2)) - g_v(x_1) g_v(x_2)}{4} \right] \\
&= \mathop{\mathbb{E}}_v \left[\frac{3 - 2\Delta\mu_v - \mathbb{S}_{\tilde\rho}(g_v)}{4} \right]
\end{aligned}$$

Plugging in Equation (32), this gives

$$\begin{aligned}
&\mathop{\mathbb{E}}_v \left[\frac{3 - 2\Delta\mu_v - \mathbb{S}_{\tilde\rho}(g_v)}{4} \right] \\
&\qquad \ge \mathop{\mathbb{E}}_v \left[\frac{2 - (1+\Delta)\mu_v - 2\Gamma_{\tilde\rho}(\mu_v)}{2} + \epsilon \right],
\end{aligned}$$

which simplifies to

$$\mathop{\mathbb{E}}_v [4\Gamma_{\tilde\rho}(\mu_v) + 2\mu_v - 1 - \mathbb{S}_{\tilde\rho}(g_v)] \quad \ge \quad 4\epsilon.$$

Note that $4\Gamma_{\tilde\rho}(\mu_v) + 2\mu_v - 1 - \mathbb{S}_{\tilde\rho}(g_v) = 2(\Gamma_{\tilde\rho}(\mu_v) + \Gamma_{\tilde\rho}(-\mu_v)) - 1 - \mathbb{S}_{\tilde\rho}(g_v) \le 2 - 1 - (-1) = 2$, so it must be the case that for a fraction of at least $\frac{3\epsilon}{2-\epsilon} \ge \epsilon$ of the vertices $v \in V$, we have

$$\mathbb{S}_{\tilde\rho}(g_v) \le 4\Gamma_{\tilde\rho}(\mu_v) + 2\mu_v - 1 - \epsilon. \qquad (33)$$

Let V_{good} be the set of all such v. Since $\tilde\rho < 0$ we have by (extended) Majority Is Stablest (Theorem A.4) that for all $v \in V_{\text{good}}$ there must be some $i \in [L]$ such that $\mathrm{Inf}^{\le k}_i(g_v) \ge \tau$, where τ and k are constants depending only on ϵ and ξ.[4] Thus, for any $v \in V_{\text{good}}$, we have

$$\begin{aligned}
\tau &\le \sum_{\substack{i \in S \\ |S| \le k}} \widehat{(g_v)}^2_S = \sum_{\substack{i \in S \\ |S| \le k}} \mathop{\mathbb{E}}_{e=\{v,w\}} \left[\widehat{(f_w)}_{\sigma^v_e S} \right]^2 \\
&\le \sum_{\substack{i \in S \\ |S| \le k}} \mathop{\mathbb{E}}_{e=\{v,w\}} \left[\widehat{(f_w)}^2_{\sigma^v_e S} \right] = \mathop{\mathbb{E}}_{e=\{v,w\}} \left[\mathrm{Inf}^{\le k}_{\sigma^v_e(i)}(f_w) \right].
\end{aligned}$$

This, and the fact that $\mathrm{Inf}^{\le k}_{\sigma^v_e(i)}(f_w) \le 1$ for all i, implies that for a fraction of at least $\frac{\tau - \tau/2}{1 - \tau/2} \ge \frac{\tau}{2}$ of the edges $e = \{v, w\} \in E(v)$, we have $\mathrm{Inf}^{\le k}_{\sigma^v_e(i)}(f_w) \ge \tau/2$.

For $v \in V$, let

$$C(v) = \{\, i \in L \mid \mathrm{Inf}^{\le k}_i(f_v) \ge \tau/2 \vee \mathrm{Inf}^{\le k}_i(g_v) \ge \tau \,\}. \qquad (34)$$

Intuitively, the criterion $\mathrm{Inf}^{\le k}_i(f_v) \ge \tau/2$ means that the purported Long Codes of the label of v suggests i as a suitable label for v, and the criterion $\mathrm{Inf}^{\le k}_i(g_v) \ge \tau$ means that many of the purported Long Codes for the neighbours of v suggests that v should have the label i. By the fact that $\sum_i \mathrm{Inf}^{\le k}_i(f_w) \le k$, we must have $|C(v)| \le 2k/\tau + k/\tau = 3k/\tau$.

We now define a labelling by picking independently for each $v \in V$ a (uniformly) random label $i \in C(v)$ (or an arbitrary label in case $C(v)$ is empty). For a vertex $v \in V_{\text{good}}$ with $\mathrm{Inf}^{\le k}_i(g_v) \ge \tau$, the probability that v is assigned label i is $1/|C(v)| \ge \tau/3k$. Furthermore, by the above reasoning and the definition of C, at least a fraction $\tau/2$ of the edges $e = \{v, w\}$ from v will satisfy $\sigma^v_e(i) \in C(w)$. For such an edge, the probability that w is assigned the label $\sigma^v_e(i)$ is $1/|C(w)| \ge \tau/3k$. Thus, the expected fraction of satisfied edges adjacent to any $v \in V_{\text{good}}$ is at least $\tau/2 \cdot (\tau/3k)^2$, and so the expected fraction of satisfied edges in total[5] is at least $\epsilon \cdot \frac{\tau^3}{18k^2}$ (note that this is a positive constant that depends only on ϵ and ξ) and thus there is an assignment satisfying at least this total weight of edges. Making sure that $\gamma < \frac{\epsilon\tau^3}{18k^2}$, we get a contradiction on the assumption of the acceptance probability (Equation (32)), implying that the soundness is at most

$$\begin{aligned}
\Pr[\mathcal{V} \text{ accepts } \Sigma] &\le \mathop{\mathbb{E}}_v \left[\frac{2 - (1+\Delta)\mu_v - 2\Gamma_{\tilde\rho}(\mu_v)}{2} + \epsilon \right] \\
&\le \max_{\mu \in [-1,1]} \frac{2 - (1+\Delta)\mu - 2\Gamma_{\tilde\rho}(\mu)}{2} + \epsilon,
\end{aligned}$$

as desired. □

[4] The dependency on ξ stems from the fact that g_v is a function from B^n_q to $\mathbb{R}$, where $q = \frac{1-\xi}{2}$.

[5] We remind the reader of the convention of Section 2.3 that the choices of random vertices and edges are according to the probability distributions induced by the weights of the edges, and so choosing a random $v \in V$ and then a random $e \in E(v)$ is equivalent to just choosing a random $e \in E$.