Towards Sharp Inapproximability For Any 2-CSP

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

Abstract

We continue the recent line of work on the connection between semidefinite programming-based approximation algorithms and the Unique Games Conjecture. Given any boolean 2-CSP (or more generally, any nonnegative objective function on two boolean variables), we show how to reduce the search for a good inapproximability result to a certain numeric minimization problem. The key objects in our analysis are the vector triples arising when doing clause-by-clause analysis of algorithms based on semidefinite programming. Given a weighted set of such triples of a certain restricted type, which are "hard" to round in a certain sense, we obtain a Unique Games-based inapproximability matching this "hardness" of rounding the set of vector triples. Conversely, any instance together with an SDP solution can be viewed as a set of vector triples, and we show that we can always find an assignment to the instance which is at least as good as the "hardness" of rounding the corresponding set of vector triples. We conjecture that the restricted type required for the hardness result is in fact no restriction, which would imply that these upper and lower bounds match exactly. This conjecture is supported by all existing results for specific 2-CSPs.

As an application, we show that MAX 2-AND *is hard to approximate within* 0.87435. *This improves upon the best previous hardness of* $\alpha_{GW} + \epsilon \approx 0.87856$, *and comes very close to matching the approximation ratio of the best algorithm known,* 0.87401. *It also establishes that balanced instances of* MAX 2-AND, *i.e., instances in which each variable occurs positively and negatively equally often, are not the hardest to approximate, as these can be approximated within a factor* α_{GW}.

1 Introduction

Predicates on two boolean variables are fundamental in the study of constraint satisfaction problems. Given a set of constraints, each being a formula on two boolean variables, it is an easy task to find an assignment satisfying all constraints, if such an assignment exists. However, determining the maximum possible number of simultaneously satisfied constraints is well-known to be NP-hard. This problem is known as the MAX 2-CSP problem. It also has some very interesting special cases, the two most well-studied of which are the MAX CUT problem and the MAX 2-SAT problem. In the MAX CUT problem, each constraint is of the form $x_i \oplus x_j$, i.e., it is true if exactly one of the inputs are true. In the MAX 2-SAT problem, each constraint is of the form $l_i \vee l_j$, i.e., a disjunction on two literals, each literal being either a variable or a negated variable.

Given that the problem is NP-hard, much research has been focused on approximating the maximum number of satisfied constraints to within some factor α. An algorithm achieves approximation ratio α if the solution found by the algorithm is guaranteed to have value at least α times the optimum. We also allow for randomized algorithms, in which we require that the expected value (over the randomness of the algorithm) of the solution found by the algorithm is α times the optimum. The arguably most trivial approximation algorithm is to simply pick a random assignment to the variables. For the general MAX 2-CSP problem, this algorithm achieves an approximation ratio of $1/4$. For the special cases of MAX CUT and MAX 2-SAT, it achieves ratios of $1/2$ and $3/4$, respectively. For several decades, no substantial improvements were made over this result, until a seminal paper by Goemans and Williamson [15], where they constructed a 0.7960-approximation algorithm for MAX 2-CSP, and 0.87856-approximation algorithms for MAX CUT and MAX 2-SAT. To do so, they relaxed the combinatorial problem at hand to a semidefinite programming problem, to which an optimal solution can be found with high precision, and then used a very clever technique to "round" the solution of the semidefinite programming back to a discrete solution for the original problem. This approach has since been succesfully applied to several other hard combinatorial optimization problems, yielding significant improvements over existing approximation algorithms. Examples include coloring graphs using as few

*Research funded by Swedish Research Council Project Number 50394001.

colors as possible [19, 6, 16, 2], MAX BISECTION [14] and quadratic programming over the boolean hypercube [8].

Some of the results by Goemans and Williamson were subsequently improved by Feige and Goemans [12], who strengthened the semidefinite relaxation using certain triangle inequalities, and considered more complicated rounding methods than [15]. They obtained 0.931-approximation for MAX 2-SAT, and 0.859-approximation for MAX 2-CSP. These results were further improved by Matuura and Matsui [26, 27], who obtained 0.935-approximation for MAX 2-SAT and 0.863-approximation for MAX 2-CSP. Shortly thereafter, Lewin et al. [25] obtained further improvements, getting a 0.94016-approximation algorithm for MAX 2-SAT and a 0.87401-approximation algorithm for MAX 2-CSP, and these stand as the current best algorithms. It should be pointed out that these last two ratios arise as the minima of two complex numeric optimization problems, and, as far as we are aware, it has not yet been proved formally that these are the actual ratios, though there seems to be very little doubt that this is indeed the case.

Meanwhile, the study of *inapproximability* has seen a lot of progress, emanating from the discovery of the celebrated PCP theorem [4, 3]. In particular, Håstad [17] showed that the generalizations of MAX 2-SAT and MAX CUT from 2 to 3 variables, MAX 3-SAT and MAX 3-LIN-MOD2,[1] are NP-hard to approximate within factors $7/8+\epsilon$ and $1/2+\epsilon$, respectively. This surprisingly demonstrates that the random assignment algorithm is the best possible for these problems, assuming $P \neq NP$. On the other hand, MAX 3-CSP can be approximated to within a factor $1/2$ [34] which is tight by the result for MAX 3-LIN-MOD2.

For optimization problems with constraints acting on two variables, however, strong inapproximability results have been more elusive. The best NP-hardness results for MAX 2-CSP, MAX 2-SAT, and MAX CUT are $9/10+\epsilon \approx 0.900$, $21/22+\epsilon \approx 0.955$, and $16/17+\epsilon \approx 0.941$, respectively [33, 17]. The most promising approach to obtaining strong results for these problems is the so-called Unique Games Conjecture (UGC), introduced by Khot [20]. The UGC has established itself as one of the most important open problems in theoretical computer science, because of the many strong inapproximability results that follow from it. Examples of such results include $2-\epsilon$ hardness for VERTEX COVER [23], superconstant hardness for SPARSEST CUT [9, 24] and MULTICUT [9], hardness of approximating MAX INDEPENDENT SET within $d/\mathrm{poly}(\log d)$ in degree-d graphs [31], and approximation resistance[2] for random predicates [18].

For MAX 2-CSP problems, Khot et al. [21] showed that the UGC implies $\alpha_{GW}+\epsilon$ hardness for MAX CUT, where $\alpha_{GW} \approx 0.87856$ is the performance ratio of the original Goemans-Williamson algorithm, and in [5], we showed that the UGC implies $\alpha_{LLZ}+\epsilon$ hardness for MAX 2-SAT, where $\alpha_{LLZ} \approx 0.94016$ is the performance ratio of the algorithm of Lewin et al. (modulo the slight possibility that the performance ratio of their algorithm is smaller than indicated by existing analyses). It is interesting that the hardness ratios yielded by the Unique Games Conjecture exactly match these somewhat "odd" constants obtained from the complex numeric optimization problems arising from the SDP-based algorithms.

There are several other cases where the best inapproximability result, based on the UGC, matches the best approximation algorithm, based on a semidefinite programming approach. Examples include the MAX k-CSP problem [7, 31] and MAX CUT-GAIN [8, 22] (which is essentially a version of the MAX CUT problem where unsatisfied constraints give negative contribution rather than zero). This line of results is not a coincidence: in most cases, the choice of optimal parameters for the so called long code test (which is at the heart of the hardness result) are derived by analyzing worst-case scenarios for the semidefinite relaxation of the problem.

1.1 Our Contribution

In this paper, we continue to explore this tight connection between semidefinite programming relaxations and the UGC. We consider a generalization of predicates on two variables to what we call *fuzzy predicates*. A fuzzy predicate P on two variables is a function $P : \{\text{true}, \text{false}\}^2 \to [0,1]$, rather than to $\{0,1\}$ as would be the case with a regular predicate. We investigate the approximability of the MAX CSP(P) problem. Following the paradigm introduced by Goemans and Williamson, we relax this problem to a semidefinite programming problem. We then consider the following approach for rounding the relaxed solution to a boolean solution: given the SDP solution, we pick the "best" rounding from a certain class of randomized rounding methods (based on skewed random hyperplanes), where "best" is in the sense of giving a boolean assignment with maximum possible expected value. Informally, let $\alpha(P)$ denote the approximation ratio yielded by such an approach. We then have the following theorem.

Theorem 1.1. *For any fuzzy predicate P and $\epsilon > 0$, the* MAX CSP(P) *problem can be approximated within $\alpha(P)-\epsilon$ in polynomial time.*

The reason that we lose an additive ϵ is that we are not, in general, able to find the *best* rounding function, but we can come arbitarily close.

Then, we turn our attention to hardness of approximation. Here, we are able to take instances which are hard

[1] Linear equations mod 2, where every equation has 3 variables.

[2] A predicate is approximation resistant if it is hard to do approximate the corresponding MAX CSP problem better than a random assignment.

to round, in the sense that the best rounding (as described above) is not very good, and translate them into a Unique Games-based hardness result. There is, however, a caveat: in order for the analysis to work, the instance needs to satisfy a certain "positivity" condition. Again, informally, let $\beta(P)$ denote the approximation ratio when restricted to instances satisfying this condition. We then have

Theorem 1.2. *If the Unique Games Conjecture is true, then for any fuzzy predicate P and $\epsilon > 0$, the* MAX CSP(P) *problem is NP-hard to approximate within $\beta(P) + \epsilon$.*

Both $\alpha(P)$ and $\beta(P)$ are the solutions to a certain numeric minimization problem. The function being minimized is the same function in both cases, the only difference is that in $\alpha(P)$, the minimization is over a larger domain, and thus, we could potentially have $\alpha(P) < \beta(P)$. However, there are strong indications that the minimum for $\alpha(P)$ is in fact obtained within the domain of $\beta(P)$, in which case they would be equal and Theorems 1.1 and 1.2 would be tight.

Conjecture 1.3. *For any fuzzy predicate P, we have $\alpha(P) = \beta(P)$.*

Because of the difficulty of actually computing the approximation ratios $\alpha(P)$ and $\beta(P)$, it may seem to be somewhat difficult to compare these results to previous results. However, previous algorithms and hardness results for MAX CUT, MAX 2-SAT, and MAX 2-CSP can all be obtained as special cases of Theorems 1.1 and 1.2. In particular, for $P(x_1, x_2) = x_1 \oplus x_2$, the XOR predicate, it can be shown that $\alpha(P) = \beta(P) = \alpha_{GW}$.

We are also able to use Theorem 1.2 to obtain new results, in the form of an improved hardness of approximation for the MAX 2-AND problem, in which every constraint is an AND of two literals. This also implies improved hardness for the MAX 2-CSP problem – as is well-known, the MAX k-CSP problem and the MAX k-AND problem are equally hard to approximate for every k (folklore, or see e.g. [32]).

Theorem 1.4. *For the predicate $P(x_1, x_2) = x_1 \wedge x_2$, we have $\beta(P) \leq 0.87435$.*

This comes very close to matching the 0.87401-approximation algorithm of Lewin et al. It also demonstrates that balanced instances, i.e., instances in which each variable occurs positively and negatively equally often, are not the hardest to approximate, as these can be approximated within $\alpha_{GW} \approx 0.87856$ [21].

Finally, as a by-product of our results, we obtain some insight regarding the possibilites of obtaining improved results by strengthening the semidefinite program with more constraints. Traditionally, the only constraints which have been useful in the design of MAX 2-CSP algorithms are triangle inequalities of a certain form (namely, those involving the vector v_0, coding the value false). It turns out that, for very natural reasons, these are exactly the inequalities that need to be satisfied in order for the hardness result to carry through. In other words, assuming Conjecture 1.3 is true, it is UG-hard to do better than what can be achieved by adding only these triangle inequalities, and thus, it is unlikely that improvements can be made by adding additional inequalities (while still using polynomial time).

1.2 Techniques and Related Work

The main new ingredients of this paper are the generalizations of the various quantities used in previous results. In e.g. the case of MAX 2-SAT [5], one only had to consider one single angle, giving rise to two configurations of a very special form, something which made the calculations a lot easier. In this paper, on the other hand, we can have an arbitrary number of angles (and this is of course the reason why it is very difficult to actually compute the approximation ratios obtained), and the "positivity" condition needed here is significantly less restrictive than the special form used for MAX 2-SAT.

The proof of Theorem 1.2 follows the same path as previous proofs for specific predicates [21, 5], using the Majority Is Stablest theorem [29]. The main difference here is that we need a generalization of the "correlation under noise" quantities involved, to functions on different probability distributions. The proof of Theorem 1.1 primarily builds upon the work of [25] for MAX 2-SAT and MAX DI-CUT, the main difference being that a rounding function is chosen based on the semidefinite solution rather than beforehand, using a discretization technique to make the search a good rounding function feasible.

2 Preliminaries

We associate the boolean values true and false with -1 and 1, respectively. Thus, a disjunction $x \vee y$ is true if $x = -1$ or $y = -1$, and a conjunction $x \wedge y$ is true if $x = y = -1$. We denote by $S^n = \{ v \in \mathbb{R}^{n+1} : ||v|| = 1 \}$ the n-dimensional unit sphere.

We denote by $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} \mathrm{d}t$ the standard normal distribution function, and by Φ^{-1} the inverse of Φ. For $\rho, \mu_1, \mu_2 \in [-1, 1]$, we define

$$\Gamma_\rho(\mu_1, \mu_2) = \Pr[X_1 \leq t_1 \wedge X_2 \leq t_2], \qquad (1)$$

where $t_i = \Phi^{-1}\left(\frac{1-\mu_i}{2}\right)$ and where $X_1, X_2 \in N(0, 1)$ with covariance ρ. In other words, Γ_ρ is just the bivariate normal distribution function with a transformation on the input.

2.1 Constraint Satisfaction Problems

A *predicate* P on two boolean variables is a function $P : \{-1,1\}^2 \to \{0,1\}$. We generalize this to the notion of *fuzzy predicates*.

Definition 2.1. A *fuzzy predicate* P on two boolean variables is a function $P : \{-1,1\}^2 \to [0,1]$.

Note that, with general objective functions from $\{-1,1\}^2$ to $\mathbb{R}$ in mind, the upper bound is without loss of generality, since we can always scale down any nonnegative objective function so that it takes values in $[0,1]$ and thus becomes a fuzzy predicate.

Definition 2.2. An instance Ψ of the MAX CSP(P) problem, for a fuzzy predicate P, consists of a set of clauses and a weight function wt. Each clause ψ is a pair of literals (l_1, l_2) (a literal is either a variable or a negation of a variable), and the weight function associates with each clause ψ a nonnegative weight $\mathrm{wt}(\psi)$. We abuse notation slightly by identifying Ψ with both the instance and the set of clauses. Given an assignment $x = (x_1, \ldots, x_n)$ to the variables occurring in Ψ, and a clause $\psi = (s_1 x_i, s_2 x_j)$ (where $s_1, s_2 \in \{-1,1\}$), we denote the restriction of x to ψ by $x|_\psi = (s_1 x_i, s_2 x_j)$. The value of an assignment x to the variables occuring in Ψ is then given by

$$\mathrm{Val}_\Psi(x) = \sum_{\psi \in \Psi} \mathrm{wt}(\psi) P(x|_\psi), \tag{2}$$

and the value of Ψ is the maximum possible value of an assignment

$$\mathrm{Val}(\Psi) = \max_x \mathrm{Val}_\Psi(x). \tag{3}$$

For convenience, we will assume (without loss of generality) that the weights are normalized so that $\mathrm{wt}(\cdot)$ is just a probability distribution on the clauses, i.e., that $\sum_{\psi \in \Psi} \mathrm{wt}(\psi) = 1$ (so $0 \le \mathrm{Val}(\Psi) \le 1$).

Definition 2.3. The MAX CSP$^+(P)$ problem is the special case of MAX CSP(P) where there are no negated literals (i.e. each clause is a pair of variables).

An example of the MAX CSP(P) problem which is of special interest for us is the MAX 2-AND problem, which is obtained by letting P be the predicate which is 1 if both of the inputs are true, and 0 otherwise. A well-known example of the MAX CSP$^+(P)$ problem is the MAX CUT problem, which is obtained by letting P be the predicate which is 1 if the inputs are different, and 0 if they are equal.

Any fuzzy predicate P can be arithmetized as $P(x_1, x_2) = \hat{P}_0 + \hat{P}_1 x_1 + \hat{P}_2 x_2 + \hat{P}_3 x_1 x_2$, for some constants $\hat{P}_0$, $\hat{P}_1$, $\hat{P}_2$ and $\hat{P}_3$. Thus, the MAX CSP(P) problem can be viewed as a certain special case of the integer quadratic programming problem. Throughout the remainder of this paper, we fix some arbitrary fuzzy predicate P and its corresponding coefficients $\hat{P}_0 \ldots \hat{P}_3$.

2.2 The Unique Games Conjecture

Definition 2.4. 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.,

$$\mathrm{Val}_X(\ell) = \sum_{\substack{e \\ \ell \text{ satisfies } e}} \mathrm{wt}(e) \tag{4}$$

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

$$\mathrm{Val}(X) = \max_\ell \mathrm{Val}_X(\ell). \tag{5}$$

Khot's Unique Games Conjecture (UGC) asserts that it is NP-hard to distinguish between UNIQUE LABEL COVER instances which are almost completely satisfiable, from those where we cannot satisfy more than a small fraction of the constraints.

Conjecture 2.5 (Unique Games Conjecture [20])**.** *For every $\eta > 0$, $\gamma > 0$, there is a constant $L > 0$ such that, given a* UNIQUE LABEL COVER *instance X with label set $[L]$, it is NP-hard to distinguish between $\mathrm{Val}(X) \le \gamma$ and $\mathrm{Val}(X) \ge 1 - \eta$.*

2.3 Influence and Correlation Under Noise

As in previous results [21, 5], the key ingredient in the proof of our hardness result is (a generalization of) the so-called Majority Is Stablest Theorem [29]. In this section, we describe this result and the exact formulation we use.

For $q \in (0,1)$, 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 write a function $f : B_q^n \to \mathbb{R}$ in terms of its Fourier coefficients, viz

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

where the coefficients $\hat{f}_S = \mathbb{E}_{x \in B_q^n}[f(x) U_q^S(x)]$ are the Fourier coefficients of the function f. Here, $U_q^S : B_q^n \to \mathbb{R}$

is defined by $U_q^S(x) = \prod_{i \in S} U_q(x_i)$ where

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

Definition 2.6. The *long code* of an integer $i \in [n]$ is the function $f : \{-1,1\}^n \to \{-1,1\}$ defined by $f(x) = x_i$.

Definition 2.7. The *influence* of the variable i on the function $f : B_q^n \to \mathbb{R}$ is

$$\mathrm{Inf}_i(f) = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathrm{Var}}_{x_i}[f(x) \,|\, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]\right] \tag{6}$$

The influence of the variable i is a measure of how much the variable i is able to change the value of f once we have fixed the other $n-1$ variables randomly (according to the distribution μ_q^{n-1}). This quantity has a particularly nice formulation in terms of the Fourier coefficients, where we have the following well-known fact

$$\mathrm{Inf}_i(f) = \sum_{\substack{S \subseteq [n] \\ i \in S}} \hat{f}_S^2. \tag{7}$$

Motivated by this formulation, we define the slightly stronger concept of low-degree influence, crucial to our application.

Definition 2.8. For $k \in \mathbb{N}$, the *low-degree influence* of the variable i on the function $f : B_q^n \to \mathbb{R}$ is

$$\mathrm{Inf}_i^{\leq k}(f) = \sum_{\substack{S \subseteq [n] \\ i \in S \\ |S| \leq k}} \hat{f}_S^2. \tag{8}$$

A nice property of the low-degree influence is the fact that for functions into $[-1,1]$, $\sum_i \mathrm{Inf}_i^{\leq k}(f) \leq k$, implying that the number of variables having low-degree influence more than, say, τ, must be small (think of k and τ as constants not depending on the number of variables n). Very informally, one can think of the low-degree influence as a measure of how close the function f is to depending on only a few variables, i.e., for the case of boolean-valued functions, how close f is to being the long code of i (or its negation). Note that a long code is the extreme case of a function with large low-degree influence, in the sense that it has one variable with $\mathrm{Inf}_i^{\leq 1}(f) = 1$, and all other variables having influence 0.

Next, we introduce the *correlation under $\tilde{\rho}$-noise* between two functions $f : B_{q_1}^n \to \mathbb{R}$ and $g : B_{q_2}^n \to \mathbb{R}$. For functions into $\{-1,1\}$, the correlation under noise measures how likely f and g are to take the same value on two random inputs with a certain correlation. For $f = g$, this is simply the well-studied *noise stability* of f.

x_i	y_i	Probability
1	1	$\frac{1+\xi_1+\xi_2+\rho}{4}$
1	-1	$\frac{1+\xi_1-\xi_2-\rho}{4}$
-1	1	$\frac{1-\xi_1+\xi_2-\rho}{4}$
-1	-1	$\frac{1-\xi_1-\xi_2+\rho}{4}$

Table 1. Distribution of x and y

Definition 2.9. The *correlation under $\tilde{\rho}$-noise* between $f : B_{q_1}^n \to \mathbb{R}$ and $g : B_{q_2}^n \to \mathbb{R}$ is given by

$$\mathbb{S}_{\tilde{\rho}}(f,g) = \mathop{\mathbb{E}}_{x,y}[f(x)g(y)], \tag{9}$$

where the i:th bits of x and y are according to the distribution in Table 1 (independently of the other bits). Here $\xi_j := 1 - 2q_j$ are the expected values of the bits of x and y, and ρ is such that $\tilde{\rho} = \frac{\rho - \xi_1\xi_2}{\sqrt{1-\xi_1^2}\sqrt{1-\xi_2^2}}$. In other words, each bit of x is picked independently with expected value ξ_1, each bit of y is picked independently with expected value ξ_2, and the i:th bits x_i and y_i have expected value $\mathbb{E}[x_i y_i] = \rho$.

In terms of the Fourier representation, it can be shown (see full version of this paper) that

$$\mathbb{S}_{\tilde{\rho}}(f,g) = \sum_{S \subseteq [n]} \tilde{\rho}^{|S|} \hat{f}_S \hat{g}_S. \tag{10}$$

For proving hardness of MAX CUT, Khot et al. [21] 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 large low-degree influence (and thus in a vague sense be similar to a Long Code). Majority Is Stablest was subsequently proved by Mossel et al. [29], using a very powerful invariance principle which, essentially, allows for considering the corresponding problem over Gaussian space instead. Subsequently, this type of invariance, and therefore also this way of extracting influential variables by studying correlation under noise, has been shown to apply in very general settings [11, 28].

For our result, we use the following formulation.

Theorem 2.10. *Let $\epsilon > 0$, $q_1, q_2 \in (0,1)$ and $\rho \in (-1,1)$. Then there are $\tau > 0$, $k \in \mathbb{N}$ such that for all functions $f : B_{q_1}^n \to [-1,1]$, $g : B_{q_2}^n \to [-1,1]$ satisfying $\mathbb{E}[f] = \mu_f$, $\mathbb{E}[g] = \mu_g$, and $\min(\mathrm{Inf}_i^{\leq k}(f), \mathrm{Inf}_i^{\leq k}(g)) \leq \tau$ for all i, we have*

$$\begin{aligned} \mathbb{S}_\rho(f,g) &\geq 4\Gamma_{-|\rho|}(\mu_f,\mu_g) + \mu_f + \mu_g - 1 - \epsilon \\ \mathbb{S}_\rho(f,g) &\leq 4\Gamma_{|\rho|}(\mu_f,\mu_g) + \mu_f + \mu_g - 1 - \epsilon. \end{aligned}$$

In the terminology of [11], the setting of Theorem 2.10 corresponds to the case of a reversible noise operator, rather than a symmetric one as was studied there. It is known that the results also hold in the reversible case [10] (and in fact even in the non-reversible case [28]), but for completeness, a proof of Theorem 2.10 (following the same lines as the proof of [11]) can be found in the full version of this paper.

3 Semidefinite Relaxation

For solving integer quadratic programming over the hypercube, where each variable is restricted to ± 1, the standard approach is to first homogenize the program by introducing a variable x_0 which is supposed to represent the value false and then replace each term x_i by $x_0 x_i$. We then relax each variable $x_i \in \{-1, 1\} = S^0$ with a vector $v_i \in S^n$ (i.e. a unit vector in $\mathbb{R}^{n+1}$), so that each term $x_i x_j$ becomes the scalar product $v_i \cdot v_j$.

In addition, we add the following inequality constraints to the program for all triples of vectors v_i, v_j, v_k.

$$\begin{array}{rl} v_i \cdot v_j + v_j \cdot v_k + v_i \cdot v_k & \geq -1 \\ -v_i \cdot v_j + v_j \cdot v_k - v_i \cdot v_k & \geq -1 \\ v_i \cdot v_j - v_j \cdot v_k - v_i \cdot v_k & \geq -1 \\ -v_i \cdot v_j - v_j \cdot v_k + v_i \cdot v_k & \geq -1 \end{array} \tag{11}$$

These are equivalent to triangle inequalities of the form $||v_i - v_j||^2 + ||v_j - v_k||^2 \geq ||v_i - v_k||^2$, which clearly hold for the case that all vectors lie in a zero-dimensional subspace of S^n (so this is still a relaxation of the original integer program), but is not necessarily true otherwise. There are of course many other valid inequalities which could also be added, considering k-tuples of variables rather than just triples. In particular, adding *all* valid constraints makes the optimum for the semidefinite program equal the discrete optimum [13] (but there are an exponential number of constraints to consider). Such higher-order constraints have not received much attention, and from what is known today, it seems that the only ones which actually help are the triangle inequalities. In particular, the only inequalities which have been used when analyzing the performance of approximation algorithms, are those of the triangle inequalities which involve the vector v_0. The results of this paper shed some light on why this is the case – these are exactly the inequalities we need in order for the hardness of approximation to work out. Thus, assuming Conjecture 1.3 and the Unique Games Conjecture, it is unlikely that adding other valid inequalities (while still being able to solve the SDP in polynomial time) will help achieve a better approximation ratio, as that would imply $P = NP$.

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

Given a vector solution $\{v_i\}_{i=0}^n$, the relaxed value of a clause $\psi \in \Psi$ depends only on the three (possibly negated) scalar products $v_0 \cdot v_i$, $v_0 \cdot v_j$, and $v_i \cdot v_j$, where x_i and x_j are the two variables occuring in ψ. Most of the time, we do not care about the actual vectors, but only be interested in these triples of scalar products.

Definition 3.1. A *scalar product configuration* θ, or just a *configuration* for short, is a triple of real numbers (ξ_1, ξ_2, ρ) satisfying

$$\begin{array}{llll} \xi_1 + \xi_2 + \rho & \geq -1 & -\xi_1 + \xi_2 - \rho & \geq -1 \\ \xi_1 - \xi_2 - \rho & \geq -1 & -\xi_1 - \xi_2 + \rho & \geq -1 \end{array} \tag{12}$$

A *family of configurations* Θ is a finite set $X = \{\theta_1, \ldots, \theta_k\}$ of configurations, endowed with a probability distribution P. We routinely abuse notation by identifying Θ both with the set X and the probability space (X, P).

A configuration can be viewed as representing three vectors v_0, v_1, v_2, where $v_0 \cdot v_i = \xi_i$, and $v_1 \cdot v_2 = \rho$. Note that the inequalities in Equation (12) then correspond exactly to those of the triangle inequalities (11) which involve v_0. It can also be shown that these inequalities ensures the existence of vectors v_0, v_1, v_2 with the corresponding scalar products.

Definition 3.2. The relaxed value of a configuration $\theta = (\xi_1, \xi_2, \rho)$ is given by

$$P_{\mathrm{relax}}(\theta) = P_{\mathrm{relax}}(\xi_1, \xi_2, \rho) = \hat{P}_0 + \hat{P}_1 \xi_1 + \hat{P}_2 \xi_2 + \hat{P}_3 \rho$$

Analogously to the notation $x|_\psi$ for discrete solutions, we denote by $v|_\psi = (s_1 v_0 \cdot v_i, s_2 v_0 \cdot v_j, s_1 s_2 v_i \cdot v_j)$ the configuration arising from the clause $\psi = (s_1 x_i, s_2 x_j)$ for the vector solution $v = \{v_i\}_{i=0}^n$. The relaxed value of the clause ψ is then simply given by $P_{\mathrm{relax}}(v|_\psi)$.

Often we view the solution to the SDP as just the family of configurations $\Theta = \{\, v|_\psi \mid \psi \in \Psi \,\}$ with the probability distribution where $\Pr_{\theta \in \Theta}[\theta = v|_\psi] = \mathrm{wt}(\psi)$. The relaxed value of an assignment of vectors $\{v_i\}_{i=0}^n$ is then given by

$$\begin{aligned} \text{SDP-Val}_\Psi(\{v_i\}) &= \sum_{\psi \in \Psi} \mathrm{wt}(\psi) P_{\mathrm{relax}}(v|_\psi) \\ &= \mathop{\mathbb{E}}_{\theta \in \Theta}[P_{\mathrm{relax}}(\theta)]. \end{aligned} \tag{13}$$

Given a vector solution $\{v_i\}$, one natural attempt at an approximation algorithm is to set x_i true with probability $\frac{1-\xi_i}{2}$ (where $\xi_i = v_i \cdot v_0$), independently—the intuition being that the linear term ξ_i gives an indication of "how true" x_i should be. This assignment has the same expected value on the linear terms as the vector solution, and the expected value of a quadratic term $x_i x_j$ is $\xi_i \xi_j$. However, typically

there is some correlation between the vectors v_i and v_j, so that the scalar product $v_i \cdot v_j$ contributes more than $\xi_i \xi_j$ to the objective function. To quantify this, write the vector v_i as

$$v_i = \xi_i v_0 + \sqrt{1 - \xi_i^2}\tilde{v}_i, \tag{14}$$

where $\xi_i = v_i \cdot v_0$, and $\tilde{v}_i$ is the part of v_i orthogonal to v_0, normalized to a unit vector (if $\xi_i = \pm 1$, we define $\tilde{v}_i$ to be a unit vector orthogonal to all other vectors v_j). Then, we can rewrite the quadratic term $v_i \cdot v_j$ as

$$v_i \cdot v_j = \xi_i \xi_j + \sqrt{1 - \xi_i^2}\sqrt{1 - \xi_j^2}\tilde{v}_i \cdot \tilde{v}_j. \tag{15}$$

As it turns out, the relevant parameter when analyzing the quadratic terms is the scalar product $\tilde{v}_i \cdot \tilde{v}_j$, i.e. how much better we do than if the variables would have been independent (scaled by an appropriate factor). Motivated by this, we make the following definition.

Definition 3.3. The *advantage* $\tilde{\rho}(\theta)$ of a configuration $\theta = (\xi_1, \xi_2, \rho)$ is

$$\tilde{\rho}(\theta) = \frac{\rho - \xi_1 \xi_2}{\sqrt{1 - \xi_1^2}\sqrt{1 - \xi_2^2}}. \tag{16}$$

In the case that $\xi_1 = \pm 1$ or $\xi_2 = \pm 1$, we define $\tilde{\rho}(\theta) = 0$.

Note that, in the notation above, the advantage is exactly the scalar product $\tilde{v}_i \cdot \tilde{v}_j$. We are now ready to define the "positivity condition", alluded to in Section 1.1.

Definition 3.4. A configuration $\theta = (\xi_1, \xi_2, \rho)$ is *positive* if $\hat{P}_3 \cdot \tilde{\rho}(\theta) \geq 0$.

Intuitively, positive configurations should be more difficult to handle, since they are the configurations where we need to do something better than just setting the variables independently in order to get a good approximation ratio.

What Goemans and Williamson [15] originally did to round the vectors back to boolean variables, was to pick a random hyperplane through the origin, and decide the value of the variables based on whether their vectors are on the same side of the hyperplane as v_0 or not. Feige and Goemans [12] suggested several generalizations of this approach, using preprocessing (e.g. first rotating the vectors) and/or more elaborate choices of hyperplanes. In particular, consider a rounding scheme where we pick a random vector $r \in \mathbb{R}^{n+1}$ and then set the variable x_i to true if

$$r \cdot \tilde{v}_i \leq T(v_0 \cdot v_i) \tag{17}$$

for some threshold function $T : [-1, 1] \to \mathbb{R}$. This scheme (and more general ones) was first analyzed by Lewin et al. [25].

To describe the performance ratio yielded by this scheme, we begin by setting up some notation.

Definition 3.5. A *rounding function* is a continuous function $R : [-1, 1] \to [-1, 1]$ which is odd, i.e. satisfies $R(\xi) = -R(-\xi)$. We denote by $\mathcal{R}$ the set of all such functions.

The reason that we require a rounding function to be odd is that a negated literal $-x_i$ should be treated the opposite way as x_i. A rounding R is in one-to-one corresponce with a threshold function T as described above by the simple relation $R(x) = 1 - 2\Phi(T(x))$, where Φ is the normal distribution function (it will turn out to be more convenient to describe the rounding in terms of R rather than in terms of T).

Definition 3.6. The *rounded value* of a configuration θ with respect to a rounding function $R \in \mathcal{R}$ is

$$P_{\text{round}}(\theta, R) = P_{\text{relax}}\big(R(\xi_1), R(\xi_2), 4\Gamma_{\tilde{\rho}(\theta)}(R(\xi_1), R(\xi_2)) + R(\xi_1) + R(\xi_2) - 1\big).$$

This seemingly arbitrary definition is motivated by the following lemma (which essentially traces back to Lewin et al. [25], though they never made it explicit).

Lemma 3.7. *There is a polynomial-time algorithm which, given a* MAX CSP(P) *instance* Ψ*, a semidefinite solution* $\{v_i\}_{i=0}^n$ *to* Ψ*, and a (polynomial-time computable) rounding function* $R \in \mathcal{R}$*, finds an assignment to* Ψ *with expected value*

$$\mathop{\mathbb{E}}_{\theta \in \Theta}\left[P_{\text{round}}(\theta, R)\right], \tag{18}$$

A proof can be found in the full version of this paper. The rounding procedure used is exactly the kind of threshold rounding described above, which is the class of roundings which Lewin et al. [25] called $\mathcal{THRESH}^-$. The rounding function R specifies an arbitrary rounding procedure from $\mathcal{THRESH}^-$.[3]

A statement similar to Lemma 3.7 holds for MAX CSP$^+(P)$, the difference being that, since there are no longer any negated literals, we can change the definition of a rounding function slightly and not require it to be odd (which could potentially give us a better algorithm). Motivated by Lemma 3.7, we make the following sequence of definitions

Definition 3.8. The approximation ratio of a rounding R for a family of configurations Θ is given by

$$\alpha_P(\Theta, R) = \frac{\mathbb{E}_{\theta \in \Theta}\left[P_{\text{round}}(\theta, R)\right]}{\mathbb{E}_{\theta \in \Theta}\left[P_{\text{relax}}(\theta)\right]} \tag{19}$$

[3] In the notation of [25], we have $S(x) = T(x)\sqrt{1 - x^2}$, or equivalently, $R(x) = 1 - 2\Phi(S(x)/\sqrt{1 - x^2})$.

Definition 3.9. The approximation ratio of a family of configurations Θ is given by

$$\alpha_P(\Theta) = \max_{R \in \mathcal{R}} \alpha_P(\Theta, R). \tag{20}$$

Definition 3.10. The approximation ratios of P for families of k configurations and families of k positive configurations, respectively, are given by (recall the definition of a positive configuration from Definition 3.4)

$$\alpha_P(k) = \min_{|\Theta|=k} \alpha_P(\Theta) \tag{21}$$

$$\beta_P(k) = \min_{\substack{|\Theta|=k \\ \text{every } \theta \in \Theta \text{ is positive}}} \alpha_P(\Theta) \tag{22}$$

We would like to point out that we do not require that the family of configurations Θ can be derived from an SDP solution to some MAX CSP(P) instance Ψ – we only require that each configuration in Θ satisfies the inequalities in Equation (12). In other words, we have a lot more freedom when searching for a Θ which makes $\alpha_P(k)$ or $\beta_P(k)$ small, than we would have when searching for MAX CSP(P) instances and corresponding vector solutions.

Finally, we define

$$\alpha(P) = \lim_{k \to \infty} \alpha_P(k), \qquad \beta(P) = \lim_{k \to \infty} \beta_P(k). \tag{23}$$

These are the approximation ratios arising in Theorems 1.1 and 1.2. Ideally, of course, we would like to prove hardness of approximating MAX CSP(P) within $\alpha(P)$ rather than $\beta(P)$, getting rid of the requirement that every $\theta \in \Theta$ must be positive. The reason that we need it shows up when we do the proof of soundness for the PCP constructed in Section 5, and we have not been able to get around this. However, as we state in Conjecture 1.3, we do not *believe* that this restriction affects the approximation ratio achieved: by the intuition above, positive configurations seem to be the ones that are hard to round, so restricting our attention to such configurations should not be a problem. And indeed, the configurations we use to show hardness for MAX 2-AND are all positive, as are all configurations which have appeared in previous proofs of hardness for 2-CSPs (e.g. for MAX CUT and MAX 2-SAT).

4 The Approximation Algorithm

The approximation algorithm for MAX CSP(P) (Theorem 1.1) is based on the following theorem.

Theorem 4.1. *For any $\epsilon > 0$, the value of a* MAX CSP(P) *instance on k clauses can be approximated within $\alpha_P(k) - \epsilon$ in time polynomial in k.*

A proof is given in the full version of this paper. Note that this theorem immediately implies Theorem 1.1 since $\alpha_P(k) \geq \alpha(P)$. We remark that the exact value of $\alpha_P(k)$ is virtually impossible to compute for large k, making it somewhat hard to compare Theorem 4.1 with existing results. However, for MAX CUT, MAX 2-SAT and MAX 2-AND, it is not hard to prove that $\alpha(P)$ is at least the performance ratio of existing algorithms.

We remark that the running time of the algorithm has a quite bad dependency on ϵ; it scales as $(1/\epsilon)^{\Omega(1/\epsilon^2)}$.

5 The PCP Reduction

Theorem 1.2 immediately follows from the following Theorem 5.1 below. Taking k large enough so that $\beta_P(k) \leq \beta(P) + \epsilon$ and invoking Theorem 5.1 gives hardness of approximating MAX CSP(P) within $\beta(P) + 2\epsilon$.

Theorem 5.1. *Assuming the Unique Games Conjecture, it is NP-hard to approximate* MAX CSP(P) *within $\beta_P(k) + \epsilon$ for any $\epsilon > 0$ and $k \in \mathbb{N}$.*

We prove Theorem 5.1 by constructing a PCP verifier which checks a supposed long coding of a good assignment to a UNIQUE LABEL COVER instance, and decides whether to accept or reject based on the evaluation of the predicate P on certain bits of these long codes. The verifier is parametrized by a family of k positive configurations $\Theta = \{\theta_1, \ldots, \theta_k\}$ and a probability distribution on Θ. Again, we point out that the requirement that the configurations of Θ are positive is by necessity rather than by choice, and if we could get rid of it, the hardness of approximation yielded would exactly match the approximation ratio from Theorem 1.1. The set Θ corresponds to a set of vector configurations for the semidefinite relaxation of MAX CSP(P). When proving soundness, i.e., in the case that there is no good assignment to the UNIQUE LABEL COVER instance, we prove that the best strategy for the prover corresponds to choosing a good rounding function R for the family of configurations Θ. Choosing a set of configurations which are hard to round, we obtain the desired result.

Since we can negate variables freely, we will assume that the purported long codes are folded over true[4] (by selecting, for each pair $(x, -x)$ of inputs one representative, say x, and then look up the value at $-x$ by reading the value at x and negating the answer). Intuitively, this ensures that the prover's rounding function is odd, i.e. that $R(\xi) = -R(-\xi)$. For a permutation $\sigma \in \mathfrak{S}_L$ and a bitstring $x \in \{-1, 1\}^L$, we denote by $\sigma x \in \{-1, 1\}^L$ the string x permuted according to σ, i.e., $\sigma x = x_{\sigma(1)} x_{\sigma(2)} \ldots x_{\sigma(L)}$. The verifier is given in Algorithm 1.

[4] A function $f : B_q^n \to \mathbb{R}$ is said to be *folded over true* if $f(-x) = -f(x)$ for all x.

Algorithm 1: The verifier $\mathcal{V}_\Theta$
$\mathcal{V}_\Theta(X, \Sigma = \{f_v\}_{v \in V})$

(1) Pick a random configuration $\theta = (\xi_1, \xi_2, \rho) \in \Theta$ according to the distribution on Θ.
(2) Pick a random $v \in V$.
(3) Pick $e_1 = \{v, w_1\}$ and $e_2 = \{v, w_2\}$ randomly from $E(v)$.
(4) Pick $x_1, x_2 \in \{-1, 1\}^L$ such that each bit of x_i is picked independently with expected value ξ_i and that the j:th bits of x_1 and x_2 are ρ-correlated for $j = 1, \ldots, L$.
(5) For $i = 1, 2$, let $b_i = f_{w_i}(\sigma^v_{e_i} x_i)$ (folded over true).
(6) Accept with probability $P(b_1, b_2)$.

The completeness and soundness of $\mathcal{V}_\Theta$ are as follows (proofs can be found in the full version of this paper).

Lemma 5.2 (Completeness)**.** *If* $\mathrm{Val}(X) \geq 1 - \eta$*, then there is a proof* Σ *such that*

$$\Pr[\mathcal{V}_\Theta(X, \Sigma) \textit{ accepts}] \geq (1 - 2\eta) \mathop{\mathbb{E}}_{\theta \in \Theta}[P_{\mathrm{relax}}(\theta)] \quad (24)$$

Lemma 5.3 (Soundness)**.** *For every* $\epsilon > 0$ *there is a* $\gamma > 0$ *such that if* $\mathrm{Val}(X) \leq \gamma$*, then for any proof* Σ*, we have*

$$\Pr[\mathcal{V}_\Theta(X, \Sigma) \textit{ accepts}] \leq \max_{R \in \mathcal{R}} \mathop{\mathbb{E}}_{\theta \in \Theta}[P_{\mathrm{round}}(\theta, R)] + \epsilon. \quad (25)$$

Combining the two lemmas and picking η small enough, we get that it is Unique Games-hard to approximate MAX CSP(P) within

$$\max_{R \in \mathcal{R}} \frac{\mathbb{E}_{\theta \in \Theta}[P_{\mathrm{round}}(\theta, R)]}{\mathbb{E}_{\theta \in \Theta}[P_{\mathrm{relax}}(\theta)]} + \mathcal{O}(\epsilon) = \alpha_P(\Theta) + \mathcal{O}(\epsilon)\,. \quad (26)$$

Picking a Θ with $|\Theta| = k$ that minimizes $\alpha_P(\Theta)$, we obtain Theorem 5.1.

6 Application to MAX 2-AND

Using the machinery developed in Sections 3 and 5, we are able to obtain an upper bound of $\beta(P) \leq 0.87435$ for the case when $P(x_1, x_2) = x_1 \wedge x_2$, i.e., the MAX 2-AND problem, establishing Theorem 1.4. We do this by exhibiting a set Θ of $k = 4$ (positive) configurations on 2 distinct non-zero ξ-values (and a probability distribution on the elements of Θ), such that $\alpha_P(\Theta) < 0.87435$. The set of configurations $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ is given in Table 2. The values of ξ_A and ξ_B are

$$\begin{aligned} \xi_A &= 0.31988 \\ \xi_B &= 0.04876. \end{aligned}$$

Configuration			Probability
$(0,$	$-\xi_A,$	$1 - \xi_A)$	0.52850
$(0,$	$\xi_A,$	$1 - \xi_A)$	0.05928
$(\xi_A,$	$-\xi_B,$	$1 - \xi_A - \xi_B)$	0.29085
$(-\xi_A,$	$\xi_B,$	$1 - \xi_A - \xi_B)$	0.12137

Table 2. Configurations used for MAX 2-AND

To compute the hardness factor given by this set of configurations, we must compute

$$\alpha_P(\Theta) = \max_{R \in \mathcal{R}} \frac{\mathbb{E}_{\theta \in \Theta}[P_{\mathrm{round}}(\theta, R)]}{\mathbb{E}_{\theta \in \Theta}[P_{\mathrm{relax}}(\theta)]}. \quad (27)$$

Since $P(x_1, x_2) = \frac{1 - x_1 - x_2 + x_1 x_2}{4}$ we have that for $\theta = (\xi_1, \xi_2, \rho)$,

$$\begin{aligned} P_{\mathrm{relax}}(\theta) &= \frac{1 - \xi_1 - \xi_2 + \rho}{4} \\ P_{\mathrm{round}}(\theta, R) &= \Gamma_{\tilde{\rho}(\theta)}(R(\xi_1), R(\xi_2)). \end{aligned}$$

To specify such an R, it is enough to specify the values of R on the two angles ξ_A and ξ_B. Figure 1 gives a contour plot of the right-hand side of Equation (27), as a function of the values of $R(\xi_A)$ and $R(\xi_B)$. There are two local maxima, one around the point $(R(\xi_A), R(\xi_B)) \approx (0.27846, 0.044376)$, and one around the point $(1, -1)$. Figure 2 gives a contour plot of the area around the first point. This maximum turns out to be approximately 0.87434075. At the point $(1, -1)$ (which is indeed the other maximum), the approximation ratio is approximately 0.87434007. Thus, we have $\alpha_P(\Theta) \leq 0.87435$.

Note that in general, given a Θ (and a probability distribution on its elements), the very problem of computing $\alpha_P(\Theta)$ is a difficult numeric optimization problem. However, for the Θ we use, the number of distinct ξ-values used is small, so that computing $\alpha_P(\Theta)$ in this case is a numeric optimization problem in 2 variables, which we are able to handle.

Using only one non-zero ξ-value one can obtain a bound of 0.87451, i.e., only slightly worse than the best bound we have been able to achieve using two ξ-values. Details about this can be found in the full version of this paper. It seems likely that additional improvements can be made by using more and more ξ-values, though these improvements will be quite small. Indeed, using larger Θ we are able to improve upon Theorem 1.4, but the improvements we have been able to make are minute (of order 10^{-5}), and it becomes a lot more difficult to verify them.

7 Concluding Remarks

We remark that it is a fairly straightforward task to adapt these results to the MAX CSP$^+(P)$ problem, obtaining statements analogous to Theorems 1.1 and 1.2. The only

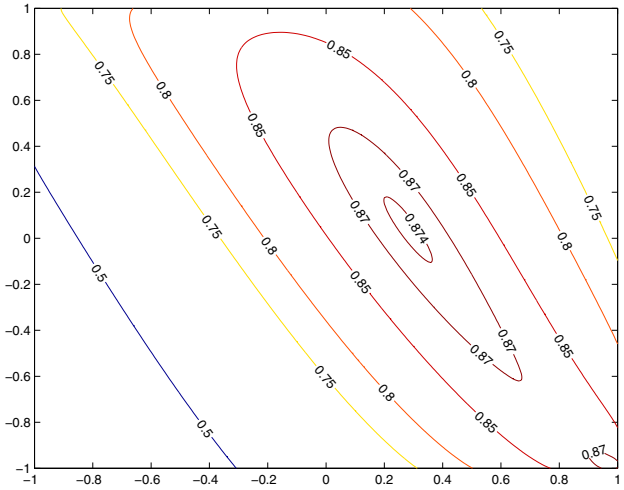

Figure 1. The entire range of R

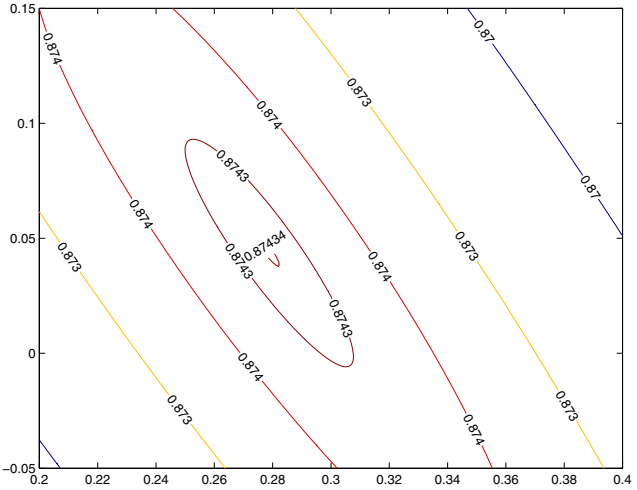

Figure 2. Restricted to the critical area

difference is that we drop the requirement that a rounding function has to be odd (since we cannot fold the long codes over true anymore, we would not be able to enforce such a constraint). However, in doing so, we also lose the possibility to force a rounding function R to satisfy $R(0) = 0$. The configurations that we use for proving hardness of MAX 2-AND rely heavily on this property, and it is for this reason that those results do not apply to the MAX DI-CUT problem directly. In other words, we are not able to obtain a statement similar to Theorem 1.4 for the MAX DI-CUT problem. Whether this is because the MAX DI-CUT problem is easier to approximate than MAX 2-AND, or whether we just have to spend some more time searching for a "bad" set of configurations, we do not know, but we conjecture that the latter is true and that they are equally hard. However, today we do not even know whether balanced instances of the MAX DI-CUT problem are the hardest or not.

If P is monotone, the MAX CSP$^+(P)$ problem is trivially solvable, so there are cases where MAX CSP$^+(P)$ is easier than MAX CSP(P). Lacking results on MAX DI-CUT, it would be interesting to determine whether there are other examples than these trivial ones. A good candidate would probably be an "almost monotone" P (recall that P is real-valued.).

Recently, O'Donnell and Wu have done a complete analysis of the "approximability curve" of the MAX CUT problem, exhibiting an algorithm, integrality gap, and UGC-based hardness result which all match [30]. It will be interesting to see whether their results can be extended to other MAX 2-CSP problems.

7.1 Acknowledgements

I would like to thank Johan Håstad for luring me into the world of 2-CSPs in the first place, as well as for his patience and many insightful comments along the way.

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. In *STOC 2007*, pages 189–197, 2007.

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

[9] 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.

[10] I. Dinur, E. Friedgut, and O. Regev. Independent sets in graph powers are almost contained in juntas. *Geometric and Functional Analysis*, 2006. to appear.

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

[12] 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.

[13] U. Feige and G. Schechtman. On the optimality of the random hyperplane rounding technique for MAX CUT. *Random Struct. Algorithms*, 20(3):403–440, 2002.

[14] A. M. Frieze and M. Jerrum. Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997.

[15] 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.

[16] E. Halperin, R. Nathaniel, and U. Zwick. Coloring k-colorable graphs using smaller palettes. In *SODA 2001*, pages 319–326, 2001.

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

[18] J. Håstad. On the approximation resistance of a random predicate. To appear in RANDOM-APPROX, 2007.

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

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

[21] 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, 2004.

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

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

[24] 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.

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

[26] S. Matuura and T. Matsui. 0.863-Approximation Algorithm for MAX DICUT. In *RANDOM-APPROX*, pages 138–146, 2001.

[27] 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.

[28] E. Mossel. Gaussian bounds for noise correlation of functions. arXiv Report math/0703683v2, 2007.

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

[30] R. O'Donnell. Personal communication, 2007.

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

[32] L. Trevisan. Parallel Approximation Algorithms by Positive Linear Programming. *Algorithmica*, 21:72–88, 1998.

[33] L. Trevisan, G. B. Sorkin, M. Sudan, and D. P. Williamson. Gadgets, Approximation, and Linear Programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000.

[34] U. Zwick. Approximation Algorithms for Constraint Satisfaction Problems Involving at Most Three Variables Per Constraint. In *SODA 1998*, 1998.