Beating a random assignment

Gustav Hast *

Abstract

MAX CSP(P) is the problem of maximizing the weight of satisfied constraints, where each constraint acts over a k-tuple of literals and is evaluated using the predicate P. The approximation ratio of a random assignment is equal to the fraction of satisfying inputs to P. If it is NP-hard to achieve a better approximation ratio for MAX CSP(P), then P is said to be *approximation resistant*. Our goal is to characterize which predicates that have this property.

A general approximation algorithm for MAX CSP(P) is introduced. For a multitude of different P, it is shown that the algorithm beats the random assignment algorithm, thus implying that P is not approximation resistant. In particular, over 2/3 of the predicates on four binary inputs are proved not to be approximation resistant, as well as all predicates on $2s$ binary inputs, that have at most $2s+1$ accepting inputs. The algorithm also has interesting implications on the characterization of almost satisfying solutions to hard instances of MAX CSP(P). For example MAX CSP(P), where P is a disjunction of three literals, is the famous MAX E3SAT problem. It is shown that an almost satisfying solution to a hard MAX E3SAT instance only makes a negligible fraction of the disjunctions have exactly two literals that are true.

We also prove a large number of predicates to be approximation resistant. In particular, all predicates of arity $2s+s^2$ with less than 2^{s^2} non-accepting inputs are proved to be approximation resistant, as well as almost 1/5 of the predicates on four binary inputs.

1 Introduction

For a constraint satisfaction problem, CSP, the objective is to satisfy a collection of constraints by finding a good assignment. In the maximization version, MAX CSP, each constraint has a weight and the objective is to maximize the weight of satisfied constraints. By restricting what types of constraints that are allowed, different types of problems can be defined. Two well-known examples are MAX CUT and MAX E3SAT.

A constraint in a MAX CUT instance consists of two binary variables, x_i and x_j, and it is satisfied iff $x_i \neq x_j$. Constraints of MAX E3SAT are disjunctions over three literals, where a literal is a variable or a negated variable. Both these problems are NP-complete. A naive algorithm for these problems, as well as all MAX CSPs, is simply assigning all variables a random value, without looking at the constraints. It is easy to see that such a random assignment on average satisfies half of the constraints in a MAX CUT instance and 7/8 of the constraints in a MAX E3SAT instance.

An algorithm is said to α-approximate a problem if the value of a produced solution is at least $\alpha \cdot OPT$, where OPT is the value of an optimal solution. For randomized algorithms, we allow this value to be an expected value over the random choices made by the algorithm. A major breakthrough in approximating various MAX CSPs was made when Goemans and Williamson used a semidefinite relaxation approach in order to 0.878-approximate MAX CUT. Before this, there was no known polynomial time $1/2+\epsilon$-approximation for any constant $\epsilon > 0$, thus no known efficient algorithms outperformed a random assignment in approximating MAX CUT. Since then, the semidefinite relaxation

*Department of Numerical Analysis and Computer Science, Royal Institute of Technology, SE - 100 44 Stockholm, Sweden. `email: ghast@nada.kth.se. phone: +46-8-790 65 40.`

technique has been used extensively in order to achieve better approximation ratios for a large number of Max CSP problems, e.g. in [14, 16, 20, 22].

But the use of semidefinite relaxation cannot be applied to all CSPs with success. Based on PCP techniques, a different line of research produced results showing NP-hardness of approximating various problems [1, 2, 6, 7, 17, 18]. Håstad [11] introduced a technique that enabled very efficient analysis of PCPs and could show that there is no hope to efficiently approximate Max E3SAT better than 7/8. Thus, in short of proving P = NP, the best possible efficient approximation algorithm for Max E3SAT, up to low order terms, is simply to pick a random assignment.

Max CSP(P) is the problem of maximizing the weight of satisfied constraints, where each constraint acts over a k-tuple of literals and is evaluated using the predicate P. As Creignou [4] and Khanna et al. [15] showed, Max CSP(P) is Max SNP-hard for all predicates depending on at least two variables. Thus, only for the unary predicate the problem can be solved optimally. An interesting distinction is whether a predicate allows a nontrivial approximation, i.e., can be approximated better than using a random assignment. If it is NP-hard to outperform a random assignment on Max CSP(P), then P is said to be *approximation resistant*. There are many reasons for studying approximation resistance. Approximation resistance is a fundamental property of a predicate which determines if anything nontrivial can be done in approximating the Max CSP. It is thus an important structural question. We believe that understanding this concept is a key to comprehending what it is that makes some problems hard to approximate. Approximation resistant predicates also play a fundamental role in the design of efficient PCPs.

There are no predicates of arity two that are approximation resistant [9], even if the input variables are non-binary [12]. Håstad [11] showed that if P depends on three binary inputs and accepts all odd parity inputs or all even parity inputs, then P is approximation resistant. Zwick [20] completed the characterization for predicates of arity three by producing approximation algorithms for Max CSP(P), for all other predicates P, that outperform a random assignment. Samorodnitsky and Trevisan [19] constructed approximation resistant predicates of arity $2s + s^2$ with only 2^{2s} accepting assignments. A predicate P implies P' if $P(y) \Rightarrow P'(y)$ for all y. In this paper we show that all predicates implied by a Samorodnitsky-Trevisan predicate are approximation resistant as well. As a consequence, all predicates of arity $2s + s^2$ with at most 2^{s^2} non-accepting inputs are also approximation resistant. On the other hand, we also show that if a predicate of arity $2s$ has at most $2s + 1$ accepting inputs, then the predicate is not approximation resistant. Thus, there is a tendency that the more inputs a predicate accepts, the harder it is to beat a random assignment for the corresponding Max CSP. It could be tempting to assume that if a predicate P is approximation resistant and P implies P', then P' is approximation resistant as well. As Zwick showed, this is true for all predicates of arity three and it is also true if P is a Samorodnitsky-Trevisan predicate. However, this is not true in general. While studying predicates of arity four, we observed that the predicate $(\overline{x}_1 \wedge (x_2 \equiv x_3)) \vee (x_1 \wedge (x_2 \equiv x_4))$, that has been shown to be approximation resistant [10], implies that $\overline{x}_2$, x_3 and x_4 cannot all be equal. Zwick [20] showed that this latter predicate is not approximation resistant.

1.1 Our contribution

Among other contributions, we introduce a general technique for approximating Max CSP instances. It works by finding a solution that gives a positive contribution on the small degree terms of the Fourier spectra of the objective function, while seeing to that the higher order terms give a smaller positive or negative contribution. By using a global analysis of the instance, we prove that the method beats a random assignment on Max CSP(P) for a multitude of different predicates P, thus showing them not to be approximation resistant. In addition, we give non-trivial conditions that optimal solutions to hard Max CSP instances have to adhere to. We also show a large number of predicates to be approximation resistant.

Method The objective function of a Max CSP(P) instance, I, is the weighted sum of the indicator variables of the constraints. In Section 2 we show that each such term can be written as a multilinear expression. The whole sum can thus also be written as a multilinear expression

$$I(x_1, \ldots x_n) = \sum_{i=1}^{m} w_i P(z_{i1}, \ldots z_{ik}) = \sum_{S \subseteq [n], |S| \leq k} c_S \prod_{i \in S} x_i \ . \tag{1}$$

In fact this is a Fourier spectra of the objective value function. Given a random assignment, where each binary variable x_i is assigned -1 or 1 randomly and independently, all terms of the above expression that includes at least one variable x_i have an expected value of zero. The only term that gives a contribution to the expectation is then the constant term, $c_\emptyset$. Thus, the expected value of a random assignment is equal to $c_\emptyset$. In order to beat a random assignment, we have to produce an assignment with an expected weight of at least $c_\emptyset + \gamma W$, for a constant $\gamma > 0$ and where W is the total weight of the instance.

If there is a large weight on the linear terms of the Fourier spectra, $\sum_{i=1}^{n} |c_{\{i\}}| = \delta W$ for a constant $\delta > 0$, then it is easy to assign values to the variables such that linear terms give a contribution of δW. The problem is, that there is no control on the higher order terms which possibly could give a negative contribution that cancel out the contribution from the linear terms. In order to reduce the contribution of the higher order terms, we therefore only give a small bias towards the solution that maximize the linear terms. With probability ε, depending on k and δ, we assign x_i to $\operatorname{sgn}(c_{\{i\}})$, and otherwise assign x_i a random value. The linear terms then give a positive contribution of $\varepsilon\delta W$ at the same time as the absolute contribution of each term of order i is decreased with a factor of ε^i. The total absolute contribution of the higher order terms can therefore be made to be at most half of the contribution of the linear terms by choosing an ε small enough. Thus, the algorithm finds an assignment with expected value of at least $c_\emptyset + \varepsilon\delta W/2$ and thereby it beats a random assignment.

We can apply the same method on the bi-linear terms, if there exists an assignment that gives a non-negligible value on the those terms. In that case, we use Charikar and Wirth's approximation algorithm for quadratic programs [3], which is based on semidefinite programming, in order to find an assignment that gives a non-negligible value on the bi-linear terms. We use that algorithm, because even if the optimal value is small it is guaranteed to outperform a the expected value of a random assignment. We note that we also could use one of the Max CUT algorithms in [8, 22].

In order to show that a predicate is not approximation resistant, it is enough to create an approximation algorithm that beats a random assignment on satisfiable and almost satisfiable instances. This is because a random assignment achieves the same expected objective value irrespective of the satisfiability of the instance. Thus, satisfiable and almost satisfiable instances are the ones where a random assignment achieves the worst approximation ratio. By combining the algorithm for almost satisfiable instances with a random assignment, we then get a better approximation ratio than using a random assignment alone. For some predicates P it is possible to show that an assignment that almost satisfies an instance of Max CSP(P) has to give a non-negligible positive value on either the linear or bi-linear terms of the Fourier spectra. Thus satisfying the condition for our algorithm to outperform a random assignment, which in turn shows that P is not approximation resistant.

Application 1 The predicates P and P' are of the same *type* if Max CSP(P) and Max CSP(P') are essentially the same problem. There are exactly 400 different non-trivial predicate types of arity four. By using the above technique we show that 270 of these are not approximation resistant. In particular, all predicates with at most six accepting inputs are not approximation resistant. Extending methods in [11] we derive that 70 predicate types are approximation resistant, thus all but 60 predicate types of arity four are characterized.

Application 2 We show that if P is a predicate of arity $2s$ and has at most $2s+1$ accepting inputs, then it is not approximation resistant. This shows that the approximation resistant predicate of Engebretsen and Holmerin [5], with arity six and eight accepting inputs, is optimal in the sense of minimizing the number of accepting inputs. The method also gives lower bounds on the soundness for a natural class of probabilistically checkable proofs. These PCPs express NP and have perfect or almost perfect completeness and soundness negligibly higher than the acceptance probability of a random proof.

Application 3 Our method also gives interesting implications for optimal solutions to hard instances of some Max CSPs. For example, Håstad produced almost satisfiable and satisfiable instances of Max E3SAT such that it is NP-hard to approximate them beyond the random assignment threshold. The proposed satisfying assignment made either one or all three of the literals true for almost all constraints. We show that this is no coincidence. In fact, if there is an almost satisfying assignment that on a non-negligible fraction of the constraints make exactly two literals true, then the instance can be approximated better than using a random assignment. This gives us some new interesting insights about the hardness of approximating Max E3SAT.

Approximation resistant predicates A Samorodnitsky-Trevisan predicate $P_{ST^{s,t}}$ consists of st different parity checks, each acting on three bits of the input,

$$P_{ST^{s,t}}(x_1,\dots x_s,x'_1,\dots x'_t,x''_1,\dots x''_{st}) \quad = \quad \bigwedge_{\substack{1\le i\le s\\ 1\le j\le t}} x_i\oplus x'_j\oplus x''_{(i-1)s+j} \; .$$

Samorodnitsky and Trevisan [19] showed that such a predicate is approximation resistant. In this paper we revisit the proof and show that all predicates implied by a Samorodnitsky-Trevisan predicate are approximation resistant as well. A consequence of this is that all predicates of arity $s+t+st$ with less than 2^{st} non-accepting inputs are approximation resistant. We also show that predicates of arity four and five, having at most two and five non-accepting inputs respectively, are approximation resistant. This is optimal in the sense that there are predicates with one more non-accepting input that we show are not approximation resistant.

1.2 Organization of this paper

Section 2 gives some notation and definitions used in the paper. In Section 3 we describe the method of exploiting the existence of a good solution on low-degree terms in order to beat a random assignment. In the following section we use this method in order to show that predicates with few accepting inputs are not approximation resistant and predicates with few non-accepting inputs are approximation resistant. In Section 5 we take a closer look at predicates of arity four. The subsequent section shows that approximation resistance is a non-monotone property. Section 7 is devoted to characterizing optimal solutions to hard instances of Max CSP(P). The paper ends with some final remarks. All proofs of theorems and lemmas have been put in the appendix.

2 Preliminaries

A predicate P maps elements from $\{\pm1\}^k$ to $\{0,1\}$. For notational convenience we let input bits have value -1, denoting true, and 1, denoting false. If P accepts an input y, then $P(y)=1$, otherwise $P(y)=0$. Thus, the set of accepting inputs to P is denoted by $P^{-1}(1)$.

Max CSP(P) is the Max CSP problem where all constraints are evaluated using a predicate $P:\{\pm1\}^k\to\{0,1\}$. More formally, an instance of Max CSP(P) consists of m weighted constraints, each one acting over a k-tuple of literals, $(z_{i1}, \dots z_{ik})$, taken from the set $\{x_1, \dots x_n, \overline{x}_1, \dots \overline{x}_n\}$. All variables in such a tuple are assumed to be distinct. A constraint is satisfied if P maps its k-tuple of literals onto one. The objective is to maximize $\sum_{i=1}^{m} w_i P(z_{i1}, \dots z_{ik})$, where w_i is the (non-negative) weight of the i:th constraint. The weight of an input $(y_1, \dots y_k)$ for an assignment a to an instance of Max CSP(P) is $\sum_{i:(z_{i1},\dots z_{ik})=y} w_i$, where each constraint tuple is evaluated on a. Two predicates, P and P', are of the same *type* iff there is a permutation π on $[k]$ and $a \in \{\pm 1\}^k$ such that $P(x_1, \dots x_k) = P'(a_1 x_{\pi(1)}, \dots a_k x_{\pi(k)})$ for all $x \in \{\pm 1\}^k$. A Max CSP(P) instance can be expressed as a Max CSP(P') instance by permuting and applying a bitmask to the constraint tuples, so clearly they are equivalent problems.

The following definition formally describes an approximation resistant predicate.

Definition 1. *A predicate P is* approximation resistant *if, for every constant $\epsilon > 0$, it is* NP*-hard to approximate* Max CSP(P) *within $2^{-k}|P^{-1}(1)| + \epsilon$.*

A predicate can be seen as a sum of conjunctions, each conjunction corresponding to an accepting input to P. If $x, y \in \{\pm 1\}^k$ then $\sum_{S\subseteq[k]} \prod_{i\in S} x_i y_i$ equals 2^k if $x = y$ and otherwise the sum is zero. Thus, a conjunction $\alpha_1 x_1 \wedge \dots \wedge \alpha_k x_k$, where $\alpha \in \{\pm 1\}^k$, can be arithmetized as

$$\psi_\alpha(x_1, \dots x_k) = 2^{-k} \sum_{S\subseteq[k]} \prod_{i\in S} \alpha_i x_i = \begin{cases} 1 & \text{if } \alpha = (x_1, \dots x_k) \\ 0 & \text{otherwise} \end{cases} .$$

A predicate can thus be expressed as a multilinear expression

$$P(x_1, \dots x_k) \quad = \sum_{\alpha\in P^{-1}(1)} \psi_\alpha(x_1, \dots x_k) \ . \tag{2}$$

We let $P^{(i)}$ denote the sum of the i-degree terms of the mulitlinear expression P, and $P^{(\geq i)}$ denotes the sum of the terms of at least degree i in P. Thus,

$$P^{(i)}(x_1, \dots x_k) \quad = \quad 2^{-k} \sum_{\alpha\in P^{-1}(1)} \sum_{S\subseteq[k],|S|=i} \prod_{i\in S} \alpha_i x_i \ .$$

3 General technique

As mentioned in the introduction, the objective function of a Max CSP(P) instance can be described as a multilinear expression, see (1). In this section we show that if there is an assignment that gives a non-negligible positive value to the linear or bi-linear terms of this expression, then we can do better than just picking a random assignment.

Consider the linear terms of the objective value function, $I^{(1)}(x_1, \dots x_n) = \sum_{i=1}^{n} c_{\{i\}} x_i$. It is easy to see that the linear terms are maximized by assigning x_i to $\operatorname{sgn}(c_{\{i\}})$, giving the value $\sum_i |c_{\{i\}}|$. However, in order to control the expected value of higher order terms Algorithm **Lin** picks an assignment that is ε-biased towards this assignment. The expected value of the i'th order terms decrease their value with a factor of ε^i. By choosing a small enough bias we ensure that the expected value of the non-constant terms is dominated by the linear terms. The following theorem quantifies the performance of Algorithm **Lin**.

Theorem 2. *Let $I(x_1, \dots x_n) = \sum_{S\subseteq[n],|S|\leq k} c_S \prod_{i\in S} x_i$ be the objective value function for an instance of* Max CSP(P)*, $P : \{\pm 1\}^k \to \{0, 1\}$. Denote the sum of all constraints' weights W. If $\sum_i |c_{\{i\}}| \geq \delta W$ for a constant $\delta > 0$, then Algorithm* **Lin***, with $\varepsilon = \delta/2k$, produces a solution of expected weight at least $c_\emptyset + \frac{\delta^2}{4k} W$.*

For each $i := 1, \ldots n$ do:

- with probability ε: assign $x_i := \begin{cases} 1 & \text{if } c_{\{i\}} \geq 0 \\ -1 & \text{otherwise} \end{cases}$,
- or else assign x_i according to an unbiased coin flip.

Figure 1: Algorithm **Lin** - for approximating MAX CSP(P).

Input: A MAX CSP(P) instance I

1. Approximate the quadratic program that contains of the bi-linear terms, $I^{(2)}$, using the Charikar-Wirth algorithm [3]. Let $a_1, \ldots a_n$ be the produced solution.
2. If $I^{(1)}(a_1, \ldots a_n) < 0$ then do: $a_i := \overline{a}_i$ for $i := 1, \ldots n$.
3. Set $\alpha = I^{(2)}(a_1, \ldots a_n)/W$, where W is the total weight in I. Set $\varepsilon = \max(\alpha/2k^{3/2}, 0)$, where k is the arity of P.
4. For $i := 1, \ldots n$ do:
   - With probability ε: assign $x_i := a_i$,
   - or else assign x_i according to an unbiased coin flip.

Figure 2: Algorithm **BiLin** - for approximating MAX CSP(P).

If there exists an assignment that makes the sum of the bi-linear terms non-negligibly positive, then Algorithm **BiLin** outperforms a random assignment.

Theorem 3. *Let $I(x_1, \ldots x_n) = \sum_{S \subseteq [n], |S| \leq k} c_S \prod_{i \in S} x_i$ be the objective value function for an instance of* MAX CSP(P), $P : \{\pm 1\}^k \to \{0, 1\}$. *Denote the sum of all constraints' weights W. If there exists an assignment such that $\sum_{1 \leq i < j \leq n} c_{\{i,j\}} x_i x_j \geq \delta W$ for a constant $\delta > 0$, then Algorithm* **BiLin** *produces a solution of expected weight at least $c_\emptyset + \kappa W$, where $\kappa > 0$ and only depends on constants δ and k.*

From Theorems 2 and 3 we derive Lemma 4 and its generalization, Lemma 5. These lemmas are used in order to show that predicates are not approximation resistant. A crucial observation in their proofs is that in order to achieve a better approximation algorithm than a random assignment, we only have to outperform a random assignment on satisfiable or almost satisfiable instances. Those instances are the ones where a random assignment achieves the worst approximation ratio.

Lemma 4. *Let $P : \{\pm 1\}^k \to \{0, 1\}$ be a predicate. If $P^{(1)}(y) + P^{(2)}(y) > 0$ for all $y \in P^{-1}(1)$, then P is not approximation resistant.*

Lemma 5. *Let $P : \{\pm 1\}^k \to \{0, 1\}$ be a predicate and $I(x_1, \ldots x_n) = \sum_{i=1}^{m} w_i P(z_{i1}, \ldots z_{ik})$ be the objective value of a* MAX CSP(P) *instance. For an assignment to $x_1, \ldots x_n$, define the weight on each input y, $u_y = \sum_{i:(z_{i1}, \ldots z_{ik}) = y} w_i$. For constants C and $\delta > 0$, there exist constants $\varepsilon > 0$ and $\kappa > 0$ such that if $\sum_{y \in P^{-1}(1)} u_y (C \cdot P^{(1)}(y) + P^{(2)}(y)) \geq \delta W$, where $W = \sum_{i=1}^{m} w_i$, then either the value of the assignment is less than $(1 - \varepsilon) W$, or Algorithm* **Lin** *or Algorithm* **BiLin** *will achieve an approximation ratio of at least $2^{-k} |P^{-1}(1)| + \kappa$.*

By using a gadget construction to MAX 2SAT and applying Zwick's algorithm for almost satisfiable MAX 2SAT instances, we show the following theorem.

Theorem 6. *Let $P : \{\pm 1\}^k \to \{0,1\}$ be a predicate. If for some constant C, $C \cdot P^{(1)}(y) + P^{(2)}(y)$ is zero for at most two $y \in P^{-1}(1)$ and strictly positive for all other $y \in P^{-1}(1)$, then P is not approximation resistant.*

4 Predicates on k variables

If a predicate has few accepting inputs, then we can show by using Lemma 4 that the prdicate is not approximation resistant.

Theorem 7. *Let $P : \{\pm 1\}^k \to \{0,1\}$, $k \geq 3$. If P has at most $2\lfloor k/2 \rfloor + 1$ accepting inputs, then P is not approximation resistant.*

Remark 8. *Engebretsen and Holmerin [5] designed an approximation resistant predicate $P : \{\pm 1\}^6 \to \{0,1\}$, with only eight accepting inputs. The above theorem is thus tight for the case $k = 6$ as it stipulates that if a predicate has at most seven accepting inputs, then it is not approximation resistant.*

In the remaining part of this section we show sufficient conditions on predicates to be approximation resistant. The following theorem shows that all predicates implied by a Samorodnitsky-Trevisan predicate is approximation resistant. It turns out that a large part of the proof is essentially a reiteration of the proof of Håstad and Wigderson [13], which is a simpler proof of the Samorodnitsky-Trevisan PCP than the original one.

Theorem 9. *Let $P : \{\pm 1\}^{s+t+st} \to \{0,1\}$, be a predicate that is implied by $P_{ST^{s,t}}$, i.e., $P^{-1}_{ST^{s,t}}(1) \subseteq P^{-1}(1)$, then P is approximation resistant.*

If a predicate P has enough accepting inputs, then there is at least one predicate of the same type as P, such that it is implied by a Samorodnitsky-Trevisan predicate. Using Theorem 9, we then conclude that P is approximation resistant.

Theorem 10. *Let $k \geq s+t+st$ and $P : \{\pm 1\}^k \to \{0,1\}$ be a predicate with at most $2^{st}-1$ non-accepting inputs, then P is approximation resistant.*

Now we consider predicates of arity five. Theorem 10, with $s = 1$ and $t = 2$, lets us conclude that if $|P^{-1}(0)| \leq 3$ then P is approximation resistant. By taking more care in the analysis we are able to show the following theorem.

Theorem 11. *Let P be a predicate on 5 binary variables with at most 5 non-accepting inputs, then P is approximation resistant.*

Remark 12. *The next section shows that there are predicates of arity four, having exactly three non-accepting inputs, which are not approximation resistant. This implies that there are also predicates of arity five, having exactly six non-accepting inputs, which are not approximation resistant. Thus, Theorem 11 is optimal in this sense.*

5 Predicates on four variables

In this section we take a closer look at predicates of arity four. Theorem 7 stipulates that if a predicate $P : \{\pm 1\}^4 \to \{0,1\}$ has at most five accepting inputs then it is not approximation resistant. That was already known as a result of the Max 4CSP 0.33-approximation algorithm of Guruswami et al. [10], because $5/2^4 < 0.33$. By applying Theorem 6 we show that no predicate that accepts exactly six inputs is approximation resistant. We also show many predicates on four variables to be approximation resistant by extending methods from [11]. Out of exactly 400 different non-trivial predicate types of arity four we show that 79 are approximation resistant and 275 are not approximation resistant. For every number of accepting inputs, the number of predicate types that have been characterized as non-trivially approximable and approximation resistant, respectively, are shown in Table 1. The methods that are used to derive the results in the table are described in Appendix J.

Table 1: Approximability of MAX CSP(P) for predicates of arity four.

# accepting inputs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	all
# non-trivially approx	1	4	6	19	27	50	50	52	27	26	9	3	1	0	0	275
# approx resistant	0	0	0	0	0	0	0	16	6	22	11	15	4	4	1	79
# unknown	0	0	0	0	0	0	6	6	23	2	7	1	1	0	0	46

6 Approximation resistance is non-monotone

There is one predicate of arity four that we know to be approximation resistant but is not implied by parity,

$$\mathrm{GLST}(x_1, x_2, x_3, x_4) = (\overline{x}_1 \wedge (x_2 \equiv x_3)) \vee (x_1 \wedge (x_2 \equiv x_4)) \ . \tag{3}$$

It was shown to be approximation resistant in [10]. Zwick [20] showed that the not all equal predicate of arity three, $\mathrm{3NAE}(x_1, x_2, x_3) = (x_1 \oplus x_2) \vee (x_1 \oplus x_3)$, is not approximation resistant. For all approximation resistant predicates P of arity three, it is true that predicates implied by P is approximation resistant as well. This is true for all Samorodnisky-Trevisan predicates as well. A natural guess could be that it is true in general. However, the observation that $\mathrm{GLST}(x_1, x_2, x_3, x_4)$ implies $\mathrm{3NAE}(\overline{x}_2, x_3, x_4)$ gives the following theorem which falsify that guess.

Theorem 13. *It is not generally true that if P is approximation resistant and P' is implied by P, then P' is approximation resistant as well.*

7 Characterization of hard instances

A hard MAX CSP instance is an instance that is satisfiable or almost satisfiable and for which it is NP-hard to approximate it significantly better than picking a random assignment. In this section we characterize optimal and near-optimal solutions to such instances by bounding the weight of particular inputs for such solutions. Our main tool for this task is Lemma 5.

Zwick characterized all predicates of arity three [20]. From this work we know that approximation resistant predicates of arity three are of the same type as either 3XOR which accepts if and only if the parity of the input bits is odd, 3NTW which accepts if and only if not exactly two of the input bits are true, $\mathrm{3OXR}(x_1, x_2, x_3) = x_1 \vee (x_2 \oplus x_3)$, or $\mathrm{3OR}(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$.

Theorem 14. *Let P be* 3NTW, 3OXR *or* 3OR, *and let I be an instance of* MAX CSP(P) *of total weight W. For every $\gamma > 0$, there exist $\varepsilon > 0$ and $\kappa > 0$ such that if there exists a $(1-\varepsilon)$-satisfying assignment with at least weight γW on even parity inputs, then I can be $(s2^{-3} + \kappa)$-approximated in probabilistical polynomial time, where s is the number of accepting inputs to P.*

The theorem shows that it is no coincidence that the PCP verifier of Håstad [11], that asks three questions, almost always accepts a correct proof due to parity. In fact, assume that a PCP verifier can express any language in NP by asking for three bits and having almost completeness one and soundness only negligibly higher than the acceptance probability of a random proof. Then Theorem

14 implies that a correct proof will almost always be accepted due to the parity of the three bits asked for.

By considering $3\mathrm{OR}^{(1)}(y)$ and $3\mathrm{OR}^{(2)}(y)$ for all 3OR's accepting inputs y we can show that an optimal solution to a hard instance of MAX CSP(3OR) must have roughly $1/4$ of the weight on input $(-1, -1, -1)$.

Theorem 15. *Let I be an instance of* MAX CSP(3OR) *with total weight W. For every $\gamma > 0$, there exist $\varepsilon > 0$ and $\kappa > 0$ such that if there exists a $(1 - \varepsilon)$-satisfying assignment with weight $u_{(-1,-1,-1)}$ on input $(-1, -1, -1)$, and either $u_{(-1,-1,-1)} \leq (1/4 - \gamma)W$ or $u_{(-1,-1,-1)} \geq (1/4 + \gamma)W$, then I can be $(7/8 + \kappa)$-approximated in probabilistical polynomial time.*

8 Final remarks

In order to reduce the number of not classified predicates of arity four, we believe that using Karloff and Zwick's method of obtaining a semidefinite relaxation [14] could be very rewarding. It should also be possible to combine this method with the method in this paper that produce conditions that optimal solutions must adhere to. However, there are problems generalizing the Karloff-Zwick method in order to apply on predicates with more than four inputs.

In this paper we have considered the MAX CSP(P) problem. If for each constraint we are allowed to choose a predicate from a set of predicates, instead of using the same predicate on every constraint, then we are able to express a wider range of MAX CSPs. We note that our methods also are applicable on such problems. For example, it is possible to show that if all constraints act over three variables, then for the instance to be hard, only approximation resistant predicates may be applied except on a small (non-constant) fraction of the constraints.

Acknowledgment I am very grateful to Johan Håstad for much appreciated help.

References

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version appeared in FOCS 1992.

[2] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability - towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[3] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending Grothendieck's inequality. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 54–60, 2004.

[4] Nadia Creignou. A dichotomy theorem for maximum generalized satisability problems. *Journal of Computer and System Sciences*, 51(3):511–522, 1995.

[5] Lars Engebretsen and Jonas Holmerin. More efficient queries in PCPs for NP and improved approximation hardness of maximum CSP. In *Proceedings of STACS 2005, Lecture Notes in Computer Science 3404*, 2004.

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

[7] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996. Preliminary version appeared in FOCS 1991.

[8] Uriel Feige and Michael Langberg. The RPR^2 rounding technique for semidefinite programs. In *Proceedings of ICALP*, pages 213–224, 2001.

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

[10] Venkatesan Guruswami, Daniel Lewin, Madhu Sudan, and Luca Trevisan. A tight characterization of NP with 3 query PCPs. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 8–17, 1998.

[11] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. Preliminary version appeared in STOC 1997.

[12] Johan Håstad. Every 2-CSP allows nontrivial approximation. In *To appear in STOC 2005*, 2005.

[13] Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and PCP. *Random Structures and Algorithms*, 22(2):139–160, 2003.

[14] Howard Karloff and Uri Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 406–415, 1997.

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

[16] Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *Proceedings of 9th IPCO, Lecture Notes in Computer Science 2337*, pages 67–82, 2002.

[17] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994. Preliminary version appeared in STOC 1993.

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

[19] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 191–199, 2000.

[20] Uri Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210, 1998.

[21] Uri Zwick. Finding almost satisfying assignments. In *Proceedings of the 30th annual ACM symposium on Theory of computing*, pages 551–560, 1998.

[22] Uri Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to max cut and other problems. In *Proceedings of the 31st annual ACM symposium on Theory of computing*, pages 679–687, 1999.

A Proof of Theorem 2

Theorem 2 *Let* $I(x_1,\dots x_n)=\sum_{S\subseteq[n],|S|\le k} c_S\prod_{i\in S}x_i$ *be the objective value function for an instance of* MAX CSP(P), $P:\{\pm1\}^k\to\{0,1\}$. *Denote the sum of all constraints' weights* W. *If* $\sum_i |c_{\{i\}}|\ge\delta W$ *for a constant* $\delta>0$, *then Algorithm* **Lin**, *with* $\varepsilon=\delta/2k$, *produces a solution of expected weight at least* $c_\emptyset+\frac{\delta^2}{4k}W$.

Proof. The expected objective value is $\mathrm{E}[I]=\mathrm{E}\left[I^{(0)}+I^{(1)}+I^{(\ge2)}\right]$. In order to get a lower bound on $\mathrm{E}[I]$, we first have

$$\mathrm{E}\left[I^{(1)}\right]=\varepsilon\sum_{i=1}^{n}|c_{\{i\}}|\ge\varepsilon\delta W \quad .$$

The following fact will help in establishing a lower bound for $\mathrm{E}\left[I^{(\ge2)}\right]$.

Lemma 16. *Let* c_S *be as in Theorem 2 and* $j\ge1$, *then*

$$\sum_{S\subseteq[n],|S|=j}|c_S|\le\frac{W}{2}\binom{k}{j}^{1/2} \quad .$$

Proof. First we observe that I is a sum of weighted predicate indicator functions, $\sum_i w_iP(z_{i1},\dots z_{ik})$. Let c_S^l be the Fourier coefficient of the l:th constraint $P(z_{l1},\dots z_{lk})$,

$$P(z_{l1},\dots z_{lk})=\sum_{S\subseteq[n],|S|\le k}c_S^l\prod_{i\in S}x_i \quad .$$

We see that $c_S=\sum_l w_lc_S^l$. Note that almost all c_S^l are equal to zero. Only if x_i appears as a part of the input for all $i\in S$, then c_S^l has a non-zero value. Thus, if $|S|=j$ there are only $\binom{k}{j}$ non-zero c_S^l. The sum of all Fourier coefficients of degree j terms is now bounded using Cauchy-Schwartz inequality,

$$\begin{aligned}\sum_{S\subset[n],|S|=j}|c_S^l| &\le \left(\sum_{S:c_S^l\neq0,|S|=j}\left(c_S^l\right)^2\right)^{1/2}\left(\sum_{S:c_S^l\neq0,|S|=j}1\right)^{1/2}\\ &= \left(2^{-k}\sum_{x\in\{\pm1\}^k}|P(x)-\frac{1}{2}|^2\right)^{1/2}\binom{k}{j}^{1/2}\\ &= \frac{1}{2}\binom{k}{j}^{1/2} ,\end{aligned}$$

where we used Parseval's identity and exploited the fact that P and $P-1/2$ have the same Fourier coefficients except for $c_\emptyset$.

By adding each constraint's Fourier coefficients we get

$$\sum_{S\subseteq[n],|S|=j}|c_S|\le\sum_{l,S\subseteq[n],|S|=j}w_l|c_S^l|\le\sum_l w_l\frac{1}{2}\binom{k}{j}^{1/2}=\frac{W}{2}\binom{k}{j}^{1/2} \quad .$$

□

The expected value of a term of degree t is given by

$$\mathrm{E}\left[\prod_{j=1}^{t} x_{i_j}\right] = \varepsilon^t \prod_{j=1}^{t} \operatorname{sgn}(c_{i_j}) \ .$$

To see this note that if any of the variables $x_{i_1}, \dots x_{i_t}$ is set according to a coin flip, then the expected value of the product is zero. The probability that none of the t variables is set to a random value is ε^t and in that case it is set to $\prod_{j=1}^{t} \operatorname{sgn}(c_{i_j})$. We can now give a lower bound of terms of degree ≥ 2.

$$\mathrm{E}\left[I^{(\geq 2)}\right] \geq \sum_{j=2}^{k} -\varepsilon^j \frac{W}{2} \binom{k}{j}^{1/2} \geq -\frac{W}{2} \sum_{j=2}^{k} \varepsilon^j k^{j/2} \geq -W\varepsilon^2 k \ , \tag{4}$$

where we in the last inequality used the fact that $\varepsilon\sqrt{k} \leq 1/2$. As $I^{(0)} = c_\emptyset$, setting $\varepsilon = \delta/2k$ concludes the proof by applying the lower bound in (4),

$$\mathrm{E}\left[I\right] \geq c_\emptyset + \varepsilon\delta W - \varepsilon^2 k W = c_\emptyset + \frac{\delta^2}{4k} W \ .$$

□

B Proof of Theorem 3

Theorem 3. *Let $I(x_1, \dots x_n) = \sum_{S \subseteq [n], |S| \leq k} c_S \prod_{i \in S} x_i$ be the objective value function for an instance of* Max CSP(P), *$P : \{\pm 1\}^k \to \{0, 1\}$. Denote the sum of all constraints' weights W. If there exists an assignment such that $\sum_{1 \leq i < j \leq n} c_{\{i,j\}} x_i x_j \geq \delta W$ for a constant $\delta > 0$, then Algorithm* **BiLin** *produces a solution of expected weight at least $c_\emptyset + \kappa W$, where $\kappa > 0$ and only depends on constants δ and k.*

Proof. Maximizing $I^{(2)}$ is an instance of a quadratic program. As a first step of Algorithm **BiLin** we run the Charikar-Wirth approximation algorithm for quadratic programs [3] trying to maximize $I^{(2)}$. The algorithm returns a solution $a_1, \dots a_n$. Let

$$I^{(2)}(a_1, \dots a_n) \quad = \quad \alpha W \ .$$

By Lemma 16, with $j = 2$, we have the following upper bound on $w_{\text{bi-lin}}$, the total weight in the quadratic program instance $I^{(2)}$:

$$w_{\text{bi-lin}} \ \leq \ \frac{W}{2}\sqrt{\frac{k(k-1)}{2}} \ < \ \frac{Wk}{2} \ .$$

The optimal gain on $I^{(2)}$ is thus at least $2\delta/k$. From [3, Lemma 5] we have a lower bound for the expected value of α over the random choices of the Charikar-Wirth algorithm:

$$\mathrm{E}\left[\alpha\right] \quad \geq \quad c_{\text{cw}}\delta / \log(2\delta/k)^{-1} \ ,$$

where c_{cw} is a positive constant. We note that the lemma we refer to is for Max CUT, but it also works for instances like $I^{(2)}$ which can be considered as generalized Max CUT instances where variables also can be negated.

If the linear terms give a negative contribution, $I^{(1)}(a_1, \dots a_n) < 0$, then the solution $a_1, \dots a_n$ is negated which does not effect $I^{(2)}$ but negates $I^{(1)}$, thus ensuring that the expected value of $I^{(1)}$ is non-negative.

The expected value of the sum of the bi-linear terms can be calculated by considering each term separately. Both variables in a term are set according to the solution $a_1, \dots a_n$ with probability ε^2. This give a contribution of $\varepsilon^2 \alpha W$. If any of the variables instead are set according to a coin flip, then the expected value is zero. Thus, we have $\mathrm{E}\left[I^{(2)}\right] = \varepsilon^2 \alpha W$. The high degree terms can be lower bounded by using a similar argument as in the proof of Theorem 2:

$$
\begin{aligned}
\mathrm{E}\left[I^{(\geq 3)}\right] &\geq -\sum_{S \subset [n], 3 \leq |S| \leq k} \varepsilon^{|S|} |c_S| \\
&\geq -\sum_{j=3}^{k} \varepsilon^j \frac{W}{2} \binom{k}{j}^{1/2} \\
&\geq -\frac{W}{2} \sum_{j=3}^{k} \varepsilon^j k^{j/2} \\
&> -W \varepsilon^3 k^{3/2} \quad ,
\end{aligned}
$$

where the last inequality holds if ε is set to a value such that $\varepsilon \sqrt{k} \leq 1/2$.

The expected objective value can now be bounded from below,

$$
\begin{aligned}
\mathrm{E}[I] &= c_\emptyset + \mathrm{E}\left[I^{(1)} + I^{(2)} + I^{(\geq 3)}\right] \\
&\geq c_\emptyset + \left(\varepsilon^2 \alpha - \varepsilon^3 k^{3/2}\right) W \quad . \qquad (5)
\end{aligned}
$$

If $\alpha > 0$ we set $\varepsilon = \alpha / 2k^{3/2}$, otherwise we set $\varepsilon = 0$. We need to ensure that $\varepsilon \sqrt{k} \leq 1/2$:

$$
\begin{aligned}
\varepsilon \sqrt{k} &= \frac{\alpha}{2k} \\
&\leq \frac{1}{8} \quad ,
\end{aligned}
$$

where we used that $\alpha \leq k/4$ according to Lemma 16 with $j = 2$. The lower bound (5) ensures that $\mathrm{E}[I]$ is at least $c_\emptyset + \left(\frac{\alpha}{2k}\right)^3 W$. Choosing $\kappa = \left(\frac{\alpha}{2k}\right)^3$, the theorem follows as the expected value of α is at least $c_{\mathrm{cw}} \delta / \log(2\delta/k)^{-1}$ and $\left(\frac{\alpha}{2k}\right)^3$ is a convex function for non-negative α. □

C Proof of Lemma 4

Lemma 4. *Let $P : \{\pm 1\}^k \to \{0, 1\}$ be a predicate. If $P^{(1)}(y) + P^{(2)}(y) > 0$ for all $y \in P^{-1}(1)$, then P is not approximation resistant.*

Proof. Let $\delta = \min_{y \in P^{-1}(1)} P^{(1)}(y) + P^{(2)}(y)$, where $\delta > 0$ according to the assumption in the lemma. Furthermore, define $\gamma = \min_{y \in \{\pm 1\}^k} P^{(1)}(y) + P^{(2)}(y)$. We know that $\gamma < 0$ as $\sum_{y \in \{\pm 1\}^k} P^{(1)}(y) + P^{(2)}(y) = 0$. Set $\varepsilon = \frac{\delta}{2(\delta - \gamma)}$. If there is no assignment that satisfies weight at least $(1 - \varepsilon)W$, then a random assignment achieves an approximation ratio larger than $2^{-k}|P^{-1}(1)|/(1 - \varepsilon) > (1 + \varepsilon)2^{-k}|P^{-1}(1)|$. Thus, the hard case is if there is an assignment that satisfies weight at least $(1 - \varepsilon)W$. Assume therefore that this is the case and calculate a lower bound on the linear and bi-linear terms of the objective value function,

$$\begin{aligned}
I^{(1)}(x_1,\dots)+I^{(2)}(x_1,\dots) &= \sum_{i=1}^{m} w_i \left(P^{(1)}(z_{i1},\dots z_{ik})+P^{(2)}(z_{i1},\dots z_{ik})\right)\\
&\geq \left(\left(1-\frac{\delta}{2(\delta-\gamma)}\right)\delta+\frac{\delta}{2(\delta-\gamma)}\gamma\right)W\\
&= \left(\delta+\frac{\gamma-\delta}{2(\delta-\gamma)}\delta\right)W\\
&= \frac{\delta}{2}W \ .
\end{aligned}$$

Thus, we know that either $I^{(1)} \geq \frac{\delta}{4}W$ or $I^{(2)} \geq \frac{\delta}{4}W$. In the first case Theorem 2 implies that P is not approximation resistant and in the latter case Theorem 3 implies that P is not approximation resistant. □

D Proof of Lemma 5

Lemma 5. *Let $P : \{\pm1\}^k \to \{0,1\}$ be a predicate and $I(x_1,\dots x_n)=\sum_{i=1}^m w_i P(z_{i1},\dots z_{ik})$ be the objective value of a* Max CSP(P) *instance. For an assignment to $x_1,\dots x_n$, define the weight on each input y, $u_y=\sum_{i:(z_{i1},\dots z_{ik})=y} w_i$. For constants C and $\delta>0$, there exist constants $\varepsilon>0$ and $\kappa>0$ such that if $\sum_{y\in P^{-1}(1)} u_y(C\cdot P^{(1)}(y)+P^{(2)}(y)) \geq \delta W$, where $W=\sum_{i=1}^m w_i$, then either the value of the assignment is less than $(1-\varepsilon)W$, or Algorithm* **Lin** *or Algorithm* **BiLin** *will achieve an approximation ratio of at least $2^{-k}|P^{-1}(1)|+\kappa$.*

Proof. Using the assumptions in the lemma we conclude that

$$\begin{aligned}
C\cdot I^{(1)}(x_1,\dots)+I^{(2)}(x_1,\dots) &= \sum_{y\in\{\pm1\}^k} u_y(C\cdot P^{(1)}(y)+P^{(2)}(y))\\
&\geq \delta W+\sum_{y\in P^{-1}(0)} u_y(C\cdot P^{(1)}(y)+P^{(2)}(y)) \ .
\end{aligned}$$

Negating an assignment negates the value of $I^{(1)}$ but does not effect $I^{(2)}$. Thus, if $C<0$, then the assignment can be negated and we set $C:=-C$, ensuring that C is non-negative. The terms of the last expression are now bounded from below using Lemma 19 (defined below in the proof of Theorem 7),

$$\begin{aligned}
C\cdot P^{(1)}(y)+P^{(2)}(y) &= C\sum_{\alpha\in P^{-1}(0)} \psi_\alpha^{(1)}(y)+\sum_{\alpha\in P^{-1}(0)} \psi_\alpha^{(2)}(y)\\
&\geq -Ck-k/2\\
&> -(C+1)k \ .
\end{aligned}$$

By setting $\varepsilon=\frac{\delta}{2(C+1)k}$, we ensure that the weight on non-accepting inputs is at most $\frac{\delta}{2(C+1)k}W$, implying that

$$C\cdot I^{(1)}(x_1,\dots)+I^{(2)}(x_1,\dots)\geq \delta W-\sum_{y\in P^{-1}(0)} u_y(C+1)k \geq \frac{\delta}{2}W \ .$$

Thus, either $I^{(1)}(x_1,\dots)\geq \frac{\delta}{4C}W$ or $I^{(2)}(x_1,\dots)\geq \frac{\delta}{4}W$. Applying Theorems 2 and 3 concludes the proof. □

E Proof of Theorem 6

Theorem 6 *Let $P : \{\pm 1\}^k \to \{0,1\}$ be a predicate. If for some constant C, $C \cdot P^{(1)}(y) + P^{(2)}(y)$ is zero for at most two $y \in P^{-1}(1)$ and strictly positive for all other $y \in P^{-1}(1)$, then P is not approximation resistant.*

Before we prove Theorem 6 we prove the following lemma which relies on Zwick's algorithm for almost satisfiable 2SAT instances [21].

Lemma 17. *Let P be a predicate with exactly two accepting inputs. If there exists an $(1-\varepsilon)$-satisfying solution to a* Max CSP(P) *instance, then a $(1 - O(\varepsilon^{1/3}))$-satisfying solution can be produced in probabilistical polynomial time.*

Proof. Assume that $a = (a_1, \dots a_k)$, $b = (b_1, \dots b_k) \in \{\pm 1\}^k$ are the two accepting inputs to P. We show that there exists a set of 2SAT clauses such that only a and b satisfy all of them. The set is simply the maximal set of clauses of size two that satisfies both a and b. Let us call this set D.

Claim 18. *Only a and b satisfies all the* 2SAT *clauses in D.*

Proof of Claim 18. Let $c \in \{\pm 1\}^k$ such that $c \neq a$, $c \neq b$. We show that there exists a clause in D that c does not satisfy. As $a \neq b$ there exists an $i \in [k]$ such that $a_i \neq b_i$. Without loss of generality we assume that $c_i = a_i$. As $a \neq c$ we also know that $a_j \neq c_j$ for some $j \in [k]$. The 2SAT clause over x_i and x_j that rejects c accepts both a and b because $a_j \neq c_j$ and $b_i \neq c_i$. This clause is contained in D. □

Each constraint in the original instance is transformed into the above set of 2SAT clauses. If an assignment satisfies an original constraint then it also satisfies all corresponding 2SAT clauses. However, if it does not satisfy an original constraint then at least one of the 2SAT clauses is not satisfied by that assignment. According to the assumption in the lemma we have that an optimal solution to the Max CSP(P) instance is at least $1 - \varepsilon$-satisfying. This implies that the same assignment also satisfies at least a $1 - \varepsilon$ fraction of the total weight in the resulting Max 2SAT instance. By using Zwick's algorithm for almost satisfiable 2SAT instances [21], we then find a $1 - O(\varepsilon^{1/3})$ satisfying solution. Each P-constraint is transformed into a constant number of 2SAT clauses, thus the solution also $1 - O(\varepsilon^{1/3})$ satisfies the original Max CSP(P) instance. This concludes the proof of Lemma 17. □

Proof of Theorem 6. Let I be an instance of Max CSP(P) with total weight W and let y_1 and y_2 be the accepting inputs to P that make $C \cdot P^{(1)}(y) + P^{(2)}(y) = 0$. If there is only one such input, let y_1 be that input and y_2 be an arbitrary accepting input that is not equal to y_1. If no accepting inputs to P make $C \cdot P^{(1)}(y) + P^{(2)}(y) = 0$, then let y_1 and y_2 be distinct arbitrary accepting inputs to P. For an optimal assignment to I, let u_y be the weight of an input y. For any constant value of $\varepsilon > 0$, we define three non-exclusive cases:

- The value of an optimal solution is less than $(1 - \varepsilon)W$.
- The weight on y_1 and y_2 is large, namely $u_{y_1} + u_{y_2} \geq (1 - 2\varepsilon)W$.
- For $\delta = \varepsilon \min_{y \in P^{-1}(1) \setminus \{y_1, y_2\}} \left\{ C \cdot P^{(1)}(y) + P^{(2)}(y) \right\}$,

$$\sum_{y \in P^{-1}(1)} u_y (C \cdot P^{(1)}(y) + P^{(2)}(y)) \geq \delta W \ .$$

We first show that every instance is covered by at least one of the above cases. Then we show that in each case we can achieve an approximation ratio of at least $2^{-k}|P^{-1}(1)| + \kappa$, for a constant $\kappa > 0$ as long as ε is small enough. This shows that P is not approximation resistant.

We assume that none of the first two cases apply on an instance and show that this implies that the last case is applicable. The first case implies that an optimal solution has value $w_{\text{opt}} = \sum_{y \in P^{-1}(1)} u_y \geq (1-\varepsilon)W$. The second case implies that $u_{y_1} + u_{y_2} < (1-2\varepsilon)W$. Thus we have

$$\sum_{y \in P^{-1}(1) \setminus \{y_1, y_2\}} u_y > \varepsilon W \ .$$

We calculate the sum specified in the third case.

$$\begin{aligned} \sum_{y \in P^{-1}(1)} u_y(C \cdot P^{(1)}(y) + P^{(2)}(y)) &= \sum_{y \in P^{-1}(1) \setminus \{y_1, y_2\}} u_y(C \cdot P^{(1)}(y) + P^{(2)}(y)) \\ &\geq \sum_{y \in P^{-1}(1) \setminus \{y_1, y_2\}} u_y \delta / \varepsilon \\ &> \delta W \end{aligned}$$

This implies that the third case is applicable.

If an instance is in the first case, a random assignment approximates the instance well enough with an approximation ratio of

$$\frac{2^{-k}|P^{-1}(1)|W}{(1-\varepsilon)W} > 2^{-k}(1+\varepsilon)|P^{-1}(1)| \ .$$

If it is in the second case we apply Lemma 17 with the predicate that only accepts y_1 and y_2. If it is in the third case we instead apply Lemma 5.

□

F Proof of Theorem 7

Theorem 7. *Let $P : \{\pm 1\}^k \to \{0, 1\}$, $k \geq 3$. If P has at most $2\lfloor k/2 \rfloor + 1$ accepting inputs, then P is not approximation resistant.*

In order to prove Theorem 7, we need two lemmas. Lemma 4 shows that if $P^{(1)}(y) + P^{(2)}(y)$ is positive for all accepting inputs y, then P is not approximation resistant. Lemma 19 is used for obtaining lower bounds on $P^{(1)}(y) + P^{(2)}(y)$.

Lemma 19. *Let $x, y \in \{\pm 1\}^k$ and $d(x, y)$ be the Hamming distance between x and y. Define $t = d(x, y) - k/2$. Then the following holds:*

1. $\psi_x^{(1)}(y) = -2t \cdot 2^{-k}$ *and*
2. $\psi_x^{(2)}(y) = (2t^2 - k/2)2^{-k}$.

Proof. Let $z \in \{\pm 1\}^k$ such that $z_i = x_i y_i$. The number of -1 in z equals the Hamming distance $d(x, y)$. Thus,

$$2^k \psi_x^{(1)}(y) = \sum_{i=1}^{k} x_i y_i = \sum_{i=1}^{k} z_i = 2^{-k}(k - 2d(x, y)) = -2t \ ,$$

establishing the first claim. The second claim is verified by simple calculation:

$$\begin{aligned}
2^k \psi_x^{(2)}(y) &= \sum_{1 \le i < j \le k} z_i z_j \\
&= \sum_{\substack{i\,:\,z_i = 1 \\ j\,:\,z_j = 1 \\ i<j}} z_i z_j + \sum_{\substack{i\,:\,z_i = -1 \\ j\,:\,z_j = -1 \\ i<j}} z_i z_j + \sum_{\substack{i\,:\,z_i = 1 \\ j\,:\,z_j = -1}} z_i z_j \\
&= \left(\frac{k}{2} - t\right)\left(\frac{k}{2} - t - 1\right)/2 + \left(\frac{k}{2} + t\right)\left(\frac{k}{2} + t - 1\right)/2 + \left(\frac{k}{2} + t\right)\left(\frac{k}{2} - t\right) \\
&= 2t^2 - \frac{k}{2} \ .
\end{aligned}$$

□

Proof. In order to prove Theorem 7 we use (2) to express the linear and bi-linear terms of a constraint $P(y)$,

$$P^{(1)}(y) + P^{(2)}(y) = \sum_{\alpha \in P^{-1}(1)} \psi_\alpha^{(1)}(y) + \psi_\alpha^{(2)}(y) \ .$$

Lemma 19 give the following lower bound for each term in the above expression,

$$\begin{aligned}
\psi_\alpha^{(1)}(y) + \psi_\alpha^{(2)}(y) &= (-2t + 2t^2 - k/2)2^{-k} \\
&\ge \begin{cases} -\frac{k}{2}2^{-k} & \text{if } k \text{ even} \\ -\frac{k+1}{2}2^{-k} & \text{if } k \text{ odd} \end{cases} \ ,
\end{aligned} \tag{6}$$

where $t = d(x, y) - k/2$, as defined in Lemma 19. In the above calculation we used the fact that $-2t + 2t^2 \ge 0$ if k is even (equality when $t = 0, 1$) and $-2t + 2t^2 \ge -1/2$ if k is even (equality when $t = 1/2$). If $\alpha = y$, it is easy to see that $\psi_\alpha^{(1)}(y) + \psi_\alpha^{(2)}(y) = \frac{(k+1)k}{2}2^{-k}$. Thus, if P accepts an input y, $y \in P^{-1}(1)$, the following lower bound is derived by applying (6)

$$\begin{aligned}
P^{(1)}(y) + P^{(2)}(y) &= \sum_{\alpha \in P^{-1}(1)} \psi_\alpha^{(1)}(y) + \psi_\alpha^{(2)}(y) \\
&\ge \left(\frac{(k+1)k}{2} - (s-1)\left\lceil \frac{k}{2} \right\rceil\right) 2^{-k} \\
&\ge \frac{k}{2}2^{-k} \begin{cases} \text{if } s \le k+1 \text{ and } k \text{ even} \\ \text{if } s \le k \text{ and } k \text{ odd} \end{cases} \ ,
\end{aligned}$$

where s is the number of accepting inputs to P. Applying Lemma 4 concludes the proof of Theorem 7. □

G Proof of Theorem 9

G.1 Preliminaries

This proof is in large part a reiteration of the analysis of the Samorodnitsky-Trevisan PCP made by Håstad and Wigderson [13]. We therefore assume that the reader is familiar with this work.

G.2 Probabilistically checkable proofs

A language L belongs to $PCP_{c,s}[r, q]$ if there exists a verifier V that only looks at q bits in the proof, π, uses at most r random bits and

1. for every $x \in L$, there exists a π such that $\Pr[V(\pi, x) = accept] \geq c$,
2. for every $x \notin L$ and every π, $\Pr[V(\pi, x) = accept] < s$.

G.3 The Samorodnitsky-Trevisan PCP

Samorodnitsky and Trevisan showed that for every constant $\epsilon > 0$ and s, $t \in Z^+$,

$$\text{NP} = PCP_{1-\epsilon, 2^{-st}+\epsilon}[O(\log n), s + t + st] \ .$$

The proof π of the PCP should contain an encoding of the answer for each possible u parallel question that the verifier can ask in the so-called two prover proof system (see [13]).

The verifier in this PCP inspects $s + t + st$ bits in the proof and decides whether or not to accept by applying the Samorodnitsky-Trevisan predicate $P_{ST^{s,t}}$ on the (possibly negated) bits in the proof. Thus, the problem of finding a proof that maximizes the accept probability of the verifier for a certain x can be expressed as a Max CSP($P_{ST^{s,t}}$) instance. The verifier is only allowed to use a logarithmic number of random bits. There are therefore only a polynomial number of possible tuples of $s + t + st$ bits that the verifier may inspect. The size of the resulting Max CSP($P_{ST^{s,t}}$) instance is thus also polynomial.

Let L be a NP-complete language. Then it is NP-hard to decide whether $x \in L$ or $x \notin L$. Thus it is NP-hard to approximate Max $P_{ST^{s,t}}$CSP within $s/c = (2^{-st} + \epsilon)/(1 - \epsilon) = 2^{-st} + \epsilon'$ for any constant $\epsilon' > 0$. Otherwise we could decide if $x \in L$ by approximating an instance of Max CSP($P_{ST^{s,t}}$).

G.4 The new verifier

The only change we do to the Samorodnitsky-Trevisan PCP is to make the verifier more generous when deciding to accept or reject. Instead of $P_{ST^{s,t}}$ we let the verifier use a P such that $P^{-1}_{ST^{s,t}}(1) \subseteq P^{-1}(1)$. It is obvious that the completeness will at least stay at the same level as in the Samorodnitsky-Trevisan PCP. The soundness needs to be analyzed more carefully.

Before analyzing the soundness of the new verifier we state the actions of both the original Samorodnitsky-Trevisan verifier and the new more generous verifier. (The following description of the actions of the Samorodnitsky-Trevisan verifier is taken from [13].)

Test ST

1. The verifier chooses a set U of u variables and s random boolean functions f_i, $i = 1, 2, \ldots s$ on U. Let A be the portion of the proof corresponding to U.
2. For $j = 1, 2, \ldots t$ the verifier repeats the following steps. For each variable in U choose a random clause containing it. Let h_j be the conjunction of the chosen clauses and let W_j be the set of variables appearing in the chosen clauses. Choose g_j to be a random boolean function on W_j. Let B_j be the portion of the proof corresponding to W_j, but folded over true and conditioned upon h_j.
3. For $i = 1, 2, \ldots s$, $j = 1, 2, \ldots t$ choose a function μ_{ij} on W_j which, independently at each point takes the value 1 with probability $1-\epsilon$ and -1 with probability ϵ. Set $g_{ij} = g_i f_i \mu_{ij}$. Test whether

4. The verifier accepts if

$$B_j(g_{ij})B_j(g_j)A(f_i) = 1$$

for $i = 1, 2, \ldots s$, $j = 1, 2, \ldots t$, otherwise it rejects.

The original verifier performs st parity checks, each containing three bits of the proof, and it accepts the proof if and only if all parity checks are passed. We consider verifiers that always accept if all parity checks are passed, but may also accept proofs where one or more parity checks fail. We get the following PCP test indexed with a predicate, P, acting on the query bits of the proof.

Test ST^P Steps 1 to 3 are performed as in Test ST.

4 The verifier accepts if

$$P\left(A(f_1), \ldots A(f_s), B_1(g_1), \ldots B_t(g_t), B_1(g_{11}), \ldots B_t(g_{st})\right),$$

otherwise it rejects.

G.5 Analysis of the soundness

A completely random proof will make the verifier accept with probability $2^{-(s+t+st)}|P^{-1}(1)|$ if P is the predicate that the verifier uses. The best soundness we can hope to achieve therefore has to be larger than $2^{-(s+t+st)}|P^{-1}(1)|$.

In Test ST, the verifier accepts if $A(f_i)B_j(g_j)B_j(g_{ij}) = 1$ for $1 \le i \le s$ and $1 \le j \le t$. Thus, the following expression is 1 if the verifier accepts and 0 otherwise.

$$\prod_{\substack{1 \le i \le s \\ 1 \le j \le t}} \frac{1 + A(f_i)B_j(g_j)B_j(g_{ij})}{2}$$

Calculating the above product we get the following sum,

$$2^{-st} \sum_{S \subseteq [s] \times [t]} \prod_{(i,j) \in S} A(f_i)B_j(g_j)B_j(g_{ij}).$$

It has been shown that if $\mathrm{E}\left[\prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j(g_{ij})\right] = \delta$ for a nonempty $S \subseteq [s] \times [t]$ (the expectation is taken over the randomness of the PCP verifier), then there is a strategy that convinces the verifier in the two prover game with probability at least $4\epsilon\delta^2$ [13, Lemma 4.6]. What is important to note is that we can make the soundness of the two prover game arbitrarily small and independent of the value of ϵ. Thus, we can choose an arbitrarily low constant threshold such that if $\mathrm{E}\left[\prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j(g_{ij})\right]$ is larger than the threshold we know that $x \in L$. When analyzing the soundness we can assume that $x \notin L$ and therefore we can bound $\mathrm{E}\left[\prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j(g_{ij})\right]$ from above with an arbitrarily low constant.

The new accept indicator function is

$$\sum_{\substack{(a_1, \ldots a_{s+t+st}) \\ \in P^{-1}(1)}} \left(\prod_{1 \le i \le s} \frac{1 + a_i A(f_i)}{2}\right) \left(\prod_{1 \le j \le t} \frac{1 + a_{s+j} B_j(g_j)}{2}\right) \left(\prod_{\substack{1 \le i \le s \\ 1 \le j \le t}} \frac{1 + a_{(1+j)s+t+i} B_j(g_{ij})}{2}\right). \tag{7}$$

We analyze the above multilinear expression by showing that the terms which neither are on the form $\prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j(g_{ij})$ or are constants, will have a zero expectation. Ignoring the sign, an arbitrary term can be written on the following form

$$2^{-(s+t+st)} \prod_{i\in I} A(f_i) \prod_{j\in J} B_j(g_j) \prod_{(i,j)\in S} B_j(g_{ij}),$$

where $I \subseteq [s]$, $J \subseteq [t]$ and $S \subseteq [s]\times[t]$. If it is not a term that shows up in the previous analysis we know that there is an i_0 (or a j_0) such that $|\{i \in I : i = i_0\}|$ has not the same parity as $|\{(i,j) \in S : i = i_0\}|$ (or that $|\{j \in J : j = j_0\}|$ has not the same parity as $|\{(i,j) \in S : j = j_0\}|$). Assume for now that there is such an i_0 and that $i_0 \in I$. Disregarding the sign the term can now be written as

$$2^{-(s+t+st)} \left(A(f_{i_0}) \prod_{(i_0,j)\in S} B_j(g_{i_0 j}) \right) C, \tag{8}$$

where the value of C is independent of the choices of f_{i_0}, $g_{i_0 1}$, $\dots$ $g_{i_0 s}$. Let us fix the choices of g_1, $\dots$ g_s. It is equally probable for a verifier to choose a function as f_{i_0} as it would have been to chose its negation. Thus the (partial) choice of the following functions $(f_{i_0}, g_1, g_2, \dots g_s, g_{i_0 1}, g_{i_0 2}, \dots g_{i_0 s})$ is as frequent as $(-f_{i_0}, g_1, g_2, \dots g_s, -g_{i_0 1}, -g_{i_0 2}, \dots - g_{i_0 s})$. Due to folding of the long-code tables we know that

$$A(f_{i_0}) \prod_{(i_0,j)\in S} B_j(g_{i_0 j}) = -A(-f_{i_0}) \prod_{(i_0,j)\in S} B_j(-g_{i_0 j}),$$

because the number of factors is odd. The expectation of expression (8) must then be zero. If $i_0 \notin I$ or there is a j_0 such that $|\{j \in J : j = j_0\}|$ has not the same parity as $|\{(i,j) \in S : j = j_0\}|$, then the same reasoning shows that the expectation of the term is zero.

The constant term in (7) is $2^{-(s+t+st)}|P^{-1}(1)|$, which also is the value of the largest possible coefficient for a term in general and for a term on the form $\prod_{(i,j)\in S} A(f_i)B_j(g_j)B_j(g_{ij})$ in particular. Thus, we can get a soundness within $2^{-(s+t+st)}|P^{-1}(1)| + \epsilon'$, for any constant $\epsilon' > 0$, by choosing a soundness in the parallel two prover game that is low enough.

G.6 Connection with Max CSP(P)

We restate the theorem that we are about to prove.

Theorem 9. *Let $P : \{\pm 1\}^{s+t+st} \to \{0,1\}$, be a predicate that is implied by $P_{ST^{s,t}}$, i.e., $P^{-1}_{ST^{s,t}}(1) \subseteq P^{-1}(1)$, then P is approximation resistant.*

Proof. Let L be a NP-complete language. Assume that P is not approximation resistant, thus we can approximate Max CSP(P), where $P^{-1}_{ST^{s,t}}(1) \subseteq P^{-1}(1)$, within $\alpha_R(P) + \epsilon$ for a constant $\varepsilon > 0$, where $\alpha_R(P)$ is the approximation ratio of a random assignment. We then show that we can decide whether $x \in L$ for an arbitrary x.

As $L \in$NP there is a PCP for $x \overset{?}{\in} L$ in which the verifier uses the Test ST^P when it decides whether to accept or not. For each possible outcome of the verifier's random bits, the verifier chooses a $s + t + st$-tuple of bits from the proof and applies P on them. This corresponds to one constraint tuple of a Max CSP(P) instance. The number of constraint tuples is polynomial because the verifier only has a logarithmically number of random bits. For an arbitrary constant $\epsilon' > 0$, we know that if $x \in L$, then the instance is $(1 - \epsilon')$-satisfiable, and if $x \notin L$, then the instance is not $(\alpha_R(P) + \epsilon')$-satisfiable. Choose $\epsilon' = \min(\epsilon/3, 1/2)$ and use the $\alpha_R(P) + \epsilon$-approximation algorithm on the Max

CSP(P) instance. The solution will at least satisfy a fraction of the constraints equal to

$$\begin{aligned}(1-\epsilon')(\alpha_R(P)+\epsilon) &= (1-\epsilon')(2^{-(s+t+st)}|P^{-1}(1)|+\epsilon) \\ &> 2^{-(s+t+st)}|P^{-1}(1)|+\epsilon'.\end{aligned}$$

If $x \in L$ we are thus guaranteed to get a solution which satisfies a fraction of the possible verifier questions that is larger than the soundness. Thus we will be able to conclude whether or not $x \in L$, which is NP-hard. This implies that P is approximation resistant. □

H Proof of Theorem 10

Theorem 10. *Let $k \geq s+t+st$ and $P : \{\pm 1\}^k \to \{0,1\}$ be a predicate with at most $2^{st}-1$ non-accepting inputs, then P is approximation resistant.*

Proof. Our goal is to design st parity relations of the right type, such that no non-accepting input fulfills all of them. Based on these parity relations and the predicate P, we construct the predicate P' which is implied by the Samorodnitsky-Trevisan predicate $P_{ST^{s,t}}$, and is of the same type as P. Using Theorem 9 we see that Max CSP(P') cannot be approximated beyond the random assignment threshold, and because P and P' are of the same type we conclude that Max CSP(P) cannot either be approximated beyond the random assignment threshold.

Let $x \in \{\pm 1\}^k$ be a possible input to P, and $par_{ij}(x)$, where $i \in [s]$ and $j \in [t]$, is the parity relation $x_i x_{s+j} x_{t+js+i}$. Initially, let $T := P^{-1}(0)$ be all non-accepting inputs of P. For each $(i,j) \in [s] \times [t]$ we repeat the following: Construct $T_{-1} = \{x \in T : par_{ij}(x) = -1\}$ and $T_1 = \{x \in T : par_{ij}(x) = 1\}$. If T_{-1} has fewer elements than T_1, set $T := T_{-1}$ and $b_{ij} := 1$, otherwise set $T := T_1$ and $b_{ij} := -1$. After each iteration, T is at most half of its previous size. The assumption of the theorem stipulates that T initially consists of at most $2^{st}-1$ elements. Thus, after st iterations T must be the empty set.

We get P' by applying a bit mask on st bits of the input to P:

$$P'(x_1, x_2, \ldots x_k) = P(x_1, \ldots x_{s+t}, b_{11}x_{s+t+1}, b_{21}x_{s+t+2}, \ldots b_{st}x_{s+t+st}, x_{s+t+st+1}, \ldots x_k) \ ,$$

for all inputs $(x_1, \ldots x_k) \in \{\pm 1\}^k$. As we have only negated some input bits on certain positions it is clear that P and P' are of the same type and thus Max CSP(P) and Max CSP(P') are equivalent problems. Furthermore, P' is implied by $P_{ST^{s,t}}$ because of the construction of the bit mask values $\{b_{ij}\}$. Applying Theorem 9 shows that P' is approximation resistant, and thereby also P. We note that if $k > s+t+st$, then $P_{ST^{s,t}}$ is considered to take inputs of length k but the last $k-(s+t+st)$ input bits will never effect the output of the predicate. □

I Proofs of Theorem 11

Theorem 11. *Let P be a predicate on 5 binary variables with at most 5 non-accepting inputs, then P is approximation resistant.*

Proof. Let $P^{-1}(0) = \{s_1, s_2, s_3, s_4, s_5\}$ where $s_j \in \{\pm 1\}^5$ for $1 \leq j \leq 5$. With $s_j[i]$ we denote the i'th bit in s_j. We start by describing a sufficient condition for P being approximation resistant.

Let a parity test $\{t,u,v\} \subset [5]$ on s be defined as $\prod_{i \in \{t,u,v\}} s[i]$. A pair of parity tests is admissible if the intersection of them contains a single element. The result of an admissible pair of parity tests is in $\{\pm 1\}^2$, thus there are four possible outcomes.

Claim 20. *If there exists an admissible pair of parity tests such that there is a possible outcome that do not occur if applied on all elements in $P^{-1}(0)$. Then P is approximation resistant.*

Proof. Let the admissible pair of parity checks be $\{t,u,v\}$ and $\{t,w,z\}$ and the outcome that not occurs (b_1,b_2). Define P' such that

$$P(x_1,x_2,x_3,x_4,x_5) = P'(x_t,-b_1x_u,-b_2x_w,x_v,x_w) \ .$$

P' is apparently of the same type as P. In addition P' is implied by $P_{ST^{1,2}}$, as

$$P_{ST^{1,2}}(x_1,x_2,x_3,x_4,x_5) = (x_1 \oplus x_2 \oplus x_4) \wedge (x_1 \oplus x_3 \oplus x_5) \ .$$

Theorem 9 shows that P' is approximation resistant and thus P is approximation resistant as well. $\square$

To prove the theorem we assume that there do not exist an admissible pair of parity checks as specified in the claim, and show that this leads to a contradiction. Therefore, assume that there exists $P^{-1}(0) = \{s_1,s_2,s_3,s_4,s_5\}$ such that, for every possible choice of admissible pair of parity checks, par_1 and par_2, for every possible outcome $(b_1,b_2) \in \{\pm1\}^2$ of par_1 and par_2, there is at least one element $s_j \in P^{-1}(0)$ such that $(par_1(s_j), par_2(s_j)) = (b_1,b_2)$.

If not every parity check on three variables will split $\{s_1,s_2,s_3,s_4,s_5\}$ into sets of size two and three, then it is easy to see that the above assumption is false. Thus, we can assume that every parity check splits $\{s_1,s_2,s_3,s_4,s_5\}$ into sets of sizes two and three. Now, w.l.o.g. we assume that

$$\prod_{i=1}^{3} s_1[i] = \prod_{i=1}^{3} s_2[i] \neq \prod_{i=1}^{3} s_3[i] = \prod_{i=1}^{3} s_4[i] = \prod_{i=1}^{3} s_5[i] \ , \tag{9}$$

because we can rearrange the values of $\{s_i\}$. Let par_1 be the parity check on variables $(1,2,3)$. Now there are three different possible parity checks for par_2, on variables $(1,4,5)$, $(2,4,5)$ or $(3,4,5)$. Each possible choice of par_2 will have to give different output when applied on s_1 and s_2.

We know that $s_1[i] = s_2[i]$ for some $1 \le i \le 3$ because of the first parity check. It can be seen that $(s_1[1],s_1[2],s_1[3]) = (s_2[1],s_2[2],s_2[3])$, because otherwise at least one of the possible parity checks will reject both s_1 and s_2. To see this, let $s_1[t] = s_2[t]$ and $s_1[u] \neq s_2[u]$ for $t,u \in \{1,2,3\}$. If $s_1[4]s_1[5] = s_2[4]s_2[5]$, then

$$\prod_{i \in \{t,4,5\}} s_1[i] = \prod_{i \in \{t,4,5\}} s_2[i]$$

and else

$$\prod_{i \in \{u,4,5\}} s_1[i] = \prod_{i \in \{u,4,5\}} s_2[i] \ .$$

Furthermore, $s_1[4] \oplus s_1[5] \neq s_2[4] \oplus s_2[5]$ because otherwise any one of the parity checks would yield the same result for s_1 and s_2. Thus, s_1 and s_2 only differ on either the fourth or the fifth element. We assume w.l.o.g. that s_1 and s_2 differ only on the last element.

We study what happens if we choose par_1 differently. Instead of $(1,2,3)$ we let par_1 check parity on variables $(3,4,5)$. This check separates s_1 from s_2. We assume w.l.o.g. that

$$\prod_{i=3}^{5} s_1[i] = \prod_{i=3}^{5} s_3[i] \neq \prod_{i=3}^{5} s_2[i] = \prod_{i=3}^{5} s_4[i] = \prod_{i=3}^{5} s_5[i] \ . \tag{10}$$

Reapplying the same arguments as above we conclude that $(s_1[3],s_1[4],s_1[5]) = (s_3[3],s_3[4],s_3[5])$ and that $s_1[1]s_1[2] \neq s_3[1]s_3[2]$. Assume w.l.o.g. that $s_1[1] \neq s_3[1]$ and $s_1[2] = s_3[2]$.

Let par_1 now act on the variables $(2,3,4)$. All these bits are identical in s_1, s_2 and s_3, thus

$$\prod_{i=2}^{4} s_4[i] = \prod_{i=2}^{4} s_5[i] \neq \prod_{i=2}^{4} s_1[i] = \prod_{i=2}^{4} s_2[i] = \prod_{i=2}^{4} s_3[i] \ .$$

Using the same argument as before, we see that $(s_4[2], s_4[3], s_4[4]) = (s_5[2], s_5[3], s_5[4])$ and $s_4[1]s_4[5] \neq s_5[1]s_5[5]$. We conclude that either $\prod_{i=1}^{3} s_4[i] \neq \prod_{i=1}^{3} s_5[i]$ or $\prod_{i=3}^{5} s_4[i] \neq \prod_{i=3}^{5} s_5[i]$. But this contradicts (9) or (10). Thus the assumption of Claim 20 is satisfied and we can conclude that P is approximation resistant which proves the theorem. $\square$

J Characterizing predicates of arity four

In this section we describe the methods that was used when creating Table 1. Almost all predicates shown to not be approximation resistant were done so by using Theorem 6. A few predicates could be shown to not be approximation resistant by using a similar approach as was done in the proof of Theorem 6.

Using results from [11] we can conclude that around thirty predicate types are approximation resistant. By also applying the following two lemmas a total of 79 predicate types are shown to be approximation resistant.

Lemma 21. *Let* $|P^{-1}(1)| = s$, $P^{-1}(1) = \{y_1, \dots y_s\}$ *and let* $h : \{\pm 1\}^4 \to \{0, 1\}$ *either be* $h(y) = (y[4] \equiv 1)$ *or* $h(y) = (y[1] \equiv y[4])$*. If* $y_i[1] \oplus y_i[2] \oplus y_i[3]$ *and* $h(y_i)$ *for all* $i \in [4]$*, and* $\sum_{y \in P^{-1}(1)} h(y) = \frac{1}{2}|P^{-1}(1)|$*, then* P *is approximation resistant.*

Proof. We show that if P is not approximation resistant, then Max CSP(3XOR) can be approximated better than a random assignment. But 3XOR is approximation resistant and thus we conclude that P is approximation resistant as well.

We first show the lemma for $h(y) = (y[4] \equiv 1)$. Given an instance of Max CSP(3XOR), we transform each constraint $z_{i_1} \oplus z_{i_2} \oplus z_{i_3}$ into four Max CSP(P) constraints

$$\begin{gathered} P(z_{i_1}, z_{1_2}, z_{i_3}, u) \\ P(\overline{z}_{i_1}, \overline{z}_{1_2}, z_{i_3}, u) \\ P(\overline{z}_{i_1}, z_{1_2}, \overline{z}_{i_3}, u) \\ P(z_{i_1}, \overline{z}_{1_2}, \overline{z}_{i_3}, u), \end{gathered}$$

where u is an auxiliary variable that is the same for each constraint that we transform.

Let us assume that $u = 1$ for an assignment a to the Max CSP(P) instance. There are $s/2 - 4$ accepting inputs y, such that $h(y)$ and not $y[1] \oplus y[2] \oplus y[3]$. If a makes $z_{i_1} \oplus z_{1_2} \oplus z_{i_3}$ true, then all four corresponding constraints are true, otherwise exactly $s/2 - 4$ of the constraints are true. Thus, if a $(1 - \epsilon)$-satisfies Max CSP(3XOR), then it satisfies a

$$\frac{4(1-\epsilon) + (s/2 - 4)\epsilon}{4} = \frac{4 - (8 - s/2)\epsilon}{4} = 1 - (2 - s/8)\varepsilon$$

fraction of the total weight of the Max CSP(P) instance. Setting $\epsilon = 1/2 - \delta$, we see that $1 - (2 - s/8)\varepsilon = s/16 + (2 - s/8)\delta$. Thus, if we can beat a random assignment on a Max CSP(P) instance, then we can also beat a random assignment on a Max CSP(3XOR) instance.

To handle the case $u = -1$ we make the following observation: For an assignment with $u = -1$, either setting $u = 1$ or negating the whole assignment including u will achieve at least the same objective value. To see this, let $s/2 - 4 \leq t \leq 4$ be the number of accepting inputs y such that $h(y)$ is false and $y[1] \oplus y[2] \oplus y[3]$ is true. Then there are $s/2 - t$ accepting inputs y such that both $h(y)$ and $y[1] \oplus y[2] \oplus y[3]$ are false. Then, if $z_{i_1} \oplus z_{i_2} \oplus z_{i_3}$ is true, then t of the four constraints are satisfied and otherwise $s/2 - t$ of them are satisfied. As $t \leq 4$ and $s/2 - t \leq 4$ it is always possible to achieve at least the same performance by setting $u = 1$, and possibly negating the rest of the assignment as well.

To prove the lemma for $h(y) = (y[1] \equiv y[4])$ we make use of a particular property of the Håstad's NP-hard Max CSP(3XOR) instances. All variables occurring at the first position in a constraint, only occurs in that position in all other constraints as well. For each such variable x_j, we introduce an auxiliary variable x'_j. If z_{i_1} is x_j or $\overline{x}_j$, then z'_{i_1} denotes x'_j or $\overline{x'}_j$ respectively. For each constraint $z_{i_1} \oplus z_{i_2} \oplus z_{i_3}$ in the hard Max CSP(3XOR) instance we introduce four constraints,

$$
\begin{gathered}
P(z_{i_1}, z_{1_2}, z_{i_3}, z'_{i_1}) \\
P(\overline{z}_{i_1}, \overline{z}_{1_2}, z_{i_3}, \overline{z'}_{i_1}) \\
P(\overline{z}_{i_1}, z_{1_2}, \overline{z}_{i_3}, \overline{z'}_{i_1}) \\
P(z_{i_1}, \overline{z}_{1_2}, \overline{z}_{i_3}, z'_{i_1}),
\end{gathered}
$$

In fact, due to the construction of the Håstad instance we only need to introduce the first of the constraints. But in order to avoid unnecessary details in the analysis, we introduce all four constraints. By setting x'_j to the same value as x_j is assigned, we get the same analysis as for the case $u = 1$ above. Thus, what remains to be shown is that we cannot win something by setting x'_j to $-x_j$. But if we win something by setting x'_j to that value, we win at least as much by changing value on x_j instead. This is realized by noting that if $z_{i_1} \oplus z_{i_2} \oplus z_{i_3}$ is true for the majority of constraints where the literal z_{i_1} contains the variable x_j, then it is best to set $x'_j = x_j$. And otherwise, if $z_{i_1} \oplus z_{i_2} \oplus z_{i_3}$ is most often false, then by changing value to x_j a better solution is obtained. Remember that x_j only occurs in the first position of the constraints. $\square$

Lemma 22. *Let* $\{(1,1,1,1), (1,-1,-1,-1), (-1,1,-1,-1), (-1,-1,1,1)\} \subset P^{-1}(1)$, *and*

$$
\sum_{(a_1,a_2,a_3,a_4) \in P^{-1}(1)} a_3 a_4 \leq 0 \ .
$$

Then P is approximation resistant.

Proof. We assume that the reader is familiar with the notation used in Håstad's paper [11]. For example, we use A, B, $\mathcal{F}_U$ and $\mathcal{F}_W$ which are defined in [11].

We define the following PCP-verifier.

- Choose $f \in \mathcal{F}_U$ and $g_1 \in \mathcal{F}_W$ independently and with uniform probability.
- Choose functions $\mu, \mu' \in \mathcal{F}_W$ by setting $\mu(y) = 1$ with probability $1 - \epsilon$ and $\mu(y) = -1$ otherwise, independently for each y. Do the same for μ'.
- Set $g_2 = f g_1 \mu$ and $g_3 = f g_1 \mu'$.
- Accept iff $P(A(f), B(g_1), B(g_2), B(g_3))$.

We consider the multilinear expression for P:

$$
P(x_1, x_2, x_3, x_4) \quad = \quad \sum_{S \subseteq [4]} c_S \prod_{i \in S} x_i \ .
$$

The acceptance probability of the verifier is

$$
\begin{aligned}
&\mathrm{E}_{f,g_1,g_2,g_3} \left[P(A(f), B(g_1), B(g_2), B(g_3)) \right] \\
&\quad = c_\emptyset + c_{\{1\}} \mathrm{E}\left[A(f)\right] + c_{\{2\}} \mathrm{E}\left[B(g_1)\right] + \ldots + c_{\{1,2,3,4\}} \mathrm{E}\left[A(f) B(g_1) B(g_2) B(g_3)\right] \ .
\end{aligned}
$$

We want to show that for every constant $\varepsilon > 0$ this probability is upper bounded by $c_\emptyset + \varepsilon$, assuming that we cannot find a good strategy in the two prover game which is hard. Thus, we analyze each term of the above expression.

Because of folding we have that

$$\mathrm{E}[A(f)] = \mathrm{E}[B(g_1)] = \mathrm{E}[B(g_2)] = \mathrm{E}[B(g_3)] = 0 \ .$$

Due to how the verifier chooses f, g_1, g_2 and g_3, we know that (f, g_1), (f, g_2) and (f, g_3) are all independent pair of functions and thus

$$\mathrm{E}[A(f)B(g_1)] = \mathrm{E}[A(f)B(g_2)] = \mathrm{E}[A(f)B(g_3)] = 0 \ .$$

The triples (f, g_1, g_2) and $(-f, g_1, -g_2)$ are equally likely to occur. This is also true for (f, g_1, g_3) and $(-f, g_1, -g_3)$ and thus

$$\mathrm{E}[B(g_1)B(g_2)] = \mathrm{E}[B(g_1)B(g_3)] = 0 \ .$$

Due to that (f, g_1, g_2, g_3) occurs with the same probability as $(f, -g_1, -g_2, -g_3)$ we can conclude that

$$\mathrm{E}[A(f)B(g_1)B(g_2)B(g_3)] = \mathrm{E}[B(g_1)B(g_2)B(g_3)] = 0 \ .$$

As (f, g_2, g_3) occurs with the same probability as $(-f, -g_2, -g_3)$ we also know that

$$\mathrm{E}[A(f)B(g_2)B(g_3)] = 0 \ .$$

Thus, the accept probability of the verifier is

$$c_\emptyset + \mathrm{E}\left[c_{\{3,4\}}B(g_2)B(g_3)\right] + c_{\{1,2,3\}}\mathrm{E}[A(f)B(g_1)B(g_2)] + c_{\{1,2,4\}}\mathrm{E}[A(f)B(g_1)B(g_3)] \ .$$

If for a constant $\delta > 0$ the absolute value of the expectation of either $\mathrm{E}[A(f)B(g_1)B(g_2)]$ or $\mathrm{E}[A(f)B(g_1)B(g_3)]$ is larger than δ, then this can be exploited in order to create a good strategy for the two player game which is assumed that we cannot do. As $c_{\{3,4\}} \leq 0$ we need to show that $\mathrm{E}[B(g_2)B(g_3)] \geq 0$ in order to get an upper bound of the accept probability of the verifier. This is easily seen by considering f and g_1 as fix functions. Then g_2 and g_3 are chosen from the same distribution and thus the expectation of their product cannot be negative. We conclude that the accept probability is at most $c_\emptyset + (c_{\{1,2,3\}} + c_{\{1,2,4\}})\delta$.

Using standard methods this implies that for any constant $\varepsilon > 0$, it is NP-complete to approximate Max CSP(P) within $c_\emptyset + \varepsilon$ and thus P is approximation resistant. $\square$

K Proof of Theorem 14

Theorem 14. *Let P be* 3NTW, 3OXR *or* 3OR, *and let I be an instance of* Max CSP(P) *of total weight W. For every $\gamma > 0$, there exist $\varepsilon > 0$ and $\kappa > 0$ such that if there exists a $(1-\varepsilon)$-satisfying assignment with at least weight γW on even parity inputs, then I can be $(s2^{-3} + \kappa)$-approximated in probabilistical polynomial time, where s is the number of accepting inputs to P.*

Proof. The multilinear expression for odd parity is

$$\mathrm{3XOR}(x_1, x_2, x_3) = \frac{4 - 4x_1x_2x_3}{8} \ .$$

For P, there are $s - 4$ accepting inputs with even parity. Using (2), it can be concluded that $P^{(0)}(x_1, x_2, x_3) = \frac{s}{8}$ and $P^{(3)}(x_1, x_2, x_3) = \frac{s-8}{8}x_1x_2x_3$. If (x_1, x_2, x_3) has odd parity, then $P^{(0)}(x_1, x_2, x_3) + P^{(3)}(x_1, x_2, x_3) = 1$ implying that $P^{(1)}(x_1, x_2, x_3) + P^{(2)}(x_1, x_2, x_3) = 0$. If $(x_1, x_2, x_3) \in P^{-1}(1)$ but have even parity, then $P^{(0)}(x_1, x_2, x_3) + P^{(3)}(x_1, x_2, x_3) = \frac{2s}{8} - 1$ implying that $P^{(1)}(x_1, x_2, x_3) + P^{(2)}(x_1, x_2, x_3) = 2 - \frac{2s}{8} > 0$. The theorem now follows from applying Lemma 5. $\square$

L Proof of Theorem 15

Theorem 14. *Let I be an instance of* Max CSP(3OR) *with total weight W. For every $\gamma > 0$, there exist $\varepsilon > 0$ and $\kappa > 0$ such that if there exists a $(1-\varepsilon)$-satisfying assignment with weight $u_{(-1,-1,-1)}$ on input $(-1,-1,-1)$, and either $u_{(-1,-1,-1)} \leq (1/4-\gamma)W$ or $u_{(-1,-1,-1)} \geq (1/4+\gamma)W$, then I can be $(7/8+\kappa)$-approximated in probabilistical polynomial time.*

Proof. First we consider the case $u_{(-1,-1,-1)} \leq (1/4-\gamma)W$. We use Lemma 5 with $P = $ 3OR, $C = 0$ and $\delta = 3\gamma/8$. We calculate the sum specified in Lemma 5 for $C = 0$:

$$\begin{aligned}
\sum_{y \in P^{-1}(1)} u_y P^{(2)}(y) &= -\frac{3}{8} u_{(-1,-1,-1)} + \sum_{y \in P^{-1}(1) \setminus \{(-1,-1,-1)\}} \frac{1}{8} u_y \\
&\geq -\frac{3}{8}(1/4-\gamma)W + \frac{1}{8}(3/4)W \\
&= (3\gamma/8)W \ .
\end{aligned}$$

In the inequality we assumed that $\varepsilon \leq \gamma$ and thus $\sum_{y \in P^{-1}(1) \setminus \{(-1,-1,-1)\}} u_y \geq 3/4$. By Lemma 5 we have that there exist positive constants ε' and κ such that either the value of an optimal assignment is less than $(1-\varepsilon)W$, or we can $(7/8+\kappa)$-approximate it. We set $\varepsilon = \min(\varepsilon', \gamma)$.

For the second case, $u_{(-1,-1,-1)} \geq (1/4+\gamma)W$, we use Lemma 5 with $P = $ 3OR, $C = 2$ and $\delta = \gamma/2$. We calculate the sum specified in Lemma 5 for $C = 2$:

$$\begin{aligned}
\sum_{y \in P^{-1}(1)} u_y (2 \cdot P^{(1)}(y) + P^{(2)}(y)) &\geq \frac{3}{8} u_{(-1,-1,-1)} + \sum_{y \in P^{-1}(1) \setminus \{(-1,-1,-1)\}} -\frac{1}{8} u_y \\
&\geq \frac{3}{8}(1/4+\gamma)W - \frac{1}{8}(3/4-\gamma)W \\
&= (\gamma/2)W
\end{aligned}$$

By Lemma 5 we know that either the value of the assignment is less than $(1-\varepsilon)W$, or we can $(7/8+\kappa)$-approximate it. $\square$