THEORY OF COMPUTING, Volume 9 (27), 2013, pp. 845–862
www.theoryofcomputing.org

SPECIAL ISSUE: ANALYSIS OF BOOLEAN FUNCTIONS

Satisfying Degree-d Equations over GF$[2]^n$

Johan Håstad*

Received June 20, 2012; Revised July 15, 2013; Published November 27, 2013

Abstract: We study the problem where we are given a system of polynomial equations defined by multivariate polynomials over GF[2] of fixed, constant, degree $d > 1$ and the aim is to satisfy the maximum number of equations. A random assignment approximates this number within a factor 2^{-d} and we prove that for any $\varepsilon > 0$, it is NP-hard to obtain a ratio $2^{-d} + \varepsilon$. When considering instances that are perfectly satisfiable we give a polynomial-time algorithm that finds an assignment that satisfies a fraction $2^{1-d} - 2^{1-2d}$ of the constraints and we prove that it is NP-hard to do better by an arbitrarily small constant. The hardness results are proved in the form of inapproximability results for MAX-CSPs where the predicate in question has the desired form and we give some immediate results on approximation resistance of some predicates.

ACM Classification: F.2.2

AMS Classification: 68Q17

Key words and phrases: polynomials, probabilistically checkable proofs, constraint satisfaction, approximation

1 Introduction

The study of polynomial equations is a basic question of mathematics. In this paper we study a problem we call MAX-d-EQ where we are given a system of m equations of degree d in n variables over GF[2]. As we consider the case of d constant, all polynomials are given in the dense representation, i. e., by a list of monomials. Many problems can be coded as polynomial equations and in particular it is easy to code

A conference version of this paper appeared in the proceedings of APPROX 2011 [6].
*Supported by ERC grant 226 203.

 DOI: 10.4086/toc.2013.v009a027

3-SAT as equations of degree 3 and thus determining whether we can simultaneously satisfy all equations is NP-complete. It is hence natural to study the question of satisfying the maximum number of equations and our interest turns to approximation algorithms. We say that an algorithm is a C-approximation algorithm if it always returns a solution which satisfies at least $C \cdot \mathrm{OPT}$ equations where OPT is the number of equations satisfied by the optimal solution.

For the problem under study, a random assignment satisfies each equation with probability at least 2^{-d} and hence obtaining an approximation within this factor must be considered folklore. When it comes to inapproximability results, it was established early on by Håstad et al. [7] that for $d = 2$ (and implicitly for any $d \geq 2$) it is NP-hard to get a constant better than $1/2$. Given the importance of the problem it is, however, natural to try to determine the exact approximability of the problem and this is the purpose of this paper.

The result of [5] proves that the optimal approximability constant for linear equations ($d = 1$) is $1/2$. This approximability is obtained by simply picking a random assignment independently of the equations at hand. To prove tightness it is established that for any $\varepsilon > 0$, it is NP-hard to approximate the answer better than within a factor $1/2 + \varepsilon$. This is proved by constructing a suitable Probabilistic Checkable Proof (PCP). It turns out that these results extend almost immediately to the higher degree case giving the optimal constant 2^{-d} for degree-d equations. We proceed to study the case when all equations can be simultaneously satisfied.

In the case of linear equations, it follows by Gaussian elimination that once it is possible to satisfy all equations one can efficiently find such a solution. The situation for higher degree equations turns out to be more interesting. Any implied affine condition can be used to eliminate a variable but this turns out to be the limit of what can be achieved. To be more precise, from a characterization of the low-weight codewords of Reed-Muller codes by Kasami and Tokura [8], it follows that any equation satisfied by a fraction lower than $2^{1-d} - 2^{1-2d}$ must imply an affine condition. This number turns out to be the sharp threshold of approximability for satisfiable instances of systems of degree-d equations.

The upper bounds is obtained by using implied affine conditions to eliminate variables and then essentially choosing an assignment at random. To make this algorithm deterministic polynomial time we use a pseudo-random generator of Viola [11].

The lower bound is proved by constructing a PCP very much inspired by [5] and indeed nothing in the current paper relies on material not known at the time of that paper. In particular, we prove standard NP-hardness results and do not use any sophisticated results in harmonic analysis.

It is interesting to note that in the lower bound construction we only use polynomials each of which depends on a constant number of variables (in fact only $6d$ variables) while our algorithms work even if each polynomial depends on all variables.

As a by-product of our proofs we make some observations in the area of Maximum Constraint Satisfaction Problems (MAX-CSPS). The problem MAX-P is given by a predicate P of arity k and an instance is given by a sequence of k-tuples of literals. The task is to find an assignment such that the maximum number of the resulting k-tuples of bits satisfy P. Let $r(P)$ be the probability that a uniformly random assignment satisfies P. Note that $r(P)$ is the approximation ratio achieved by the algorithm that simply picks a uniformly random assignment independent of the instance under consideration. A predicate for which you cannot do significantly better (this is formally defined in the next section) in polynomial time, unless $\mathsf{P} = \mathsf{NP}$, is said to be *approximation resistant*. This property is equivalent to, for any $\varepsilon > 0$, it being NP-hard to distinguish instances of MAX-P where you can satisfy a fraction $1-\varepsilon$ of the constraints from those where you can satisfy only a fraction $r(P)+\varepsilon$.

Given this formulation it is natural to formulate a stronger notion. Namely, that it is NP-hard to distinguish instances of MAX-P where all constraints can be satisfied simultaneously from those where only a fraction $r(P)+\varepsilon$ of the constraints can be satisfied simultaneously. In this case P is called *approximation resistant on satisfiable instances*.

It is technically easier to prove approximation resistance while the extension to satisfiable instances might be more complicated, unknown, or simply false. One example to distinguish the two notions is parity of at least three variables which is known to be approximation resistant [5] but which is easy for satisfiable instances due to Gaussian elimination.

In the current paper we extend this to give an example of a predicate which is approximation resistant on satisfiable instances but which has a different, and still non-trivial, approximation ratio for satisfiable instances.

An outline of the paper is as follows. In Section 2 we give some preliminaries and the rather easy result for imperfect completeness is given in Section 3. The most technically interesting part of the paper is given in Section 4 where we study systems of equations where all equations can be satisfied simultaneously. We give the results on MAX-CSPs in Section 5 and end with some final remarks in Section 6.

This is the full version of the conference paper [6].

2 Preliminaries

We are interested in polynomials over GF[2]. Most polynomials we use are of degree d but also polynomials of degree one, that we call "affine forms" play a special role. We are not interested in the polynomials as formal polynomials, but rather as functions mapping GF$[2]^n$ to GF[2] and hence we freely use that $x_i^2 = x_i$ and thus any monomial in our polynomials can be taken to be multi-linear. We start with the following standard result which we, for completeness, even prove.

Theorem 2.1. *Any multivariate polynomial P of degree d that is nonzero takes the value 1 for at least a fraction 2^{-d} of the inputs.*

Proof. The proof is by induction over n and d, with the base case of $d = 1$ which is true as each linear polynomial is unbiased.

For the induction step, suppose $P(x) = P_0(x) + x_1 P_1(x)$ and let us consider what happens for the two possible values of x_1. If both P_0 and $P_0 + P_1$ are non-zero we are done by induction. If not, as P_1 is of degree at most $d-1$, the polynomial of P_0 and $P_0 + P_1$ that is non-zero is of degree at most $d-1$. Hence this polynomial takes the value 1 for at least a fraction 2^{1-d} of *its* inputs. As the set of inputs of this polynomial constitutes half of the inputs of P, the result follows also in this case. □

It is not difficult to see that this result is tight by considering $P(x) = \prod_{i=1}^{d} x_i$, or, more generally, products of d linearly independent affine forms. It is important for us that these are the only cases of tightness. This follows from a characterization by Kasami and Tokura [8] of all polynomials that are non-zero for at most a fraction 2^{1-d} of the inputs. A consequence of their characterization is the following theorem.

Theorem 2.2. *Let P be a degree-d polynomial over* GF[2] *which factors as*

$$P(x) = Q(X)\prod_{i=1}^{r} A_i(x)$$

where A_i are linearly independent affine forms and Q does not contain any affine factor. Then the fraction of points on which $P(x) = 1$ is at least

$$2^{-r}(2^{1-(d-r)} - 2^{1-2(d-r)}),$$

if $d \neq r$ and 2^{-d} if $d = r$.

Given that we do not need their full characterization and the proof of the part that we need is shorter than the proof of the full characterization, we give the proof of Theorem 2.2 in an appendix.

We make use of the Fourier transform and, as we are dealing with polynomials over GF[2], we let the inputs come from $\{0,1\}^n$. For any $\alpha \subseteq [n]$ we have the character χ_α defined by

$$\chi_\alpha(x) = (-1)^{\sum_{i\in\alpha} x_i}$$

and the Fourier expansion of a real-valued function f is given by

$$f(x) = \sum_{\alpha\subseteq[n]} \hat{f}_\alpha \chi_\alpha(x).$$

Suppose that $R \leq L$ and we are given a projection π mapping $[L]$ to $[R]$. We define a related operator π_2 acting on sets as follows. For $\beta \subseteq [L]$ we let $\pi_2(\beta)$ be set of the elements in $[R]$ with an odd number of preimages under π that belong to β. The reason this is a useful definition for us is that if we have $x \in \{0,1\}^R$ and define $y \in \{0,1\}^L$ by setting $y_i = x_{\pi(i)}$ then $\chi_\beta(y) = \chi_{\pi_2(\beta)}(x)$.

As is standard we use the long code introduced by Bellare et al. [4]. If $v \in [L]$ then the corresponding long code is a function $A : \{0,1\}^L \to \{-1,1\}$ where $A(x) = (-1)^{x_v}$. We require our tables to be folded, which means that they only contain values for inputs with $x_1 = 1$. The value when $x_1 = 0$ is defined to be $-A(\bar{x})$. This ensures that the function is unbiased and hence that the Fourier coefficient corresponding to the empty set is 0.

We are interested in Maximum Constraint Satisfaction Problems (MAX-CSPs) given by a predicate P of some constant arity k. An instance is given by a set of k-tuples of literals and the task is to find an assignment that maximizes the number of the resulting k-tuples of bits that satisfy the predicate P. As mentioned in the introduction we let $r(P)$ be the probability that a random assignment satisfies P. We have the following two definitions mentioned briefly in the introduction.

Definition 2.3. A predicate P is *approximation resistant* if, for any $\varepsilon > 0$, it is NP-hard to approximate MAX-P within $r(P) + \varepsilon$.

Definition 2.4. A predicate P is *approximation resistant on satisfiable instances* if, for any $\varepsilon > 0$, it is NP-hard to distinguish instances of MAX-P where all constraints can be satisfied simultaneously from those where only a fraction $r(P) + \varepsilon$ of the constraints can be satisfied simultaneously.

We define P^L to be the predicate of arity $3k$ defined on (x_j^i), $1 \leq i \leq 3$ and $1 \leq j \leq k$, by setting $y_j = x_j^1 + x_j^2 + x_j^3$ and defining $P^L(x)$ to be true iff $P(y)$ is true. Note that if P is given by a polynomial equation of degree d then so is P^L. By slightly abusing language, in this situation we do not distinguish the predicate P from the polynomial defining the predicate.

3 The case of imperfect completeness

We start with the algorithm.

Theorem 3.1. *Given a system of m polynomial equations of degree d over* GF[2]*, it is possible, in polynomial time, to find an assignment that satisfies at least $m2^{-d}$ equations.*

Proof. In fact, by Theorem 2.1, a random assignment satisfies each equation with probability 2^{-d} and thus just picking a random assignment gives a randomized algorithm fulfilling the claim of the theorem in expectation.

To get a deterministic algorithm we use the method of conditional expectations. The idea is to assign values to the variables in order and we choose values for the variables such that the expected number of satisfied equations, if the remaining variables are set randomly, never drops below $m2^{-d}$. When all variables are set, any equation is satisfied with probability either 0 or 1 and hence at least $m2^{-d}$ equations are satisfied.

To make this procedure efficient we, at each time, use the lower estimate that at least a fraction $2^{-d'}$ of the inputs satisfy any nontrivial equation that is currently of degree d'. For trivial equations we naturally give the score 1 if they are satisfied and 0 if they are falsified. Looking at the proof of Theorem 2.1 we can set the variables one by one making sure that this lower bounds estimate never decreases. The value of this lower bound is initially at least $m2^{-d}$ and at the end exactly the number of satisfied equations. The theorem follows. $\square$

The lower bound follows rather immediately from known results.

Theorem 3.2. *For any $\varepsilon > 0$ it is NP-hard to approximate* MAX-d-EQ *within $2^{-d} + \varepsilon$.*

Proof. In [5] it is proved that, for any $\delta > 0$, it is NP-hard to distinguish systems of linear equations where a fraction $1-\delta$ of the equations can be satisfied from those where only a fraction $1/2+\delta$ can be satisfied. Suppose we are given an instance of this problem with m equations which, possibly by adding one to both sides of the equation, can be be assumed to be of the form

$$A_i(x) = 1\,, \qquad 1 \le i \le m. \tag{3.1}$$

Taking all d-wise products of equations from (3.1) we end up with m^d equations, each of the form

$$\prod_{j=1}^{d} A_{i_j}(x) = 1\,,$$

which clearly is a polynomial equation of degree at most d. This system has the same optimal solution as the linear system and if it satisfies τm linear equations, then it satisfies $\tau^d m^d$ degree-d equations. As it is NP-hard to determine whether $\tau = 1-\delta$ or $\tau = 1/2+\delta$, it is NP-hard to approximate the optimal value for the degree-d system with a ratio better than

$$\left(\frac{\frac{1}{2}+\delta}{1-\delta}\right)^d$$

and by making δ sufficiently small this can be made smaller than $2^{-d}+\varepsilon$. The theorem follows. $\square$

We remark that, by appealing to the results by Moshkovitz and Raz [9], we can even obtain results for non-constant values of ε.

4 Completely satisfiable systems

When studying systems where it is possible to simultaneously satisfy all equations the situation changes and let us start with the positive results.

4.1 Finding a good assignment

Suppose we have an equation of the form $P(x) = 1$ and that this equation implies the affine condition $A(x) = 1$. Then, as the system is satisfiable, we can use this equation to eliminate one variable from the system, preserving the degrees of all equations. This is done by taking some variable x_i that appears in A and replacing it by $x_i + A(x) + 1$ (note that this function does not depend on x_i as the two occurrences of this variable cancel). This substitution preserves the satisfiability of the system and the process stops only when none of the current equations implies an affine condition.

Using Theorem 2.2 we see that when this process ends each equation is satisfied by at least a fraction $2^{1-d} - 2^{1-2d}$ of the inputs and thus the following theorem should not come as a surprise.

Theorem 4.1. *There is a polynomial time algorithm that, given a system of m simultaneously satisfiable equations of degree d over* GF[2]*, finds an assignment that satisfies at least $\lceil (2^{1-d} - 2^{1-2d})m \rceil$ equations.*

Proof. There are two points in the outlined argument that require closer inspection. The first is the question of how to actually determine whether a polynomial equation implies an affine condition and the second is to make sure that once the process of finding implied affine conditions has ended we can indeed deterministically find a solution that satisfies the expected number of equations. Let us first address the issue of determining whether a given equation implies an affine condition.

Suppose $P(x) = 1$ implies $A(x) = 1$ for some unknown affine function A. Let us assume that x_1 appears in A with a nonzero coefficient. We may write

$$P(x) = P_0(x) + P_1(x)x_1$$

where neither P_0 nor P_1 depends on x_1. We want to prove that $P(x) = A(x)P_1(x)$ and towards this consider

$$Q(x) = P(x) + A(x)P_1(x). \tag{4.1}$$

As x_1 appears with coefficient one in A it follows that Q does not depend on x_1 and let us assume that Q is not identically 0. Choose any values for $x_2, x_3 \ldots x_n$, to make $Q(x) = 1$ and set x_1 to make $A(x) = 0$. It follows from (4.1) that $P(x) = 1$ and thus we have found a counterexample to the assumed implication. We can hence conclude that $Q \equiv 0$ and we have

$$P(x) = A(x)P_1(x).$$

We claim furthermore that this procedure is entirely efficient. Namely given P and the identity of one variable occurring in A, P_1 is uniquely defined. Once P_1 is determined the rest of the coefficients of A can be found by solving a linear system of equations. As there are only n candidates for a variable in A and solving a linear system of equations can be done in polynomial time we conclude that the entire process of finding implied affine conditions can be done in polynomial time.

Once this process halts we need to find an assignment that satisfies the expected number of equations. This could possibly be done in a manner similar to how it was done in the proof of Theorem 3.1 but for the general case we do not know how to, in deterministic polynomial time, find a suitable bound for the probability that an equation is satisfied. It is true that the bound of Theorem 2.2 can be calculated deterministically, but as we are not able to prove it by induction, setting only one variable at the time, it is not clear how to use it to guide an algorithm. We turn to a different approach and we need the following result by Viola [11].

Theorem 4.2 ([11])**.** *For any constant d and $\varepsilon > 0$ there exists a polynomial time pseudo-random generator G with seed length $O(d \log n + d2^d \log(1/\varepsilon))$ and output in $\{0,1\}^n$ such that*

$$|\Pr[P(x) = 0] - \Pr[P(G(y)) = 0]| \le \varepsilon$$

for any polynomial P of degree d. Here x is a uniformly random string in $\{0,1\}^n$ and y is a uniformly random seed.

This theorem implies that if we are willing to sacrifice a small ε it is enough to consider the candidate assignments $G(y)$ for all possible seeds y and taking the assignment from this set that satisfies the maximum number of equations. Let us be more precise.

We know that the expected number of satisfied equations when we consider a uniformly random input is $(2^{1-d} - 2^{1-2d})m$ and, by Theorem 4.2, the same number, when we consider a random output from G, is at least $(2^{1-d} - 2^{1-2d} - \varepsilon)m$. Setting $\varepsilon = 2^{-2d}m^{-1}$ and observing that the maximum number of equations satisfied is an integer we see that this number is at least

$$\lceil (2^{1-d} - 2^{1-2d})m - 2^{-2d} \rceil .$$

Finally, as $(2^{1-d} - 2^{1-2d})m$ is an integer multiple of 2^{1-2d}, we have that

$$\lceil (2^{1-d} - 2^{1-2d})m - 2^{-2d} \rceil = \lceil (2^{1-d} - 2^{1-2d})m \rceil ,$$

and the proof is complete. $\square$

Note that the generator of Viola is only needed to derandomize the algorithm and had we been content with a randomized algorithms we could simply replace the last step by random sampling, obtaining a much more efficient algorithm.

We turn to establishing that Theorem 4.1 is essentially tight by supplying a matching lower bound in the next section.

4.2 Inapproximability results

In this section we establish the following lower bound result.

Theorem 4.3. *For any $\varepsilon > 0$ it is NP-hard to distinguish satisfiable instances of* MAX-d-EQ *from those where the optimal solution satisfies a fraction $2^{1-d} - 2^{1-2d} + \varepsilon$ of the equations.*

Proof. Consider the predicate, Q, on $6d$ variables given by

$$\prod_{i=1}^{d} L_i(x) + \prod_{i=d+1}^{2d} L_i(x) = 1\,, \tag{4.2}$$

where $L_i(x) = x_{3i-2} + x_{3i-1} + x_{3i}$, i. e., each L_i is the sum of three independent variables. Note that Q is of the form P^L according to the definition given in Section 2 where P is defined by the equation

$$\prod_{i=1}^{d} y_i + \prod_{i=d+1}^{2d} y_i = 1\,. \tag{4.3}$$

Theorem 4.3 now follows from Theorem 4.4 below as the probability that a random assignment satisfies P is exactly $2^{1-d} - 2^{1-2d}$. $\square$

Theorem 4.4. *The predicate Q defined by (4.2) is approximation resistant on satisfiable instances.*

Proof. We below give a general PCP where the acceptance criteria is given by a predicate of the type P^L. The aim of the PCP is to prove the existence of a good labeling for a projective label cover problem and let us start by defining this problem. For comparison we note that this is the same starting point as in [5] but we formulate it in more modern terms.

We are given a bipartite graph with vertices U and V. Each vertex $u \in U$ should be given a label $\ell(u) \in [L]$ and each vertex $v \in V$ should be given a label $\ell(v) \in [R]$. For each edge (u, v) there is a mapping $\pi_{u,v}$ and a labeling satisfies this edge iff $\pi_{u,v}(\ell(u)) = \ell(v)$.

As stated in [5] (and based on [1] and [10]) it is known that for any constant $\varepsilon > 0$ there are constant values for L and R such that it is NP-hard to determine whether the optimal labeling satisfies all constraints or only a fraction ε of the constraints. Using [9] one can extend this to non-constant size domains, but let us ignore this point.

As is standard, we transform the label cover instance into a PCP by long-coding a good assignment, and for each vertex u we have a table $g_u(y)$ for $y \in \{0,1\}^L$, and similarly we have a table $f_v(x)$ for $x \in \{0,1\}^R$ for each $v \in V$. As mentioned in the preliminaries we assume that these tables are folded and hence each table is unbiased.

The PCP can use an arbitrary predicate of the form P^L of arity $3k$ (P is thus of arity k) and is also parametrized by a probability distribution D on k-bit strings.

Test $T_{P,D}$

1. Pick uniformly at random an edge (u, v) which comes with a projection constraint $\pi_{u,v} : [L] \mapsto [R]$.
2. Pick $x^{(i)} \in \{0,1\}^R$ and $y^{(i)} \in \{0,1\}^L$ uniformly at random, $1 \le i \le k$.
3. For each $j \in [L]$ pick an element $\mu^{(j)}$ from the distribution D and construct $z^{(i)}$ by setting

$$z_j^{(i)} = x_{\pi_{u,v}(j)}^{(i)} + y_j^{(i)} + \mu_i^{(j)} \mod 2\,.$$

4. Read the $3k$ bits[1] corresponding to $f_v(x^{(i)})$, $g_u(y^{(i)})$, and $g_u(z^{(i)})$. Set
$$w_i = \text{XOR}(f_v(x^{(i)}), g_u(y^{(i)}), g_u(z^{(i)}))$$
for $1 \le i \le k$ and accept iff the k-bit string w satisfies P.

We first have the easy completeness. Let E_D denote the expected value operator when the input is chosen with distribution D.

Lemma 4.5. *If there is a labeling that satisfies all the constraints in the underlying label cover instance then there is a proof for $T_{P,D}$ that makes the verifier of this proof accept with probability at least* $\mathrm{E}_D[P(x)]$.

Proof. As indicated above, given a satisfying labeling for the label cover problem we use the long code to construct a good proof for the PCP. In other words, if the node v is given the label ℓ_v then f_v is set to be the long code for ℓ_v and similarly g_u is the long code of the label of u. It follows, more or less by definition, that for such a proof the string w equals $\mu^{(\ell_u)}$ and as this string was chosen according to the distribution D, the lemma follows. $\square$

Let us turn to soundness. Let m_D be its maximum of the absolute value for a Fourier coefficient of the measure D that corresponds to a non-constant character. In other words, we let
$$m_D = \max_{\alpha \neq 0^k} \left| \mathrm{E}_D[\chi_\alpha(x)] \right| . \tag{4.4}$$
Note that $m_D < 1$ unless D is completely supported on an affine subspace of $\{0,1\}^k$.

Lemma 4.6. *If the verifier in test $T_{P,D}$ accepts with probability at least $r(P) + \varepsilon$ then there is a labeling in the label cover problem that satisfies at least a fraction $c_k(1-m_D)\varepsilon^2$ of the constraints of this problem. Here $c_k > 0$ is a constant depending only on k.*

Before we prove this lemma let us give some high level intuition what mechanisms are behind the proof. That the exclusive-or of several (at least three) bits is easy to analyze using the Fourier transform was clear already in [5]. This is the reason why we work with P^L. The other important property needed is that the entire construction is not linear. This was achieved by adding random noise in [5] while here we have the distribution D with the key property that its support is not contained in an affine subspace.

Proof of Lemma 4.6. For notational convenience let us drop the subscripts on f, g and π. Expand the predicate P by its multi-linear expansion. Since the constant term, $\hat{P}_\emptyset$, equals $r(P)$, we conclude that given the assumption of the lemma there are sets S_1, S_2 and S_3, at least one of which is non-empty such that
$$\left| E\left[\prod_{i \in S_1} f(x^{(i)}) \prod_{i \in S_2} g(y^{(i)}) \prod_{i \in S_3} g(z^{(i)}) \right] \right| \ge c_k \varepsilon , \tag{4.5}$$
for some constant c_k depending only on k.

First note that if $S_2 \neq S_3$ the expectation in (4.5) is zero as for any i in the symmetric difference we get a factor $g(y^{(i)})$ or $g(z^{(i)})$ that is independent of the other factors and, as g is folded, the expectation of

[1] We interpret -1 as the bit 1 and 1 as the bit 0.

such a term is 0. Note here that $y^{(i)}$ and $z^{(i)}$ are in fact symmetric and in particular if the value of $y_j^{(i)}$ is not constrained, then $z_j^{(i)}$ is uniformly chosen and independent of all other variables.

To get a non-zero value we also need $S_1 = S_3$ as otherwise negating $x^{(i)}$ in the symmetric difference we get canceling terms. Thus we need to study

$$E\left[\prod_{i\in S} f(x^{(i)})g(y^{(i)})g(z^{(i)})\right]. \tag{4.6}$$

Expanding each function by the Fourier transform we get the expectation

$$E\left[\prod_{i\in S}\left(\sum_{\alpha^i,\beta^i\gamma^i} \hat{f}_{\alpha^i}\hat{g}_{\beta^i}\hat{g}_{\gamma^i}\chi_{\alpha^i}(x^{(i)})\chi_{\beta^i}(y^{(i)})\chi_{\gamma^i}(z^{(i)})\right)\right]. \tag{4.7}$$

If we mentally expand this product of sums and look at the expectation of each term we see that terms with $\gamma^i \neq \beta^i$ give contribution 0. This follow as if $j \in \beta^i/\gamma^i$ then $y_j^{(i)}$ is independent of all other occurring variables. By symmetry this argument applies to $z_j^{(i)}$ if $j \in \gamma^i/\beta^i$.

If $\pi_2(\beta^i) \neq \alpha^i$ then, for any j in the symmetric difference, negating $x_j^{(i)}$ we negate the resulting term and we can conclude that also in this case the expectation is 0. To analyze the remaining terms, let μ_i denote the vector $(\mu_i^{(j)})_{j=1}^L$ then

$$\chi_{\pi_2(\beta^i)}(x^{(i)})\chi_{\beta^i}(y^{(i)})\chi_{\beta^i}(z^{(i)}) = \chi_{\pi_2(\beta^i)}(x^{(i)})\chi_{\beta^i}(y^{(i)})\chi_{\beta^i}(x_\pi^{(i)} + y^{(i)} + \mu_i) = \chi_{\beta^i}(\mu_i),$$

and thus (4.7) reduces to

$$E\left[\prod_{i\in S}\left(\sum_{\beta^i} \hat{f}_{\pi_2(\beta^i)}\hat{g}^2_{\beta^i}\chi_{\beta^i}(\mu_i)\right)\right]. \tag{4.8}$$

We have

$$\prod_{i\in S}\chi_{\beta^i}(\mu_i) = \prod_{j\in\cup_i\beta^i}(-1)^{\sum_i \mu_i^{(j)}} \tag{4.9}$$

where the sum in the exponent is over the set of i such that $j \in \beta^i$. As $\mu^{(j)}$ is chosen independently for different values of j and each factor in the product corresponds to a non-trivial character the absolute value of the expectation of (4.9) is bounded, in absolute value, by

$$m_D^{|\cup_i\beta^i|} \le m_D^{\sum_{i\in S}|\beta^i|/k},$$

and hence we can conclude from (4.5) that

$$\mathrm{E}_{u,v}\left[\prod_{i\in S}\left(\sum_{\beta^i}\left|\hat{f}_{\pi_2(\beta^i)}\right|\hat{g}^2_{\beta^i} m_D^{|\beta^i|/k}\right)\right] \ge c_k\varepsilon. \tag{4.10}$$

As S is nonempty and any factor is bounded from above by one we conclude that

$$\mathrm{E}_{u,v}\left[\sum_{\beta}\left|\hat{f}_{\pi_2(\beta)}\right|\hat{g}_{\beta}^2 m_D^{|\beta|/k}\right] \geq c_k\varepsilon\,. \tag{4.11}$$

The Cauchy-Schwarz inequality implies that

$$\sum_{\beta}\left|\hat{f}_{\pi_2(\beta)}\right|\hat{g}_{\beta}^2 m_D^{|\beta|/k} \leq \left(\sum_{\beta}\hat{g}_{\beta}^2\right)^{1/2}\left(\sum_{\beta}\hat{f}_{\pi_2(\beta)}^2\hat{g}_{\beta}^2 m_D^{2|\beta|/k}\right)^{1/2} \tag{4.12}$$

$$\leq \left(\sum_{\beta}\hat{f}_{\pi_2(\beta)}^2\hat{g}_{\beta}^2 m_D^{2|\beta|/k}\right)^{1/2}. \tag{4.13}$$

Using (4.11), and $\mathrm{E}[X^2] \geq \mathrm{E}[X]^2$, we conclude that

$$\mathrm{E}_{u,v}\left[\sum_{\beta}\hat{f}_{\pi_2(\beta)}^2\hat{g}_{\beta}^2 m_D^{2|\beta|/k}\right] \geq c_k^2\varepsilon^2\,. \tag{4.14}$$

We now extract a probabilistic labeling using the standard procedure. For each u we choose a set β with probability $\hat{g}_{\beta}^2$ and return a random element in β. Similarly for each v we choose a set α with probability $\hat{f}_{\alpha}^2$ and return a random element in α. The expected fraction of satisfied constraints under this strategy is clearly at least

$$\mathrm{E}_{u,v}\left[\sum_{\beta}\hat{f}_{\pi_2(\beta)}^2\hat{g}_{\beta}^2\frac{1}{|\beta|}\right]. \tag{4.15}$$

We have that $e^{m_D-1} \geq m_D$ for any $0 \leq m_D \leq 1$ and $te^{-t} \leq e^{-1}$ for all $t \geq 0$ and using these two inequalities we have

$$\frac{1}{x} \geq \frac{2e(1-m_D)}{k}e^{2x(m_D-1)/k} \geq \frac{2e(1-m_D)}{k}m_D^{2x/k}$$

for any $x \geq 1$. Applying this last inequality with $x = |\beta|$ and inserting this into (4.15) and comparing this to (4.14) we see that the expectation of (4.15) is at least

$$\frac{2e(1-m_D)c_k^2}{k}\varepsilon^2\,.$$

Finally, adjusting the value of c_k, this completes the proof of Lemma 4.6. □

We now prove Theorem 4.4 using the standard translation of PCPs where the verifier uses $O(\log n)$ bits of randomness and has acceptance criteria P to a lower bound for MAX-P. In view of Lemma 4.5 and Lemma 4.6 all we have to do is to supply a distribution D supported on inputs such that $P(y) = 1$ (where P is defined by (4.3)) such that $m_D < 1$.

Let us describe D by a sampling procedure. First flip a bit b and if $b = 0$ set $y_i = 1$ for $1 \leq i \leq d$ while $(y_i)_{i=d+1}^{2d}$ are picked uniformly from the $2^d - 1$ different d bit strings that are not all-one. If $b = 1$ then the roles of the two halves are interchanged.

Clearly this distribution is fully supported on strings accepted by P. Let $\alpha \subseteq [2d]$ be a nonempty set defining a character. If α is contained in one of the two halves we observe that the distribution on this half is obtained by picking a string from the uniform distribution with probability $(1/2)(1+(2^d-1)^{-1})$ and otherwise picking the all one string. It follows that in this case

$$\left|\mathrm{E}_{D_\mu}\left[(-1)^{\sum_{i\in S}\mu_i}\right]\right| = \frac{1}{2}\left(1-(2^d-1)^{-1}\right) < \frac{1}{2}\,.$$

If, on the other hand, α contains inputs from both halves then by conditioning on which half gets the all one assignment it follows that

$$\left|\mathrm{E}_{D_\mu}\left[(-1)^{\sum_{i\in S}\mu_i}\right]\right| \leq (2^d-1)^{-1} < \frac{1}{2}\,.$$

We conclude that $m_D \leq 1/2$.

This completes the proof of Theorem 4.4. □

5 Consequences for MAX-CSPS

Let us draw some conclusions from Lemma 4.5 and Lemma 4.6 relating to MAX-CSPs.

Theorem 5.1. *For any non-trivial predicate P that accepts at least one input, the predicate P^L is approximation resistant.*

Proof. Let $\alpha \in \{0,1\}^k$ be an input accepted by P. For an arbitrary $\delta > 0$, define a distribution D by setting $\mu_i = \alpha_i$ with probability $1-\delta$ and otherwise $\mu_i = \overline{\alpha_i}$, independently for each i. The theorem now follows for Lemma 4.5 and Lemma 4.6 as $\mathrm{E}_D[P(x)] \geq 1-k\delta$ and $m_D \leq 1-\delta$ and δ might be arbitrarily small. □

It is not difficult to see that for any P, P^L supports a measure that is pairwise independent. This implies that the results of Austrin and Mossel [3] would have been sufficient to give approximation resistance but this conclusion relies on the unique games conjecture. In our case we get NP-hardness which is an advantage and it is also possible to get a general theorem with perfect completeness.

Theorem 5.2. *For any predicate P such that $P^{-1}(1)$ is not contained in a $(k-1)$-dimensional affine subspace of $\{0,1\}^k$, the predicate P^L is approximation resistant for satisfiable instances.*

Proof. Given the above discussion we need only define a suitable distribution D and we let it be the uniform distribution on inputs accepted by P. The PCP is now given by $T_{P,D}$ and by the assumption of the theorem we have $m_D < 1$. □

Note that Theorem 4.4 is a special case of Theorem 5.2 where we spelled out the details of the proof more carefully.

It is tempting to guess that for any P that does imply an affine condition, and hence Theorem 5.2 does not apply, P^L would not be approximation resistant on satisfiable instances. This does not seem to be obviously true and let us outline the problems.

We can use the implied affine conditions to eliminate some variables as we did in the proof of Theorem 4.1. The final stage when we have no more implied affine constraints is, however, more difficult to control. The resulting constraints are given by affine constraints in conjunction with the original P. By the assumption on perfect satisfiability we can conclude that each equation is still satisfiable but not much more. In particular we have no immediate estimate on the expected number of equations that will be satisfied if we give uniformly random values to the remaining free variables.

If, however, our predicate is of limited degree when viewed as a polynomial we have more information on the result. Clearly during the process of eliminating affine constraints, the degrees of the equations do not increase, and in fact, they decrease when we remove the known affine factor within each polynomial. We get the following conclusion.

Theorem 5.3. *Suppose predicate P of arity k is given by a polynomial of degree d that contains r linearly independent affine factors. Then if P accepts less than a fraction $2^{1-(d-r)} - 2^{1-2(d-r)}$ of the inputs, P^L is approximation resistant but not approximation resistant on satisfiable instances, unless $\mathsf{NP} \neq \mathsf{P}$.*

Proof. The predicate is approximation resistant by Theorem 5.1. On perfectly satisfiable instances we can run the algorithm of Theorem 4.1, and as we remove affine constraints the resulting degree is at most $d - r$. $\square$

The simplest example of a predicate for which this theorem applies is the predicate, P, given by the equation

$$x_1(x_2x_3 + x_4x_5) = 1$$

which has $d = 3$ and $r(P) = 3/16$. For this instantiation of P, P^L is approximation resistant but not approximation resistant for satisfiable instances. To get a hardness result for satisfiable constraints we can use Theorem 4.4 for the predicate

$$x_2x_3 + x_4x_5 = 1$$

which is approximation resistant with factor $3/8$ on satisfiable instances. We get a matching algorithm as the affine factor can be removed and the equations that remain are of degree 2.

Let us finally point out that all our approximation resistance results establish the stronger property of "uselessness" introduced by Austrin and Håstad [2]. This follows as we are able to bound arbitrary non-trivial characters and not only the characters appearing in multivariate expression of the considered predicates.

6 Final words

The current paper gives optimal approximability results for satisfying the maximum number of low-degree equations over GF[2]. The methods used in the proofs are more or less standard and thus the main contribution of this paper is to obtain tight results for a natural problem. There is a provable difference between perfectly satisfiable and almost-perfectly satisfiable systems in that we can satisfy strictly more equations in the former case. The difference is not as dramatic as in the linear case, but still striking.

For the case of MAX-CSPs we obtained a few approximation resistance results for, admittedly, non-standard predicates. We feel, however, that the examples give, a not major but nonempty, contribution towards understanding the difference of approximation resistant predicates and those predicates that have this property also on satisfiable instances. Our example of an approximation resistant predicate which has another, nontrivial, approximation constant on satisfiable instances is the first of its kind. Although not surprising this result gives another piece in the puzzle to understand MAX-CSPs.

Our problem is not a CSP in the traditional sense as the only restriction we put on our polynomials is that they are of bounded degree and in particular each constraint can depend on all variables. This implies that an inapproximability for our basic problem does not even imply the PCP theorem and thus could potentially be proved by simpler methods, not relying on the PCP theorem. Indeed the results of Håstad, Phillips and Safra [7] mentioned in the introduction do not rely on the PCP theorem. It is conceivable that it is possible to prove the optimal bounds (potentially even with stronger error terms) using such methods.

A natural extension of the current work is to study what happens over fields other than GF[2]. This is clearly an interesting problem which deserves to be studied. We have not made a serious attempt to extend the results of the current paper but feel that such an extension would require some work as we have made use of many properties that are not true over larger fields. As a simple example note that $P(x)$ might be a non-constant polynomial of degree two and still $P(x) = 0$ might have no solution modulo 3.

Acknowledgement I thank Parikshit Gopalan for alerting me to the paper [8] and providing me with an electronic version of that paper. I am also grateful to Srikanth Srinivasan and Shachar Lovett who independently pointed out the fact that the generator by Viola can be used to derandomize the main algorithm. I also thank Dieter van Melkebeek for alerting me to the fact that I initially used the generator in a strange way and pointing out the best way to get the derandomization. Finally I am grateful to three referees for a careful reading of the manuscript.

A Proof of Theorem 2.2

The goal of the current section is the prove Theorem 2.2. We remind the reader that this result follows from the characterization by Kasami and Tokura [8] of all codewords of the Reed-Muller code that have weight at most twice the minimal weight.

First note that the bound of Theorem 2.2 is sharp as it is obtained by

$$x^{\alpha}(x^{\beta} + x^{\gamma})$$

where α, β, and γ are disjoint multi-indices where the size of α is r and the sizes of β and γ both are $d-r$.

Proof. We prove the statement by induction and we establish $d = 2$ as the base case below to avoid degenerate cases. The following easy observations are useful for us.

- As $d-r$ cannot equal 1, it follows that for any degree-d polynomial that is not a product of affine factors, the fraction of inputs on which it takes the value one is at least $(3/2)\,2^{-d}$.
- The lower bound to be proved never exceeds 2^{1-d}, and thus this bound is always sufficient (but not always possible).

The statement for general d and r follows from the case of $d-r$ and 0 and hence we may focus on the case of $r=0$ (but of course use arbitrary r in the inductive statements).

A fact that needs to be taken into account is, as we are using that $x_i^2 = x_i$, that the factorization is not unique. In particular if A and A' are two affine factors of a polynomial P then another factor is $1+A+A'$ as

$$(1+A+A')A = A'A\,.$$

Thus if we have several affine factors we can construct new affine factors by taking the sum of of an even number of such factors added with the constant 1 or the sum of an odd number of such factors. Let us point out that the statement of Theorem 2.2 that Q does not contain any affine factor is intended to mean that there is no factorization of P that contains the given affine factors and one more linearly independent affine factor.

Another useful fact is that an affine function $A(x)$ is a factor of a polynomial P iff $A(x)=0$ implies $P(x)=0$ or equivalently if $P(y)=1$ implies $A(y)=1$. The non-obvious direction of this statement was established during the proof of Theorem 4.1 when identifying implied affine constraints.

For the reader worried about the non-unique factorization, let us point out the following fact that we leave to the reader to verify. The number of affine factors is the co-dimension of the affine hull of all points such that $P(x)=1$. The factors that may appear in the factorization are the affine equations defining this affine hull. In the full factorization we may take any full set of linearly independent equations.

Let us address the case of $d=2$, which can be established by the normal form of degree-two polynomials, but let us follow a different path to prepare for the general proof. As stated above, the interesting case is when P has no affine factor and let us write $P(x) = P_0(x) + x_1 P_1(x)$.

Consider setting x_1 to its two values and as P does not contain an affine factor neither of the induced polynomials can be identically 0. Furthermore if neither of these settings result in a polynomial with an affine factor we are done by induction. Let us finally assume that $x_1 = 0$ results in an affine factor, which we, by an affine change of coordinates, can assume is x_2. Thus we can assume that

$$P(x) = x_2 A_2(x) + x_1 A_1(x)\,,$$

for two affine functions A_1 and A_2 where we also can assume that x_i does not appear in $A_i(x)$ as it can be replaced by 1 giving the same result. The main case is that the collection of $x_1, x_2, A_1(x)$, and $A_2(x)$ form linearly independent affine functions and in this case the fraction of inputs for which P is one is exactly $3/8$ which is the claimed bound. We have a number of cases to consider when the four functions are not linearly independent.

1. $A_1(x) \equiv 1$. If $A_2(x) = x_1$, then x_1 is a factor of P while if $A_2(x) = 1+x_1$ then P is one with probability $3/4$. Finally if $A_2(x)$ is linearly independent of x_1 this probability is $1/2$.

2. $A_1(x) = x_2$ makes x_2 a factor of P.

3. $A_1(x) = 1+x_2$ makes $Pr[P(x)=1] = 1/2$ unless we have $A_2(x) \equiv 1$ when this probability is $3/4$.

4. $A_1(x)$ is linearly independent of x_2. In this case, by an affine change of variables, we can assume that $A_1(x) = x_3$. Now since $A_2(x)$ does not contain x_2, is linearly dependent of x_1 and x_3, and is not a factor of $x_1 x_3$ (ruling out also $(1+x_1+x_3)$), $A_2(x)$ must equal one of the functions 1, $1+x_1$ or $1+x_3$. In each of these cases it is easy to check that $Pr[P(x)=1]$ is $1/2$.

This finishes the case $d=2$ end we turn to the general case. Not surprisingly, also here we end up analyzing a number of cases.

As in the case $d=2$ we can assume that P has no affine factors but the polynomial resulting when substituting $x_1=0$ gives a polynomial with at least one affine factor. Picking a full set of linearly independent factors and making an affine transformation we can assume that

$$P(x)=x_2x^{\beta}P_2(x)+x_1P_1(x)$$

where β is a possible empty multi-index and P_2 has no affine factors. First let us consider affine factors in P_1.

Let $\prod_{i=1}^{r}A_i(x)$ be the affine factors that appear in P_1. We have two cases depending whether each A_i that might appear in the factorization, together with the affine forms that appear in the first product (i. e., the coordinate functions given by x_2 and the elements of β) are linearly independent.

Suppose these functions are not linearly independent and hence that we can choose the factorization such that $A_1(x)$ only depends on x_2 and the variables in β. Let us look at the point x^0 where x_2 and all elements of β equals 1. If $A_1(x^0)=1$ then $A_1(x)$ is a factor of P and this is a contradiction of the assumptions. If, on the other hand $A_1(x_0)=0$ then the sets of points where $x_2x^{\beta}P_2(x)=1$ and $x_1P_1(x)=1$ are disjoint and, as each of these sets is of density at least 2^{-d}, we get $\Pr[P(x)=1]\geq 2^{1-d}$ and the theorem follows also in this case.

Now assume that any affine form appearing in the factorization of P_1 is linearly independent of x_2 and the variables in β. Then, by an affine transformation, we can assume that

$$P(x)=x_2x^{\beta}P_2(x)+x_1x^{\gamma}P_1'(x)\,, \tag{A.1}$$

where β and γ are disjoint multi-indices and no more affine factors can be pulled out of P_2 or P_1'.

Let us analyze what happens for the four possible simultaneous assignments of values to x_1 and x_2. When both are 0 we get a function that is identically 0 which is not good for us but in the other cases we get non-zero polynomials of degree $d-1$ and we now analyze the structure of these polynomials.

When $x_1=0$ and $x_2=1$ we get $x^{\beta}P_2(x)$, when $x_1=1$ and $x_2=0$ we get $x^{\gamma}P_1'(x)$, and in the final case we have

$$W(x)=x^{\beta}P_2(x)+x^{\gamma}P_1'(x)\,.$$

Note that W is not identically 0 as (x_1+x_2) then would have been an affine factor of P and furthermore that $x^{\beta}P_2$, $x^{\gamma}P_1'$, W all are of degree at most $d-1$.

Suppose first that neither P_2 nor P_1' is the constant one. Then, using the bound that a degree-$(d-1)$ polynomial that is not a pure product of affine factors is one with probability at least $(3/2)\,2^{1-d}$, we have that $\Pr[P(x)=1]$ is at least

$$\frac{1}{4}\left(\frac{3}{2}2^{1-d}+\frac{3}{2}2^{1-d}+2^{1-d}\right)=2^{1-d}$$

proving the bound in this case. By symmetry we may thus assume that $P_2\equiv 1$. If $\gamma=\emptyset$ or W does not contain any affine factor, then $\Pr[P(x)=1]$ is at least

$$\frac{1}{4}(2^{1-d}+2^{1-d}+2^{2-d}-2^{3-2d})\,,$$

which exactly equals the claimed bound. Thus we can assume that γ is non-empty and W contains an affine factor $A(x)$ and remember that

$$W(x) = x^{\beta} + x^{\gamma} P_1'(x). \tag{A.2}$$

Suppose A only depends on variables in β. If it is fixed to 1 by setting all variables in β to one, then $A(x)$ is a factor of x^{β} and hence also of $W(x) + x^{\beta} = x^{\gamma} P_1'(x)$ and hence also of $P(x)$, contradicting assumptions. If $A(x)$ is forced to 0 by this assignment then setting any variable in γ (remember it is non-empty and disjoint from β) to 0 and we get $W(x) = 0$ while $x^{\beta} = 1$ and $x^{\gamma} P_1'(x) = 0$ contradicting (A.2). Thus we can assume that $A(x)$ depends on some variable outside β.

If we can fix some variable in γ to 0, the variables of β to one and A to zero we get a contradiction to (A.2) as $x^{\beta} = 1$ while $W(x) = 0$ and $x^{\gamma} = 0$. If the size of γ is at least 2 or W contains at least two different affine factors we claim that this must be possible. Suppose first that the size of γ is at least 2.

As $A(x)$ does not only depend on variables in β we can fix these variables to one, and then pick a suitable variable in γ to fix to 0 without fixing the value of $A(x)$. We can then fix additional variables to make $A(x) = 0$ obtaining the desired contradiction.

Now suppose that W contains at least 2 affine factors and let A' be a factor distinct from A. In this case, one of A, A' and $1 + A + A'$ is a factor of W and does not depend on the first variable of γ (and some variable outside β). It follows again that we can make $x^{\beta} = 1$, $x^{\gamma} = 0$ and $W(x) = 0$, again obtaining a contradiction.

The only remaining case is when γ is of size one and W has one affine factor. In this case, by induction $\Pr[P(x) = 1]$ is at least

$$\frac{1}{4}\left(2^{1-d} + 2 \cdot \frac{1}{2}\left(2^{3-d} - 2^{5-2d}\right)\right).$$

For d at least 4 this is at least 2^{1-d} and we are done. Finally note that in the case $d = 3$, P_1' as well as the co-factor of A in W are of degree at most one and hence if they do not contain an additional factor they must be the constant 1 and the theorem follows also in this case. □

References

[1] SANJEEV ARORA, CARSTEN LUND, RAJEEV MOTWANI, MADHU SUDAN, AND MARIO SZEGEDY: Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. Preliminary version in FOCS'92. See also at ECCC. [doi:10.1145/278298.278306] 852

[2] PER AUSTRIN AND JOHAN HÅSTAD: On the usefulness of predicates. *ACM Trans. Computation Theory*, 5(1):1:1–1:24, 2013. Preliminary version in CCC'12. [doi:10.1145/2462896.2462897] 857

[3] PER AUSTRIN AND ELCHANAN MOSSEL: Approximation resistant predicates from pairwise independence. *Comput. Complexity*, 18(2):249–271, 2009. Preliminary version in CCC'08. See also at ECCC. [doi:10.1007/s00037-009-0272-6] 856

[4] MIHIR BELLARE, ODED GOLDREICH, AND MADHU SUDAN: Free bits, PCPs, and nonapproximability—towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998. Preliminary version in FOCS'95. See also at ECCC. [doi:10.1137/S0097539796302531] 848

[5] JOHAN HÅSTAD: Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. Preliminary version in STOC'97. [doi:10.1145/502090.502098] 846, 847, 849, 852, 853

[6] JOHAN HÅSTAD: Satisfying degree-d equations over $GF[2]^n$. In *Proc. 14th Internat. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'11)*, pp. 242–253. Springer, 2011. [doi:10.1007/978-3-642-22935-0_21] 845, 847

[7] JOHAN HÅSTAD, STEVEN PHILLIPS, AND SHMUEL SAFRA: A well-characterized approximation problem. *Information Processing Letters*, 47(6):301–305, 1993. Preliminary version in ISTCS'93. [doi:10.1016/0020-0190(93)90076-L] 846, 858

[8] TADAO KASAMI AND NOBUKI TOKURA: On the weight structure of Reed-Muller codes. *IEEE Trans. Inform. Theory*, 16(6):752–759, 1970. [doi:10.1109/TIT.1970.1054545] 846, 847, 858

[9] DANA MOSHKOVITZ AND RAN RAZ: Two-query PCP with subconstant error. *J. ACM*, 57(5):29:1–29:29, 2010. Preliminary version in FOCS'08. See also at ECCC. [doi:10.1145/1754399.1754402] 850, 852

[10] RAN RAZ: A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. Preliminary version in STOC'95. [doi:10.1137/S0097539795280895] 852

[11] EMANUELE VIOLA: The sum of D small-bias generators fools polynomials of degree D. *Comput. Complexity*, 18(2):209–217, 2009. Preliminary version in CCC'08. [doi:10.1007/s00037-009-0273-5] 846, 851

AUTHOR

Johan Håstad
Professor
KTH - Royal Institute of Technology
johanh@kth.se
http://www.csc.kth.se/~johanh

ABOUT THE AUTHOR

JOHAN HÅSTAD received his Bachelor of Science from Stockholm University in 1981, his Master of Science from Uppsala University in 1984, and his Ph. D. from MIT in 1986 under the supervision of Shafi Goldwasser. Johan was appointed Associate Professor at the Royal Institute of Technology in Stockholm, Sweden in 1988 and advanced to the level of Professor in 1992. He was elected a member of the Swedish Royal Academy of Sciences in 2001. He has research interests within several subareas of Theory of Algorithms and Complexity theory but has mainly focused on the approximability of NP-hard optimization problems.