Is Constraint Satisfaction Over Two Variables Always Easy?

Lars Engebretsen[1,⋆] and Venkatesan Guruswami[2,⋆⋆]

[1] Department of Numerical Analysis and Computer Science
Royal Institute of Technology
SE-100 44 Stockholm
SWEDEN

[2] University of California at Berkeley
Miller Institute for Basic Research in Science
Berkeley, CA 94720
USA

Abstract. By the breakthrough work of Håstad, several constraint satisfaction problems are now known to have the following *approximation resistance* property: satisfying more clauses than what picking a random assignment would achieve is **NP**-hard. This is the case for example for Max E3-Sat, Max E3-Lin and Max E4-Set Splitting. A notable exception to this extreme hardness is constraint satisfaction over two variables (2-CSP); as a corollary of the celebrated Goemans-Williamson algorithm, we know that every Boolean 2-CSP has a non-trivial approximation algorithm whose performance ratio is better than that obtained by picking a random assignment to the variables. An intriguing question then is whether this is also the case for 2-CSPs over larger, non-Boolean domains. This question is still open, and is equivalent to whether the generalization of Max 2-SAT to domains of size d, can be approximated to a factor better than $(1 - 1/d^2)$.

In an attempt to make progress towards this question, in this paper we prove, firstly, that a slight restriction of this problem, namely a generalization of linear inequations with two variables per constraint, *is not* approximation resistant, and, secondly, that the Not-All-Equal Sat problem over domain size d with three variables per constraint, *is* approximation resistant, for every $d \geq 3$. In the Boolean case, Not-All-Equal Sat with three variables per constraint is equivalent to Max 2-SAT and thus has a non-trivial approximation algorithm; for larger domain sizes, Max 2-SAT can be reduced to Not-All-Equal Sat with three variables per constraint. Our approximation algorithm implies that a wide class of 2-CSPs called *regular 2-CSPs* can all be approximated beyond their random assignment threshold.

⋆ Research partly performed while the author was visiting MIT with support from the Marcus Wallenberg Foundation and the Royal Swedish Academy of Sciences.

⋆⋆ Supported by a Miller Research Fellowship.

1 Introduction

In a breakthrough paper, Håstad [6] studied the problem of giving approximate solutions to maximization versions of several constraint satisfaction problems. An instance of a such a problem is given as a set of variables and a collection of constraints, i.e., functions from some domain to $\{0, 1\}$, on certain subsets of variables, and the objective is to find an assignment to the variables that satisfies as many constraints as possible. An approximate solution of a constraint satisfaction program is simply an assignment that satisfies roughly as many constraints as possible. In this setting we are interested in proving either that there exists a polynomial time algorithm producing approximate solutions, i.e., solutions that are at most some constant factor worse compared to the optimum, or that no such algorithms exist.

Typically, each individual constraint depends on a fixed number k of the variables—this case is usually called the Max k-CSP problem. The complexity of the constraint satisfaction problem (CSP) is determined by the precise set of constraints that may be posed on subsets of k variables, and accordingly we get various families of Max k-CSP problems. For each such CSP, there exists a very naive algorithm that approximates the optimum within a constant factor: The algorithm that just guesses a solution at random. In his paper, Håstad [6] proved the very surprising fact that this algorithm is essentially the best possible efficient algorithm for several constraint satisfaction problems, unless $\mathbf{P} = \mathbf{NP}$. Håstad [6] suggests that predicates with the property that the naive randomized algorithm is the best possible polynomial time approximation algorithm should be called *non-approximable beyond the random assignment threshold*; we also use the phrase *approximation resistant* to refer to the same phenomenon.

Definition 1. *A solution to a maximization problem is* α-approximate *if it is feasible and has weight at least* α *times the optimum. An approximation algorithm has* performance ratio α *if it delivers* α*-approximate solutions in polynomial time.*

Definition 2. *A CSP is said to be* approximation resistant *or* non-approximable beyond the random assignment threshold *if, for any constant* $\varepsilon > 0$*, it is* ***NP****-hard to compute a* $(\rho + \varepsilon)$*-approximate solution, where* ρ *is the expected fraction of constraints satisfied by a solution guessed uniformly at random.*

Clearly, understanding which predicates are approximation resistant is an important pursuit. The current knowledge is that for Boolean CSPs, which understandably have received the most attention so far, there is a precise understanding of which CSPs on *exactly three* variables are approximation resistant: All predicates that are implied by parity have this property [6, 11]. It is known that *no* Boolean CSP over two variables is approximation resistant; this is a corollary of the breakthrough Goemans-Williamson algorithm [4]. For the case of four or more variables, very little is known; therefore it seems to be a good approach to first understand the situation for two and three variables.

Accordingly, we are interested in the situation for CSPs with two and three variables over larger, non-Boolean, domains. In particular, it is a really intriguing question whether every CSP over two variables can be approximated better than random, no matter what the domain size is. The central aim of this paper is to study this question. We are not able to resolve it completely, but we conjecture that the answer to the question is yes.

1.1 Formal Definitions of Some CSPs

Before discussing our results, we will need to define some of the CSPs that we will be concerned with in this paper. A specific k-CSP problem is defined by the family of constraints that may be imposed on subsets of k variables. Allowing arbitrary constraints gives the most general problem, which we call Max Ek-CSP(d). In this paper, d refers to the domain size from which the variables may take values, with $d = 2$ corresponding to the Boolean case. Over domain size d, a constraint is simply a function $f\colon [d]^k \to \{0, 1\}$, where $[d] = \{0, 1, \ldots, d-1\}$. Equivalently, a constraint f can be viewed as a subset of $[d]^k$ consisting of all inputs which it maps to 1.

The Max Ek-Sat(d) problem is defined by the constraint family $\{f \subseteq [d]^k : |f| = d^k - 1\}$, i.e., the family of all constraints having just one non-satisfying assignment. Max Ek-NAE-Sat(d) is the problem where the constraints assert that the specific variables are not all equal, except that we also allow translates of variables, e.g., for the two variable case, a constraint can assert $x_1 + 1 \neq x_2 + 3$ (the addition being done modulo d); this is the analog of complementation of Boolean variables. In the Max Ek-Lin(d) problem, the constraint family is given by all linear constraints: $\{\mathrm{Lin}(\alpha_1, \ldots, \alpha_k, c) : \alpha_i, c \in [d]\}$ where $\mathrm{Lin}(\alpha_1, \ldots, \alpha_k, c) = \{(x_1, \ldots, x_k) : \sum_i \alpha_i x_i = c \bmod d\}$. The Max E$k$-LinInEq($d$) problem is defined by the family of all linear inequations: $\{f \subseteq [d]^k : f^c \text{ is a linear constraint}\}$— here $f^c = [d]^k \setminus f$ denotes the complement of the constraint f.

For the two variable case, we define the constraint satisfaction problems Max BIJ(d) and Max Co-BIJ(d) which are generalizations of Max E2-Lin(d) and Max E2-LinInEq(d) respectively. Let S_d be the set of all bijections from $[d]$ to $[d]$. For each $\pi \in S_d$, define the 2-ary constraint $f_{\pi,d} = \{(a, b) \in [d]^2 : b = \pi(a)\}$. Now define the family $\mathrm{BIJ}(d) = \{f_{\pi,d} : \pi \in S_d\}$; we call the CSP associated with this family Max BIJ(d). The problem Max Co-BIJ(d) is obtained by constraints which are complements of those in BIJ(d), i.e., a constraint is of the form $\pi(x_1) \neq x_2$ for some bijection π defined over $[d]$. It is clear that these problems generalize Max E2-Lin(d) and Max E2-LinInEq(d) respectively.

For the three variable case, we define the problem Max E3-NAE-Sat(G) for finite Abelian groups G. For each triple $(g_1, g_2, g_3) \in G^3$ define the constraint $N_{g_1,g_2,g_3} = \{(x_1, x_2, x_3) \in G^3 : \neg(g_1 x_1 = g_2 x_2 = g_3 x_3)\}$. Now define the family $\mathrm{NAE}(G) = \{N_{g_1,g_2,g_3} : (g_1, g_2, g_3) \in G^3\}$; we denote by Max E3-NAE-Sat(G) the CSP associated with this family of constraints. Note that the group structure is indeed present in the problem since the constraints involve multiplication by elements from G. In fact, we are able to prove in this paper that Max E3-

NAE-Sat($\boldsymbol{Z}_4$) is approximation resistant while we are unable to determine the approximability of Max E3-NAE-Sat($\boldsymbol{Z}_2 \times \boldsymbol{Z}_2$).

It is an interesting open question to determine what kind of hardness holds for the restricted version of Max E3-NAE-Sat(G) where group multipliers are not allowed; for this problem the group structure is, of course, not present at all. Recently, Khot [8] has shown that Max E3-NAE-Sat($\boldsymbol{Z}_3$) is approximation resistant even without group multipliers.

1.2 Our Results

Preliminaries: First, we make explicit the easily seen result that an approximation algorithm for Max E2-Sat(d) with performance ratio better than $1 - 1/d^2$, i.e., better than the random assignment threshold, implies that any CSP over 2 variables can be approximated to within better than *its respective* random assignment threshold. In other words, Max E2-Sat(d) is the hardest problem in this class, and if there is some Max E2-CSP(d) which is approximation resistant, then Max E2-Sat(d) has to be approximation resistant.

Consequently, our interest is in the approximability of Max E2-Sat(d), specifically to either find a polynomial time approximation algorithm with performance ratio greater than $1 - 1/d^2$ or to prove a tight hardness result that the trivial $1-1/d^2$ is the best one can hope for. While we are unable to resolve this question, we consider and prove results for two predicates whose difficulty sandwiches that of solving Max E2-Sat(d): namely Max Co-BIJ(d) and Max E3-NAE-Sat($\boldsymbol{Z}_d$). The former problem is (in a loose sense) the natural 2-CSP which is next in "easiness" after Max E2-Sat(d) as far as approximating better than the random assignment threshold is concerned. There is an approximation preserving reduction from Max E2-Sat(d) to Max E3-NAE-Sat($\boldsymbol{Z}_d$), implying that Max E3-NAE-Sat($\boldsymbol{Z}_d$) is a harder problem than Max E2-Sat(d).

Algorithms: For the Max Co-BIJ(d) problem, we prove that it is *not* approximation resistant by presenting a polynomial time approximation algorithm with performance ratio $1 - d^{-1} + 0.07d^{-4}$. This result implies that a large class of 2-CSPs, called *regular 2-CSPs* (defined below), are *not* approximation resistant. Viewing a 2-ary constraint C over domain size d as a subset of $[d] \times [d]$, the constraint is said to be r*-regular* if for each $a \in [d]$, $|\{x : (x, a) \in C\}| = |\{y : (a, y) \in C\}| = r$ (the term regular comes from the fact that the bipartite graph defined by C is regular). The constraint is *regular* if it is r-regular for some $1 \leq r < d$. A 2-CSP is regular if all the constraints in the CSP are regular and it is r-regular if all the constraints are r-regular.

Our result for regular 2-CSPs includes as a special case the result of Frieze and Jerrum [3] that Max d-Cut can be approximated to better than its random threshold. Our performance ratio is weaker, but our analysis is simpler and gives a more general result. Another special case is the result for Max E2-Lin(d) where our result actually improves the approximation ratio of Andersson *et al* [2]. Recently, Khot [9] gave a simpler algorithm that beats the random assignment threshold for Max E2-Lin(d) as well as the more general Max BIJ(d) problems—his result is actually more general and can find a near-satisfying assignment given a near-satisfiable instance, i.e., an instance where the optimum solution satisfies a fraction $1-\varepsilon$ of constraints. Our approximation algorithm for Max Co-BIJ(d) is based on a semidefinite programming relaxation, similar to that used for Max BIJ(d) by Khot [9], combined with a rounding scheme used by Andersson [1] to construct an approximation algorithm for Max d-Section, the generalization of Max Bisection to domains of size d. Technically, we view this algorithmic result as the main contribution of this paper.

Inapproximability results: For the Boolean case, $d = 2$, it is known that Max E3-NAE-Sat can be approximated to better than random. The GW-algorithm [4] for Max E2-Sat essentially gives such an algorithm, and the performance ratio was later improved by Zwick [12]. We prove that for larger domains, the problem becomes approximation resistant; in other words, it is **NP**-hard to approximate Max E3-NAE-Sat($\boldsymbol{Z}_d$) to better than $(1 - 1/d^2 + \varepsilon)$ for any $d \geq 3$ and any $\varepsilon > 0$. This result rules out the possibility of a non-trivial algorithm for Max E2-Sat(d) that works by reducing it to Max E3-NAE-Sat($\boldsymbol{Z}_d$). In fact, we prove that for any finite group G which is not of the form $\boldsymbol{Z}_2 \times \boldsymbol{Z}_2 \times \cdots \times \boldsymbol{Z}_2$, Max E3-NAE-Sat($G$) is hard to approximate within a factor $(1 - 1/|G|^2 + \varepsilon)$.

We remark that the above hardness results hold with *perfect completeness*; in other words, the stated approximation factors are hard to obtain even on satisfiable instances of the concerned constraint satisfaction problems.

Conclusions: We are not able to completely resolve the status of the 2-CSP problem over larger domains. Using reductions, we prove a hardness result of $1 - \Omega(1/d^2)$ for Max E2-Sat(d), which compares reasonably well with the $(1 - 1/d^2)$ random assignment threshold. For satisfiable instances of Max E2-Sat(d), we prove a hardness result of $1 - \Omega(1/d^3)$. Nevertheless, we conjecture that there is an approximation algorithm beating the random assignment threshold for Max E2-Sat(d), and hence for all instances of 2-CSP.

Organization: We begin with a brief Section 2 highlighting why Max E2-Sat(d) is the hardest Max E2-CSP(d) problem in terms of beating the random assignment threshold. Next, in Section 3 we prove that every Max Co-BIJ(d) problem admits an algorithm that beats the random assignment threshold, and record some of its consequences. In Section 4, we prove that Max E3-NAE-Sat(G) is approximation resistant for most groups, including $G = \boldsymbol{Z}_d$ (the case of most interest in the context of Max E2-Sat(d)). Finally, we record results that directly apply to Max E2-Sat(d) in Section 5.

2 The "Universality" of Max E2-Sat(d)

We note that the existence of an approximation algorithm that beats the random threshold for every 2-CSP is equivalent to the existence of such an algorithm for

Max E2-Sat(d). Thus, an algorithm for Max E2-Sat(d) with performance ratio better than $(1-1/d^2)$ will imply that no 2-CSP is approximation resistant, thus resolving our conjecture that every 2-CSP is "easy".

This claim is seen by a "gadget" reducing an arbitrary CSP(d) to Max E2-Sat(d). Given an instance of any 2-CSP, construct an instance of Max E2-Sat(d) by repeating the following for every constraint C in the original 2-CSP: For every non-satisfying assignment to C, add one 2SAT(d) constraint which has precisely this non-satisfying assignment. If an assignment satisfies C then it also satisfies all the 2SAT(d) constraints in the gadget, and otherwise it satisfies precisely all but one of the 2SAT(d) constraints. Using this fact, it is straightforward to show that if Max E2-Sat(d) can be approximated beyond the random threshold, the above procedure gives an approximation algorithm that approximates an arbitrary 2-CSP beyond the random threshold. Conversely, if any 2-CSP at all is approximation resistant, then Max E2-Sat(d) must be approximation resistant.

3 Approximation Algorithm for Max Co-BIJ(d)

To construct an approximation algorithm for Max Co-BIJ(d) we combine a modification of the semidefinite relaxation used by Khot [9] for the Max BIJ(d) problem with a modification of the randomized rounding used by Andersson [1] for the Max d-Section problem. Recall that a specific clause in the Max Co-BIJ(d) problem is of the form (x, x', π), where x and x' are variables in the Max Co-BIJ(d) instance and π is a permutation, and that the clause is satisfied unless $x = j$ and $x' = \pi(j)$ for some j. In our semidefinite relaxation of Max Co-BIJ(d) there are d vectors $\{u_0^x, \ldots, u_{d-1}^x\}$ for every variable x in the Max Co-BIJ(d) instance. Intuitively, the vector u_j^x sets the value of the variable x to j. It is straightforward to see that the semidefinite program in Fig. 1 and that the *barycenter* $b = \frac{1}{d}\sum_{j=0}^{d-1} u_j^x$ is independent of x for any feasible solution to the program. To establish a bound on the performance ratio of the algorithm in Fig. 2, we use local analysis:

Lemma 1. *For any clause (x, x', π) in the Max Co-BIJ(d) instance, the algorithm in Fig. 2 satisfies (x, x', π) with probability at least*

$$\left(1 - \sum_{j=0}^{d-1} \langle u_j^x, u_{\pi(j)}^{x'} \rangle \right) \int_B \left(1 - \frac{1}{d} + \frac{K^2 r_1^2}{d}\right) dP(r)$$

where K is any positive constant, the vectors u_j^x and $u_{j'}^{x'}$ are as described in the algorithm, $B = \{r \in \boldsymbol{R}^{2d} : |r| \le 1/Kd\}$, and P is the probability distribution of a $2d$-dimensional Gaussian with mean zero and identity covariance matrix.

Proof. Consider an arbitrary clause (i, i', π) and the corresponding values q_{xj} computed by the algorithm. Let $B = \{r \in \boldsymbol{R}^{2d} : |r| \le 1/Kd\}$. When $r \in B$, both q_{xj} and $q_{x'j}$ are in the interval $[0, 2/d]$; hence the clause (i, i', π) is satisfied

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{x,x',\pi} w_{x,x',\pi}\left(1-\sum_{j=0}^{d-1}\langle u_j^x, u_{\pi(j)}^{x'}\rangle\right) & \\
\text{such that} & \langle u_j^x, u_{j'}^{x'}\rangle \geq 0 & \forall x \in X, x' \in X, j \in \boldsymbol{Z}_d, j' \in \boldsymbol{Z}_d \\
& \langle u_j^x, u_{j'}^{x}\rangle = 0 & \forall x \in X, (j,j') \in \boldsymbol{Z}_d^2 : j \neq j' \\
& \displaystyle\sum_{j=0}^{d-1}\langle u_j^x, u_j^x\rangle = 1 & \forall x \in X \\
& \displaystyle\sum_{j=0}^{d-1}\sum_{j'=0}^{d-1}\langle u_j^x, u_{j'}^{x'}\rangle = 1 & \forall (x,x') \in X^2 : x \neq x'
\end{array}
$$

Fig. 1. Semidefinite relaxation of Max Co-BIJ(d) with variable set X. A clause in the Max Co-BIJ(d) instance is denoted by (x, x', π) where $x \in X$ and $x' \in X$ are variables and $\pi\colon \boldsymbol{Z}_d \to \boldsymbol{Z}_d$ is a permutation. The clause is satisfied unless $x = j$ and $x' = \pi(j)$ for some $j \in \boldsymbol{Z}_d$. Each clause (x, x', π) has a non-negative weight $w_{x,x',\pi}$ associated with it.

1. Solve the semidefinite program in Fig. 1.
2. Denote by u_j^x the vectors obtained from the solution.
3. For every $(x, j) \in X \times \boldsymbol{Z}_d$, let $v_j^x = u_j^x - \frac{1}{d}\sum_{j=0}^{d-1} u_j^x$.
4. Select r from a dn-dimensional Gaussian distribution.
5. Set $q_{xj} = \frac{1}{d} + K\langle r, v_j^x\rangle$ for all $(x, j) \in X \times \boldsymbol{Z}_d$.
6. For each $x \in X$,
   - set $p_{xj} = q_{xj}$ if $q_{xj} \in [0, 2/d]$ for all $j \in \boldsymbol{Z}_d$;
   - set $p_{xj} = 1/d$ for all $j \in \boldsymbol{Z}_d$ otherwise.
7. For each $x \in X$, let $x = j$ with probability p_{xj}.

Fig. 2. Approximation algorithm for Max Co-BIJ(d) with variable set X. The algorithm is parameterized by the positive constant K.

with probability

$$
\begin{aligned}
1-\sum_{j=0}^{d-1} p_{xj}p_{x',\pi(j)} &= 1-\sum_{j=0}^{d-1}\left(\frac{1}{d}+K\langle r, v_j^x\rangle\right)\left(\frac{1}{d}+K\langle r, v_{\pi(j)}^{x'}\rangle\right) \\
&= 1-\frac{1}{d}-K^2\sum_{j=0}^{d-1}\langle r, v_j^x\rangle\langle r, v_{\pi(j)}^{x'}\rangle
\end{aligned}
$$

given r in this case. Using the definition of v_j^x from the algorithm, and integrating over B, we can lower bound the probability that the clause (i, i', π) is accepted by

$$
\int_B \left(1-\frac{1}{d}+K^2\left(d\langle r, b\rangle\langle r, b\rangle - \sum_{j=0}^{d-1}\langle r, u_j^x\rangle\langle r, u_{\pi(j)}^{x'}\rangle\right)\right) dP(r).
$$

where b is the barycenter of the vectors $\{u_j^x : j \in \boldsymbol{Z}_d\}$, which is independent of x. To compute the integral of $\langle r, b\rangle\langle r, b\rangle$, introduce an orthonormal basis $\{e_k\}$ such that $b = e_1/d$ and write $r = \sum_k r_k e_k$ in this basis. Then $\int_B \langle r, b\rangle\langle r, b\rangle\, dP(r) = \frac{1}{d^2}\int_B r_1^2\, dP(r)$, where the last integral is actually independent of the basis since both P and B are spherically symmetric. To compute the integral of $\langle r, u_j^x\rangle\langle r, u_{\pi(j)}^{x'}\rangle$, we proceed similarly: Introduce an orthonormal basis $\{e_k\}$ such that $u_j^x = x_1 e_1$ and $u_{\pi(j)}^{x'} = y_1 e_1 + y_2 e_2$. Then

$$\int_B \langle r, u_j^x\rangle\langle r, u_{\pi(j)}^{x'}\rangle\, dP(r) = x_1 y_1 \int_B r_1^2\, dP(r) = \langle u_j^x, u_{\pi(j)}^{x'}\rangle \int_B r_1^2\, dP(r).$$

To conclude, we can write the probability that the clause is satisfied as $a - cx$ where

$$\begin{aligned} a &= \int_B \left(1 - \frac{1}{d} + \frac{K^2 r_1^2}{d}\right) dP(r), \\ c &= K^2 \int_B r_1^2\, dP(r), \\ x &= \sum_{j=0}^{d-1} \langle u_j^x, u_{\pi(j)}^{x'}\rangle. \end{aligned}$$

Since $a \geq c > 0$, $a - cx > a(1-x)$; therefore the clause is satisfied with probability at least $a(1-x)$, which equals

$$\left(1 - \sum_{j=0}^{d-1} \langle u_j^x, u_{\pi(j)}^{x'}\rangle\right) \int_B \left(1 - \frac{1}{d} + \frac{K^2 r_1^2}{d}\right) dP(r).$$

Using standard calculus, it can be shown that

$$\int_B \left(1 - \frac{1}{d} + \frac{K^2 r_1^2}{d}\right) dP(r) \geq 1 - \frac{1}{d} + \frac{0.07}{d^4}$$

with the parameter choice $K = 1/\sqrt{13d^3}$. Together with the above lemma, this proves our main theorem:

Theorem 1. *The algorithm in Fig. 2 with $K = 1/\sqrt{13d^3}$ is a randomized polynomial time approximation algorithm for Max Co-BIJ(d) with expected performance ratio $1 - d^{-1} + 0.07d^{-4}$ and thus better than the random assignment threshold.*

3.1 An Approximation Algorithm for Max E2-Lin(d)

We can use the above algorithm for Max Co-BIJ(d) to construct an algorithm also for Max E2-Lin(d): Simply replace an equation $ax + by = c$ with the $d - 1$ inequations $ax + by \neq c_i$ for all $c_i \neq c$. Then an assignment that satisfies a linear equation satisfies all of the corresponding linear inequations and an assignment that does not satisfy a linear equation satisfies $d-2$ of the $d-1$ corresponding linear equations.

Run the following two algorithms and take the assignment producing the largest weight as the result: The first algorithms selects a random assignment to the variables; the second algorithm runs the above algorithm for Max E2-Lin(d). This algorithm gives a performance ratio of $1/d+\Omega(1/d^4)$ which improves significantly on the previously best known ratio of $1/d+\Omega(1/d^{14})$ [2]. The formal proof is omitted from this extended abstract.

Theorem 2. *For all $d \geq 4$, the above algorithm is a randomized polynomial time approximation algorithm for Max E2-Lin(d) with expected performance ratio $d^{-1} + 0.05d^{-4}$.*

Corollary 1. *For all $d \geq 2$, there is a polynomial time approximation algorithm for Max E2-Lin(d) with expected performance ratio $d^{-1}+0.05d^{-4}$ and thus better than the random assignment threshold.*

Proof. Algorithms for $d = 2$ and $d = 3$ have been provided by Goemans and Williamson [4, 5], for $d \geq 4$ the result follows by Theorem 2

3.2 An Approximation Algorithm for Regular 2-CSPs

We can obtain an approximation algorithm for all regular CSPs by a straightforward generalization of the ideas from the previous section. Given an r-regular 2-CSP, we proceed as follows for every relation R defining the CSP: Decompose R^c, the "bipartite complement" of the graph defined by R, into $(d-r)$ perfect matchings $\pi_R^1, \pi_R^2, \ldots, \pi_R^{d-r}$. Then let these matchings define the Max Co-BIJ(d) instance. An assignment that satisfies R satisfies all of the $d-r$ matchings while an assignment that does not satisfy R satisfies $d-r-1$ of them. Run the following two algorithms and take the assignment producing the largest weight as the result: The first algorithms selects a random assignment to the variables; the second algorithm runs the above algorithm for Max Co-BIJ(d). The formal proof is, again, omitted.

Theorem 3. *For all $d \geq 2$ and all $1 \leq r \leq d-1$, the above algorithm is a randomized polynomial time approximation algorithm for r-regular CSPs with expected performance ratio $r/d + \Omega(d^{-4})$.*

It is not necessary that the various constraints be r-regular for the same r, and a similar argument also shows that every regular 2-CSP can be approximated in polynomial time beyond its random assignment threshold.

4 Hardness Results for Max E3-NAE-Sat($\boldsymbol{Z}_d$)

In this section, our aim is to prove that unlike the Boolean case, for every $d \geq 3$, Max E3-NAE-Sat($\boldsymbol{Z}_d$) is approximation resistant. We will actually prove that Max E3-NAE-Sat(G) is approximation resistant for pretty much every finite Abelian group. Specifically, we will prove:

Theorem 4. *For every constant $\varepsilon > 0$ and every finite Abelian group G that is not isomorphic to $\mathbf{Z}_2^m$ for any positive integer m, it is* **NP***-hard to distinguish instances of Max E3-NAE-Sat(G) that are satisfiable from instances where at most a fraction $(1 - |G|^{-2} + \varepsilon)$ of the constraints are simultaneously satisfiable.*

Inevitably, the proof of the above theorem will involve the machinery of probabilistically checkable proofs (PCP). We first give a rapid introduction to the necessary background, and then indicate the kind of PCP verifiers that yield Theorem 4.

4.1 Background on PCP constructions

Our hardness of approximation results are proved by constructions of suitable probabilistically checkable proofs (PCPs) for **NP**. Our PCP constructions follow the by now standard paradigm used by Håstad; we refer the reader to Håstad's paper [6] for the complete details. The basic construction methodology is to start with an instance of 3SAT, called μ-gap E3-Sat(5), where there is a gap in the optimum (either the formula is satisfiable or at most a fraction μ of clauses are satisfied by any assignment) and each variable occurs in exactly five clauses. One then uses Raz's parallel repetition theorem to obtain a two-prover one-round (2P1R) system with completeness 1 and arbitrarily low soundness. The verifier in the 2P1R system picks a set W of u clauses of the 3SAT instance at random and also picks a set U of u variables, one in each of the picked clauses, at random. The first prover P_1 is expected to give an assignment to the variables in U and the second prover P_2 is expected to give an assignment that satisfies all the clauses in W. The verifier accepts if the answer of the P_2 is consistent with the answer of P_1. This protocol has soundness at most c^u for some absolute constant c.

In order to prove that several constraint satisfaction programs are non-approximable beyond the random assignment threshold, we use a verifier whose acceptance predicate is closely related to the particular CSP we want to analyze. The final verifier expects as proof encodings of the answers of P_1 and P_2 in the Raz 2P1R, and then checks very efficiently, by making very few queries, that the proof is close to valid encodings of answers that would have made the 2P1R verifier accept with good probability. To get hardness results for CSPs over domain size d, the specific encoding used is the *Long G-Code* where G is an Abelian group of order d.

Definition 3. *Let U be a set of variables and denote by $\{-1,1\}^U$ the set of assignments to the variables in U. The* long G-code of some assignment x to the variables in U *is a function $A_{U,x}\colon \{-1,1\}^U \to G$ defined by $A_{U,x}(f) = f(x)$.*

Definition 4. *Let W be a set of clauses and denote by SAT^W the set of satisfying assignments to the clauses in W. The* long G-code of some satisfying assignment y to the clauses in W *is a function $A_{W,y}\colon \mathrm{SAT}^W \to G$ defined by $A_{W,y}(h) = h(y)$.*

The proof is a standard written G-proof with parameter u:

The verifier acts as follows:

1. Select a sequence W of clauses, each clause uniformly and independently at random from Φ.
2. Select a sequence U of variables by selecting one variable from each clause in W, uniformly and independently.
3. Let $\pi\colon \mathrm{SAT}^W \to \{-1,1\}^U$ be the function that creates an assignment in $\{-1,1\}^U$ from an assignment in SAT^W.
4. Let $F = G^{\{-1,1\}^U}$ and $H = G^{\mathrm{SAT}^W}$.
5. Select $f \in F$ and $h \in H$ uniformly at random.
6. Select $e \in H$ such that independently for every $y \in \mathrm{SAT}^W$, $e(y)$ is uniformly distributed in $G \setminus \{\mathbf{1}\}$.
7. Accept if $A_U(f)$, $A_W(h)$ and $A_W((f \circ \pi)^{-1}h^2e)$ are not all equal; Reject otherwise.

Fig. 3. The PCP used to prove optimal approximation hardness for Max E3-NAE-Sat($\boldsymbol{Z}_d$) for odd d. The PCP is parameterized by the constant u and tests if a μ-gap E3-Sat(5) formula Φ is satisfiable.

The proof expected by the PCP verifier now consists of purported Long G-Codes of the assignments to the u variables in U and the u clauses in W for each possible choice U, W of the 2P1R verifier. We call such a proof in such a format the *Standard Written G-proof with parameter u*.

The PCP design task now reduces to designing an "inner" verifier to check if two purported Long G-Codes encode assignments which are consistent answers for P_1 and P_2 in the 2P1R. One designs such a verifier with an acceptance predicate closely tied to the problem at hand, and its performance is analyzed using Fourier analysis. The basic strategy here is to show how proofs that make the "inner" verifier accept with high probability can be used to extract good strategies for P_1 and P_2 in the 2P1R protocol.

4.2 Intuition behind our PCP constructions

For lack of space, we defer the analysis of our PCPs to the full version of our paper and only give the intuition behind the constructions here. The first construction turns out to work for all Abelian groups of odd order. A verifier in a PCP typically first selects sets U and W uniformly at random and then checks a small number of positions in tables corresponding to U and W. Specifically, the standard way to get a PCP with three queries is to query one position in a table corresponding to U and two positions in a table corresponding to W. The three values obtained are then tested to see if they satisfy some given constraint—such a construction gives a hardness result for the CSP corresponding to the type of constraint checked. To get a hardness result for Max E3-NAE-Sat(G) the constraint checked by the verifier therefore has to be a not-all-equal constraint. Moreover, we want the verifier to have perfect completeness, i.e., to always verify

The proof is a standard written $\boldsymbol{Z}_d$-proof with parameter u:

The verifier acts as follows:

Steps 1–5 are as in Fig. 3 applied to $G = \boldsymbol{Z}_d$.

6. Select $e_1 \in H$ such that independently for every $y \in \mathrm{SAT}^W$, $e_1(y)$ is uniformly distributed in $\{\omega^{4i}, \omega^{4i+1}\}_{i=0}^{d/4-1}$.
   Select $e_2 \in H$ such that independently for every $y \in \mathrm{SAT}^W$, $e_2(y)$ is uniformly distributed in $\{\omega^{4i+1}, \omega^{4i+2}\}_{i=0}^{d/4-1}$.
   Let $e = e_1 e_2$.
7. Accept if $A_U(f)$, $A_W(h)$ and $A_W((f \circ \pi)^{-1} h^2 e)$ are not all equal; Reject otherwise.

Fig. 4. The PCP used to prove optimal approximation hardness for Max E3-NAE-Sat($\boldsymbol{Z}_d$) where $d = 2^m$ for integers $m \geq 2$. The group $\boldsymbol{Z}_d$ is represented by $\{\omega^i\}_{i=0}^{d-1}$ where $\omega = e^{2\pi i/d}$ and multiplication is the group operator. The PCP is parameterized by the constant u and tests if a μ-gap E3-Sat(5) instance Φ is satisfiable.

The proof is a standard written G-proof with parameter u:

The verifier acts as follows:

Steps 1–5 are as in Fig. 3.

6. Select e by selecting independently the components of $e(y)$ according to Step 6 in Figs. 3 and 4, respectively.
7. Accept if $A_U(f)$, $A_W(h)$ and $A_W((f \circ \pi)^{-1} h^2 e)$ are not all equal; Reject otherwise.

Fig. 5. The PCP used to prove optimal approximation hardness for Max E3-NAE-Sat(G) for any finite Abelian group $G \cong G_o \times \boldsymbol{Z}_{2^{\alpha_1}} \times \boldsymbol{Z}_{2^{\alpha_2}} \times \cdots \times \boldsymbol{Z}_{2^{\alpha_s}}$ where G_o is a finite Abelian group of odd order and $\alpha_i > 1$ for all i. The PCP is parameterized by the constant u and tests if a μ-gap E3-Sat(5) formula Φ is satisfiable.

a correct proof. We accomplish this by querying the positions $A_U(f)$, $A_W(h)$ and $A_W(f^{-1}h^2e)$ where f and h are selected uniformly at random and e is selected such that $e(y)$ is selected independently and uniformly at random from $G \setminus \{\mathbf{1}\}$ (see Figure 3). Here, the function $f^{-1}h^2e$ is the map $y \mapsto (f(y|_U))^{-1}(h(y))^2 e(y)$. For a correct proof of a satisfying assignment, the answers to these queries will be $f(y|_U)$, $h(y)$ and $(f(y|_U))^{-1}(h(y))^2 e(y)$ where y is a satisfying assignment to the clauses in W. These three values can never be all equal, since $f(y|_U) = h(y)$ implies that $(f(y|_U))^{-1}(h(y))^2 e(y) = h(y)e(y) \neq h(y)$. Therefore the verifier always accepts a correct proof and we prove in the full version of our paper that the verifier accepts a proof corresponding to an unsatisfying assignment with probability at most $1 - |G|^{-2} + \varepsilon$, where $\varepsilon > 0$ is an arbitrary constant.

To obtain results for the Abelian groups $\boldsymbol{Z}_d$, where $d = 2^m$ for some $m \geq 2$, we need to change the verifier slightly. The reason that $|G|$ being odd was very useful in the above mentioned analysis is the following: For a random function h with range in G, h^2 is also a random function when $|G|$ is odd. This is no longer the case when $|G|$ is even. We get around this obstacle by a different, clever choice of the error function e. The PCP verifier is described in Figure 4. This verifier also has perfect completeness and we prove in the full version of our paper that it has soundness $1 - |G|^{-2} + \varepsilon$, where $\varepsilon > 0$ is an arbitrary constant.

Finally, in order to obtain a hardness result for every Abelian group except powers of $\boldsymbol{Z}_2$, we construct a verifier that essentially runs several of the verifiers described previously in parallel. This verifier is described in Fig. 5. We prove in the full version of our paper that the analysis of the previous two types of verifiers can be combined into an analysis that works in this case, which establishes the proof of Theorem 4.

5 The Status of Max E2-Sat(d)

For the Max E2-Sat(d) problem itself, we present some hardness results below. We first prove a result for domain size 3 and then prove a result for general domains.

Lemma 2. *For every constant $\varepsilon > 0$, the predicate $(x \neq a) \vee (y = b)$ over domains of size 3 is hard to approximate within $(23/24 + \varepsilon)$ with perfect completeness.*

Proof. Consider the following 2P1R interactive proof system for 3SAT: The first prover is given a variable and returns an assignment to that variable, the second prover is given a clause and returns an index of a literal that makes the clause satisfied. The verifier selects a clause at random, then a variable in the clause at random, sends the variable to P_1, the clause to P_2 and accepts unless P_2 returns the index of the variable sent to P_1 and P_1 returned an assignment that does not satisfy the literal. It is known that there are satisfiable instances of 3SAT such that it is **NP**-hard to satisfy more than $7/8 + \varepsilon$ of the clauses, for any constant $\varepsilon > 0$ [6]. When the above protocol is applied to such an instance of 3SAT, the test has perfect completeness and soundness $(1 - 1/24 + \varepsilon)$. To obtain the hardness for the claimed constraint satisfaction problem, we just use the following reduction: x specifies the name of a clause, y specifies a variable in this clause, a specifies the location of y in x (encoded as 0,1 or 2), and b specifies a Boolean assignment to y (encoded over 0,1,2, where 2 is meaningless).

Theorem 5. *For every constant $\varepsilon > 0$, it is **NP**-hard to approximate Max E2-Sat(3) within $47/48 + \varepsilon$ with perfect completeness.*

Proof. Follows from Lemma 2 since $(x \neq a) \vee (y = b)$ can be written as a two E2-Sat(3) clauses.

Theorem 6. *For every $d \geq 3$ and every constant $\varepsilon > 0$, Max E2-Sat(d) is hard to approximate within a factor of $(1 - d^{-4} + \varepsilon)$ with perfect completeness.*

Proof. We reduce Max E3-Sat(d), which is known to be hard to approximate within $(1 - d^{-3} + d\varepsilon)$ with perfect completeness to Max E2-Sat(d). A constraint SAT(x, y, z), which requires that at least one of x, y, z does not equal

0, is replaced with the constraints $\mathrm{SAT}(x,t)$, $\mathrm{SAT}(x,t+1)$, $\mathrm{SAT}(x,t+2)$, $\ldots$, $\mathrm{SAT}(x,t+d-3)$, $\mathrm{SAT}(y,t+d-2)$, $\mathrm{SAT}(z,t+d-1)$, where t is an auxiliary variable specific to this constraint and the additions are done modulo d. If all d 2SAT clauses are satisfied, the 3SAT clause has to be satisfied; if the 3SAT clause is not satisfied we can satisfy $d-1$ of the 2SAT clauses. Therefore it is hard to distinguish the case when all the constraints are satisfied from the case when a fraction $\frac{1}{d}(d(1-d^{-3}+d\varepsilon)+(d-1)(d^{-3}-d\varepsilon)) = (1-d^{-4}+\varepsilon)$ of the constraints are satisfied.

Theorem 7. *Max E2-Sat(d) is hard to approximate within a factor $1-\Omega(d^{-2})$ with non-perfect completeness. It is also hard to approximate within a factor $1-\Omega(d^{-3})$ with perfect completeness for all $d \geq 3$.*

Proof. We reduce d-CUT, which is hard to approximate within $1-1/34d+\varepsilon$ with non-perfect completeness [7], to Max E2-Sat(d). A clause $\mathrm{CUT}(x,y)$ is replaced with the clauses $\mathrm{SAT}(x+i,y+i)$ for all i from 0 to $d-1$. A d-CUT instance with n constraints corresponds to a 2SAT(d) instance with dn constraints and an assignment satisfying all but k 2SAT(d) constraints satisfies all but k d-CUT constraints. The hardness result with perfect completeness follows by a reduction from Max 3-CUT on 3-colorable graphs (i.e., Max 3-CUT with perfect completeness), which is known to hard to approximate within a factor γ for some absolute constant $\gamma < 1$ [10], to Max d-CUT for $d \geq 3$. Such a reduction which preserves perfect completeness and shows the hardness of approximating Max d-CUT within $1-\Omega(1/d^2)$ exists. Combining with the above gadget that reduces d-CUT to 2SAT(d), we get the claimed $1-\Omega(1/d^3)$ hardness for satisfiable instances of Max E2-Sat(d) for all $d \geq 3$. We omit the details.

The result of Theorem 6 is not entirely subsumed by the result of Theorem 7 for satisfiable instances, since the constant in front of $1/d^3$ implies that the result of Theorem 6 will actually be stronger for small values of d. An interesting question is whether a factor $(1-\Omega(d^{-2}))$ hardness can be shown for satisfiable instances of Max E2-Sat(d).

Conjecture. Although we did not completely resolve the status of Max E2-Sat(d), we conjecture at this point that the problem is *not* approximation resistant.

Acknowledgments

We would like to thank Subhash Khot for providing us with a copy of [9]; his algorithm for Max BIJ(d) directly inspired our algorithm for Max Co-BIJ(d).

References

1. Gunnar Andersson. *Some New Randomized Approximation Algorithms.* Doctoral dissertation, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, May 2000.

2. Gunnar Andersson, Lars Engebretsen, and Johan Håstad. A new way of using semidefinite programming with applications to linear equations mod p. *Journal of Algorithms*, 39(2):162–204, May 2001.
3. Alan Frieze and Mark Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18:67–81, 1997.
4. Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.
5. Michel X. Goemans and David P. Williamson. Approximation algorithms for Max-3-Cut and other problems via complex semidefinite programming. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 443–452. Hersonissos, Crete, Grece, 6–8 July 2001.
6. Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, July 2001.
7. Viggo Kann, Sanjeev Khanna, Jens Lagergren, and Alessandro Panconesi. On the hardness of approximating Max k-Cut and its dual. *Chicago Journal of Theoretical Computer Science*, 1997(2), June 1997.
8. Subhash Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. To appear in *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*. Vancouver, Canada, 16–19 November 2002.
9. Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775. Montréal, Québec, Canada, 19–21 May 2002.
10. Erez Petrank. The hardness of approximation: Gap location. *Computational Complexity*, 4(2):133–157, 1994.
11. Uri Zwick. Approximation algorithms for constraint satisfaction programs involving at most three variables per constraint. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210. San Francisco, California, 25–27 January 1998.
12. 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 Thirty-First Annual ACM Symposium on Theory of Computing*, pages 679–687. Atlanta, Georgia, 1–4 May 1999.