Hardness of Max 3SAT with No Mixed Clauses

Venkatesan Guruswami[*]
Dept. of Computer Science and Engineering
University of Washington
Seattle, WA
venkat@cs.washington.edu

Subhash Khot
College of Computing
Georgia Institute of Technology
Atlanta, GA
khot@cc.gatech.edu

Abstract

We study the complexity of approximating Max NM-E3SAT, *a variant of* Max 3SAT *when the instances are guaranteed to not have any mixed clauses, i.e., every clause has either all its literals unnegated or all of them negated. This is a natural special case of* Max 3SAT *introduced in [7], where the question of whether this variant can be approximated within a factor better than 7/8 was also posed. We prove that it is NP-hard to approximate* Max NM-E3SAT *within a factor of $7/8 + \varepsilon$ for arbitrary $\varepsilon > 0$, and thus this variant is no easier to approximate than general* Max 3SAT. *The proof uses the technique of multilayered PCPs, introduced in [3], to avoid the technical requirement of* folding *of the proof tables. Circumventing this requirement means that the PCP verifier can use the bits it accesses without additional negations, and this leads to a hardness for* Max 3SAT *without any mixed clauses.*

1 Introduction

We study the approximability of a natural variant of the satisfiability problem where all clauses are restricted to have either all literals unnegated or all of them negated. We refer to the version of Max SAT where all clauses have at most k literals and none of the clauses have both negated and unnegated literals as Max NM-kSAT (here NM-SAT stands for non-mixed satisfiability). The version where all the clauses have exactly k literals will be referred to as Max NM-EkSAT.

This problem appears to be a fairly natural variant of Max SAT, but was studied only recently in [7], where the question of resolving its approximability was posed. Our main result in this paper is a tight inapproximability result for Max NM-E3SAT: we show that it is NP-hard to approximate within any factor better than 7/8, and the hardness holds even when the input instance is satisfiable.[1]

[1]Throughout this paper we deal only with maximization problems and

Such a tight inapproximability result for Max NM-EkSAT for $k \geq 4$ was already known, as we now explain. Define Max Ek-Set Splitting to be the following problem: given a k-uniform hypergraph, 2-coloring it so that a maximum number of hyperedges are not monochromatic. Consider the following reduction from Max Ek-Set Splitting to Max NM-EkSAT: replace a constraint requiring that $x_1, x_2, \ldots, x_k$ be split by the two clauses $(x_1 \vee x_2 \vee \cdots \vee x_k)$ and $(\bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k)$. For $k \geq 4$ and arbitrary $\varepsilon > 0$, together with the tight factor $(1 - 2^{-k+1} + \varepsilon)$ hardness result for approximating Max Ek-Set Splitting [8, 7], this yields a factor $1 - 2^{-k} + \varepsilon$ inapproximability bound for Max NM-EkSAT. This bound is tight since a random assignment satisfies a fraction $(1 - 2^{-k})$ of the clauses in expectation. For $k = 3$, a tight hardness result for Max E3-Set Splitting is not known, and in fact Max E3-Set Splitting can be approximated better than the factor $3/4$ achieved by simply picking a random 2-coloring. Therefore, one cannot obtain a tight result for Max NM-E3SAT in this way.

A factor $(7/8 + \varepsilon)$ inapproximability result for Max 3SAT was shown in the celebrated work of Håstad [8], but this bound does not apply for Max NM-E3SAT. However, by modifying the underlying PCP for Max 3SAT from [8] and using a further simple reduction, it was shown in [7] that Max NM-3SAT and Max NM-E3SAT are hard to approximate within factors better than $11/12$ and $15/16$ respectively.

There is a basic reason why the PCP techniques of [8] cannot produce a non-mixed instance of 3SAT. The proof in the PCP for 3SAT is divided into two disjoint parts say A and B, corresponding to answers of the two provers in an "outer" 2-prover 1-round proof system. Each 3SAT constraint checked by the verifier queries one variable from part A, and two from the other part B. Therefore, if none of the 3SAT clauses checked by the verifier has both negated and unnegated literals, an assignment which sets all variables in

part A to False and all variables in part B to True will satisfy all the 3SAT constraints. This "bipartite" structure of the queries prevents one from getting a tight hardness result for Max NM-E3SAT in this way.

We get over this predicament using *multilayered PCPs*, which were first introduced in [3] for proving hardness results for hypergraph vertex cover. Here the PCP verifier expects proofs which are split not into two parts but rather L parts (or layers) for rather large L. There are constraints between every pair of layers which resemble the constraints for the two layered PCP. This gets around the above-mentioned bipartiteness problem — for example, for assignments which are constant on layers, there must be at least half the layers which get assigned the same value and then a good fraction (in fact $1/2$) of the constraints just amongst those variables will be violated. In the limit of large L, we are able to prove that the resulting PCP has soundness as close to $7/8$ as we desire, giving a tight hardness result for approximating Max NM-E3SAT. Another way to view the role of multilayered PCPs in our construction is that it enables working with proof tables (purported long codes) that are not *folded*. Folding is a technical requirement in certain PCP constructions, and it corresponds to negations of variables in the corresponding constraint satisfaction problem for which inapproximability is being shown. Further discussion of folding appears in Section 2.3.

The technique of multilayered PCPs has also been used for showing hardness results for coloring 3-uniform hypergraphs [4, 9]. These results also need a "weak expansion property" of the underlying constraint graph which is not important for our application here. In this respect, our result is similar to the tight factor $(1-1/k^2)$ inapproximability result in [9], when $k \geq 3$, for the Max k-colorable 3-uniform hypergraph problem. In this paper, we demonstrate the utility of multilayered PCPs in a particularly simple setting: the complexity of Boolean Satisfiability. Thus, our result might serve as a good introduction to multilayered PCPs and perhaps spur its use for studying other constraint satisfaction problems like Max E3-Set Splitting whose exact approximability remains unknown.

We also use the smooth projections idea from [9] to present a simpler soundness analysis than in [8]. Our PCP has perfect completeness, which implies that it is NP-hard to satisfy more than fraction $7/8$ of clauses in a *satisfiable* Max NM-E3SAT instance. If we do not insist on perfect completeness, then a simpler proof that does not need smooth projections (but still relies on multilayered PCPs) can be given.[2]

We conclude the introduction with the following comment. In general, it seems to be an intriguing question

---

[2]This is similar to the simpler proof in Håstad's paper [8] for Max 3SAT without perfect completeness.

when a certain inapproximability result also holds for the version of the constraint satisfaction problem with the restriction that the constraints are non-mixed (i.e., do not have both negated and unnegated literals). For example, the non-mixed version of Max NAE-3SAT is precisely the problem Max E3-Set Splitting, and the best known inapproximability factor for the latter is $19/20 + \varepsilon$ which is weaker than the factor $15/16 + \varepsilon$ known for Max NAE-3SAT [7]. A similar situation holds for Max E2-LIN2 (maximizing the number of satisfied linear equations in two variables mod 2), whose (only interesting) non-mixed variant is the Max CUT problem. Max E2-LIN2 is known to be hard to approximate within $11/12 + \varepsilon$ while the bound for Max CUT is $16/17 + \varepsilon$ [8, 12]. On the algorithmic side, however, the existing approximation algorithms do not take advantage of any restrictions on negations (like non-mixed constraints), at least for Boolean constraint satisfaction problems. This is the case since they are typically semidefinite programming (SDP) based, and negations "come for free" in SDP relaxations (if vector v_i represents a variable x_i, then $-v_i$ represents its negation $\overline{x}_i$). For instance, in the above-mentioned examples, the best known approximation factor achievable in polynomial time for Max E2-LIN2 and Max CUT is the same (achieved by the Goemans-Willamson algorithm [6]), and this is the case for Max NAE-3SAT and Max E3-Set Splitting as well [13, 14]. The recent work [10] on hardness results for Max CUT and other two variable constraint satisfaction problems based on the Unique Games conjecture, indicates that the restriction on negations does *not* help in terms of the approximability factor for these problems. Our work can then be viewed as one instance of this general phenomenon where we are in fact able to use some recent advances in PCPs to *unconditionally* establish that non-mixed constraints do not help.

2 Background on PCPs

In this section, we introduce the basic components used to construct PCPs. These include the PCP Theorem, 2-Prover-1-Round protocols, the Parallel Repetition Theorem, Long Codes and Fourier analysis. We also present a construction of so-called "multi-layered smooth Outer PCP" from [9] that builds on work in [3].

2.1 The Basic PCP Outer Verifier

The celebrated PCP theorem [2, 1] is equivalent to saying that for some absolute constant $\rho < 1$, it is NP-hard to distinguish between instances of Max 3SAT that are satisfiable and those where at most a fraction ρ of the clauses can be satisfied by any assignment. With a suitable transformation, one can show that the hardness holds for 3SAT formulae with a regular structure, i.e. every clause contains

exactly 3 variables and every variable appears in exactly 5 clauses (see [5]). We call such formulae to be instances of 3-SAT-5. Thus we have the theorem :

Theorem 2.1 *For some absolute constant $c < 1$, it is NP-hard to distinguish whether a 3-SAT-5 instance is satisfiable (called the YES instance) or there is no assignment satisfying a fraction c of the clauses (called the NO instance).*

Using the above, we can get powerful constant prover 1-round proof systems (also called outer verifiers) as follows. Let u be a parameter, thought of as a large constant. The verifier picks a set of u clauses at random, say $W = \{C_i \mid i = 1, 2, \ldots, u\}$, and asks the first prover to give a satisfying assignment to these clauses. For every i, the verifier picks a variable x_i at random from the clause C_i. Let $U = \{x_i \mid i = 1, 2, \ldots, u\}$ be the set of these variables. The verifier asks the second prover to give an assignment to the set U. The verifier accepts iff the answers of the two provers agree on the set of variables U. If we denote $\pi^{U,W}$ to be the projection that maps an assignment to W to its sub-assignment to U, the verifier accepts iff the answer of the first prover is σ and the answer of the second prover is $\pi^{U,W}(\sigma)$.

Clearly, if the 3-SAT-5 formula is satisfiable, the provers have a strategy to make the verifier accept with probability 1 (completeness property). The Parallel Repetition Theorem of Raz [11] gives the soundness property:

Theorem 2.2 *There is an absolute constant $c_0 < 1$ for which the following holds. If the 3-SAT-5 instance is a NO instance, then the provers in the above 2-prover 1-round protocol have no strategy that makes the verifier accept with probability more than c_0^u.*

2.2 The Multilayered Smooth PCP

The PCP verifier we use is constructed by extending the above 2-prover system. The construction is identical to the one in [9, Sec. 3] and we present it here for the sake of completeness. It uses two ideas: the multilayered PCP due to [3] and the use of smooth projections which behave like bijections on average (these are used as the tests between pairs of provers).

Let L and T be integer parameters (think of them as large constants). In what follows, we will have $(L + 1)$ layers, with the i'th layer corresponding to choices of $(L - i)u$ variables and $(T + i)u$ clauses, with edges between layers imposing consistency checks between the common variables. (The additional Tu clauses in each block are the ones which will ensure that the projections imposing consistency are smooth, i.e., almost bijections on average.)

Definition 2.3 *For $0 \leq i \leq L$, a type-i block is defined to be the union of a set of $(L - i)u$ variables and a set of*

$(T + i)u$ *clauses. The family of all type-i blocks is denoted by $\mathcal{V}_i$.*

Remarks : (a) A block is union of a set of variables and a set of clauses. The fact that the components are **sets** and not **tuples** is important. By definition, there is no order associated with elements of a set. (b) The parameter L above refers to the number of layers. The parameter T is much bigger than L, u, and will enable ensuring that the projections are bijections on average, as formalized in Lemma 2.8.

Definition 2.4 *For $0 \leq i < j \leq L$, we say that a block $U \in \mathcal{V}_i$ is a sub-block of a block $W \in \mathcal{V}_j$ if one can obtain block U by replacing $(j - i)u$ clauses $\{C_l \mid l = 1, 2, .., (j - i)u\}$ in W by $(j - i)u$ variables $\{x_l \mid l = 1, 2, ..., (j - i)u\}$ such that the variable x_l is contained in the clause C_l for $1 \leq l \leq (j - i)u$.*

For $W \in \mathcal{V}_j$, a random sub-block of W of type-i is obtained by choosing $(j - i)u$ clauses at random from the $(T + j)u$ clauses in W and replacing each clause by one of the variables appearing in that clause picked at random.

A block $U \in \mathcal{V}_i$ contains $(L - i)u$ variables and $(T + i)u$ clauses, and thus a total of $(L - i)u + 3(T + i)u$ variables. A satisfying assignment to the block U is an assignment to these $(L - i)u + 3(T + i)u$ variables such that the assignment satisfies all the clauses in U. Let $\mathcal{M}_U$ denote the set of all satisfying assignments to the block U.

If U is a sub-block of W, then every satisfying assignment to W can be restricted to a satisfying assignment to U. Let the map $\pi^{U,W} : \mathcal{M}_W \mapsto \mathcal{M}_U$ denote this operation of taking a sub-assignment/restriction.

Definition 2.5 *We construct a multi-layered graph $\mathcal{G} = \mathcal{G}(L, T, u)$ as follows :*

1. *The graph has $(L + 1)$ layers numbered 0 through L and $\mathcal{V}_i$ is the set of vertices in the ith layer. Thus the vertices in the ith layer are the blocks of type-i.*

2. *There are no edges between the vertices of the same layer.*

3. *For $0 \leq i < j \leq L$, the edges between layers i and j are defined as follows : For a pair of blocks $U \in \mathcal{V}_i$ and $W \in \mathcal{V}_j$ such that U is a sub-block of W, let (U, W) be an edge of the graph.*

Since the 3-SAT-5 instance is regular, the bipartite graph between every pair of layers is regular.

Definition 2.6 *An assignment Φ to the graph $\mathcal{G}(L, T, u)$ assigns a satisfying assignment $\Phi(U) \in \mathcal{M}_U$ to every block U. If U is a sub-block of W, then we say that Φ satisfies an*

edge (U, W) of the graph if $\Phi(U)$ is a restriction of $\Phi(W)$. *In other words*

$$\Phi \text{ satisfies the edge } (U, W) \text{ if } \pi^{U,W}(\Phi(W)) = \Phi(U)$$

Let $OPT(\mathcal{G}, i, j)$ denote the maximum fraction of the edges satisfied between layers i, j by any assignment Φ to these layers.

The lemma below formalizes the "near-bijection" property of the above projection maps. For completeness, we reproduce its short proof from [9].

Lemma 2.7 ([9]) *Let $0 \leq i < j \leq L$ and W be a block of type-j. Let $\sigma, \sigma' \in \mathcal{M}_W$ be two distinct satisfying assignments to W. If U is a random sub-block of W of type-i, then*

$$\Pr_U \left[\pi^{U,W}(\sigma) = \pi^{U,W}(\sigma') \right] \leq \frac{L}{T} \ .$$

Proof: σ, σ' differ in at least one bit. The projection $\pi^{U,W}$ preserves the variables in W, but replaces some clauses by variables. If σ, σ' differ on a variable in W, their projections under $\pi^{U,W}$ are still distinct. Otherwise they differ on some clause, say clause C_0. For a choice of a random sub-block U, one replaces at random $(j - i)u$ clauses out of the $(T + j)u$ clauses in W. With probability $1 - \frac{j-i}{T+j} \geq 1 - \frac{L}{T}$, the clause C_0 is **not** replaced and hence projections of σ, σ' are distinct. ∎

We will state a simple corollary to the above which will be the form we will use this property in our analysis.

Lemma 2.8 *Let $0 \leq i < j \leq L$ and W be a block of type-j. Let $\beta \subseteq \mathcal{M}_W$ be non-empty and let $\sigma \in \beta$. If U is a random sub-block of W of type-i, then with probability at least $1 - \frac{|\beta|L}{T}$ (over the choice of U), we have*

$$\forall \ \sigma' \in \beta, \ \sigma' \neq \sigma, \quad \pi^{U,W}(\sigma) \neq \pi^{U,W}(\sigma') \ . \quad (1)$$

Furthermore, if $\pi^{U,W}(\beta) \subseteq \mathcal{M}_U$ is the projected set, then

$$E_U \left[\frac{1}{|\pi^{U,W}(\beta)|} \right] \leq \frac{1}{|\beta|} + \frac{L}{T} \ . \quad (2)$$

Proof: Equation (1) clearly follows immediately from Lemma 2.7. It remains to show (2). Using the shorthand $\pi = \pi^{U,W}$, we have

$$
\begin{aligned}
E_U \left[\frac{1}{|\pi(\beta)|} \right] &\leq E_U \left[\Pr_{x,y \in \beta}[\pi(x) = \pi(y)] \right] \\
&= E_U \left[\frac{1}{|\beta|} + \Pr_{\substack{x,y \in \beta, \\ x \neq y}}[\pi(x) = \pi(y)] \right] \\
&= \frac{1}{|\beta|} + E_{\substack{x,y \in \beta, \\ x \neq y}} \left[\Pr_U[\pi(x) = \pi(y)] \right] \\
&\leq \frac{1}{|\beta|} + \frac{L}{T}
\end{aligned}
$$

where in the last step we used Lemma 2.7. ∎

Theorem 2.2 can be used to prove the following result, which will be the outer verifier that will be used in our construction.

Theorem 2.9 *For any choice of parameters L, T, u, the multilayered graph $\mathcal{G}(L, T, u)$ has the following properties:*

- *(Completeness:) If the 3-SAT-5 instance is a YES instance, there exists an assignment Φ that satisfies every edge in the graph $\mathcal{G}$.*

- *(Soundness:) If the 3-SAT-5 instance is a NO instance, $OPT(\mathcal{G}, i, j) \leq 2^{-\Omega(u)}$ for every pair of layers $i < j$.*

- *The smoothness property given by Lemma 2.8 is satisfied.*

The size of the graph $\mathcal{G}(L, T, u)$ and time to construct it are bounded by $n^{O((L+T)u)}$ where n is the size of the 3-SAT-5 instance.

2.3 Long Codes and Fourier Analysis

The *Long Code* over a domain M is indexed by all Boolean functions $f : M \mapsto \{1, -1\}$ where we use 1 to represent False and -1 to represent True. Let $\mathcal{F} := \{f \mid f : M \mapsto \{1, -1\}\}$. The long code A of an element $a \in M$ is defined as

$$A(f) = f(a) \quad \forall f \in \mathcal{F}$$

Consider the space of all real-valued "tables" $A : \mathcal{F} \mapsto \mathbb{R}$ that are indexed by functions in $\mathcal{F}$. In particular, a long code is one such table. Consider the characters χ_α where $\alpha \subseteq M$. There is one such character for every $\alpha \subseteq M$, and it is defined as :

$$\chi_\alpha(f) := \prod_{x \in \alpha} f(x)$$

The characters $\{\chi_\alpha\}_{\alpha \subseteq M}$ form an orthonormal basis of the space of real-valued functions on $\mathcal{F}$ under the following definition of inner product of tables. For tables A_1, A_2, define

$$\langle A_1, A_2 \rangle := \frac{1}{2^{|M|}} \sum_{f \in \mathcal{F}} A_1(f) A_2(f) = E_f \left[A_1(f) A_2(f) \right]$$

It follows that any table can be expanded as

$$A = \sum_\alpha \widehat{A}_\alpha \chi_\alpha \quad (3)$$

where the $\widehat{A}_\alpha$ are the Fourier coefficients with $\sum_\alpha |\widehat{A}_\alpha|^2 = \langle A, A \rangle$. When $A : \mathcal{F} \mapsto \{1, -1\}$, we have Parseval's identity $\sum_\alpha \widehat{A}_\alpha^2 = 1$, which in turn implies that all Fourier coefficients are in the range $[-1, 1]$. The Fourier coefficients of a table are given by

$$\widehat{A}_\alpha = \langle A, \chi_\alpha \rangle = E_f \left[A(f) \chi_\alpha(f) \right] \ .$$

In particular when $\alpha = \emptyset$, the value of the coefficient $\widehat{A}_\emptyset$ is just $E_f[A(f)]$.

Folding of proof tables. A table A is said to be *folded* if $A(f) = -A(-f)$ for every $f \in \mathcal{F}$. If A is folded, then $\widehat{A}_\alpha = 0$ whenever $|\alpha|$ is even, and in particular $\widehat{A}_\emptyset = 0$. If A is a legal codeword of the long code, then clearly A is folded. Therefore, one often assumes that the proof tables in a PCP are folded (this holds for several PCPs in [8], including the one for Max 3SAT). A PCP based on folded tables leads to a hardness result (for the associated constraint satisfaction problem) with negations that cannot be controlled easily. Hence, for our result for Max NM-E3SAT, we *cannot* assume that the tables are folded, and in fact removing the requirement of folding can be viewed as the main technical aspect of our work.

3 Hardness of Max NM-E3SAT

In this section, we prove our main result stated below.

Theorem 3.1 *For every constant $\gamma > 0$, it is NP-hard to distinguish between satisfiable instances of* Max NM-E3SAT *and instances where at most a fraction $(7/8 + \gamma)$ of the clauses can be satisfied.*

To prove the above, we will construct a PCP verifier for 3-SAT-5 that, based on its random coin tosses, reads exactly 3 bits from the proof and its acceptance predicate is always of the form $(a \vee b \vee c)$ or $(\bar{a} \vee \bar{b} \vee \bar{c})$ where a, b, c are the 3 bits queried. If the 3-SAT-5 instance is satisfiable, there exists a proof that the verifier accepts with probability 1. If the 3-SAT-5 instance is not satisfiable, the verifier does not accept any proof with probability greater than $7/8 + \gamma$. By creating a Max NM-E3SAT instance consisting of all those clauses checked by the verifier over all its coin tosses, weighted by the probability that each such clause is checked, immediately yields Theorem 3.1. (This will yield a weighted instance of Max NM-E3SAT but we can get the same hardness bound for the unweighted variant using standard gadgets.) So, in the rest of the section we will describe and analyze the PCP verifier with the stated behavior.

3.1 The PCP verifier

Given a 3-SAT-5 instance ϕ, let $\mathcal{G}(L, T, u)$ be the multilayered graph constructed from ϕ, as in Definition 2.5. The parameters L, T, u are large constants (depending on γ) that will be fixed later. We expect the proof to contain, for every block U of the multilayered graph $\mathcal{G}(L, T, u)$, the long code A_U of a supposed satisfying assignment to U. Accordingly, the table A_U will be indexed by all functions in $\mathcal{F}_U$, the set of Boolean functions on $\mathcal{M}_U$. We will use the following

notation for operations on Boolean functions h_1, h_2 on the same domain. The product $h_1 \cdot h_2$ of two functions over the same domain is defined pointwise, namely $(h_1 \cdot h_2)(y) = h_1(y)h_2(y)$. Note that the product is simply the XOR of the two functions. The AND of two functions h_1 and h_2, denoted $h_1 \wedge h_2$, is defined pointwise as $(h_1 \wedge h_2)(y) = (1 + h_1(y) + h_2(y) - h_1(y)h_2(y))/2$ — this represents the Boolean AND operation as we are using the ± 1 notation to denote truth values.

Given access to a collection of tables as described above as proof, our verifier $V^{\mathrm{NM-SAT}}(\varepsilon)$ operates as follows (it uses a parameter $\varepsilon > 0$ depending on γ; it will be chosen later):

1. Pick a random block W_L of type-L.
   For $i = L, L-1, \ldots, 1$, let W_{i-1} be a random sub-block of type-$(i-1)$ of the (already chosen) type-i block W_i.

2. Let $\mathcal{M}_{W_i}$ be the set of satisfying assignments to the block W_i.
   Let $A_i = A_{W_i}$ be the supposed long code of a satisfying assignment to the block W_i.
   Recall that these long codes are indexed by Boolean functions mapping $\mathcal{M}_{W_i}$ to $\{1, -1\}$.
   For $0 \le i < j \le L$, let $\pi^{i,j} = \pi^{W_i, W_j} : \mathcal{M}_{W_j} \mapsto \mathcal{M}_{W_i}$ be the projection function between W_i and W_j.

3. Pick i, j where $0 \le i < j \le L$ at random.

4. Pick uniformly at random functions $f : \mathcal{M}_{W_i} \to \{1, -1\}$ and $g : \mathcal{M}_{W_j} \to \{1, -1\}$.

5. Pick a function $h : \mathcal{M}_{W_j} \to \{1, -1\}$ by defining for every $y \in \mathcal{M}_{W_j}$
$$h(y) = \begin{cases} -1 & \text{with probability } 1 - \varepsilon \\ 1 & \text{with probability } \varepsilon \end{cases}$$

6. Perform one of the following two tests, picked at random with probability $1/2$ each:

   (i) Let g_1 denote the function $\left(-g \cdot (f \circ \pi^{i,j} \wedge h)\right)$. Check whether $A_i(f) \vee A_j(g) \vee A_j(g_1)$ is true, and accept if so.

   (ii) Let g_2 denote the function $\left(-g \cdot ((-f \circ \pi^{i,j}) \wedge h)\right)$. Check whether $\overline{A_i(f)} \vee \overline{A_j(g)} \vee \overline{A_j(g_2)}$ is true, and accept if so.

Note that a verifier like the one above cannot have good soundness when there are only two layers (i.e., the case $L = 1$). This is because a proof that sets all the A_0-tables to 1's and all the A_1 tables to -1's will satisfy every 3SAT constraint that is tested. This shows that the multilayering that we employ is critical for our result stated below.

Theorem 3.2 *The above PCP verifier $V^{\mathrm{NM-SAT}}(\varepsilon)$ has perfect completeness for all $\varepsilon \in [0,1]$. Moreover, for every $\gamma > 0$, there is a choice of $\varepsilon > 0$ and parameters L, T, u for which the $V^{\mathrm{NM-SAT}}(\varepsilon)$ has soundness at most $\frac{7}{8} + \gamma$.*

The rest of this section is devoted to proving the above theorem. We first argue completeness, followed by the soundness analysis in Section 3.3.

3.2 Completeness

When the 3-SAT-5 instance ϕ is satisfiable, consider the proof where for each block W, A_W is the correct long code of $a \in \mathcal{M}_W$ obtained by restricting a fixed satisfying assignment s of ϕ. Then for the tests involving $A_{W_i} = A_i$ and $A_{W_j} = A_j$, A_i will be the long code of some $a \in \mathcal{M}_{W_i}$ and A_j will be the long code of some $b \in \mathcal{M}_{W_j}$ with $\pi^{i,j}(b) = a$. In such a case, we have

$$A_i(f) = f(a), \ A_j(g) = g(b),$$
$$A_j(g_1) = -g(b)(f(a) \wedge h(b)), \text{ and}$$
$$A_j(g_2) = -g(b)(-f(a) \wedge h(b)) .$$

Consider the disjunction $A_i(f) \vee A_j(g) \vee A_j(g_1)$. If $A_i(f) = -1$, this clearly evaluates to true. If $A_i(f) = 1$, then $f(a) = 1$ and so $A_j(g_1) = -g(b)(1 \wedge h(b)) = -g(b) = -A_j(g)$, and hence one of $A_j(g_1)$ and $A_j(g)$ must be true, and once again the disjunction being verified is true. Similarly it can be checked that the disjunction $(\overline{A_i(f) \vee A_j(g) \vee A_j(g_2)})$ is also true. Therefore this proof is accepted by the verifier with probability 1.

3.3 Soundness

Let $\vec{W} = \langle W_L, W_{L-1}, \ldots, W_0 \rangle$ denote the blocks chosen by the verifier in the first step. Arithmetizing the acceptance behavior of the verifier, we see that the probability that the verifier *rejects* is given *exactly* by the expectation

$$E_{\vec{W}, i, j, f, g, h} \left[\frac{(1 + A_i(f))(1 + A_j(g))(1 + A_j(g_1))}{16} \right.$$
$$\left. + \ \frac{(1 - A_i(f))(1 - A_j(g))(1 - A_j(g_2))}{16} \right]$$

where A_i denotes the table A_{W_i}, and A_j the table A_{W_j}. We can simplify the above expression by collecting together terms which are based on identical distributions – for example, the distribution of (f, g) is identical to that of (f, g_1) as well as (f, g_2); the distribution of (g, g_1) is identical to that of (g, g_2), etc. This yields that the rejection probability above equals

$$\frac{1}{8} + \frac{1}{4} E\big[A_i(f) A_j(g)\big] + \frac{1}{8} E\big[A_j(g) A_j(g_1)\big] + \qquad (4)$$
$$+ \ \frac{1}{16} \Big(E\big[A_i(f) A_j(g) A_j(g_1) - A_i(f) A_j(g) A_j(g_2)\big] \Big) ,$$

where the expectations are over $\vec{W}, i, j, f, g, h$ and this is hidden for notational convenience. We wish to show that when the 3-SAT-5 instance is not satisfiable, the above is at least $1/8 - \gamma$.

Now we analyze and bound each of the above terms in turn. We bound the first term using the multi-layered structure. This term corresponds to a cheating strategy where all bits in a layer are set to the same value (1 or -1). The multi-layered structure ensures that such cheating strategies do not succeed.

We bound the third term using the standard approach. This term can be used to extract an assignment to the multilayered graph. If the graph is chosen to have sufficiently small soundness, then this term is negligible in magnitude. The smoothness property of the multilayered construction makes the analysis of both second and third terms easier than corresponding analysis of Håstad [8].

3.3.1 The first term

We begin with the expectation $E[A_i(f) A_j(g)]$. Fix a choice of $\vec{W}$ and we will bound the expectation taken over only i, j, f, g.

Lemma 3.3 *For each choice of $\vec{W} = \langle W_L, W_{L-1}, \ldots, W_0 \rangle$, the following holds:*

$$E_{i,j,f,g}\big[A_i(f) A_j(g)\big] \geq -\frac{1}{L} . \qquad (5)$$

Proof: For $0 \leq l \leq L$, let us define $a_l = E_h[A_l(h)]$ where the expectation is taken over a random function $h : \mathcal{M}_{W_l} \to \{1, -1\}$. Note that $-1 \leq a_l \leq 1$ for every l. For each choice of i, j, $0 \leq i < j \leq L$, we have $E[A_i(f) A_j(g)] = E_f[A_i(f)] E_g[A_j(g)] = a_i a_j$. Therefore

$$E_{i,j,f,g}\big[A_i(f) A_j(g)\big] = \frac{1}{\binom{L+1}{2}} \sum_{0 \leq i < j \leq L} a_i a_j . \qquad (6)$$

Since each a_i is in the range $[-1, 1]$,

$$2 \sum_{i<j} a_i a_j = \Big(\sum_i a_i\Big)^2 - \sum_i a_i^2 \geq -\sum_i a_i^2 \geq -(L+1) . \qquad (7)$$

Combining (6) and (7) gives the inequality (5). ∎

Therefore the expectation $E\big[A_i(f) A_j(g)\big]$ in (4) is at least $-1/L$.

3.3.2 The second term

We now analyze the second expectation $E\big[A_j(g) A_j(g_1)\big]$. Fix a choice of i, j ($0 \leq i < j \leq L$) and a type-j block W_j. We will prove that for each such choice, the expectation over W_i, f, g, h is small where W_i is a random sub-block

of W_j of type-i. For notational convenience, let $W = W_j$, $U = W_i$, $B = A_j$, and $\pi = \pi^{U,W}$. We now proceed with analyzing $E_{U,f,g,h}\big[B(g)B(g_1)\big]$. This can be expanded as in [8, Equation 36] as

$$E_U\Big[\sum_{\beta \subseteq \mathcal{M}_W} \widehat{B}_\beta^2 \prod_{x \in \pi(\beta)} (\tfrac{1}{2}((-1)^{s_x} + (1 - 2\varepsilon)^{s_x}))\Big]$$

where for $x \in \pi(\beta)$, s_x denotes the number of elements of β that project to x. We thus have

$$E_{U,f,g,h}\big[B(g)B(g_1)\big] = \widehat{B}_\emptyset^2 + \tag{8}$$
$$E_U\Big[\sum_{\substack{\beta \subseteq \mathcal{M}_W \\ \beta \neq \emptyset}} \widehat{B}_\beta^2 \prod_{x \in \pi(\beta)}(\tfrac{1}{2}((-1)^{s_x} + (1 - 2\varepsilon)^{s_x}))\Big] .$$

We will bound the absolute value of the second term in the right hand side of (8) above by $O(\varepsilon)$. Fix $T = \frac{L\log(1/\varepsilon)}{\varepsilon^3}$ for the rest of the argument.

Fix a **non-empty** β and consider the expression

$$E_U\Big[\Big|\prod_{x \in \pi(\beta)}(\tfrac{1}{2}((-1)^{s_x} + (1 - 2\varepsilon)^{s_x}))\Big|\Big] \tag{9}$$

We will argue that this is small for every β by considering the cases of small $|\beta|$ and large $|\beta|$ separately. Let $S \stackrel{\text{def}}{=} \frac{\log(1/\varepsilon)}{\varepsilon^2}$.

Case (i) : $|\beta| \leq S$. Using Lemma 2.8, except with probability $\frac{|\beta|L}{T} \leq \varepsilon$ over the choice of the sub-block U, there exists $x \in \pi(\beta)$ with $s_x = 1$. Whenever this happens, the expression (9) is bounded by $|\tfrac{1}{2}(-1 + (1 - 2\varepsilon))| = \varepsilon$.

Case (ii) : $|\beta| \geq S$. Using Lemma 2.8,

$$E_U\Big[\frac{1}{|\pi(\beta)|}\Big] \leq \frac{1}{|\beta|} + \frac{L}{T} \leq \frac{2\varepsilon^2}{\log(1/\varepsilon)} .$$

Hence, with probability except 2ε, we have $|\pi(\beta)| \geq \frac{\log(1/\varepsilon)}{\varepsilon}$. Note that the term corresponding to every $x \in \pi(\beta)$ in expression (9) is at most $(1-\varepsilon)$ in magnitude and so the expression is at most $(1 - \varepsilon)^{|\pi(\beta)|}$ in magnitude. When $|\pi(\beta)| \geq \frac{\log(1/\varepsilon)}{\varepsilon}$, the expression is bounded by ε.

By (8), this proves that

$$\Big|E[B(g)B(g_1)] - \widehat{B}_\emptyset^2\Big| \leq O(\varepsilon) .$$

It follows that $E[B(g)B(g_1)] \geq -O(\varepsilon)$.

3.3.3 The third term

It remains to analyze the expectation $E\big[A_i(f)A_j(g)A_j(g_1) - A_i(f)A_j(g)A_j(g_2)\big]$. By the definition of g_1, g_2, it is easily seen that

$$E\big[A_i(f)A_j(g)A_j(g_2)\big] = E\big[A_i(-f)A_j(g)A_j(g_1)\big] . \tag{10}$$

Fix a choice of $i < j$, and let us take expectations over blocks $W = W_j$ and a random type-i sub-block $U = W_i$. For ease of notation, again denote $\pi = \pi^{U,W}$, $A = A_i$ and $B = A_j$. Rewriting the expectations using Fourier expansions exactly as in [8], we get

$$E_{U,W,f,g,h}[A_i(f)A_j(g)A_j(g_1)]$$
$$= E_{U,W}\Big[\sum_{\substack{\beta \subseteq \mathcal{M}_W \\ \alpha \subseteq \pi(\beta)}} \widehat{A}_\alpha \widehat{B}_\beta^2 p(\alpha, \beta)\Big]$$
$$E_{U,W,f,g,h}[A_i(-f)A_j(g)A_j(g_1)]$$
$$= E_{U,W}\Big[\sum_{\substack{\beta \subseteq \mathcal{M}_W \\ \alpha \subseteq \pi(\beta)}} (-1)^{|\alpha|}\widehat{A}_\alpha \widehat{B}_\beta^2 p(\alpha, \beta)\Big] ,$$

where

$$p(\alpha, \beta) = \prod_{x \in \alpha}(\tfrac{1}{2}((-1)^{s_x} - (1 - 2\varepsilon)^{s_x})) \cdot$$
$$\cdot \prod_{x \in \pi(\beta)\setminus\alpha}(\tfrac{1}{2}((-1)^{s_x} + (1 - 2\varepsilon)^{s_x})) .$$

Together with (10), this gives

$$\Big| E_{U,W,f,g,h}\big[A(f)B(g)B(g_1) - A(f)B(g)B(g_2)\big]\Big|$$
$$= \Big|E_{U,W}\Big[\sum_{\substack{\beta \subseteq \mathcal{M}_W \\ \alpha \subseteq \pi(\beta)}}(1 - (-1)^{|\alpha|})\widehat{A}_\alpha \widehat{B}_\beta^2 p(\alpha, \beta)\Big]\Big|$$
$$\leq 2E_{U,W}\Big[\sum_{\substack{\beta, \alpha \subseteq \pi(\beta) \\ |\alpha| \text{ odd}}} |\widehat{A}_\alpha|\, \widehat{B}_\beta^2\, |p(\alpha, \beta)|\Big] \tag{11}$$

As shown by Håstad [8], for any β,

$$\sum_{\alpha \subseteq \pi(\beta)} p(\alpha, \beta)^2 \leq (1 - \varepsilon)^{|\pi(\beta)|} . \tag{12}$$

We will use the above to show that the contribution to (11) from terms with large $|\beta|$, specifically when $|\beta| \geq S \stackrel{\text{def}}{=} \frac{\log(1/\varepsilon)}{\varepsilon^2}$, is small. In fact, this will be the case even when terms with even $|\alpha|$ are allowed. Indeed, as in **Case (ii)** above, for such β, we have $|\pi(\beta)| \geq \frac{\log(1/\varepsilon)}{\varepsilon}$ except with

probability 2ε. For terms with $|\beta| \geq S$, we have

$$E_{U,W}\left[\sum_{\alpha \subseteq \pi(\beta), |\beta| \geq S} |\widehat{A}_\alpha|\, \widehat{B}_\beta^2\, |p(\alpha,\beta)|\right]$$

$$\leq E_{U,W}\left[\sqrt{\sum_{\substack{|\beta| \geq S \\ \alpha \subseteq \pi(\beta)}} \widehat{A}_\alpha^2 \widehat{B}_\beta^2} \sqrt{\sum_{|\beta| \geq S} \widehat{B}_\beta^2 \sum_{\alpha \subseteq \pi(\beta)} p(\alpha,\beta)^2}\right]$$

$$\leq E_{U,W}\left[\sqrt{\sum_{|\beta| \geq S} \widehat{B}_\beta^2 (1-\varepsilon)^{|\pi(\beta)|}}\right]$$

$$\leq \sqrt{E_{U,W}\left[\sum_{|\beta| \geq S} \widehat{B}_\beta^2 (1-\varepsilon)^{|\pi(\beta)|}\right]},$$

where in the first step we used Cauchy-Schwartz, and in the second step we used Parseval's identity to bound the first product term, and (12) to bound the second product term. Now, except with probability 2ε, we have $|\pi(\beta)| \geq \frac{\log(1/\varepsilon)}{\varepsilon}$. Hence we get a bound of $O(\sqrt{\varepsilon})$.

Now consider the terms with $|\beta| \leq S$.

$$E_{U,W}\left[\sum_{\substack{\alpha \subseteq \pi(\beta), |\beta| \leq S \\ |\alpha| \text{ odd}}} |\widehat{A}_\alpha|\, \widehat{B}_\beta^2\, |p(\alpha,\beta)|\right] \quad (13)$$

$$\leq E_{U,W}\left[\sqrt{\sum_{\substack{\alpha \subseteq \pi(\beta), |\beta| \leq S \\ |\alpha| \text{ odd}}} \widehat{A}_\alpha^2 \widehat{B}_\beta^2} \sqrt{\sum_{\substack{|\beta| \leq S, \\ \alpha \subseteq \pi(\beta)}} \widehat{B}_\beta^2 p(\alpha,\beta)^2}\right]$$

$$\leq E_{U,W}\left[\sqrt{S \cdot \sum_{\substack{\alpha \subseteq \pi(\beta) \\ |\alpha| \text{ odd}}} \widehat{A}_\alpha^2 \widehat{B}_\beta^2 \frac{1}{|\beta|}}\right]$$

$$\leq \sqrt{S} \cdot \sqrt{E_{U,W}\left[\sum_{\substack{\alpha \subseteq \pi(\beta) \\ |\alpha| \text{ odd}}} \widehat{A}_\alpha^2 \widehat{B}_\beta^2 \frac{1}{|\beta|}\right]} \quad (14)$$

To bound the last expectation (for each fixed $i < j$), we use the fact that the 3-SAT-5 instance ϕ is not satisfiable, and hence $OPT(\mathcal{G}, i, j) \leq 2^{-\Omega(u)}$. Consider the following assignment to the type-i and type-j blocks. If U (resp. W) is a type-i (resp. type-j) block with proof table $A = A_U$ (resp. $B = A_W$), assign $\Phi(U) \in \mathcal{M}_U$ to U (resp. $\Phi(W) \in \mathcal{M}_W$ to W) as follows: Pick $\alpha \subseteq \mathcal{M}_U$ (resp. $\beta \subseteq \mathcal{M}_W$) with probability $\widehat{A}_\alpha^2$ (resp. $\widehat{B}_\beta^2$), and set $\Phi(U)$ (resp. $\Phi(W)$) to be a random element in α (resp. β) if these sets are non-empty, and to an arbitrary element otherwise. Note that whenever $\alpha \subseteq \pi^{U,W}(\beta)$ and $\alpha \neq \emptyset$, once α, β are picked, there is at least a $1/|\beta|$ chance that $\pi^{U,W}(\Phi(W)) = \Phi(U)$. It follows

that this randomized procedure satisfies, in expectation, at least a fraction

$$E_{U,W}\left[\sum_{\alpha \subseteq \pi(\beta),\ \alpha \neq \emptyset} \widehat{A}_\alpha^2 \widehat{B}_\beta^2 \frac{1}{|\beta|}\right]$$

of the edges between the i'th and j'th layers. By the soundness condition of Theorem 2.9 and the upper bound (14), it follows that the contribution to (11) from terms with $|\beta| \leq S$ is at most $\frac{\sqrt{\log(1/\varepsilon)}}{\varepsilon} 2^{-\Omega(u)}$ which can be assumed to be at most ε by picking u large enough.

Thus, the third expectation in (4) is at most $O(\sqrt{\varepsilon})$ in absolute value.

3.3.4 Summing up

Using the analysis of the individual terms of (4) from Sections 3.3.1, 3.3.2 and 3.3.3, we conclude that the verifier rejects a NO instance with probability at least

$$\frac{1}{8} - \frac{1}{4L} - O(\varepsilon) - O(\sqrt{\varepsilon}) .$$

Pick $L \geq 1/\varepsilon$. Pick $\varepsilon > 0$ small enough so that the verifier has soundness at most $7/8 + \gamma$, for any desired $\gamma > 0$. This completes the proof of Theorem 3.2.

Acknowledgments

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[2] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[3] I. Dinur, V. Guruswami, S. Khot, and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. In *Proceedings of 35th Annual ACM Symposium on Theory of Computing*, pages 595-601, 2003. To appear in *SIAM Journal on Computing*.

[4] I. Dinur, O. Regev, and C. Smyth. The Hardness of 3-uniform Hypergraph Coloring. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 33-42, 2002.

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

[6] M. Goemans and D. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, 42(6): 1115-1145, 1995.

[7] V. Guruswami. Inapproximability results for set splitting and satisfiability problems with no mixed clauses. *Algorithmica*, 38: 451–469, 2004.

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

[9] S. Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 23-32.

[10] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for Max Cut and other 2-variable CSPs. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, 2004.

[11] R. Raz. A parallel repetition theorem. *SIAM J. of Computing*, 27(3):763–803, 1998.

[12] L. Trevisan, G. Sorkin, M. Sudan, and D. Williamson. Gadgets, Approximation, and Linear Programming. *SIAM J. Comput.*, **29**(6): 2074-2097 (2000).

[13] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proc. of 9th SODA*, pages 201-210, 1998.

[14] U. Zwick. Outward Rotations: A Tool for Rounding Solutions of Semidefinite Programming Relaxations, with Applications to MAX CUT and Other Problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 679-687, 1999.