Inapproximability Results for Set Splitting and Satisfiability Problems with no Mixed Clauses*

Venkatesan Guruswami

MIT Laboratory for Computer Science, NE43-370
200 Technology Square
Cambridge, MA 02139.
Email: venkat@theory.lcs.mit.edu

Abstract

We prove hardness results for approximating set splitting problems and also instances of satisfiability problems which have no "mixed" clauses, i.e., every clause has either all its literals unnegated or all of them negated. Results of Håstad [9] imply tight hardness results for set splitting when all sets have size exactly $k \geq 4$ elements and also for non-mixed satisfiability problems with exactly k literals in each clause for $k \geq 4$. We consider the case $k = 3$. For the MAX E3-SET SPLITTING problem in which all sets have size exactly 3, we prove an NP-hardness result for approximating within any factor better than 19/20. This result holds even for satisfiable instances of MAX E3-SET SPLITTING, and is based on a PCP construction due to Håstad [9]. For "non-mixed MAX 3SAT", we give a PCP construction which is a variant of one in [8] and use it to prove the NP-hardness of approximating within a factor better than 11/12, and also a hardness factor of $15/16 + \varepsilon$ (for any $\varepsilon > 0$) for the version where each clause has *exactly* 3 literals (as opposed to up to 3 literals).

Keywords: Hardness of approximations, Set splitting, PCP, Gadgets.

1 Introduction

We study the approximability of set splitting problems and satisfiability problems whose clauses are restricted to have either all literals unnegated or all of them negated. The latter seems to be a natural variant of the fundamental satisfiability problem.

1.1 Set Splitting Problems

We first discuss the set splitting problems we consider and the prior work on them. In the general MAX SET SPLITTING problem, we are given a universe U and a family $\mathcal{F}$ of subsets of U, and the goal is to find a partition of U into two disjoint (not necessarily equal sized) sets as $U = U_1 \cup U_2$ that maximizes the number of subsets in $\mathcal{F}$ that are *split* (where a set $S \subseteq U$ is said to be split by the partition $U = U_1 \cup U_2$ if $S \cap U_1 \neq \emptyset$ and $S \cap U_2 \neq \emptyset$). The version when all subsets in the family $\mathcal{F}$ are of size *exactly* k is referred to as MAX Ek-SET SPLITTING. For any

*A preliminary version of this paper [6] appears in the Proceedings of the 3rd International Workshop on Approximations Algorithms for Combinatorial Optimization Problems (APPROX), September 2000.

fixed $k \geq 2$, MAX Ek-SET SPLITTING was shown to be NP-hard by Lovász [13]. Obviously, MAX E2-SET SPLITTING is exactly the extensively studied MAX CUT problem. MAX CUT is known to be NP-hard to approximate within $16/17 + \varepsilon$, for any $\varepsilon > 0$ [9, 17], and Goemans and Williamson, in a major breakthrough, used semidefinite programming to give a factor 0.878-approximation algorithm for MAX CUT [5].[1] Here we investigate the approximability of MAX Ek-SET SPLITTING for $k \geq 3$.

The MAX SET SPLITTING problem is related to the constraint satisfaction problem MAX NAE SAT, which is a variant of MAX SAT, but where the goal is to maximize the total weight of the clauses that contain both true and false literals. MAX SET SPLITTING is simply a special case of MAX NAE SAT where all literals appear unnegated (i.e., MAX SET SPLITTING is the same problem as monotone MAX NAE SAT). Similarly MAX Ek-SET SPLITTING is just the monotone version of MAX NAE-Ek-SAT, and in fact is the same problem as MAX 2-COLORABLE HYPERGRAPH on k-uniform hypergraphs (i.e., given a k-uniform hypergraph, find a 2-coloring of its vertices that maximizes the number of hyperedges which are not monochromatic).

Prior Work. We discuss below the status of the MAX Ek-SET SPLITTING and MAX NAE-Ek-SAT problems for $k \geq 3$. These problems are all NP-hard and MAX SNP-hard. Moreover, it was shown that for each $k \geq 3$, there is a constant $\varepsilon_k > 0$, such that is is NP-hard to distinguish between MAX Ek-SET SPLITTING instances where *all* sets can be split by some partition (which we call *satisfiable* instances in the sequel), and those where no partition splits more than a $(1 - \varepsilon_k)$ fraction of the sets [14].

Following the striking inapproximability results of Håstad [9], it has become possible to prove reasonable explicit bounds on the inapproximability ratios of MAX Ek-SET SPLITTING and MAX NAE-Ek-SAT by construction of appropriate *gadgets* (see [17] for formal definitions of gadgets; we freely use this terminology throughout the paper). In particular, the result for $k = 2$ (i.e., MAX CUT) mentioned above follows this approach, and so does the $11/12 + \varepsilon$ hardness result for MAX NAE-E2-SAT [9]. In the same paper [9], Håstad proved a tight inapproximability bound of $1 - 2^{-k} + \varepsilon$, for an arbitrary constant $\varepsilon > 0$, for satisfiable instances of MAX k-SAT, for $k \geq 3$. It follows that even satisfiable instances of MAX NAE-Ek-SAT, for $k \geq 4$, are hard to approximate within $1 - 2^{-k+1} + \varepsilon$, for an arbitrary constant $\varepsilon > 0$. Note that this result is *tight*, since a random truth assignment will "satisfy" a fraction $1 - 2^{-k+1}$ of the clauses of a MAX NAE-Ek-SAT instance. This leaves only the case $k = 3$, and this turns out to be an intriguing case where a tight result is not yet in sight. We now review the results that are known for $k = 3$.

For MAX NAE-E3-SAT, a hardness of approximation within $15/16 + \varepsilon$, even for satisfiable instances, follows from Håstad's inapproximability result for MAX 3-SAT and an easy 2-gadget from MAX 3-SAT to MAX NAE-E3-SAT [14, 18]. On the algorithmic side, the best known approximation algorithm, due to Zwick [19], achieves a ratio of 0.908. (This bound is as yet only based on numerical evidence, the best proven bound is 0.87868 [11, 18], which is slightly better than the Goemans and Williamson approximation guarantee for MAX CUT [5].) For satisfiable instances of MAX NAE-E3-SAT, an approximation ratio of 0.91226 can be achieved in polynomial time [18].

Turning again to set splitting problems, Håstad [9] has shown the tight result that it is NP-hard to approximate (even satisfiable instances of) MAX E4-SET SPLITTING within $7/8+\varepsilon$, for any $\varepsilon > 0$. For MAX E3-SET SPLITTING, by the results of Zwick [18, 19] mentioned above, there exist approximation algorithms achieving a ratio of 0.908 (resp. 0.912) for general (resp. satisfiable) instances. As regards hardness results for MAX E3-SET SPLITTING, no explicit bound in the literature ap-

[1]Throughout this paper we deal only with maximization problems and by an α factor approximation we mean a solution whose value is at least α times that of an optimum solution. Consequently, all factors of approximation we discuss will be less than 1.

pears to be correct. Inapproximability within a factor of (approximately) 0.987 is claimed in [10]; and it is mentioned in [1] that the 9-gadget reducing a PC_0 constraint to 3-Set Splitting constraints that appears in [10] (a PC_0 constraint is of the form $x \oplus y \oplus z = 0$ where x, y, z are *unnegated* variables), together with Håstad's inapproximability result for MAX 3-PARITY, implies a hardness of $17/18 + \varepsilon$ for MAX E3-SET SPLITTING. These claims suffer from the "well-known" flaw in early gadget results that use PC_0 gadgets without giving explicit PC_1 gadgets (a PC_1 constraint checks if $x \oplus y \oplus z = 1$) to conclude hardness results for approximating monotone constraint satisfaction problems (like MAX CUT, MAX E3-SET SPLITTING, etc). The problem is that when the target problem is monotone, one cannot "convert" a PC_1 constraint to a PC_0 constraint by simply negating a variable, and one has to pay an explicit cost in the gadget for negating a variable. This error occurs in early versions of [2] and in [10, 1]. For the case of MAX CUT, the error can be (and was) fixed in [2] who construct a PC_1 gadget from a PC_0 gadget by negating a variable at a unit extra cost. One can similarly fix the error for MAX E3-SET SPLITTING by incurring an extra cost of 4 for the PC_1 gadget, and this will give a hardness result for approximating MAX E3-SET SPLITTING to better than a factor $21/22$ (this result is reported formally in an earlier version of this paper [6]).

1.2 Satisfiability with no Mixed Clauses

The set splitting problem is just a special case of NAE-SAT where one adds the restriction that in every clause all literals appear unnegated (or, equivalently, all appear negated), i.e., no clause is "mixed". This leads us to consider the corresponding "non-mixed" version of the even more fundamental problem of satisfiability where one adds the restriction that none of the clauses in the instance are "mixed". 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 k-NM-SAT (here NM-SAT stands for non-mixed satisfiability). The version where all the clauses have exactly k literals will be referred to as MAX Ek-NM-SAT. This problem appears to be a fairly natural variant of SAT, and does not appear to have been explicitly considered in the literature.

Known results on approximating MAX k-NM-SAT: Clearly, any algorithm that approximates MAX k-SAT within factor α_k also approximates MAX k-NM-SAT within the same factor; in particular approximation factors of $\alpha_2 = 0.931$ and $\alpha_3 = 7/8$ can be achieved in this way [4, 11]. For MAX Ek-NM-SAT, an approximation factor of $1 - 2^{-k}$ can be achieved trivially for all $k \geq 3$, by simply picking a random truth assignment. There are no algorithms known which perform any better on non-mixed clauses than on general satisfiability instances. For $k \geq 4$, a recent result of Håstad [9] shows that MAX Ek-SET SPLITTING is hard to approximate within a factor of $1 - 2^{-k+1} + \varepsilon$ for any $\varepsilon > 0$. Since there is a trivial 2-gadget reducing MAX Ek-SET SPLITTING to MAX Ek-NM-SAT (namely, replace the constraint $\mathrm{split}(x_1, x_2, \ldots, x_k)$ by the two clauses $(x_1 \vee x_2 \vee \cdots \vee x_k)$ and $(\bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k)$), this implies that MAX Ek-NM-SAT (and hence also MAX k-NM-SAT) is NP-hard to approximate within a factor better than $(1 - 2^{-k})$. Hence, for $k \geq 4$, the naive algorithms that work for the more general MAX Ek-SAT are really the best possible for MAX Ek-NM-SAT as well. As in the case of set splitting problems, our focus, therefore, is on the case $k = 3$.

1.3 Our Main Results

For MAX E3-SET SPLITTING, we prove that for every $\varepsilon > 0$, it is NP-hard to find a partition that splits more than a $19/20 + \varepsilon$ fraction of the 3-sets of a *satisfiable* MAX E3-SET SPLITTING instance. This result is proved by demonstrating a simple 5-gadget reducing the predicate $\mathsf{SNE}_4(x, y, z, w) = (x \neq y) \vee (z \neq w)$ to 3-set splitting constraints. The gadget is in fact optimal, and hence this is the best hardness one can obtain for MAX E3-SET SPLITTING by a reduction from SNE_4. We also consider the MAX 2-2-SET SPLITTING problem, which is a variant of MAX E4-SET SPLITTING where the goal is to find a partition that maximizes the number of 4-sets that are split evenly (i.e., have a 2-2-split), and prove an inapproximability ratio of $11/12 + \varepsilon$ for satisfiable instances, and $8/9 + \varepsilon$ for near-satisfiable instances.

For MAX 3-NM-SAT, one can prove an inapproximability ratio of $13/14 + \varepsilon$ by starting with a hard to approximate instance of MAX 3-SAT, and use a 2-gadget to replace each *mixed* clause with clauses that only have either all negated or all unnegated literals (for example, replace a clause $(a \vee b \vee \bar{c})$ with $(a \vee b \vee t)$ and $(\bar{t} \vee \bar{c})$ and a clause $(a \vee \bar{b} \vee \bar{c})$ with $(a \vee t)$ and $(\bar{t} \vee \bar{b} \vee \bar{c})$). For MAX E3-NM-SAT, this method gives hardness within a factor of $19/20 + \varepsilon$. Both these hardness results apply for satisfiable instances of non-mixed SAT as well.

We improve these results by giving a PCP construction following the one in [8]; our PCP makes 3 queries, has perfect completeness and has soundness $1/2 + \varepsilon$. This new PCP construction is essentially the same as the one in [8] – we show that, with one simple modification, one of the two proof tables the verifier reads in their construction need not be *folded* (folding is a technical requirement in PCP constructions which will be elaborated later in the paper). This modified PCP construction enables us to prove a hardness of $11/12 + \varepsilon$ for MAX 3-NM-SAT and a hardness of $15/16 + \varepsilon$ for MAX E3-NM-SAT. Note that no polynomial time algorithm with better performance ratio than $7/8$ is known for either of these problems, and progress in closing this gap remains an open problem.

2 Hardness of Approximating MAX E3-SET SPLITTING

Gadgets: A brief discussion: The hardness results of this section are proven by giving appropriate gadgets reducing constraint satisfaction problems already known to be hard to approximate, to MAX E3-SET SPLITTING. We use the definitions of gadgets following [2, 12, 17]: an α-gadget reducing a boolean function f on variables $x_1, x_2, \ldots, x_k$ to a constraint family $\mathcal{F}$ is a finite collection of (rational) weights w_j and constraints C_j from $\mathcal{F}$ over $x_1, x_2, \ldots, x_k$ and auxiliary variables $y_1, y_2, \ldots, y_p$ such that for each assignment $\vec{a} = a_1, a_2, \ldots, a_k$ to the x_i's that satisfies f, there is an assignment $\vec{b}$ to the y_i's such that a total weight α of the constraints C_j are satisfied by $(\vec{a}, \vec{b})$, and if $\vec{a}$ does not satisfy f, then for every assignment $\vec{b}$ to the y_i's, the weight of the constraints C_j satisfied by $(\vec{a}, \vec{b})$ is at most $\alpha - 1$ (see [12, 17] for further details). The quantity α is a measure of the quality of the reduction, a smaller value of α implies a better approximation preserving reduction.

Theorem 1 *For any $\varepsilon > 0$, it is NP-hard to distinguish between instances of* MAX E3-SET SPLITTING *where all the sets can be split by some partition and those where any partition splits at most a $\frac{19}{20} + \varepsilon$ fraction of the sets.*

The starting point for proving the above is the following powerful hardness result which follows from a PCP construction by Håstad [9] (namely the one he used to show the hardness result for 4-set splitting).

Definition: (MAX SNE_4) An instance of MAX SNE_4 consists on Boolean variables $\{x_1, x_2, \ldots, x_n\}$ and a collection $\mathcal{C}$ of constraints of the form

$$\mathsf{SNE}_4(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}) = (x_{i_1} \neq x_{i_2}) \vee (x_{i_3} \neq x_{i_4})$$

(no negations allowed) defined on certain 4-element subsets of these variables.

Theorem 2 ([9]) *For any $\varepsilon > 0$, it is NP-hard to distinguish between instances of* MAX SNE_4 *where all the constraints can be satisfied by some (boolean) assignment to the variables and where no assignment satisfies more than a $3/4 + \varepsilon$ fraction of the constraints. In other words, even satisfiable instances of* MAX SNE_4 *are NP-hard to approximate within a factor better than $3/4$.*

Proof of Theorem 1: The result will be proven by construction of an appropriate gadget that reduces SNE_4 constraints to 3-Set Splitting constraints. Indeed we claim the following is a 5-*gadget* reducing $\mathsf{SNE}_4(a, b, c, d)$ to 3-set splitting constraints: Introduce auxiliary variables x, y, z (specific just to this constraint) and replace $\mathsf{SNE}_4(a, b, c, d)$ with the five constraints $\mathsf{SPLIT}(a, b, x)$, $\mathsf{SPLIT}(a, b, y)$, $\mathsf{SPLIT}(c, d, y)$, $\mathsf{SPLIT}(c, d, z)$, and $\mathsf{SPLIT}(x, y, z)$ (the constraint $\mathsf{SPLIT}(p, q, r)$ represents the condition that not all of p, q, r are equal).

Firstly when $\mathsf{SNE}_4(a, b, c, d)$ is satisfied by some assignment to a, b, c, d; we wish to verify that all five constraints in the gadget can be satisfied by an appropriate assignment to the auxiliary variables. Indeed, let us assume that, say, $(a \neq b)$ without loss of generality (since the gadget is symmetric in $\{a, b\}$ and $\{c, d\}$); then the assignment $y = z = \bar{c}$ and $x = c$ satisfies all the five 3-set splitting constraints. On the other hand, if $\mathsf{SNE}_4(a, b, c, d)$ is not satisfied by some assignment to a, b, c, d, then we have $a = b$ *and* $c = d$. Now two cases arise: (i) $(a = b) \neq (c = d)$, in which case one of the two constraints $\mathsf{SPLIT}(a, b, y)$ and $\mathsf{SPLIT}(c, d, y)$ is violated irrespective of the assignment to y, and (ii) $a = b = c = d$, in which case the only way to satisfy all the four constraints $\mathsf{SPLIT}(a, b, x)$, $\mathsf{SPLIT}(a, b, y)$, $\mathsf{SPLIT}(c, d, y)$ and $\mathsf{SPLIT}(c, d, z)$ is to set $x = y = z = \bar{a}$ and in this case $\mathsf{SPLIT}(x, y, z)$ is violated. Thus at most four of the five constraints can be satisfied for any assignment to x, y, z if $\mathsf{SNE}_4(a, b, c, d)$ is not satisfied. In other words, the gadget above is a 5-gadget.

Starting from a satisfiable instance of SNE_4 constraints that is hard to approximate within $3/4 + \varepsilon$ (as in Theorem 2), this gives a hardness result of approximating satisfiable instances of MAX E3-SET SPLITTING to within $\frac{19}{20} + \frac{\varepsilon}{5}$, and since $\varepsilon > 0$ is arbitrary, this gives the desired result of Theorem 1. □

Optimality of the Gadget. Sorkin [16] has shown that there is no α-gadget reducing an SNE_4 constraint to CUT constraints for $\alpha < 5$. Since there is a 1-gadget reducing a 3-set splitting constraint to CUT constraints, this also shows that there is no α-gadget reducing an SNE_4 constraint to 3-set splitting constraints for any $\alpha < 5$, and thus the gadget construction we used in the proof of Theorem 1 is in fact optimal.

3 k-Set Splitting for $k \geq 4$

3.1 Hardness of MAX Ek-SET SPLITTING for $k \geq 4$

Håstad [9] proves the tight result that even satisfiable instances of MAX E4-SET SPLITTING are hard to approximate within any factor better than $7/8$. This result does not seem to imply a tight hardness result for MAX Ek-SET SPLITTING for $k \geq 5$ by simply using a gadget. It is possible, however, to modify Håstad's PCP construction to work also for MAX Ek-SET SPLITTING for $k \geq 5$, and this gives the following result:

Theorem 3 ([9]) *For $k \geq 4$, for any $\varepsilon > 0$, it is NP-hard to distinguish between instances of* MAX Ek-SET SPLITTING *where all the sets can be split by some partition and those where any partition splits at most a $1 - 2^{-k+1} + \varepsilon$ fraction of the sets.*

3.2 "Restricted" 4-set splitting

For MAX E4-SET SPLITTING, if the goal is only to split as many sets as possible, then Håstad's result shows that the best one can do is a factor of $7/8$, and this is tight. One can be more specific about requirements on the partition, and for example require that that we maximize the number of 4-sets which have a 1-3 split under the partition, or similarly require that we maximize the number of sets that have a 2-2 split under the partition. Thus we put restrictions on the way in which a set can be split for it to count towards the solution. We call these "restricted" 4-set splitting problems MAX 1-3-SET SPLITTING and MAX 2-2-SET SPLITTING respectively. It is easy to see that these versions are also NP-hard, and in this section we will discuss their approximability. Other than being natural questions, one motivation to consider these "restricted" set splitting questions is their potential applicability to improving the hardness result for approximating MAX CUT; we will discuss this possibility in more detail at the end of this section.

It is easy to see that a MAX 1-3-SET SPLITTING instance can be cast as a system of linear equations over GF(2) each of the form $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} \oplus x_{i_4} = 1$ (no negations involved in the literals), and by yet another powerful result in [9], this problem is hard to approximate within any factor better than $1/2$ (of course picking a random partition will satisfy half of these linear constraints, so this result is again tight). We focus, therefore, on MAX 2-2-SET SPLITTING and prove the following result:

Theorem 4 *For any $\varepsilon > 0$, it is NP-hard to distinguish between instances of* MAX 2-2-SET SPLITTING *which are satisfiable (i.e., all the sets can be 2-2-split by some partition) and those where any partition 2-2-splits at most a $(\frac{11}{12} + \varepsilon)$ fraction of the sets.*

Proof: The proof proceeds along similar lines to that of Theorem 1, namely via a gadget construction from SNE_4. Specifically we will give a 3-gadget reducing an SNE_4 constraint to 2-2-set splitting constraints, and together with the hardness result for MAX SNE_4 of Theorem 2, this implies the claimed result for MAX 2-2-SET SPLITTING.

The gadget is in fact very simple. Given a constraint $\mathsf{SNE}_4(a,b,c,d)$, introduce auxiliary variables x,y,z,w and replace $\mathsf{SNE}_4(a,b,c,d)$ with the three constraints: 2-2-SPLIT(a,b,x,y), 2-2-SPLIT(c,d,y,z) and 2-2-SPLIT(x,y,z,w). We claim that this is a 3-gadget. Indeed suppose that $\mathsf{SNE}_4(a,b,c,d)$ is satisfied. We consider two cases: (i) $(a \neq b) \wedge (c = d)$, and (ii) $(a \neq b) \wedge (c \neq d)$ (the third case $(a = b) \wedge (c \neq d)$ is similar to case (i)). In case (i), we can satisfy all three 2-2-SPLIT constraints by setting $y = z = \bar{c}$ and $x = w = c$. In case (ii), we can satisfy all three 2-2-SPLIT constraints by setting $x = z = 0$ and $y = w = 1$.

Now consider the case when $\mathsf{SNE}_4(a,b,c,d)$ is not satisfied, so that $(a = b) \wedge (c = d)$. In order to satisfy both 2-2-SPLIT(a,b,x,y) and 2-2-SPLIT(c,d,y,z), one is forced to pick $x = y = \bar{a}$ and $y = z = \bar{c}$. Now either $a \neq c$ in which case the two forced assignments to y are not consistent, or we have $x = y = z$ which violates 2-2-SPLIT(x,y,z,w). Hence at most two of the three constraints can be satisfied in this case. Thus the above gadget is indeed a 3-gadget and the proof is complete. $\Box$

If we relax the requirement that the instance of MAX 2-2-SET SPLITTING in Theorem 4 above be satisfiable, then we can prove a better hardness result.

Theorem 5 *For any $\varepsilon > 0$, it is NP-hard to distinguish between instances of* MAX 2-2-SET SPLITTING *where a $(1 - \varepsilon)$ fraction of all the sets can be 2-2-split by some partition and those where any partition 2-2-splits at most a $(\frac{8}{9} + \varepsilon)$ fraction of the sets.*

Proof: The proof is again based on gadget constructions, this time from 3-parity constraints (PC_0 and PC_1) and we only sketch the details. We will first present gadgets from 3-parity to a constraint family ONE where $\mathsf{ONE}(a,b,c)$ is TRUE if and only if exactly one of the three Boolean variables a,b,c takes on the value 1 (we do not allow any negations in the ONE constraints), and prove that approximating MAX ONE within $\frac{8}{9}+\varepsilon$ is NP-hard for any $\varepsilon > 0$. Given an instance $\mathcal{I}$ of MAX ONE consider the MAX 2-2-SET SPLITTING instance $\mathcal{I}'$ produced by adding one global element T to each ONE constraint of $\mathcal{I}$, i.e., replace $\mathsf{ONE}(a,b,c)$ with 2-2-$\mathsf{SPLIT}(a,b,c,T)$. Clearly the optimum value of $\mathcal{I}$ is the same as that of $\mathcal{I}'$ and any solution to $\mathcal{I}'$ that satisfies Q constraints can be tranformed efficiently into a solution to $\mathcal{I}$ that satisfies the same number Q of ONE constraints. Thus the hardness result for MAX ONE implies our claimed hardness result.

To prove the hardness result for MAX ONE, we give gadgets from PC_0 and PC_1. Given a PC_1 constraint $a \oplus b \oplus c = 1$, the gadget transforms it into ONE constraints as follows: Introduce auxiliary variables x,y,z and the three constraints $\mathsf{ONE}(a,x,y)$, $\mathsf{ONE}(b,y,z)$ and $\mathsf{ONE}(c,z,x)$. It is easy to check that this is a 3-gadget. Given a PC_0 constraint $a \oplus b \oplus c = 0$, note that this is equivalent to $\bar{a} \oplus \bar{b} \oplus \bar{c} = 1$, and thus one can use the PC_1 gadget if one can somehow force three variables t_1,t_2,t_3 to equal $\bar{a},\bar{b},\bar{c}$ respectively (recall that we do *not* allow negations in our constraints). To achieve this we use a global auxiliary variable F that is shared across all constraints; we think of F as being set to 0. The gadget now consists of the six ONE constraints: $\mathsf{ONE}(a,t_1,F)$, $\mathsf{ONE}(b,t_2,F)$, $\mathsf{ONE}(c,t_3,F)$, $\mathsf{ONE}(t_1,x,y)$, $\mathsf{ONE}(t_2,y,z)$ and $\mathsf{ONE}(t_3,z,x)$ (here t_1,t_2,t_3,x,y,z are auxiliary varaibles specific to PC_0 constraint $a \oplus b \oplus c = 0$ and F is a global auxiliary variable). The purpose of the first three constraints is to ensure (assuming $F = 0$) that $t_1 = \bar{a}$, $t_2 = \bar{b}$ and $t_3 = \bar{c}$ and the last three check that $t_1 \oplus t_2 \oplus t_3 = 1$. If we force $F = 0$, it is therefore easy to see that the above is a 6-gadget. If $F = 1$, then the only way first three ONE constraints can be satisfied is if $t_1 = t_2 = t_3 = 0$, and in this case at most two of the last three ONE constraints can be satisfied. Hence, the above is indeed a 6-gadget. We thus have a 3-gadget from PC_1 and a 6-gadget from PC_0. Using this (as in [17]) together with the tight hardness result for 3-parity from [9], we get the hardness of approximating MAX ONE within $\frac{8}{9}+\varepsilon$ for any $\varepsilon > 0$. □

Algorithms. On the algorithmic side, the current best approximation algorithm for MAX 2-2-SET SPLITTING seems to be reducing the problem to MAX E3-SET SPLITTING or picking a random solution, and returning the better of the two solutions. This achieves an approximation ratio of about 0.4035 (which is better than the factor $3/8$ achieved by a random solution). The current gap between the hardness and algorithms is thus quite large (we believe a semidefinite programming based algorithm should be able to achieve much better performance for this problem).

Discussion. One motivation for studying MAX 2-2-SET SPLITTING is a quest to obtain an improved hardness result for MAX CUT. The strong hardness results for set splitting obtained recently, for example the one for MAX E4-SET SPLITTING, are not directly amenable to reduction to CUT constraints by a "good" gadget, and the reason for this is precisely the lack of stringent requirements on how the set must be split in the set splitting constraint. On the other hand, there is a simple 4-gadget reducing the constraint 2-2-$\mathsf{SPLIT}(a,b,c,d)$ to CUT constraints: namely include the $\binom{4}{2}$ CUT constraints on all pairs of elements drawn from $\{a,b,c,d\}$. This raises the hope that new, strong inapproximability results for some suitable "restricted" set splitting problem (akin to Håstad's recent results [9] for MAX E4-SET SPLITTING), if they can be proved, would give us a "new" shot at improving the inapproximability results for MAX CUT (and may be even MAX E3-SET SPLITTING).

To get such strong hardness results, it seems that one would have to design and analyze special purpose PCPs whose acceptance predicate is close to the concerned set splitting question (just like Håstad's result for MAX E4-SET SPLITTING), and this appears to be an extremely challenging task. Of course, in addition to MAX 2-2-SET SPLITTING, one could also consider other "restricted" set splitting problems; for example a good candidate could be MAX 2-3-SET SPLITTING. For this problem, an inapproximability result for a factor $(\frac{7}{8}+\varepsilon)$ follows by a trivial reduction from MAX E4-SET SPLITTING. On the algorithmic side, a simple reduction to MAX E3-SET SPLITTING gives an algorithm with approximation ratio roughly 0.64, which is better than the trivial factor $5/8$ achieved by a random solution. Once again there is a simple 3-gadget reducing 2-3-SPLIT constraints to CUT constraints, so a hardness result for approximating MAX 2-3-SET SPLITTING within a factor, say 0.8, would give an improvement on the factor $(\frac{16}{17}+\varepsilon)$ hardness result for MAX CUT.

4 Hardness of approximating MAX 3-NM-SAT

We will prove the following theorems in this section.

Theorem 6 *For any $\varepsilon > 0$, it is NP-hard to distinguish between satisfiable instances of* MAX 3-NM-SAT *and those where at most a fraction $11/12+\varepsilon$ of the clauses can be satisfied.*

Theorem 7 *For any $\varepsilon > 0$, it is NP-hard to distinguish between satisfiable instances of* MAX E3-NM-SAT *and those where at most a fraction $15/16+\varepsilon$ of the clauses can be satisfied.*

Sketch of Idea: The instances of 3SAT which are proved hard to approximate within a factor of $7/8+\varepsilon$ have the property that (at least) $3/4$ of the clauses are "mixed" (i.e., have both positive and negative literals). In proving a hardness result for, say, MAX 3-NM-SAT, one converts these clauses into non-mixed clauses using a gadget of some cost (and this method gives a factor $13/14+\varepsilon$ hardness for MAX 3-NM-SAT). In order to obtain an improved hardness bound, our approach will be to get a similar hardness result for 3SAT when the fraction of mixed clauses is smaller, and then use the same gadget approach to get hardness for MAX 3-NM-SAT. To this end, we will first prove:

Theorem 8 *For any $\varepsilon > 0$, given a* MAX E3-SAT *instance in which at most half the clauses are mixed, it is NP-hard to distinguish between instances which are satisfiable and those where every assignment satisfies at most a fraction $7/8+\varepsilon$ of the clauses.*

Proof: The proof follows from the construction of a PCP system for NP that makes 3 (adaptive) queries, has perfect completeness, and a soundness of $1/2+\varepsilon$ for any $\varepsilon > 0$. The hardness for MAX E3-SAT as claimed then follows by suitable gadgets reducing the constraints checked by the PCP verifier to 3SAT constraints. The PCP will be a simple modification of the one in [8, 7]; we will be heavily relying on the treatment and terminology of [7]. We provide below a high-level description of the PCP construction; while by no means complete, this should give some sense of the ideas used in the construction.

Interlude: PCP constructions follow the paradigm of proof composition. In its most modern and convenient to use form, one starts with an *outer proof system* which is a *2-Prover 1-Round proof system* (2P1R) construction for NP due to Raz [15]. Raz's construction works as follows. Given a 3SAT instance, the verifier picks u variables at random, and for each variable picks a clause in which it occurs at random. The verifier then asks the prover P_1 for the truth assignment to the u variables and the prover P_2 for the truth assignment to the $3u$ variables in all the clauses it picked. (We ignore issues like two picked clauses sharing a variable as these have $o(1)$ probability of

```
Inner Verifier V_p^{A,B} (u, π)
```

$\texttt{Inner Verifier}\ \mathrm{V}_p^{A,B}\ (u,\pi)$

Choose uniformly at random $f \in \mathcal{F}_{[2]^u}$, $g \in \mathcal{F}_{[7]^u}$

Choose at random $h \in \mathcal{F}_{[7]^u}$ such that $\forall b \in [7]^u$, $\mathbf{Pr}[h(b) = 1] = p$

if $A(f) = 1$ **then accept** iff $B(g) \neq B(-g(f \circ \pi \wedge h))$

if $A(f) = -1$ **then accept** iff $B(g) \neq B(-g(-f \circ \pi \wedge h))$

$\texttt{Inner Verifier}\ \mathrm{IV3}_\delta{}^{A,B}\ (u,\pi)$

Set $t = \lceil 1/\delta \rceil$, $\varepsilon_1 = \delta^2$ and $\varepsilon_i = \varepsilon_{i-1}^{2c/\varepsilon_{i-1}}$

Choose $p \in \{\varepsilon_1, \ldots, \varepsilon_t\}$ uniformly at random

Run B-MBC$_p{}^{A,B}$ (u,π).

Figure 1: The inner verifier V_p and our final inner verifier $\mathrm{IV3}_\delta$.

occurring.) The verifier then accepts if the assignment given by P_2 satisfies all the clauses it picked and also is consistent with the assignment returned by P_1. This requirement can be captured as the answers a and b of P_1 and P_2 satisfying a "projection" requirement $\pi(b) = a$. Raz's parallel-repetition theorem proves that the soundness of this 2P1R goes down as c^u for some absolute constant $c < 1$.

In the final PCP system, the proof is expected to be the encodings of all possible answers of the two provers of the outer 2P1R proof system using some suitable error-correcting code. For efficient constructions the code used is the *long code* of [2]. The long code of a string of k bits is simply the value of all the 2^{2^k} k-ary boolean functions on that string (for example the long code of a k-bit string a is a string A which has one coordinate for each k-ary boolean function f and the entry of A in coordinate f, denoted $A(f)$ satisfies $A(f) = f(a)$) . The construction of a PCP now reduces to the construction of a good *inner verifier* that given a pair of strings A, B which are purportedly long codes, and a projection function π, checks if these strings are the long codes of two consistent strings (as per the projection). Referring the reader to [7, 9] for details, we delve into the specification of our inner verifier. [End Interlude]

First, some remarks on our notation. For an integer n, $[n]$ stands for the set $\{1, 2, \ldots, n\}$. We represent Boolean values by $\{1, -1\}$ with 1 standing for FALSE and -1 for TRUE – this has the nice feature that Xor becomes multiplication. For any domain D, $\mathcal{F}_D$ stands for the space of all functions $f : D \to \{1, -1\}$.

The inner verifier is given input an integer u and a projection function $\pi : [7]^u \to [2]^u$ and has oracle access to tables $A : \mathcal{F}_{[2]^u} \to \{1, -1\}$ which is *folded* (i.e., $A(-f) = -A(f)$ for all functions in $\mathcal{F}_{[2]^u}$) and $B : \mathcal{F}_{[7]^u} \to \{1, -1\}$ (which is *not* required to be folded), and aims to check that A (resp. B) is the long code of a (resp. b) which satisfy $\pi(b) = a$. The formal specification of our inner verifier $\mathrm{IV3}_\delta{}^{A,B}$ (u, π) is given in Figure 1. The (weird) distribution according to which (the bias parameter) p is picked is identical to the one used in [7] and the constant c used in its specification is an absolute (small) constant which can be figured out from our proofs.

The main (and in fact only) difference of this inner verifier from the one in [8] is that the table B is not assumed to be folded (this will be critical to our application), and the conditions checked are "inequalities" as opposed to "equalities" (i.e., of the form $x \neq y$ instead of $x = y$). Note that this change is clearly necessary, as otherwise one could simply set $B(g) = 1\ \forall g$, and this will satisfy all checks (this will be a valid table as B is *not* required to be folded).

It is clear that this inner verifier has perfect completeness, since when A is the long code of a, B the long code of b and $\pi(b) = a$, the inner verifier accepts with probability 1. Assume now that we can prove that for some small $\varepsilon > 0$ (which can be made as small as we seek), the combined PCP verifier constructed by *composing* the standard outer verifier due to Raz with the above inner verifier has soundness $1/2 + \varepsilon$.[2] We now claim that this will imply the result claimed about MAX E3-SAT with at most half the number of clauses being mixed. It is easy to see that, for each random choice of the inner verifier, the boolean function checked by the inner verifier is of the form

$$(a \vee b \vee c) \wedge (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee b \vee c') \wedge (\bar{a} \vee \bar{b} \vee \bar{c'})$$

by appropriate identifications of $B(g)$ with b, $A(f)$ with a or $\bar{a}$ depending upon whether the folded table contains an entry for the function f or for its complement $-f$, and suitable identifications $B(-g(f \circ \pi \wedge h))$ and $B(-g(-f \circ \pi \wedge h))$ with c, c'. This actually gives a 4-gadget reducing the acceptance predicate of the PCP to 3SAT constraints.

Hence the acceptance condition of the PCP can be viewed as a 3SAT instance in which at most half the clauses are mixed (note that only two of the four clauses in the gadget above are mixed). Since the soundness of the PCP is $1/2 + \varepsilon$, together with the 4-gadget, this gives a hardness of approximating MAX E3-SAT with at most half the number of clauses being mixed, as desired. We therefore need to bound the soundness of the inner verifier by $1/2 + \varepsilon$. The analysis of the soundness of the inner verifier follows the same sequence of lemmas as the proof for the original inner verifier IV3$_\delta$ in [9, 7]. The only change is that we must now rework the proofs without assuming that B is folded. The full details of the soundness analysis are presented in Appendix A. This completes the proof of Theorem 8. □ (*Theorem 8*)

Proof of Theorem 6: There is a simple reduction from E3SAT to non-mixed 3SAT obtained by replacing a clause $(a \vee b \vee \bar{c})$ with two non-mixed clauses $(a \vee b \vee t)$ and $(\bar{t} \vee \bar{c})$ where t is a new variable used only in these two clauses, and by similarly replacing a clause $(a \vee \bar{b} \vee \bar{c})$ with two non-mixed clauses $(a \vee t)$ and $(\bar{t} \vee \bar{b} \vee \bar{c})$. Starting from a hard instance of E3SAT (3SAT with all clauses having *exactly* three literals) as in Theorem 8, satisfiable instances of E3SAT get mapped to satisfiable instances of MAX 3-NM-SAT, while instances where at most a fraction $7/8 + \varepsilon$ of the clauses are satisfiable get mapped to instances of MAX 3-NM-SAT where at most a fraction $11/12 + 2\varepsilon/3$ of the clauses are satisfiable. Since $\varepsilon > 0$ is arbitrary, the result of Theorem 6 follows. □ (*Theorem 6*)

Proof of Theorem 7: The proof is similar to the above, except now a clause $(a \vee b \vee \bar{c})$ is replaced by the three non-mixed clauses $(a \vee b \vee t_1)$, $(a \vee b \vee t_2)$ and $(\bar{c} \vee \bar{t_1} \vee \bar{t_2})$, and a clause $(a \vee \bar{b} \vee \bar{c})$ is replaced by the three non-mixed clauses $(\bar{b} \vee \bar{c} \vee \bar{t_1})$, $(\bar{b} \vee \bar{c} \vee \bar{t_2})$ and $(a \vee t_1 \vee t_2)$ where t_1, t_2 are new variables used only in these three clauses. Using the result of Theorem 8, this implies a hardness of approximating (even satisfiable instances of) MAX E3-NM-SAT within any factor strictly greater than $15/16$. □ (*Theorem 7*)

5 Concluding Remarks

The exact approximability of MAX E3-SET SPLITTING remains an intriguing open question, though by now the gap between the positive and negative results is quite small (there is a 0.908 approximation algorithm, and it is NP-hard to get a better than 0.95 approximation; in fact the hardness

[2] The exact definition of soundness of an inner verifier turns out to be a tricky issue, we once again refer the reader to [8, 7] for the definitions.

holds for satisfiable instances as well for which a 0.912 approximation algorithm exists). A similar situation exists with MAX CUT where a 0.878 approximation algorithm exists, while approximating within a factor of 0.942 is NP-hard.

We established the tight result that approximating MAX E3-SAT where at most half the number of clauses are mixed to within any factor better than $7/8$ is NP-hard. We then used this result to deduce hardness of approximating MAX 3-NM-SAT and MAX E3-NM-SAT within factors better than $11/12$ and $15/16$ respectively. It is not clear how to algorithmically exploit the non-mixed nature of all clauses and devise an algorithm with performance ratio better than $7/8$ for either of these problems; in fact it is very well possible that these problems are hard to approximate within $7/8 + \varepsilon$, and if this is the case, we believe an interesting new approach might be required to establish such a result.

Shedding further light on the approximability of MAX 2-2-SET SPLITTING (and similarly MAX SET SPLITTING23) are other interesting open questions arising from our work.

Acknowledgments

I would like to thank Johan Håstad, Greg Sorkin, Madhu Sudan and Uri Zwick for useful discussions. I am grateful to Greg Sorkin for his proof that the 5-gadget from SNE_4 to 3-set splitting is in fact optimal.

References

[1] G. Andersson and L. Engebretsen. Better approximation algorithms for SET SPLITTING and NOT-ALL-EQUAL SAT. *Information Processing Letters*, 65:305-311, 1998.

[2] M. Bellare, O. Goldreich and M. Sudan. Free bits, PCP's and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804-915, 1998. Preliminary version in *Proc. of FOCS'95*.

[3] P. Crescenzi, R. Silvestri and L. Trevisan. To weight or not to weight: Where is the question? *Proc. of 4th Israel Symposium on Theory of Computing and Systems*, pp. 68-77, 1996.

[4] U. Feige and M. Goemans. Approximating the value of two prover proof systems, with applications to MAX-2SAT and MAX-DICUT. *Proc. of the 3rd Israel Symposium on Theory and Computing Systems, Tel Aviv*, pp. 182-189, 1995.

[5] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115-1145, 1995.

[6] V. Guruswami. Inapproximability results for set splitting and satisfiability problems with no mixed clauses. *Proc. of APPROX 2000*.

[7] V. Guruswami. *Query-efficient Checking of Proofs and Improved PCP Characterizations of NP*. S.M Thesis, MIT, May 1999.

[8] V. Guruswami, D. Lewin, M. Sudan and L. Trevisan. A tight characterization of NP with 3 query PCPs. *ECCC Technical Report* TR98-034, 1998. Preliminary Version in *Proc. of FOCS'98*.

[9] J. Håstad. Some optimal inapproximability results. *ECCC Technical Report TR97-37*, 1997. Preliminary version in *Proc. of STOC'97*.

[10] V. Kann, J. Lagergren and A. Panconesi. Approximability of maximum splitting of k-sets and some other APX-complete problems. *Information Processing Letters*, 58:105-110, 1996.

[11] H. Karloff and U. Zwick. A $(7/8-\varepsilon)$-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th FOCS*, 1997.

[12] S. Khanna, M. Sudan and D. Williamson. A complete classification of the approximability of maximization problems derived from Boolean constraint satisfaction. *Proc. of the 29th STOC*, 1997.

[13] L. Lovász. Coverings and colorings of hypergraphs. *Proc. 4th Southeastern Conf. on Combinatorics, Graph Theory, and Computing*, pp. 3-12, Utilitas Mathematica Publishing, Winnipeg, 1973.

[14] E. Petrank. The hardness of approximation: Gap location. *Computational Complexity*, 4:133-157, 1994.

[15] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998. Preliminary version in *Proc. of STOC'95*.

[16] G. Sorkin. Private Communication, July 2000.

[17] L. Trevisan, G. Sorkin, M. Sudan and D. Williamson. Gadgets, approximation and linear programming. *Proceedings of the 37th FOCS*, pp. 517-626, 1996.

[18] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[19] U. Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proceedings of STOC'99*.

A Soundness Analysis of Inner Verifier IV3$_\delta$

This section presents the analysis of the soundness of the inner verifier IV3$_\delta$ from the proof of Theorem 8. Let A, B, π and p be fixed, and let $X = X_{A,B,\pi,p}$ be the random variable whose value is 1 when $V_p^{A,B}(u,\pi)$ accepts and 0 otherwise[3]. The acceptance probability of $V_p^{A,B}(u,\pi)$ is $\mathbf{E}[X]$, which equals

$$\mathop{\mathbf{E}}_{f,g,h}\left[\left(\frac{(1+A(f))}{2}\right)\left(\frac{(1-B(g)B(-g(f\circ\pi\wedge h)))}{2}\right)\right.$$
$$\left.+\left(\frac{(1-A(f))}{2}\right)\left(\frac{(1-B(g)B(-g(-f\circ\pi\wedge h)))}{2}\right)\right]$$

[3]The sample space of $X_{A,B,\pi,p}$ is given by the possible choices of f, g, and h.

Since A is folded, we always have $-A(f) = A(-f)$. Using the linearity of expectation and the fact that f and $-f$ are identically distributed in the uniform distribution, we get $\mathbf{E}[X]$ equals

$$
\begin{aligned}
& 2 \mathop{\mathbf{E}}_{f,g,h}\left[\left(\tfrac{(1+A(f))}{2}\right)\left(\tfrac{(1-B(g)B(-g(f\circ\pi\wedge h)))}{2}\right)\right] \\
= \quad & \tfrac{1}{2} + \tfrac{1}{2} \mathop{\mathbf{E}}_{f,g,h}[-B(g)B(-g(f\circ\pi\wedge h))] \\
& + \tfrac{1}{2} \mathop{\mathbf{E}}_{f,g,h}[-A(f)B(g)B(-g(f\circ\pi\wedge h))] \qquad (1)
\end{aligned}
$$

(there would also be a term $\frac{1}{2}\mathop{\mathbf{E}}_{f}[A(f)]$ which equals zero owing to the foldedness of A).

To complete the soundness analysis, we prove that the second term in Equation (1) can be made very small (at most 3δ) when we take expectations over p, π as well. We also prove that whenever the expectation of the third term over p, π is "large", A and B can be "decoded" (using a decoding strategy that will be specified shortly) into two "consistent" strings (strings that cause the outer verifier to accept) with non-negligible probability. Appealing to the soundness of the outer verifier, exactly as in [8, 7], this implies that the soundness of the inner verifier is close to $1/2$.

Hence the soundness analysis can now be completed provided the two technical lemmas (Lemmas 9 and 10) are proved (the reader can find the exact and full details of why these lemmas suffice to complete the proof in the description of [7]). We first define some terms which are necessary for stating our lemmas.

Definition 1 [Smooth Distribution]*: Let Σ, R be arbitrary finite sets, and $\mathcal{D} = \{D_n\}_{n\geq 1}$ be an ensemble of distributions over functions $\pi : \Sigma^n \to R^n$. Then $\mathcal{D}$ is said to be* smooth *if there exists an absolute constant c such that for all large enough n, the following holds: for every k and every set $S \subseteq \Sigma^n$,*

$$|S| \geq |\Sigma|^k \Rightarrow \mathop{\mathbf{Pr}}_{\pi\in D_n}[|\pi(S)| < \sigma k/2] \leq e^{-c\cdot k} \quad .$$

We now (briefly) review some discrete Fourier analysis which will be useful in our analysis. It is useful to view a function $A : \mathcal{F}_D \to \{-1, 1\}$ as a real-valued function $A : \mathcal{F}_D \to \mathcal{R}$. The set of functions $A : \mathcal{F}_D \to \mathcal{R}$ is a vector space over the reals of dimension $2^{|D|}$. We can define a scalar product between functions as

$$A \cdot B = \frac{1}{2^{|D|}} \sum_{f\in\mathcal{F}_D} A(f)B(f) = \mathop{\mathbf{E}}_{f}[A(f)B(f)] \, .$$

For each $\alpha \subseteq D$, the linear function l_α is defined by: $l_\alpha(f) = \prod_{x\in\alpha} f(x)$ for $f \in \mathcal{F}_D$ The set of linear functions is easily seen to form an orthonormal basis for the set of functions $A : \mathcal{F}_D \to \mathcal{R}$ (with respect to the above dot product). This implies that for any such function A we have the *Fourier expansion*

$$A(f) = \sum_{\alpha} \hat{A}_\alpha l_\alpha(f), \qquad (2)$$

where for $\alpha \subseteq D$, $\hat{A}_\alpha = A \cdot l_\alpha$ is the *Fourier coefficient* of A w.r.t α. Because the range of A is $\{1, -1\}$, we have $\sum_\alpha \hat{A}_\alpha^2 = 1$ (this is the famous Parseval's identity). We are now in a position to describe the "decoding procedure" we will use to "decode" purported long codes into the corresponding codeword.

Definition 2 [Decoding Procedure]*: The decoding procedure* Decode *takes a string $A : \mathcal{F}_D \to \{1, -1\}$, and returns an element of the domain D by randomly picking a subset α of D with probability equal to $\hat{A}_\alpha^2$ (this is a valid probability distribution because of Parseval's identity), and then returning a random element of α. If $\alpha = \emptyset$,* Decode *simply returns some fixed element of D.*[4]

Lemma 9 *If $t = \lceil \delta^{-1} \rceil$, $\varepsilon_1 = \delta^2$ and $\varepsilon_i = \varepsilon_{i-1}^{2c/\varepsilon_{i-1}}$ for $1 < i \le t$, and $p \in \{\varepsilon_1, \cdots, \varepsilon_t\}$ is chosen uniformly at random, then for large enough positive integers u, for all $B : \mathcal{F}_{[7]^u} \to \{-1, 1\}$,*

$$\mathop{\mathbf{E}}_{p,\pi,f,g,h}[-B(g)B(-g \cdot (f \circ \pi \wedge h))] \le 2\varepsilon_1^{1/2} + \frac{1}{t} \le 3\delta$$

assuming π is picked according to a smooth distribution. (The parameter p is implicitly used in bias of the random choice of the function h.)

Lemma 10 *For every $\delta, p > 0$, there exists a constant $\gamma = \gamma_{\delta,p} > 0$, such that for all large enough u, for all strings $B : \mathcal{F}_{[7]^u} \to \{1, -1\}$ and* folded strings *$\{A^\pi : \mathcal{F}_{[2]^u} \to \{1, -1\}\}_{\pi \in_R \mathcal{D}(B)}$, if*

$$\mathop{\mathbf{E}}_{\pi,f,g,h}[-A^\pi(f)B(g)B(-g \cdot (f \circ \pi \wedge h))] \ge \delta,$$

then $\mathbf{Pr}\left[\mathsf{Decode}(A^\pi) = \pi(\mathsf{Decode}(B))\right] \ge \gamma$ where the probability is taken over the choice of π from the smooth distribution $\mathcal{D}(B)$ and the coin tosses of Decode*. (Note that the parameter p is implicitly used in bias of the random choice of the function h.)*

We now set out to prove Lemmas 9 and 10. The following fact will be very useful in the proofs of the above two Lemmas. (The simple proof of this fact can be found in [7]; the result there is stated for folded B tables, but that was clearly not necessary as the proof only uses Parseval's identity.)

Lemma 11 *Let $B : \mathcal{F}_{[7]^u} \to \{1, -1\}$, and let $\mathcal{D}(B)$ be a smooth distribution on projection functions $\pi : [7]^u \to [2]^u$. Then, for any p, $0 < p < 1$,*

$$\mathop{\mathbf{E}}_{\pi \in_R \mathcal{D}(B)}\Big[\sum_{\beta : |\beta| \ge K} \hat{B}_\beta^2 (1-p)^{|\pi(\beta)|}\Big] \le \delta$$

provided $K \ge 2^{\Omega(p^{-1} \ln \delta^{-1})}$.

We now prove Lemmas 9 and 10. The proofs follow the ones in [7] closely except for some parts where additional arguments are used to take care of the fact that B is not assumed to be folded.

Proof of Lemma 9: Using the Fourier expansions of $B(g)$ and $B(-g \cdot (f \circ \pi \wedge h))$ as in Equation (2), the properties of linear functions, and using linearity of expectation, we transform the given expectation, for each fixed p, into

$$\sum_{\beta_1,\beta_2} \hat{B}_{\beta_1}\hat{B}_{\beta_2} \mathop{\mathbf{E}}_{\pi,f,g,h}\Big[-(-1)^{|\beta_2|} l_{\beta_1 \Delta \beta_2}(g) l_{\beta_2}(f \circ \pi \wedge h)\Big].$$

[4] In the case when A is folded, the Fourier coefficient $\hat{A}_\emptyset = 0$, and so this case will never arise.

Since g is picked uniformly and independently at random, the inner expectation is 0 unless $\beta_1 = \beta_2 = \beta$. Let us take expectation fixing π also for the moment. For $x \in \pi(\beta)$, we define $\beta_x = \{y \in \beta : \pi(y) = x\}$. We need to compute, for each β,

$$\begin{aligned}
\mathop{\mathbf{E}}_{f,g,h}\Big[(-1)^{|\beta|+1}\prod_{y\in\beta}(f(\pi(y)) \wedge h(y))\Big] &= \prod_{x\in\pi(\beta)}(-1)^{|\beta|+1}\mathop{\mathbf{E}}_{f,h}\Big[\prod_{y\in\beta_x}(f(x)\wedge h(y))\Big] \\
&= \prod_{x\in\pi(\beta)}(-1)^{|\beta|+1}\Big(\frac{1}{2}+\frac{1}{2}\cdot(2p-1)^{|\beta_x|}\Big).
\end{aligned}$$

For β with $|\beta|$ even, this expectation is clearly negative (since $p > 0$). Hence we only worry about terms with $|\beta|$ odd, and we need to upper bound the sum

$$\sum_{\beta:|\beta|\ \text{odd}}\prod_{x\in\pi(\beta)}\Big(\frac{1}{2}+\frac{1}{2}\cdot(2p-1)^{|\beta_x|}\Big).$$

For each fixed p the we need to estimate the expectation:

$$\mathop{\mathbf{E}}_{\pi}\Big[\sum_{\beta:|\beta|\ \text{odd}} -\hat{B}_\beta^2\prod_{x\in\pi(\beta)}\big(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\big)\Big]\,. \tag{3}$$

One hopes to estimate this sum as a function of p which tends to 0 as p tends to 0. Unfortunately such a estimate does not exist; it turns out, however, that it is easy to bound the expectation of Equation (3) above for small β and large β and this is very useful as it will allow us to vary p as per some distribution so that one can bound (3) (actually even its absolute value) when we take expectations over p as well. Let $c > 1$ be a small absolute constant to be determined. We have

Claim 1 *For each fixed $p > 0$ and B,*

$$\Big|\mathop{\mathbf{E}}_{\pi}\Big[\sum_{\beta:|\beta|\ odd}\hat{B}_\beta^2\prod_{x\in\pi(\beta)}\big(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\big)\Big]\Big| \le 2p^{1/2} + \sum_{\beta|p^{-1/2}<|\beta|<p^{-c/p}}\hat{B}_\beta^2\,.$$

Proof: We split the sum of Equation (3) into three parts depending upon which of the three disjoint intervals $[1, p^{-1/2}]$, $(p^{-1/2}, p^{-c/p})$ and $[p^{-c/p}, \infty)$, $|\beta|$ lies in. The middle interval need not be estimated as it appears on the right hand side of the estimate stated in the Claim.

Let us now consider β with $|\beta|$ small, i.e $|\beta| \le p^{-1/2}$. Since $|\beta|$ is odd, there must exist $x \in \pi(\beta)$ with $|\beta_x|$ odd (also $|\beta_x| \le |\beta| \le p^{-1/2}$). For such an x

$$0 \ge \big(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\big) \ge \frac{1}{2}\big(-1+(1-2p|\beta_x|)\big) = -p|\beta_x| \ge -p^{1/2}\,. \tag{4}$$

Hence, when $|\beta| \le p^{-1/2}$,

$$\Big|\prod_{x\in\pi(\beta)}\Big(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\Big)\Big| \le p^{1/2}$$

(using (4) and the fact that all factors are bounded by 1 in absolute value), and so

$$\Big|\sum_{\beta:|\beta|\le p^{-1/2}}\hat{B}_\beta^2\prod_{x\in\pi(\beta)}\Big(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\Big)\Big| \le \sum_{\beta:|\beta|\le p^{-1/2}}\hat{B}_\beta^2 p^{1/2} \le p^{1/2}\,. \tag{5}$$

Let us now consider the β's with $|\beta| \geq p^{-c/p}$. For these the relevant expectation to bound is:

$$\begin{aligned}
\left|\mathop{\mathbf{E}}_{\pi}\Big[\sum_{\beta:|\beta|\geq p^{-c/p}} \hat{B}_\beta^2 \prod_{x\in\pi(\beta)} (\frac{(-1)^{|\beta_x|}}{2} + \frac{(1-2p)^{|\beta_x|}}{2})\Big]\right| &\leq \mathop{\mathbf{E}}_{\pi}\Big[\sum_{\beta:|\beta|\geq p^{-c/p}} \hat{B}_\beta^2 (1-p)^{|\pi(\beta)|}\Big] \\
&= \sum_{\beta:|\beta|\geq p^{-c/p}} \hat{B}_\beta^2 \mathop{\mathbf{E}}_{\pi}[(1-p)^{|\pi(\beta)|}] \\
&\leq \sum_{\beta:|\beta|\geq p^{-c/p}} \hat{B}_\beta^2 p^{1/2} \\
&\leq p^{1/2} \,. \qquad (6)
\end{aligned}$$

where the last but one step follows using Lemma 11 together with an appropriate choice of the (absolute) constant $c > 1$.

The statement of our Claim now follows from Equations (5) and (6). $\square$ (*Claim 1*)

All that remains to be done is to now take expectations over the choice of p as well in (3). Recall that p is drawn uniformly at random from $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_t\}$ for $t = \lceil \delta^{-1} \rceil$ and where $\varepsilon_i = \varepsilon_{i-1}^{2c/\varepsilon_{i-1}}$ for $1 < i \leq t$. Hence using Claim 1, we have

$$\begin{aligned}
\left|\mathop{\mathbf{E}}_{p,\pi}\Big[\sum_{\beta:|\beta| \text{ odd}} -\hat{B}_\beta^2 \prod_{x\in\pi(\beta)} (\frac{(-1)^{|\beta_x|}}{2} + \frac{(1-2p)^{|\beta_x|}}{2})\Big]\right| &\leq \mathop{\mathbf{E}}_{p}[2p^{1/2} + \sum_{\beta:p^{-1/2}<|\beta|<p^{-c/p}} \hat{B}_\beta^2] \\
&\leq 2\varepsilon_1^{1/2} + \frac{1}{t},
\end{aligned}$$

where the last step follows because $\varepsilon_1 > \varepsilon_i$ for $1 < i \leq t$ and because the ranges of $|\beta|$ for the different values of p are disjoint and $\sum_\beta \hat{B}_\beta^2 = 1$. Since $\varepsilon_1 = \delta^2$ and $1/t \leq \delta$, $2\varepsilon_1^{1/2} + 1/t \leq 3\delta$, and the Lemma follows. $\square$ (*Lemma 9*)

Proof of Lemma 10: Using the Fourier expansions of $A^\pi(f)$, $B(g)$ and $B(-g\cdot(f\circ\pi \wedge h))$ as in Equation (2), the properties of linear functions, and using linearity of expectation, we transform the given expectation, for each fixed π, into

$$\sum_{\alpha,\beta_1,\beta_2} \hat{A}_\alpha \hat{B}_{\beta_1} \hat{B}_{\beta_2} \mathop{\mathbf{E}}_{f,g,h}\Big[-(-1)^{|\beta_2|} l_\alpha(f) l_{\beta_1\Delta\beta_2}(g) l_{\beta_2}(f\circ\pi\wedge h)\Big].$$

(We have omitted the superscript π on A for notational convenience.)

Since g is picked uniformly and independently at random, the inner expectation is 0 unless $\beta_1 = \beta_2 = \beta$ (say). Since f is also picked uniformly and independently at random, we can also conclude $\alpha \subseteq \pi(\beta)$. Using this, our expression simplifies, once a π is fixed, to

$$\begin{aligned}
\mathop{\mathbf{E}}_{f,g,h}\Big[-A(f)B(g)B(-g\cdot(f\circ\pi\wedge h))\Big] &= \sum_{\beta,\ \alpha\subseteq\pi(\beta)} \hat{A}_\alpha \hat{B}_\beta^2 (-1)^{|\beta|+1} \mathop{\mathbf{E}}_{f,h}\Big[l_\alpha(f) l_\beta(f\circ\pi\wedge h)\Big] \\
&= \sum_{\substack{\beta \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha \hat{B}_\beta^2 (-1)^{|\beta|+1} \prod_{x\in\alpha} \mathbf{E}\Big[f(x) \prod_{y\in\beta_x} (f(x)\wedge h(y))\Big]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\substack{\beta \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha \hat{B}_\beta^2 (-1)^{|\beta|+1} \prod_{x\in\alpha}\left(\frac{1}{2}+\frac{-(2p-1)^{|\beta_x|}}{2}\right) \\
&\qquad \prod_{x\in\pi(\beta)\setminus\alpha}\left(\frac{1}{2}+\frac{(2p-1)^{|\beta_x|}}{2}\right) \\
&= -\sum_{\substack{\beta \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha \hat{B}_\beta^2 \prod_{x\in\alpha}\left(\frac{(-1)^{|\beta_x|}}{2}-\frac{(1-2p)^{|\beta_x|}}{2}\right) \\
&\qquad \prod_{x\in\pi(\beta)\setminus\alpha}\left(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\right) \qquad (7)
\end{aligned}
$$

Define $h(\alpha,\beta)$ to be the quantity

$$
\prod_{x\in\alpha}\left(\frac{(-1)^{|\beta_x|}}{2}-\frac{(1-2p)^{|\beta_x|}}{2}\right)\prod_{x\in\pi(\beta)\setminus\alpha}\left(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\right)
$$

It is not difficult to see that

$$
\begin{aligned}
\sum_{\alpha\subseteq\pi(\beta)} h^2(\alpha,\beta) &= \prod_{x\in\pi(\beta)}\left[\left(\frac{(-1)^{|\beta_x|}}{2}-\frac{(1-2p)^{|\beta_x|}}{2}\right)^2+\left(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\right)^2\right] \\
&\leq (1-p)^{|\pi(\beta)|} \qquad (8)
\end{aligned}
$$

The last step follows from the fact that if $|a|,|b|\leq 1-p$ and $|a|+|b|=1$, then $a^2+b^2\leq(1-p)$. It is also easy to see that

$$
\sum_{\alpha\subseteq\pi(\beta)}|h(\alpha,\beta)|=1. \qquad (9)
$$

Using this with Equation (7), we have, for each fixed π and p,

$$
\begin{aligned}
\mathop{\mathbf{E}}_{f,g,h}\Big[-A(f)B(g)B(-g\cdot(f\circ\pi\wedge h))\Big] &\leq \sum_{\substack{\beta \\ \alpha\subseteq\pi(\beta)}} \hat{B}_\beta^2\Big(|\hat{A}_\alpha||h(\alpha,\beta)|\Big) \\
&\leq \sum_{\beta:|\beta|\geq K}\hat{B}_\beta^2\Big(\sum_{\alpha\subseteq\pi(\beta)}\hat{A}_\alpha^2\Big)^{1/2}\Big(\sum_{\alpha\subseteq\pi(\beta)}h^2(\alpha,\beta)\Big)^{1/2} + \\
&\quad + \sum_{\substack{\beta:|\beta|\leq K \\ \alpha\subseteq\pi(\beta),|\hat{A}_\alpha|\leq\delta/4}} \hat{B}_\beta^2|\hat{A}_\alpha||h(\alpha,\beta)| + \\
&\quad + \sum_{\substack{\beta:|\beta|\leq K \\ \alpha\subseteq\pi(\beta),|\hat{A}_\alpha|\geq\delta/4}} \hat{B}_\beta^2|\hat{A}_\alpha||h(\alpha,\beta)| \\
&\leq \sum_{\beta:|\beta|\geq K}\hat{B}_\beta^2\Big(\sum_{\alpha\subseteq\pi(\beta)}h^2(\alpha,\beta)\Big)^{1/2}+\frac{\delta}{4}\sum_{\beta:|\beta|\leq K}\hat{B}_\beta^2 + \\
&\quad + \frac{4}{\delta}\sum_{\substack{\beta:|\beta|\leq K \\ \alpha\subseteq\pi(\beta),|\hat{A}_\alpha|\geq\delta/4}}\hat{A}_\alpha^2\hat{B}_\beta^2 \text{ (using Equation 9))}
\end{aligned}
$$

$$\leq \sum_{\beta:|\beta|\geq K} \hat{B}_\beta^2(1-p)^{|\pi(\beta)|/2} + \frac{\delta}{4} + \frac{4}{\delta} \sum_{\substack{\beta:|\beta|\leq K \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha^2 \hat{B}_\beta^2 \quad (10)$$

where the last step follows using (8). We now take expectations over the projection π (which is picked from a smooth distribution $\mathcal{D}(B)$). Using Lemma 11 we conclude

$$\mathop{\mathbf{E}}_{\pi}\Big[\sum_{\beta:|\beta|\geq K} \hat{B}_\beta^2(1-p)^{|\pi(\beta)|/2}\Big] \leq \frac{\delta}{4}, \quad (11)$$

provided $K = 2^{\Omega(p^{-1}\ln\delta^{-1})}$. Such a choice of K is possible provided $7^u \geq K$, and we assume that u is large enough for this purpose. Now, combining (10) and (11), together with the hypothesis of the Lemma that $\mathop{\mathbf{E}}_{\pi,f,g,h}[-A^\pi(f)B(g)B(-g\cdot(f\circ\pi\wedge h))] \geq \delta$, we get

$$\mathop{\mathbf{E}}_{\pi}\Big[\sum_{\substack{\beta:|\beta|\leq K \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha^2 \hat{B}_\beta^2\Big] \geq \frac{\delta}{4}\cdot\frac{\delta}{2} = \frac{\delta^2}{8}\,. \quad (12)$$

We now estimate the probability of success of the decoding strategy.

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\pi,\text{coin tosses of }\mathsf{Decode}}[\mathsf{Decode}(A^\pi) = \pi(\mathsf{Decode}(B))] &\geq \mathop{\mathbf{E}}_{\pi}\Big[\sum_{\substack{\alpha,\beta \\ \alpha\cap\pi(\beta)\neq\emptyset}} \hat{A}_\alpha^2\hat{B}_\beta^2\frac{1}{|\alpha|}\frac{1}{|\beta|}\Big] \\ &\geq \mathop{\mathbf{E}}_{\pi}\Big[\sum_{\substack{\beta:\beta\leq K \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha^2\hat{B}_\beta^2\frac{1}{K^2}\Big]\,. \quad (13)\end{aligned}$$

(The last step above is valid because when $\alpha \subseteq \pi(\beta)$, we may assume $\alpha\cap\pi(\beta) = \alpha \neq \emptyset$ as $\hat{A}_\emptyset = 0$. It is critical for this step that A is folded.)
Combining (12) and (13) we get

$$\mathop{\mathbf{Pr}}_{\pi,\text{coin tosses of }\mathsf{Decode}}[\mathsf{Decode}(A^\pi) = \pi(\mathsf{Decode}(B))] \geq \frac{\delta^2}{8K^2}. \quad (14)$$

Recalling that $K = 2^{\Omega(p^{-1}\ln\delta^{-1})}$, the success probability depends only on p, δ, and the proof is complete. □ (*Lemma 10*)