Electronic Colloquium on Computational Complexity, Report No. 43 (1999)

The Approximability of Set Splitting Problems and Satisfiability Problems with no Mixed Clauses

Venkatesan Guruswami*

MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 01239.
Email: venkat@theory.lcs.mit.edu

October, 1999

Abstract

We prove hardness results for approximating set splitting problems and also instances of satisfiability problems which have no "mixed" clauses, i.e all clauses have either all their literals unnegated or all of them negated. Results of Håstad [8] 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 problem MAX E3-SET SPLITTING in which sets have size exactly 3, and prove, by constructing simple gadgets from 3-parity constraints, that MAX E3-SET SPLITTING is hard to approximate within any factor strictly better than 21/22. We also prove that even satisfiable instances of MAX E3-SET SPLITTING are NP-hard to approximate better than 27/28; this latter result uses a recent PCP construction of [7]. We give a PCP construction which is a variant of the one in [7] and use it to prove a hardness result of $11/12 + \varepsilon$ for approximating non-mixed MAX 3SAT, and also a hardness of $15/16 + \varepsilon$ for the version where each clause has *exactly* 3 literals (as opposed to up to 3 literals).

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 (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 fixed $k \geq 2$, MAX Ek-SET

*Research supported by an IBM Graduate Fellowship.

ISSN 1433-8092

SPLITTING was shown to be NP-hard by Lovász [12]. 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 $\varepsilon > 0$ [8, 15], and Goemans and Williamson, in a major breakthrough, used semidefinite programming to give a factor 0.878-approximation algorithm for MAX CUT [5]. 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.

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 [13].

Following the striking inapproximability results of Håstad [8], 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 [15] 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 [8]. In the same paper [8], 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. 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 [13, 16]. On the algorithmic side, the best known approximation algorithm, due to Zwick [17], achieves a ratio of 0.908. (This bound is as yet only based on numerical evidence, the best proven bound is 0.87868 [10, 16], 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 [16].

Turning again to set splitting problems, recently, Håstad, in the final version of [8], proves the tight result that it is NP-hard to approximate (even satisfiable instances) MAX E4-SET SPLITTING within $7/8 + \varepsilon$, for any $\varepsilon > 0$. (The result is *tight*, because a random partition will split a $\frac{7}{8}$th fraction of the sets.)

For MAX E3-SET SPLITTING, by the results of Zwick [16, 17] 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 appears to be correct. Inapproximability within a factor of (approximately) 0.987 is claimed in [9]; and it is mentioned in [1] that the 9-gadget reducing a PC_0 constraint to 3-Set Splitting constraints that appears in [9] (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 [9, 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. The question we therefore address is whether it is possible to do the same for MAX E3-SET SPLITTING, and we are in fact able to achieve this by paying an extra cost of 4 for the PC_1 gadget.

In light of the work of Trevisan *et al.* [15] on methods for finding optimal gadgets, it is interesting to ask why we cannot use their techniques to search for and find the optimal gadgets for set splitting, say for example, an optimal gadget reducing PC_0 to MAX E3-SET SPLITTING. It turns out that it is not possible to guarantee an optimal gadget by the means in [15], because 3-Set splitting constraints are not hereditary (a constraint family is hereditary if identifying two variables of a function results in a new function that is either in the family or is the all 0 or all 1 function; see [15] for details). The linear programs involved in getting the best gadget in even some reasonable subclass of gadgets are too big to be solved[1], and it is probably not worthwhile to pursue this approach as one is not guaranteed a proof of optimality of the gadget anyway.

Remark. It is easy to see that MAX 2-SAT reduces to MAX NAE-E3-SAT, and that MAX E3-SET SPLITTING reduces to MAX CUT in an approximation preserving way. However, the best algorithm known for MAX 2-SAT [4] achieves a ratio of 0.931, while only a weaker ratio of 0.908 [17] is known for MAX NAE-E3-SAT. Similarly, while the best approximation ratio to date for MAX CUT is 0.878 [5], a better factor of 0.908 is known for MAX E3-SET SPLITTING (using the same algorithm as the one for MAX NAE-E3-SAT [17]). It is not clear therefore any of the above two reductions go the opposite way as well, i.e. whether MAX 2-SAT is as hard to approximate as MAX NAE-E3-SAT or whether MAX E3-SET SPLITTING is as hard to approximate as MAX CUT.

1.2 Satisfiability with no Mixed Clauses

The set splitting problem is a special case of NAE-SAT in which in all clauses all literals appear unnegated (or, equivalently, all appear negated). This leads us to consider the corresponding question for the even more fundamental problem of satisfiability where 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, 10]. 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 [8] shows that MAX Ek-SET SPLITTING is hard to approximate within a factor of $1 - 2^{-k+1} + \varepsilon$ for any

[1] This was pointed out to us by Greg Sorkin.

$\varepsilon > 0$. Since there is a trivial 2-gadget reducing MAX Ek-SET SPLITTING to MAX Ek-NM-SAT (namely, replace the constraint 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 the set splitting problems, we design a 13-gadget reducing PC_1 to 3-Set Splitting constraints which can be used in conjunction with the 9-gadget from PC_0, to yield, on combining with the hardness result for MAX 3-PARITY [8], an inapproximability bound of $21/22 + \varepsilon$ for MAX E3-SET SPLITTING. Note that this exactly equals the best known inapproximability bound for MAX 2-SAT; it is not clear whether this is simply a coincidence. This result, however, does not hold for satisfiable instances of MAX E3-SET SPLITTING. For such instances we prove a weaker inapproximability bound of $27/28 + \varepsilon$, for any $\varepsilon > 0$; this result uses a recent PCP construction of [7] together with gadgets reducing the predicates tested in their PCP to 3-Set Splitting constraints.

For MAX Ek-SET SPLITTING with higher values of k, note that the case $k = 4$ is completely resolved by Håstad's results mentioned above. Unlike most other problems like MAX k-PARITY or MAX k-SAT, this result alone does not seem to automatically, via a simple gadget, imply a tight inapproximability result for MAX Ek-SET SPLITTING, $k \geq 5$, as well. It turns out, however, that one can slightly modify the PCP construction of Håstad to also prove a tight hardness result for MAX Ek-SET SPLITTING for $k \geq 5$ (we record this fact without proof in Section 3).

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 [7]; 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 [7] – 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 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. While it is not clear how to exploit the structure of non-mixed SAT to design approximation algorithms with guarantee better than $7/8$ for MAX 3-NM-SAT (or even MAX E3-NM-SAT), proving a (what would be tight) hardness result for approximating better than $7/8$ seems quite difficult as well. Progress in closing this gap should be an exciting direction for future work.

Remark. Our hardness result for MAX E3-SET SPLITTING is proved for the weighted version of the problem where each of the sets has a (small constant) weight and the goal is to find a partition that maximizes the total weight of all the split sets. By using results from [3], the same hardness bound holds for the unweighted version of the problem as well. All the algorithmic guarantees mentioned above hold for the weighted versions of the problems as well.

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, 11, 15][2] : 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 [11, 15] 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 approximate* Max E3-Set Splitting *within a factor of* $21/22 + \varepsilon$.

Proof: The result will be proven by construction of appropriate gadgets that reduce PC_0 and PC_1 to 3-Set Splitting constraints. As is explicitly stated in [15], it follows from the result of Håstad [8], that for any family $\mathcal{F}$ of constraints, if there exists an α_0 gadget reducing PC_0 to $\mathcal{F}$ and an α_1 gadget reducing PC_1 to $\mathcal{F}$, then, for any $\varepsilon > 0$, Max $\mathcal{F}$ is NP-hard to approximate within $1 - \frac{1}{\alpha_0+\alpha_1} + \varepsilon$. For Max E3-Set Splitting, we will give a 9-gadget from PC_0 and a 13-gadget from PC_1; using the above result, this will imply an hardness of $21/22 + \varepsilon$ for Max E3-Set Splitting.

It remains to construct the gadgets for Max E3-Set Splitting. The gadget, in addition to the usual *primary* and *auxiliary* variables, will use a special auxiliary element T which will be *shared* by all the gadgets for the various parity constraints. The special element T is for the purpose of interpreting variables as either true or false depending on whether they fall in the same side or the opposite side of T in the partition of the universe.

We first describe the gadget from PC_0 (this is the same gadget presented in [9]). For a constraint $a \oplus b \oplus c = 0$, the gadget will consist of the 3-sets $\{Tab\}, \{Tbc\}, \{Tca\}, \{abc\}$, each with weight $1/3$, and the 3-sets $\{abx\}$, $\{bcx\}$, $\{cax\}$, $\{aTx\}$, $\{bTx\}$ and $\{cTx\}$ each with a weight of $4/3$. We claim that this is a 9-gadget. Indeed, if a, b, c satisfy $a \oplus b \oplus c = 0$, then, interpreting T as "true", exactly three of the elements $\{T, a, b, c\}$ are in one side of the partition which places all variables set to true along with T on one side, and the remaining variables (that are set to false) on the other side. Thus this partition splits exactly three of the sets $\{Tab\}, \{Tbc\}, \{Tca\}, \{abc\}$, and we can place x in the side of the partition that has only one element of $\{T, a, b, c\}$ to split all the sets that contain x. This partition therefore splits sets with total weight $6 \times \frac{4}{3} + 3 \times \frac{1}{3} = 9$. In any partition that does not split the elements $\{T, a, b, c\}$ as one element in one side and three in the other, the total weight of the split sets cannot be more than 8, as either one of the sets containing x will not be split, or none of the sets $\{Tab\}, \{Tbc\}, \{Tca\}, \{abc\}$ will be split.

We now describe the gadget from PC_1. For a constraint $a \oplus b \oplus c = 1$, the gadget will use the same "global" auxiliary element T as the PC_0 gadget, and other auxiliary elements $x, \bar{a}, p, q, r$ specific to just this gadget. (Note $\bar{a}$ is simply another element of the 3-Set Splitting instance we create, its label refers to the fact that we want it to be on the other side of the element a in the

[2] A bit of history on the formalization of gadgets: Bellare *et al* [2] gave specific examples of gadgets and made explicit the notion of a gadget; the definition of a gadget was formalized in [11] (they called it implementation); and Trevisan *et al* [15] focused on construction of optimal gadgets for various constraint families by casting the search for gadgets as appropriate linear programs.

optimal partition.) The gadget will comprise the same 3-sets and weights as the PC_1 gadget except that the element $\bar{a}$ will be used in place of a. In addition, the gadget will use four 3-sets, each with weight 1, to force $a, \bar{a}$ to be on opposite sides in any "good" partition. These 3-sets are $\{p, q, r\}$, $\{p, a, \bar{a}\}$, $\{q, a, \bar{a}\}$ and $\{r, a, \bar{a}\}$. Using arguments similar to those used for PC_0, it is easy to check that this gives a 13-gadget. □

Remark: Using a "global" auxiliary element F that stands for "false" (analogous to the element T used in the above proof) we can give a 9-gadget from PC_1 to 3-set splitting by simply using F in place of T in the PC_0 gadget. One can then force T and F to be on opposite sides of the partition by using four 3-sets (P, Q, R), (T, F, P), (T, F, Q) and (T, F, R) of large enough weight, and this gives the same inapproximability bound as the above proof.[3] While this method is cleaner and has the advantage of uniformity between the PC_0 and PC_1 gadgets, we adopted the treatment in the proof of Theorem 1 for two reasons: (a) It involves using only small constant weights as opposed to the large weights required above; and (b) it is better to highlight the PC_1 gadget problem if one wants to improve the result.

Theorem 2 *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 $27/28 + \varepsilon$ fraction of the sets.*

Proof: The proof proceeds by constructing suitable gadgets reducing various forms of *monomial basis check constraints* $RMBC_{ij}$ to 3-Set Splitting constraints that preserve satisfiability (see [2] or [15] for definitions of $RMBC_{ij}$ for $i, j \in \{0, 1\}$, we also define it below for completeness). Specifically we will construct 12-gadgets reducing $RMBC_{00}$ and $RMBC_{10}$ to 3-Set Splitting constraints, and 16-gadgets reducing $RMBC_{01}$ and $RMBC_{11}$ to 3-Set Splitting constraints. These gadgets put together give a "gadget" that behaves on the average like a 14-gadget from RMBC to 3-Set Splitting. This gadget, upon using standard arguments together with the recent (adaptive 3-query) PCP construction of [7] that performs a single RMBC test and has perfect completeness and soundness $1/2+\varepsilon$, yields a hardness of approximation within $27/28+\varepsilon$ for *satisfiable* instances of MAX E3-SET SPLITTING.

We now define the constraint families $RMBC_{ij}$ and specify our gadgets. For variables a, b, b', c,

$$\mathrm{RMBC}_{ij}(a, b, b', c) = \begin{cases} 1 & \text{if } a = 0 \text{ and } c = b \oplus i \\ 1 & \text{if } a = 1 \text{ and } c = b' \oplus j \\ 0 & \text{otherwise.} \end{cases}$$

Let us first take an $RMBC_{00}(a, b, b', c)$ constraint. Written in 3CNF form, this constraint is the same as

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

It is now easy to give a 8-gadget from $RMBC_{00}(a, b, b', c)$ to MAX NAE-E3-SAT – the gadget consists of the constraints $NAE(a, b, w)$, $NAE(w, c, \bar{F})$, $NAE(a, c, x)$, $NAE(x, b, \bar{F})$, $NAE(a, c, y)$, $NAE(y, b', F)$, $NAE(a, b', z)$ and $NAE(z, c, F)$. Here w, x, y, z are auxiliary variables specific to this gadget while F is shared auxiliary variable across all gadgets and represents the assignment FALSE. We obtained this gadget by using that fact that $NAE(a, b, r)$, $NAE(\bar{r}, c, F)$ is a 2-gadget from the 3SAT constraint $(a \vee b \vee c)$, and that $NAE(\bar{p}, \bar{q}, \bar{r})$ is equivalent to the constraint $NAE(p, q, r)$. Note that all of the above eight NAE constraints except two of them which use $\bar{F}$ are *monotone*

[3]We thank Johan Håstad who also pointed this out.

and can therefore directly be used as 3-Set Splitting constraints. The way to handle the $\bar{F}$ is the same as the one used in the proof of Theorem 1 to ensure two elements $a, \bar{a}$ are in opposite sides of the partition. We create a copy of the element $\bar{F}$ *just for this gadget* (and **not** shared across gadgets, unlike F), and also further auxiliary elements p, q, r together with the four 3-sets $\{F, \bar{F}, p\}$, $\{\bar{F}, F, q\}$, $\{\bar{F}, F, r\}$ and $\{p, q, r\}$. Our final gadget will then contain these four 3-sets and the eight NAE constraints written as 3-Set Splitting constraints (with the element $\bar{F}$ taking the place of negation of the variable F in the NAE constraints). It is straightforward to check that this will be a 12-gadget from RMBC_{00} to MAX E3-SET SPLITTING.

We next describe the 16-gadget from RMBC_{11}. Note that $\text{RMBC}_{00}(a, b, b', c) \equiv \text{RMBC}_{11}(a, b, b', \bar{c})$, and hence we can get a 16-gadget from RMBC_{11} by using the 12-gadget described above from RMBC_{00} together with an extra four 3-sets to force the "element" $\bar{c}$ of the set splitting constraints (that will simulate the literal $\bar{c}$ in the RMBC constraint) to be on opposite side of c.

Similarly, by noting that $\text{RMBC}_{01}(a, b, b', c) \equiv \text{RMBC}_{00}(a, b, \bar{b'}, c)$ and that $\text{RMBC}_{10}(a, b, b', c) \equiv \text{RMBC}_{00}(a, \bar{b}, b', c)$, we can get 16-gadgets from RMBC_{10} and RMBC_{o1} as well. It turns that we can in fact do better for RMBC_{10}, as we describe below.

Written in 3CNF form, $\text{RMBC}_{10}(a, b, b', c)$ is equivalent to

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

Thus we have the 8-gadget to MAX NAE-E3-SAT comprising of the constraints: $\text{NAE}(b, c, w)$, $\text{NAE}(a, w, F)$, $\text{NAE}(a, b, x)$, $\text{NAE}(c, \bar{x}, F)$, $\text{NAE}(a, c, y)$, $\text{NAE}(b', y, F)$, $\text{NAE}(a, b', z)$ and $\text{NAE}(c, z, F)$. As usual, we can "simulate" the literal $\bar{x}$ with an element $\bar{x}$ and four new 3-Set Splitting constraints. This gives us a 12-gadget just as in the RMBC_{00} case, and we are done. □

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

Håstad [8] 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 imply a tight hardness result for MAX Ek-SET SPLITTING for $k \geq 5$ by just using a gadget, one can, however, easily 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 ([8]) *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.*

4 Hardness of approximating MAX 3-NM-SAT

We will prove the following theorems in this section.

Theorem 4 *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 5 *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 6 *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 [7, 6]; we will be heavily relying on the treatment and terminology of [6]. 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 form, one starts with an *outer proof system* which is a *2-Prover 1-Round proof system* (2P1R) construction for NP due to Raz [14]. 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 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. Thus for any soundness parameter $s < 1$, one can design a 2P1R with completeness 1 and soundness at most s and answers of sizes u and $3u$ bits respectively from the two provers P_1 and P_2 for a *constant* u (that depends on s, actually $u = O(\log 1/s)$ suffices).

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 [6] for details, we delve into the specification of our inner verifier. [End Interlude]

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 $A : \mathcal{F}_{[2]^u} \to \{1, -1\}^4$ 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.

[4]The notation $\mathcal{F}_D$ stands for the space of all functions $f : D \to \{1, -1\}$ and $[n]$ stands for the set $\{1, , 2, \ldots, n\}$. We also use $\{1, -1\}$ for representing boolean values, with 1 standing for FALSE and -1 for TRUE.

`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 $B(g) \neq B(-g(-f \circ \pi \wedge h))$

`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 $\text{B-MBC}_p{}^{A,B}\ (u,\pi)$.

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

B) is the long code of a (resp. b) which satisfy $\pi(b) = a$. The formal specification of our inner verifier $\mathrm{IV3}_\delta{}^{A,B}\ (u,\pi)$ is given in Figure 1 (the constant c used in its specification is an absolute (small) constant which can be figured out from our proofs).

The only difference of this inner verifier from the one in [7] 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 $1/2 + \varepsilon$.[5] 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 PCP's acceptance predicate 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. It therefore remains 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 $\mathrm{IV3}_\delta$ in [8, 6]. The only change is that we must now rework the proofs without assuming that B is folded.

[5]The exact definition of soundness of an inner verifier turns out to be a tricky issue, we once again refer the reader to [7, 6] for the definitions.

We now estimate the acceptance probability of our inner verifier. 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[6]. The acceptance probability of $V_p^{A,B}(u,\pi)$ is $\mathbf{E}[X]$, which equals

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

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

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

(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", the decoding strategy described earlier produces "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 [7, 6], 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 6.1 and 6.2) are proved (the reader can find the exact and full details of why these lemmas suffice to complete the proof in the description of [6]). 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 $A : \mathcal{F}_D \to \mathcal{R}$

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

(with respect to the above dot product). This implies that for any such function A we have the *Fourier expansion*

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

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 equality), and then returning a random element of α. If $\alpha = \emptyset$,* Decode *simply returns some fixed element of D.*[7]

Lemma 6.1 *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 $B : \mathcal{F}_{[7]^u} \to \{-1,1\}$,*

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

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 6.2 *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*

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

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.)*

The proof of Theorem 6 is now complete modulo the proofs of Lemmas 6.1 and 6.2. □ (*Theorem 6*)

We now set out to prove Lemmas 6.1 and 6.2. 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 [6]; the result there is stated for folded B tables, but that was clearly not necessary as the proof only uses Parseval's identity.)

Lemma 6.3 *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 $K \ge 2^{\Omega(p^{-1} \ln \delta^{-1})}$.

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

We now prove Lemmas 6.1 and 6.2. The proofs follow the ones in [6] 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 6.1: 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

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

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

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

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

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

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,*

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

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

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

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

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

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

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

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

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

where the last but one step follows using Lemma 6.3 together with an appropriate choice of the (absolute) constant $c > 1$.

The statement of our Claim now follows from Equations (5) and (6). □ (*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 $1 < i \le t$. Hence using Claim 1, we have

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

where the last step follows because $\varepsilon_1 > \varepsilon_i$ for $1 < i \le t$ 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 $1/t \le \delta$, $2\varepsilon_1^{1/2} + 1/t \le 3\delta$, and the Lemma follows. □ (*Lemma 6.1*)

Proof of Lemma 6.2: 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

$$\begin{aligned}
\mathop{\mathbf{E}}_{f,g,h}\Big[-A(f)B(g)B(-g\cdot(f\circ\pi\wedge h))\Big] &= \sum_{\beta,\ \alpha\subseteq\pi(\beta)} \hat{A}_\alpha \hat{B}_\beta^2 (-1)^{|\beta|+1} \mathop{\mathbf{E}}_{f,h}\Big[l_\alpha(f) l_\beta(f\circ\pi\wedge h)\Big] \\
&= \sum_{\substack{\beta \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha \hat{B}_\beta^2 (-1)^{|\beta|+1} \prod_{x\in\alpha} \mathbf{E}\Big[f(x)\prod_{y\in\beta_x}(f(x)\wedge h(y))\Big] \\
&= \sum_{\substack{\beta \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha \hat{B}_\beta^2 (-1)^{|\beta|+1} \prod_{x\in\alpha}\left(\frac{1}{2}+\frac{-(2p-1)^{|\beta_x|}}{2}\right) \\
&\qquad \prod_{x\in\pi(\beta)\setminus\alpha}\left(\frac{1}{2}+\frac{(2p-1)^{|\beta_x|}}{2}\right) \\
&= -\sum_{\substack{\beta \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha \hat{B}_\beta^2 \prod_{x\in\alpha}\left(\frac{(-1)^{|\beta_x|}}{2}-\frac{(1-2p)^{|\beta_x|}}{2}\right) \\
&\qquad \prod_{x\in\pi(\beta)\setminus\alpha}\left(\frac{(-1)^{|\beta_x|}}{2}+\frac{(1-2p)^{|\beta_x|}}{2}\right) \qquad (7)
\end{aligned}$$

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

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

The last step follows from the fact that if $|a|,|b| \le 1-p$ and $|a|+|b| = 1$, then $a^2+b^2 \le (1-p)$. 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,

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

$$
\begin{aligned}
& + \sum_{\substack{\beta:|\beta|\le K \\ \alpha\subseteq\pi(\beta),|\hat{A}_\alpha|\ge\delta/4}} \hat{B}_\beta^2 |\hat{A}_\alpha||h(\alpha,\beta)| \\
\le & \sum_{\beta:|\beta|\ge K} \hat{B}_\beta^2 \Big(\sum_{\alpha\subseteq\pi(\beta)} h^2(\alpha,\beta)\Big)^{1/2} + \frac{\delta}{4} \sum_{\beta:|\beta|\le K} \hat{B}_\beta^2 + \\
& + \frac{4}{\delta} \sum_{\substack{\beta:|\beta|\le K \\ \alpha\subseteq\pi(\beta),|\hat{A}_\alpha|\ge\delta/4}} \hat{A}_\alpha^2 \hat{B}_\beta^2 \text{ (using Equation 9))} \\
\le & \sum_{\beta:|\beta|\ge K} \hat{B}_\beta^2 (1-p)^{|\pi(\beta)|/2} + \frac{\delta}{4} + \frac{4}{\delta} \sum_{\substack{\beta:|\beta|\le K \\ \alpha\subseteq\pi(\beta)}} \hat{A}_\alpha^2 \hat{B}_\beta^2 \quad (10)
\end{aligned}
$$

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 6.3 we conclude

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

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

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

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

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

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

Combining (12) and (13) we get

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

Recalling that $K = 2^{\Omega(p^{-1}\ln\delta^{-1})}$, the success probability depends only on p,δ, and the proof is complete. □ (*Lemma 6.2*)

Proof of Theorem 4: 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 $(\bar{t}\vee\bar{c}\vee\bar{c})$. Starting from a hard instance of E3SAT (3SAT with all clauses having *exactly* three literals) as in Theorem 6, 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 4 follows. $\square$ (*Theorem 4*)

Proof of Theorem 5: 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 6, this implies a hardness of approximating (even satisfiable instances of) MAX E3-NM-SAT within any factor strictly greater than 15/16. $\square$ (*Theorem 5*)

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 0.955 approximation). A similar situation exists with MAX CUT where an 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 $7/8 + \varepsilon$, though an approach which could potentially establish such a result has eluded us.

We close by mentioning an open question that is related to MAX E4-SET SPLITTING. If the goal is only to split as many sets as possible, then Håstad's result gives that the best one can do is a factor of 7/8. One can ask, more specifically, that we wish to maximize the number of 4-sets which have a 1-3 split under the partition, and similarly maximize the number of sets that have a 2-2 split under the partition. It turns out that the former problem can be cast as a system of linear equations over GF(2) each of the form $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} \oplus x_{i_4} = 1$, and by yet another powerful result in [8], 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 tight). The 2-2 splitting problem, however, turns out to be another instance where a tight result is still unknown. It is easy to see that our hardness result for MAX E3-SET SPLITTING implies that this problem is hard to approximate within a factor better than 21/22; no approximation algorithm achieving better performance ratio than reducing the problem to MAX E3-SET SPLITTING or picking a random solution, and returning the better of the two solutions, seems to be known for this problem.

Acknowledgments

I would like to thank Greg Sorkin, Madhu Sudan and Uri Zwick for useful discussions. Thanks also to Johan Håstad whose tight hardness result for 4-set splitting prompted this work.

References

[1] Gunnar Andersson and Lars Engebretsen. Better approximation algorithms for Set Splitting and Not-All-Equal Sat. *Information Processing Letters*, 65:305-311, 1998.

[2] Mihir Bellare, Oded Goldreich and Madhu 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] Pierluigi Crescenzi, Riccardo Silvestri and Luca 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] Uri Feige and Michel 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] Michel Goemans and David Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115-1145, 1995.

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

[7] Venkatesan Guruswami, Daniel Lewin, Madhu Sudan and Luca Trevisan. A tight characterization of NP with 3 query PCPs. *ECCC Technical Report* TR98-034, 1998. Preliminary Version in *Proc. of FOCS'98.*

[8] Johan Håstad Some optimal inapproximability results. Technical Report TR97-37, Electronic Colloquium on Computational Complexity, 1997. Preliminary version in *Proc. of STOC'97.*

[9] Viggo Kann, Jens Lagergren and Alessandro Panconesi. Approximability of maximum splitting of k-sets and some other APX-complete problems. *Information Processing Letters*, 58:105-110, 1996.

[10] Howard Karloff and Uri Zwick. A $(7/8 - \varepsilon)$-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, 1997.

[11] Sanjeev Khanna, Madhu Sudan and David Williamson. A complete classification of the approximability of maximization problems derived from Boolean constraint satisfaction. *Proc. of the 29th ACM Symposium on Theory of Computing*, 1997.

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

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

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

[15] Luca Trevisan, Greg Sorkin, Madhu Sudan and David Williamson. Gadgets, approximation and linear programming. *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pp. 517-626, 1996.

[16] Uri 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.

[17] Uri 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.*

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'