Inapproximability Results for Set Splitting 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

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).

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 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, 16], 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 [16] 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, 17]. On the algorithmic side, the best known approximation algorithm, due to Zwick [18], achieves a ratio of 0.908. (This bound is as yet only based on numerical evi-

[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.

dence, the best proven bound is 0.87868 [11, 17], 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 [17].

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 [17, 18] 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 [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 gives 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]).

In light of the work of Trevisan *et al.* [16] on methods for finding optimal gadgets, it is natural to ask why we cannot use their techniques to search for and find the optimal gadgets for set splitting from various constraint families. It turns out that it is not possible to guarantee an optimal gadget by the means in [16], 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 [16] for details). The linear programs involved in getting the best gadget in even some reasonable subclass of gadgets are too big to be solved[2], and it is probably not worthwhile to pursue this approach as one is not guaranteed a proof of optimality of the gadget anyway. This makes the question of pin-pointing the approximability of MAX E3-SET SPLITTING all the more intriguing.

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

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

a ratio of 0.931, while only a weaker ratio of 0.908 [18] 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 [18]).

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 question for the even more fundamental problem of satisfiability with 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 $\text{split}(x_1, x_2, \ldots, x_k)$ 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 gadget reducing the predicate $\text{SNE}_4(x, y, z, w) = (x \neq y) \vee (z \neq w)$ to 3-set splitting constraints. This improves the hardness factor of $27/28 + \varepsilon$ for approximating satisfiable instances of MAX E3-SET SPLITTING that was reported in an earlier version of this paper [6], and is in addition a lot simpler.

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, 16]: 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 $\boldsymbol{a} = a_1, a_2, \ldots, a_k$ to the x_i's that satisfies f, there is an assignment $\boldsymbol{b}$ to the y_i's such that a total weight α of the constraints C_j are satisfied by $(\boldsymbol{a}, \boldsymbol{b})$, and if $\boldsymbol{a}$ does not satisfy f, then for every assignment $\boldsymbol{b}$ to the y_i's, the weight of the constraints C_j satisfied by $(\boldsymbol{a}, \boldsymbol{b})$ is at most $\alpha - 1$ (see [12, 16] 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 $19/20+\varepsilon$ 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

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

(no negations allowed) defined on certain 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 $\mathrm{SNE}_4(a, b, c, d)$ to 3-set splitting constraints: Introduce auxiliary variables x, y, z (specific just to this constraint) and replace $\mathrm{SNE}_4(a, b, c, d)$ 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 $\mathrm{SNE}_4(a, b, c, d)$ 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 $\mathrm{SNE}_4(a, b, c, d)$ 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 $\mathsf{SPLIT}(z, y, z)$ is violated. Thus at most four of the five constraints can be satisfied for any assignment to x, y, z if $\mathrm{SNE}_4(a, b, c, d)$ 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. □

3 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 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 **([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.*

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 [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 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] 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 $A : \mathcal{F}_{[2]^u} \to \{1, -1\}$ which is *folded*[3] (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 constant c used in its specification is an absolute (small) constant which can be figured out from our proofs).

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

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

The 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

[3] 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.

$x \neq y$ 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 $1/2 + \varepsilon$.[4] 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. 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 $\mathrm{IV3}_\delta$ 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 can be found in the full version of the paper [6].[5] $\square$ (*Theorem 6*)

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 $(\bar{t} \vee \bar{b} \vee \bar{c})$. 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*)

[4] 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.

[5] The results in this paper on set splitting are stronger than the ones claimed in [6], but the results on non-mixed SAT are identical in both versions.

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. □ (*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 better than 0.95 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 discussing some interesting questions 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 [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 also tight). The 2-2 splitting problem, however, is another instance where a tight result is not known. It is easy to prove, using a simple 3-gadget from Max SNE_4, that this problem is hard to approximate within $11/12 + \varepsilon$ for any $\varepsilon > 0$, even on satisfiable instances. The current best approximation algorithm 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). A significantly better hardness result for the 2-2 splitting problem would be interesting, as it might give insights on how to improve the inapproximability bounds for Max E3-Set

SPLITTING and possibly even MAX CUT – in fact a hardness factor better than 0.794 for the 2-2 splitting problem would immediately improve the current best hardness result for MAX CUT.

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. 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. The approximability of set splitting problems and satisfiability problems with no mixed clauses. *ECCC Technical Report TR-99-043*, 1999.
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. L. Trevisan, G. Sorkin, M. Sudan and D. Williamson. Gadgets, approximation and linear programming. *Proceedings of the 37th FOCS*, pp. 517-626, 1996.
17. 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.
18. 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*.