Scheme-theoretic Approach to Computational Complexity IV. A New Perspective on Hardness of Approximation

Ali Çivril[*]

November 24, 2022

Abstract

We provide a new approach for establishing hardness of approximation results, based on the theory recently introduced by the author. It allows one to directly show that approximating a problem beyond a certain threshold requires super-polynomial time. To exhibit the framework, we revisit two famous problems in this paper. The particular results we prove are:

- $\mathsf{MAX\text{-}3\text{-}SAT}(1, \frac{7}{8} + \epsilon)$ requires exponential time for any constant ϵ satisfying $\frac{1}{8} \geq \epsilon > 0$. In particular, the gap exponential time hypothesis (Gap-ETH) holds.
- $\mathsf{MAX\text{-}3\text{-}LIN\text{-}2}(1 - \epsilon, \frac{1}{2} + \epsilon)$ requires exponential time for any constant ϵ satisfying $\frac{1}{4} \geq \epsilon > 0$.

1 Introduction

The classical approach in computational complexity is to establish relative hardness results, assuming certain hypotheses. The stereotypical example is the assumption $\mathsf{P} \neq \mathsf{NP}$ (now a fact by [2]), which is used to show that solving a problem exactly is NP-complete or NP-hard via the concept of an efficient reduction [3, 6]. The corresponding starting point for establishing hardness results for approximating computational problems is the PCP Theorem [1].

There has been substantial progress in the field of hardness of approximation in the past three decades since the introduction of the PCP Theorem. On a global scale, two important themes have emerged:

1. One needs to construct increasingly sophisticated reductions to establish strong hardness results for problems of interest. In particular, the framework using the Long Code and its variants together with discrete Fourier analysis have been widespread [5].

2. It has been witnessed that the original assumption $\mathsf{P} \neq \mathsf{NP}$ is not enough to yield a meaningful picture of the landscape of approximability of computational problems. One often needs to resort to various stronger assumptions, such as NP does not have quasi-polynomial algorithms, or other more involved conjectures [7].

The introduction of a new approach for computational complexity by the author [2] allows one to establish absolute hardness results, without relying on any hypotheses. This is quite natural from an epistemological point of view: Everything about a computational problem, including its complexity, is already encoded in it. Reductions and hypotheses are external constructs, and there is no inherent reason why we need them. With this in mind, one might even consider the two themes listed above as defects of the classical theory, which almost entirely relies on reductions, and impose the following two:

[*] Atlas University, Computer Engineering Department, Kagithane, Istanbul Turkey, e-mail: ali.civril@atlas.edu.tr

1. It should ideally be possible to establish a direct hardness result for the problem at hand by avoiding convoluted intermediate steps and other problems as much as possible.
2. A hardness result should not necessarily rely on hypotheses. Completeness or hardness results relative to a class are still important, but once we are able to establish an absolute complexity result for a problem, they are mainly of structural interest.

This paper aims to exhibit these themes on two well-known problems. Given an unweighted constraint satisfaction problem $\mathsf{\Pi}$ with optimal value $c \cdot M$, we denote the problem of finding a solution with value at least $s \cdot M$ by $\mathsf{\Pi}(c, s)$, where M is the total number of constraints. Recall that $\mathsf{MAX\text{-}3\text{-}SAT}$ is the problem of finding an assignment to the variables of a 3CNF Boolean formula so as to maximize the number of satisfied clauses. $\mathsf{MAX\text{-}3\text{-}LIN\text{-}2}$ is the problem of finding an assignment to the variables of a linear system over $\mathbb{F}_2$ with each equation consisting of three variables, so that the number of satisfied equations is maximum.

Theorem 1. *There exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\delta > 0$, the problem $\mathsf{MAX\text{-}3\text{-}SAT}(1, \frac{7}{8} + \epsilon)$ cannot be deterministically solved in time less than $2^{(1-\delta)3\epsilon n}$, where n is the number of variables in the $\mathsf{MAX\text{-}3\text{-}SAT}$ instance, and ϵ is a constant, or any monotonically non-increasing function $\epsilon(n)$ satisfying $\frac{1}{8} \geq \epsilon \geq 0$.*

A few notes are in order about the expressiveness of this theorem. Note first that for $\epsilon = 0$, we have $2^{3\epsilon n} = 1$, and the theorem naturally does not imply a strong hardness for $\mathsf{MAX\text{-}3\text{-}SAT}(1, \frac{7}{8})$. For $\epsilon = \frac{1}{8}$, we recover the hardness of $\mathsf{MAX\text{-}3\text{-}SAT}(1, 1)$, i.e. solving $\mathsf{MAX\text{-}3\text{-}SAT}$ exactly from [2]. The theorem also implies a recently introduced hypothesis in [4, 8].

Corollary 2. *The gap-exponential time hypothesis (Gap-ETH) holds against deterministic algorithms, i.e. there exists a constant $\epsilon > 0$ such that $\mathsf{MAX\text{-}3\text{-}SAT}(1, \frac{7}{8} + \epsilon)$ has no $2^{o(n)}$-time deterministic algorithm, where n is the number of variables in the $\mathsf{MAX\text{-}3\text{-}SAT}$ instance.*

Proof. Select $\epsilon = \frac{1}{16}$, so that $2^{3\epsilon n} = 2^{3n/16}$. □

Theorem 1 might also be considered as a *Half-PCP Theorem*, directly establishing an optimal hardness result for $\mathsf{MAX\text{-}3\text{-}SAT}$, albeit by dropping the NP-hardness. Given this, if one is only interested in the complexity of the problem at hand in terms of the minimum number of operations required to solve it, the theorem bypasses the involved proof of the PCP Theorem, establishing hardness of $\mathsf{MAX\text{-}3\text{-}SAT}(1, 1 - \epsilon)$ for some small constant $\epsilon > 0$ together with the reduction given in [5], which amplifies this to $\mathsf{MAX\text{-}3\text{-}SAT}(1, \frac{7}{8} + \epsilon)$. In contrast to these heavy machinery, the proof of this theorem relies only on simple combinatorial reasoning, modulo the Fundamental Lemma of [2]. We would like to finally note that by this theorem, one can select $\epsilon(n) = \frac{(\log n)^{1+\gamma}}{n}$ for some small constant $\gamma > 0$, and can still get super-polynomial hardness results, in a regime which is very close to the tractable case.

Theorem 3. *There exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\delta > 0$, the problem $\mathsf{MAX\text{-}3\text{-}LIN\text{-}2}(1 - \epsilon, \frac{1}{2} + \epsilon)$ cannot be deterministically solved in time less than $2^{(1-\delta)\epsilon n}$, where n is the number of variables in the $\mathsf{MAX\text{-}3\text{-}LIN\text{-}2}$ instance, ϵ is a constant, or any monotonically non-increasing function $\epsilon(n)$ satisfying $\frac{1}{4} \geq \epsilon \geq 0$.*

2 Preliminaries

We assume the reader is familiar with the first two sections of [2]. We repeat some important definitions. All the following definitions are with regard to a computational problem $\mathsf{\Pi}$. A computational problem defined via a set of instances that is a subset of the instances of Π is called a *sub-problem.* A sub-problem Λ is called a *simple sub-problem* if the instances of Λ have the same Hilbert polynomial. Two instances with distinct solution sets are said to be *distinct.* A sub-problem Λ is said to be *homogeneous* if the instances of Λ are pair-wise distinct. Given two distinct instances I_1 and I_2, a computational procedure transforming I_1 to I_2 is called a *unit instance operation.* Two unit instance operations are said to be *distinct* if they result in distinct instances when applied on the same instance. They are said to be *disparate* if they are distinct and one is not a subset of another. In this case we also say that one operation is *disparate from* the other. A sub-problem Λ is said to be *prime* if the unit instance operations between all pairs of its instances are pair-wise disparate.

We define $\tau(\Pi)$ to be the minimum number of *deterministic* operations required to solve Π. Given a prime homogeneous simple sub-problem Λ, we denote the number of instances of Λ by $b(\Lambda)$. Over all such sub-problems Λ, we denote by $\kappa(\Pi)$ the maximum value of $b(\Lambda)$.

Lemma 4. [2] $\tau(\Pi) \geq \kappa(\Pi)$.

3 Proofs of Theorem 1 and Theorem 3

The following theorem implies Theorem 1 by Lemma 4.

Theorem 5. *There exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\delta > 0$, we have*

$$\kappa\left(\mathsf{MAX\text{-}3\text{-}SAT}\left(1, \frac{7}{8} + \epsilon\right)\right) \geq 2^{(1-\delta)3\epsilon n},$$

where n is the number of variables in the 3-SAT *instance, and ϵ is a constant, or any monotonically non-increasing function $\epsilon(n)$ satisfying $\frac{1}{8} \geq \epsilon \geq 0$.*

Proof. We will use the construction from [2]. In particular, we first consider a homogeneous simple sub-problem of 3-SAT with 3^r instances, each having $4r$ variables and $8r$ clauses, for $r \geq 1$, where each instance consists of r *blocks.* For $r = 1$, a block of an instance is defined via 4 variables x_1, x_2, x_3, x_4, and 8 clauses. These instances are pair-wise distinct. Moreover, as proved in [2], the Hilbert polynomials of the instances are the same. Thus, they form a homogeneous simple sub-problem.

In the inductive step of the argument in [2], we introduce 4 new variables $x_{4r+1}, x_{4r+2}, x_{4r+3}, x_{4r+4}$, and 3 new blocks on these variables each consisting of 8 clauses with the exact form as in Table 1. Appending these blocks to each of the 3^r instances of the induction hypothesis, we obtain 3^{r+1}

Clause	Instance 1	Instance 2	Instance 3
1	$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee x_3$
2	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$x_2 \vee \overline{x_3} \vee \overline{x_4}$
3	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_2} \vee x_3 \vee \overline{x_4}$
4	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$
5	$\overline{x_1} \vee x_2 \vee \overline{x_4}$	$\overline{x_1} \vee x_2 \vee \overline{x_4}$	$\overline{x_1} \vee x_2 \vee \overline{x_4}$
6	$x_1 \vee x_3 \vee x_4$	$\overline{x_1} \vee x_3 \vee x_4$	$\overline{x_1} \vee \overline{x_2} \vee x_4$
7	$\overline{x_1} \vee \overline{x_3} \vee x_4$	$x_2 \vee x_3 \vee x_4$	$\overline{x_1} \vee \overline{x_3} \vee x_4$
8	$\overline{x_2} \vee \overline{x_3} \vee x_4$	$\overline{x_2} \vee \overline{x_3} \vee x_4$	$x_2 \vee \overline{x_3} \vee x_4$

Table 1: The clauses of the 3 instances in the base case of the construction

instances. The constructed sub-problem is a homogeneous simple sub-problem. We then make it into a prime homogeneous simple sub-problem by mixing certain literals across blocks, in particular modifying one specific clause of a block depending on which type of instance this block together with its next block are defined via. In doing so, we ensure that x_{4i} is either 0 or 1, which allows a rather neat construction for ensuring that we have a simple sub-problem. We do not repeat this procedure here and refer the reader to the first paper of the series. Suffice it to say that one can construct $\binom{r}{r/2} \cdot 2^{r/2}$ instances forming a prime homogeneous simple sub-problem.

Suppose first that we are only required to satisfy 7 clauses in a block of an instance, so that we may leave one of the clauses unsatisfied. One then easily sees that with this relaxation x_4 does not necessarily belong to the set $\{0,1\}$, which is crucially required to construct a simple sub-problem as argued in [2]. Thus, we cannot derive a strong hardness result for approximating 3-SAT within factor $\frac{7}{8}$, which is the expected case.

Suppose now for an ϵ given in the statement of the theorem, one is required to satisfy at least $\frac{7}{8}+\epsilon$ fraction of the clauses. Then out of $r = n/4$ blocks of a given instance, one must satisfy all the 8 clauses of some $8\epsilon r$ blocks, each of which we call a *special block*. Here we assume $8\epsilon r = 2\epsilon n$ is an integer. The complexity of the problem is then the minimum complexity over all choices of $8\epsilon r$ special blocks out of r. By the argument mentioned above, one can construct for each such choice a prime homogeneous simple sub-problem of size $\binom{8\epsilon r}{4\epsilon r} \cdot 2^{4\epsilon r} = \binom{2\epsilon n}{\epsilon n} \cdot 2^{\epsilon n} > 2^{(1-\delta)3\epsilon n}$ as n tends to infinity by the Stirling approximation. This completes the proof. □

The following theorem implies Theorem 3 by Lemma 4.

Theorem 6. *There exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\delta > 0$, we have*

$$\kappa\left(\mathsf{MAX\text{-}3\text{-}LIN\text{-}2}\left(1-\epsilon, \frac{1}{2}+\epsilon\right)\right) \geq 2^{(1-\delta)\epsilon n},$$

where n is the number of variables in the MAX-3-LIN-2 *instance, ϵ is a constant, or any monotonically non-increasing function $\epsilon(n)$ satisfying $\frac{1}{4} \geq \epsilon \geq 0$.*

Proof. Consider the two instances given in Table 2, defined on the variables x_1, x_2, x_3, x_4. The solution set of Instance 1 is $\{(\alpha_1, \alpha_2, \alpha_1+\alpha_2, \alpha_1+\alpha_2)\}$, where $\alpha_1, \alpha_2 \in \overline{\mathbb{F}}_2$. There is no solution satisfying both of the equations of Instance 2. However, the solution set for Instance 1 satisfies its first equation. In addition, the solution set $\{(\alpha_1, \alpha_2, \alpha_1+\alpha_2, \alpha_1+\alpha_2+1)\}$ satisfies its second equation. For simplicity, we call the type of block defining Instance 1 an A block, the type defining Instance 2 a B block.

Given these and the approach introduced in the proof of Theorem 5, we can construct a sub-problem of MAX-3-LIN-2 with each instance having $1-\epsilon$ fraction of its equations satisfied as follows. Each instance has $r = n/4$ blocks. We consider all possible instances with $2\epsilon r$ B blocks and $r - 2\epsilon r$ A blocks. Here and throughout the proof we assume ϵr is an integer. This construction ensures that there are in total $2r$ equations and $2\epsilon r$ of them cannot be satisfied.

We will use the same argument as in the proof of Theorem 5 to construct a prime homogeneous simple sub-problem. Suppose one is required to satisfy at least $\frac{1}{2}+\epsilon$ fraction of the equations of an

Equation	Instance 1	Instance 2
1	$x_1+x_2+x_3=0$	$x_1+x_2+x_3=0$
2	$x_1+x_2+x_4=0$	$x_1+x_2+x_3=1$

Table 2: The equations of the 2 instances in the base case of the construction

instance. Then there are $r + 2\epsilon r$ equations to be satisfied. Assume without loss of generality that each block already satisfies one of its equations, thus without fixing any of the variables. With this assumption, there remain $2\epsilon r$ equations to be satisfied. Fix $4\epsilon r$ blocks with $2\epsilon r$ A blocks whose both equations are satisfied and $2\epsilon r$ B blocks, and consider the restriction of the instances to these blocks. The Hilbert polynomial defined by these sub-instances are uniform, as they have the same number of A blocks and B blocks. They are also pair-wise distinct by definition, so that we can construct a homogeneous simple sub-problem of size $\binom{4\epsilon r}{2\epsilon r}$. Notice that there are indeed at least $2\epsilon r$ A blocks since $r - 2\epsilon r \geq 2\epsilon r$ by the fact that $\epsilon \leq \frac{1}{4}$.

We now describe a procedure that makes the considered sub-problem into a prime homogeneous simple sub-problem without changing the solution sets. For simplicity, we describe the procedure for $r = 2$. The construction is easily extended to the general case. If the first block is an A block and the second block is also an A block, we replace x_1 in the first equation of the first block with x_5. If the second block is a B block, we replace x_2 in the first equation with x_6. If the first block is a B block and the second block is also a B block, we replace x_2 in the second equation of the first block with x_6. If the second block is an A block, we replace x_1 in the second equation with x_5.

In extending this to the general case in which the operation is to be performed on each block, the second block is generalized as the next block to the current one, and x_5 and x_6 are generalized as the smallest and the second smallest index of the next block, respectively. If the current block is the last block, then the next block is defined to be the first block, so that the operations between blocks complete a cycle. Upon these operations, a specific equation of each block depending on its type contains variables of the next block so that they are distinguished by the type of the next block. This is an important characteristic of all the constructions we have introduced in this theory,

A	A
$x_1 + x_2 + x_3 = 0$	$x_5 + x_6 + x_7 = 0$
$x_1 + x_2 + x_4 = 0$	$x_5 + x_6 + x_8 = 0$

A	A
$\mathbf{x_5} + x_2 + x_3 = 0$	$\mathbf{x_1} + x_6 + x_7 = 0$
$x_1 + x_2 + x_4 = 0$	$x_5 + x_6 + x_8 = 0$

Table 3: The operations between blocks A-A to create a prime sub-problem

A	B
$x_1 + x_2 + x_3 = 0$	$x_5 + x_6 + x_7 = 0$
$x_1 + x_2 + x_4 = 0$	$x_5 + x_6 + x_7 = 1$

A	B
$x_1 + \mathbf{x_6} + x_3 = 0$	$x_5 + x_6 + x_7 = 0$
$x_1 + x_2 + x_4 = 0$	$\mathbf{x_1} + x_6 + x_7 = 1$

Table 4: The operations between blocks A-B to create a prime sub-problem

B	B
$x_1 + x_2 + x_3 = 0$	$x_5 + x_6 + x_7 = 0$
$x_1 + x_2 + x_3 = 1$	$x_5 + x_6 + x_7 = 1$

B	B
$x_1 + x_2 + x_3 = 0$	$x_5 + x_6 + x_7 = 0$
$x_1 + \mathbf{x_6} + x_3 = 1$	$x_5 + \mathbf{x_2} + x_7 = 1$

Table 5: The operations between blocks B-B to create a prime sub-problem

B	A
$x_1 + x_2 + x_3 = 0$	$x_5 + x_6 + x_7 = 0$
$x_1 + x_2 + x_3 = 1$	$x_5 + x_6 + x_8 = 0$

B	A
$x_1 + x_2 + x_3 = 0$	$x_5 + \mathbf{x_2} + x_7 = 0$
$\mathbf{x_5} + x_2 + x_3 = 1$	$x_5 + x_6 + x_8 = 0$

Table 6: The operations between blocks B-A to create a prime sub-problem

ensuring that we have a prime sub-problem.

All possible cases for $r = 2$ are depicted in Table 4-Table 6, where the replaced variables are in bold. We would like to point as a side note that for the case in which only one equation of an A block is satisfied, one cannot guarantee the mixing property, which is crucial to ensure a prime sub-problem. This is to be expected since approximating MAX-3-LIN-2 within factor $\frac{1}{2}$ is easy.

Recall now that the size of the problem we have constructed is $\binom{4\epsilon r}{2\epsilon r}$, which is at least $2^{(1-\delta)4\epsilon r} = 2^{(1-\delta)\epsilon n}$ by the Stirling approximation. This completes the proof. □

References

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

[2] A. Çivril. Scheme-theoretic approach to computational complexity I. The separation of P and NP. *arXiv e-prints*, page arXiv:2107.07386, 2021.

[3] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, 1971*, pages 151–158. ACM, 1971.

[4] I. Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electron. Colloquium Comput. Complex.*, page 128, 2016.

[5] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[6] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[7] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, 2002*, pages 767–775. ACM, 2002.

[8] P. Manurangsi and P. Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, volume 80 of *LIPIcs*, pages 78:1–78:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.