Scheme-theoretic Approach to Computational Complexity III. SETH

Ali Çivril*

September 7, 2022

Abstract

We show that there exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\epsilon > 0$, any deterministic algorithm to solve k-SAT for $k \geq 3$ must perform at least $(2^{k-\frac{3}{2}-\epsilon})^{\frac{n}{k+1}}$ operations, where n is the number of variables in the k-SAT instance.

1 Introduction

The purpose of this paper is to prove the following theorem, establishing the strong exponential time hypothesis (SETH) [2] against deterministic algorithms.

Theorem 1. *For any integer $k \geq 3$, there exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\epsilon > 0$, any deterministic algorithm must perform at least*

$$\left(2^{k-\frac{3}{2}-\epsilon}\right)^{\frac{n}{k+1}}$$

operations to solve the problem k-SAT, where n is the number of variables in the k-SAT instance.

We assume the reader is familiar with the first two sections of [1]. We repeat some important definitions. All the following definitions are with regard to $\Pi := k\text{-SAT}$. Given an instance I and $S \subseteq \{x_1, \ldots, x_n\}$, the *sub-instance* of I induced by S is the instance consisting of the set of all equations of I in which an element of S appears. Given a subset S of the instances, a computational problem whose instances form a set of sub-instances of the instances in S induced by the same set $S \subseteq \{x_1, \ldots, x_n\}$, 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 sub-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 of a sub-problem Λ of Π, a computational procedure transforming I_1 to I_2 is called a *unit operation*. Two unit operations are said to be *distinct* if the instances they result in are distinct 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 defined via the set of instances $\{I_1, \ldots, I_p\}$ is said to be *prime* if there exists an ordering π of its instances such that there are unit operations from $I_{\pi(i)}$ to $I_{\pi(i+1)}$ for $i = 1, \ldots, p-1$ that 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 2. [1] $\tau(\Pi) \geq \kappa(\Pi)$.

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

2 Proof of Theorem 1

Denote by k-SAT(n, m) the problem k-SAT with n variables and m clauses. The following theorem establishes Theorem 1 by Lemma 2.

Theorem 3. *For any integer $k \geq 3$, there exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\epsilon > 0$, we have*

$$\kappa\left(k\text{-SAT}\left(n, \left(\frac{2^k}{k+1}\right) n\right)\right) \geq \left(2^{k-\frac{3}{2}-\epsilon}\right)^{\frac{n}{k+1}}.$$

Proof. We construct a prime homogeneous simple sub-problem of k-SAT with $\binom{r}{r/2} \cdot 2^{r/2} \cdot (2^{k-3})^r$ instances, each having $(k+1)r$ variables and $2^k r$ clauses, for $r \geq 1$.

Each instance consists of r *blocks*. For $r = 1$, a block of an instance is defined via $k+1$ variables $x_1, \ldots, x_{k+1}$, and 2^k clauses. We will first describe a homogeneous simple sub-problem, which we will later make into a prime homogeneous simple sub-problem by mixing certain clauses between blocks. We start by constructing 2^{k-3} instances corresponding to each of the following solution sets over $\mathbb{F}_2$ with coordinates in the variables x_1, x_2, x_3, x_{k+1} in order:

Instance 1	Instance 2	Instance 3
$(0,0,1,0)$	$(0,0,1,0)$	$(0,1,0,0)$
$(1,0,0,0)$	$(0,1,0,0)$	$(0,1,1,0)$
$(1,1,0,0)$	$(1,0,1,0)$	$(1,0,0,0)$

These are exactly the same initial solution sets as introduced in the first paper for the variables x_1, x_2, x_3, x_4, where we prove the main theorem establishing hardness for 3-SAT. They will be constructed by using the same clauses (and hence polynomials), so that they also extend to the same solution sets over $\overline{\mathbb{F}}_2$. Thus, instances constructed so have the same Hilbert polynomial, resulting in a homogeneous simple sub-problem. Each of the 2^{k-3} instances corresponding to such a solution is identified by a distinct Boolean assignment of the vector $(x_4, \ldots, x_k)$. Furthermore, the solution sets over $\overline{\mathbb{F}}_2$ will also fix these variables either to 0 or 1. It follows that the cohomology of all the instances are identical to those of these 3 instances: a single closed point, and an affine line, as established for 3-SAT. This implies that we have a homogeneous simple sub-problem consisting of $3 \cdot 2^{k-3}$ instances.

We now give the clauses to fix the solutions sets as described above. In doing so, we consider the corresponding polynomials, but speak of the solution over $\overline{\mathbb{F}}_2$. The first clause is

$$(x_1 \vee x_2 \vee \cdots \vee x_k),$$

which forces at least one of x_i to 1, for $i = 1, \ldots, k$. The next 2^{k-1} clauses are

$$\begin{gathered}
\overline{x_1} \vee x_2 \vee \cdots \vee x_{k-1} \vee \overline{x_{k+1}} \\
x_1 \vee \overline{x_2} \vee \cdots \vee x_{k-1} \vee \overline{x_{k+1}} \\
\vdots \\
\overline{x_1} \vee \overline{x_2} \vee \cdots \vee \overline{x_{k-1}} \vee \overline{x_{k+1}} \\
\overline{x_k} \vee x_1 \vee \cdots \vee x_{k-2} \vee \overline{x_{k+1}},
\end{gathered}$$

where in the first $2^{k-1} - 1$ of them, the first $k-1$ literals encode all possible Boolean vectors of length $k-1$ except the all 1s vector, and in the last one, the variables $x_1, \ldots, x_{k-2}$ all appear as

positive literals. Given these clauses, assume $x_{k+1} \neq 0$, which will eventually derive a contradiction. Assume further that $x_i = 1$ for some $i = 1, \ldots, k-1$. Then the clause

$$x_1 \vee \cdots \vee \overline{x_i} \vee \cdots \vee x_{k-1} \vee \overline{x_{k+1}}$$

in which all x_j appear as a positive literal except $j \in \{i, k+1\}$, implies that at least one such x_j is also 1. Then by the definition of the first $2^{k-1} - 1$ clauses, one can find another clause forcing yet another variable to 1, and one can continue this procedure to set all $x_i = 1$ for $i = 1, \ldots, k-1$. This however contradicts the last of the first $2^{k-1} - 1$ clauses, which implies that at least one of these variables must be 0. In turn, we have that none of the x_i for $i = 1, \ldots, k-1$ can be 1. Then we must have $x_k = 1$ by the very first clause we have introduced, which contradicts the last clause listed above. We thus arrive at the conclusion that $x_{k+1} = 0$.

We now describe a set of clauses fixing $x_i = 0$ for $i = 4, \ldots, k$ in $k-3$ groups, each fixing one variable. Replacing these groups by the negation of the target variable, one can construct all the 2^{k-3} instances encoding a distinct Boolean assignment to these $k-3$ variables, thereby completing our construction. We only describe the all 0s case. The first group consists of the following 2^{k-2} clauses

$$\begin{gathered} x_1 \vee x_2 \vee \cdots \vee x_{k-2} \vee x_{k+1} \vee \overline{x_k} \\ \overline{x_1} \vee x_2 \vee \cdots \vee x_{k-2} \vee x_{k+1} \vee \overline{x_k} \\ \vdots \\ \overline{x_1} \vee \overline{x_2} \vee \cdots \vee \overline{x_{k-2}} \vee x_{k+1} \vee \overline{x_k}, \end{gathered}$$

where the first $k-2$ literals of the clauses encode all possible Boolean vectors of length $k-2$. Recalling that $x_{k+1} = 0$, this forces $x_k = 0$. In a similar fashion, the next 2^{k-3} clauses are

$$\begin{gathered} x_k \vee x_1 \vee x_2 \vee \cdots \vee x_{k-3} \vee x_{k+1} \vee \overline{x_{k-1}} \\ x_k \vee \overline{x_1} \vee x_2 \vee \cdots \vee x_{k-3} \vee x_{k+1} \vee \overline{x_{k-1}} \\ \vdots \\ x_k \vee \overline{x_1} \vee \overline{x_2} \vee \cdots \vee \overline{x_{k-3}} \vee x_{k+1} \vee \overline{x_{k-1}}, \end{gathered}$$

which force $x_{k-1} = 0$. Notice that the clauses contain x_k, which was set to 0 in the previous group. Continuing to introduce a newly set variable in the next group, and hence decreasing the number of clauses by half at each step, we finally have the following 4 clauses setting x_4 to 0:

$$\begin{gathered} x_k \vee x_{k-1} \vee \cdots \vee x_5 \vee x_1 \vee x_2 \vee x_{k+1} \vee \overline{x_4} \\ x_k \vee x_{k-1} \vee \cdots \vee x_5 \vee \overline{x_1} \vee x_2 \vee x_{k+1} \vee \overline{x_4} \\ x_k \vee x_{k-1} \vee \cdots \vee x_5 \vee x_1 \vee \overline{x_2} \vee x_{k+1} \vee \overline{x_4} \\ x_k \vee x_{k-1} \vee \cdots \vee x_5 \vee \overline{x_1} \vee \overline{x_2} \vee x_{k+1} \vee \overline{x_4}. \end{gathered}$$

The total number of clauses we have introduced thus far is $1 + (2^{k-1}) + (2^{k-2} + 2^{k-3} + \cdots + 4) = 2^k - 3$. The next 3 clauses complete the definition of the solution set by assigning x_1, x_2, x_3 their values. For Instance 1, we have the clauses

$$\begin{gathered} x_1 \vee x_3 \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k \\ \overline{x_1} \vee \overline{x_3} \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k \\ \overline{x_2} \vee \overline{x_3} \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k. \end{gathered}$$

For Instance 2, we have

$$\begin{gathered} \overline{x_1} \vee x_3 \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k \\ x_2 \vee x_3 \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k \\ \overline{x_2} \vee \overline{x_3} \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k. \end{gathered}$$

For Instance 3, we have

$$\overline{x_1} \vee \overline{x_2} \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k$$
$$\overline{x_1} \vee \overline{x_3} \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k$$
$$x_2 \vee \overline{x_3} \vee x_{k+1} \vee x_4 \vee \cdots \vee x_k.$$

The first two literals of all the introduced clauses are the same as in the proof of the main theorem of [1]. These instances are distinct, and having the same Hilbert polynomial, they form a homogeneous simple sub-problem. Thus, we have constructed such a sub-problem with $3 \cdot 2^{k-3}$ instances on $k+1$ variables, completing the base case of the induction.

Assume the induction hypothesis that there exists a homogeneous simple sub-problem of size $(3 \cdot 2^{k-3})^r$, for some $r \geq 1$. In the inductive step, we introduce $k+1$ new variables, namely $x_{(k+1)r+1}, \ldots, x_{(k+1)(r+1)}$, and $3 \cdot 2^{k-3}$ new blocks on these variables each consisting of 2^k clauses with the form as described above. Appending these blocks to each of the $(3 \cdot 2^{k-3})^r$ instances of the induction hypothesis, we obtain $(3 \cdot 2^{k-3})^{r+1}$ instances. The constructed sub-problem is a homogeneous simple sub-problem. We next make this into a prime homogeneous simple sub-problem.

We first note that the solution sets for the aforementioned 3 instances are as follows over (x_1, x_2, x_3). Instance 1: $\{(0,0,1)\} \cup \{(1,\alpha,0)\}$, Instance 2: $\{(0,1,0)\} \cup \{(\alpha,0,1)\}$, Instance 3: $\{(1,0,0)\} \cup \{(0,1,\alpha)\}$, where $\alpha \in \overline{\mathbb{F}}_2$. We shall not prove this easy fact here, which was already established in the main theorem of [1].

For ease of illustration, we describe the procedure to construct a prime homogeneous simple sub-problem for $r = 2$, which can easily be extended to the general case. Suppose that the first block is defined via Instance 1. We perform the following operation: Consider the clauses numbered from 2 to 2^{k-1}. Retain the literal $\overline{x_{k+1}}$ in all the clauses and the clauses containing the following group of literals, where they appear together:

$$x_2 \vee \overline{x_3}$$
$$\overline{x_2} \vee x_3$$
$$\overline{x_1} \vee x_2$$

For all the other clauses, replace the literals involving the variables x_1, x_2 and x_3 with the literals of the second block depending on which instance it is defined via. If the second block is defined via Instance 1, replace any expression involving these literals with $x_{k+2} \vee \overline{x_{k+3}} \vee x_{k+4}$. If it is defined via Instance 2, replace them with $x_{k+2} \vee \overline{x_{k+3}} \vee \overline{x_{k+4}}$. If it is defined via Instance 3, replace them with $\overline{x_{k+2}} \vee \overline{x_{k+3}} \vee x_{k+4}$. In extending this to the general case, the variables used for replacement is from the next block to the current one, and their indices are the ones respectively corresponding to $k+2, k+3$ and $k+4$.

If the first block is defined via Instance 2 or Instance 3, we disregard blocks defined via Instance 1 in defining the second block, i.e., Instance 2 and Instance 3 blocks are not mixed with Instance 1 blocks. With this exception, if the first block is defined via Instance 2, we perform the aforementioned operation by retaining the following group of literals:

$$x_2 \vee \overline{x_3}$$
$$\overline{x_2} \vee x_3$$
$$\overline{x_2} \vee \overline{x_3}$$

If it is defined via Instance 3, the operation is performed by retaining the following group of literals:

$$\overline{x_2} \vee x_3$$
$$\overline{x_2} \vee \overline{x_3}$$
$$\overline{x_1} \vee x_2$$

In the general case, all these operations are also performed for the last block for which the next block is defined to be the first block (or the first block that is not an Instance 1 block for Instance 2 and Instance 3), so that the operations complete a cycle over the blocks. Upon these operations, a specific set of clauses of each block depending on its type contain literals of the next block so that they are distinguished by the type of the next block. We call this the *mixing property* of the construction.

We next observe the following for the first block, which also holds for all the other blocks by the construction. Assume $x_{k+1} \neq 0$ and $x_{k+1} \neq 1$. We will show that this leads to a contradiction, so that $x_{k+1} \neq 0$ implies $x_{k+1} = 1$. Consider the case in which the first block is defined via Instance 1. By looking at the clauses on which we have retained the aforementioned group of literals, we then have

$$\begin{aligned}(1-x_2)x_3 &= 0.\\ x_2(1-x_3) &= 0.\\ x_1(1-x_2) &= 0.\end{aligned}$$

As noted, the solution set implied by the last 3 equations of the first block for $x_{k+1} \neq 1$ sets at least one of x_1, x_2 and x_3 to 1. By checking each case then, we have that the solution set to these equations is $\{(\alpha, 1, 1)\}$. This contradicts the aforementioned solution set: $\{(0, 0, 1)\} \cup \{(1, \alpha, 0)\}$.

Suppose now that the first block is defined via Instance 2. By looking at the corresponding group of literals retained, we get

$$\begin{aligned}(1-x_2)x_3 &= 0.\\ x_2(1-x_3) &= 0.\\ x_2x_3 &= 0.\end{aligned}$$

By the same token, the solution set to these equations is $\{(1, 0, 0)\}$. This contradicts the solution set implied by the last 3 equations of Instance 2 for $x_{k+1} \neq 1$: $\{(0, 1, 0)\} \cup \{(\alpha, 0, 1)\}$.

Finally, suppose that the first block is defined via Instance 3. We then obtain

$$\begin{aligned}x_2(1-x_3) &= 0.\\ x_2x_3 &= 0.\\ x_1(1-x_2) &= 0.\end{aligned}$$

With a similar argument to those above, the solution set to these equations is $\{(0, 0, 1)\}$. This contradicts the solution set implied by the last 3 equations of Instance 3 for $x_{k+1} \neq 1$: $\{(1, 0, 0)\} \cup \{(0, 1, \alpha)\}$. Thus, we have either $x_{k+1} = 0$ or $x_{k+1} = 1$.

Observe that the replaced clauses are satisfiable. Assume $x_{k+1} \neq 0$. If the second block is defined via Instance 1, $x_{k+2} \vee \overline{x_{k+3}} \vee x_{k+4}$ does not contradict the solution set for Instance 1, which is $\{(0, 0, 1)\} \cup \{(1, \alpha, 0)\} \cup \{(\alpha, 1, 1)\}$. Similarly, if the second block is defined via Instance 2, $x_{k+2} \vee \overline{x_{k+3}} \vee \overline{x_{k+4}}$ does not contradict the solution set for Instance 2, which is $\{(1, 0, 0)\} \cup \{(0, 1, 0)\} \cup \{(\alpha, 0, 1)\}$. If the second block is defined via Instance 3, $\overline{x_{k+2}} \vee \overline{x_{k+3}} \vee x_{k+4}$ does not contradict the solution set for Instance 3, which is $\{(0, 0, 1)\} \cup \{(1, 0, 0)\} \cup \{(0, 1, \alpha)\}$. Here the solution sets are expressed over $(x_{k+2}, x_{k+3}, x_{k+4})$.

We have already mentioned that for $x_{k+1} = 0$, the solution sets associated to three different types of blocks have the same cohomology. For $x_{k+1} = 1$, the solution sets associated to these blocks are the ones computed in the discussion above. For Instance 1, it is $(\alpha, 1, 1, 1)$. For Instance 2, it is $(1, 0, 0, 1)$. For Instance 3, it is $(0, 0, 1, 1)$, where the solutions are expressed over (x_1, x_2, x_3, x_{k+1}). Thus, the Hilbert polynomials associated to Instance 2 and Instance 3 are the same, whereas Instance 1 differs from them. Consider the instances having $r/2$ blocks defined via Instance 1 and $r/2$ blocks defined via Instance 2 or Instance 3. These instances have the same Hilbert polynomial,

so that we have a homogeneous simple sub-problem. The number of such instances is $\binom{r}{r/2} \cdot 2^{r/2}$. Using the Stirling approximation, we obtain

$$\binom{r}{r/2} = \frac{r!}{(r/2)!(r/2)!} > \sqrt{\frac{2}{\pi r}} \cdot 2^r \cdot \exp\left(\frac{1}{12r+1} - \frac{1}{3r}\right),$$

so that for all $\epsilon > 0$, we have

$$\binom{r}{r/2} \cdot 2^{r/2} > 2^{\left(\frac{3}{2}-\epsilon\right)r},$$

as r tends to infinity. Noting that $r = n/4$, there remains to see that we have a prime sub-problem. This has already been established in the proof of the main theorem of [1], which easily follows from the mixing property of our construction and an easy fact about the structure of the binary Gray code. □

References

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

[2] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.