Scheme-theoretic Approach to Computational Complexity III. SETH

Ali Çivril*

March 29, 2022

Abstract

We show that any deterministic algorithm to solve k-SAT must perform at least $(3 \cdot 2^{k-3})^{\frac{n}{k+1}}$ operations in the worst-case.

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. *Any deterministic algorithm must perform at least*

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

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

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. Given two sub-instances I_1 and I_2 defined via the same variable set $\{x_i\}_{i \in I}$, I_2 is said to be a *variant* of I_1 if it can be obtained from I_1 by replacing each x_i in a subset of I by $1 - x_i$ in its equations, followed by a permutation of the variables. Two sub-instances are said to be *distinct* if they are not variants of each other. A sub-instance is said to be *homogeneous* if the set of its equations cannot be partitioned into two sets, such that the set of variables appearing in the partitions are disjoint. A simple sub-problem Λ is said to be *homogeneous* if the instances of Λ are pair-wise distinct, and each instance of Λ is homogeneous.

We define $\tau(\Pi)$ to be the minimum number of *deterministic* operations required to solve Π. Given a homogeneous simple sub-problem Λ, we denote the number of instances of Λ by $b(\Lambda)$. We denote by $\overline{\Lambda}$ the computational problem, which contains all the variants of the instances of Λ. A sub-problem Γ is called a *normal sub-problem* if $\Gamma = \prod_{i=1}^{r} \overline{\Lambda_i}$, for some $r \geq 1$, where Λ_i are homogeneous simple sub-problems defined via disjoint sets of variables, and the product of computational problems are defined in the usual sense, i.e., the set of instances of the product is the Cartesian product of the sets of instances of the problems. Over all such normal sub-problems Γ, let $\kappa(\Pi)$ denote the maximum value of $\sum_{i=1}^{r} b(\Lambda_i)$.

*e-mail: ali.civril@istinye.edu.tr

Lemma 2. *[1]* $\tau(\Pi) \geq \kappa(\Pi)$.

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*

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

Proof. We construct by induction on r, a homogeneous simple sub-problem of k-SAT with $\left(3 \cdot 2^{k-3}\right)^r$ 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 construct 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 solution sets as those of the first paper, 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, they correspond to the same Hilbert polynomial, resulting in a simple sub-problem. We shall not prove this fact here, and refer the reader to the proof of the main theorem of [1]. 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: two distinct points, and an affine line, as established for 3-SAT. This implies that we have a 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 and 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. With all of these, we 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]. Thus, these instances are distinct, and having the same Hilbert polynomial, they form a homogeneous simple sub-problem. Given this, we have constructed a homogeneous simple sub-problem with $3 \cdot 2^{k-3}$ instances on $k+1$ variables, completing the base case of the induction.

The rest of the proof is similar to the one in [1]. Assume the induction hypothesis for some $r \geq 1$. In the inductive step, we introduce $k+1$ new variables $x_{(k+1)r+1}, \ldots, x_{(k+1)(r+1)}$, 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 $\left(3 \cdot 2^{k-3}\right)^r$ instances of the induction hypothesis, we obtain $\left(3 \cdot 2^{k-3}\right)^{r+1}$ instances. The constructed sub-problem is a simple sub-problem with distinct instances. Assume we have a homogeneous simple sub-problem as the induction hypothesis, which is clearly satisfied for $r = 1$. Replace the positive literal $x_{(k+1)r}$, where it appears in the last $2^{k-1} - 1$ clauses of its block by the positive literal $x_{(k+1)(r+1)}$ in all of the instances. This operation does not change the solution set and hence the Hilbert polynomial, as $x_{(k+1)r}$ is already forced to be 0 for all r by the other clauses. It also ensures that we have a homogeneous simple sub-problem, completing the induction and the proof. □

3 Final Remarks

The cohomology of the solution sets of the instances we have constructed are identical to those of 3-SAT. We essentially double the number of instances at each step, thereby creating $3 \cdot 2^{k-3}$ instances. We also increase the number of variables by 1, thus generalizing the parameter 4 for 3-SAT to $k+1$. We suspect however that there must exist a set of instances specific to k-SAT, whose cohomology is more intricate, in particular getting more complicated as k increases. In more quantitative terms, the correct generalization of the number of instances from 3-SAT to k-SAT should be from 3 to $2^{k-1} - 1$, rather than $3 \cdot 2^{k-3}$.

Conjecture 4. *For any integer $k \geq 3$, there exist infinitely many $n \in \mathbb{Z}^+$, such that*

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

References

[1] A. Çivril. Scheme-theoretic approach to computational complexity I. The separation of P and NP. Manuscript, 2021.

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