Scheme-theoretic Approach to Computational Complexity I. The Separation of P and NP

Ali Çivril*

September 20, 2021

1 Introduction

This paper introduces the rudiments of a new theory for algorithms and computational complexity via schemes. One of the most important consequences of the theory is the resolution of the conjecture $\mathsf{P} \neq \mathsf{NP}$.

The theory envisions a computational problem as a (moduli) scheme formed by its instances, and an algorithm as a morphism contracting the computational problem at hand to a single point. For concreteness, consider a satisfiable instance of 3-SAT represented by the formula ϕ with variables $x_1, \ldots, x_n$. We associate this instance with all the solutions that make ϕ satisfiable, which can be expressed as the zeros of a polynomial $\phi(x_1, \ldots, x_n)$ over $\mathbb{F}_2$. We then identify this information by considering the closed subscheme Proj $\overline{\mathbb{F}}_2[x_0, x_1, \ldots, x_n]/(\phi(x_0, x_1, \ldots, x_n))$. The global scheme corresponding to the computational problem 3-SAT is the Hilbert scheme parameterizing these closed subschemes together with a set of others to ensure connectedness.

The next step is to unify the notion of a reduction and an algorithm in the new setting. Consider 1-SAT $\in$ P. In order to separate P and NP, one needs to rule out a polynomial-time reduction f satisfying $x \in$ 3-SAT $\Leftrightarrow f(x) \in$ 1-SAT. We extend this line of thinking by introducing the trivial object in the category of computational problems: the trivial problem asking "Is '0'=0?", where the first '0' is a polynomial. The associated scheme of this problem is the final object Spec $\overline{\mathbb{F}}_2$ in the category of schemes over $\overline{\mathbb{F}}_2$. In our new language, solving a problem is nothing but reducing it to the trivial problem. One then needs to show that, in geometric terms we will later formalize, it is impossible to reduce the scheme of 3-SAT to the single point Spec $\overline{\mathbb{F}}_2$ with polynomial number of unit contractions.

2 Computational Problems and the Amplifying Functor

A computational problem consists of a set of *instances*. In this paper we impose that each instance consists of a finite set of polynomial equations over $\mathbb{F}_2$. We thus use a *polynomial system* as a synonym for an instance. The synonym for a single polynomial equation is a *clause*. One seeks, given an instance, an assignment to the variables in $\mathbb{F}_2$ satisfying all the equations of the instance. Throughout the paper an instance is one which has such a solution. We give below examples by listing the possible set of polynomials that might be considered for an equation. The simplest problem is what we call TRIVIAL or T for short, defined via an equation with an empty set of

*Istinye University, Computer Engineering Department, Zeytinburnu, Istanbul 34010 Turkey, e-mail: ali.civril@engr.istinye.edu.tr, website: www.alicivril.com

variables. If the set of instances of a problem is empty, it is assumed to be T. The simplest problem after T is UNIT or U for short, a special case of $\mathsf{1\text{-}SAT}$ and $\mathsf{3\text{-}SAT}$.

Problem: $\mathsf{TRIVIAL}$ or T
Polynomials: $p(\cdot) \in \{0\}$.

Problem: UNIT or U
Polynomials: $p(x) \in \{x, 1-x\}$.

Problem: $\mathsf{1\text{-}SAT}$
Polynomials: All $\{p(x_1, \dots, x_n)\}$ with $p(x_1, \dots, x_n) = t$, where $t = x_\ell$ or $t = 1 - x_\ell$ for some $\ell \in \{1, \dots, n\}$.

Problem: $\mathsf{3\text{-}SAT}$
Polynomials: All $\{p(x_1, \dots, x_n)\}$ with $p(x_1, \dots, x_n) = t_1 t_2 t_3$, where $t_j = x_\ell$ or $t_j = 1 - x_\ell$ for some $\ell \in \{1, \dots, n\}$, for $j \in \{1, 2, 3\}$.

For the sake of explicitness, we give the following examples regarding instances.

Problem: $\mathsf{TRIVIAL}$ or T
Logical form: $\{\}$.
Algebraic form: $\{0 = 0\}$.

Problem: UNIT or U
Logical form: $\{x\}, \{\overline{x}\}$.
Algebraic form: $\{1 - x = 0\}, \{x = 0\}$.

Problem: $\mathsf{1\text{-}SAT}$
Logical form: $\{x_1 \wedge \overline{x_2} \wedge x_3\}$.
Algebraic form: $\{1 - x_1 = 0, x_2 = 0, 1 - x_3 = 0\}$.

Problem: $\mathsf{3\text{-}SAT}$
Logical form: $\{(x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_5})\}$.
Algebraic form: $\{(1 - x_1)x_3(1 - x_4) = 0, x_2(1 - x_3)x_5 = 0\}$.

We consider the computational problem $\mathsf{\Pi} := k\text{-}\mathsf{SAT}$. We will define the *amplifying functor* from $\mathsf{\Pi}$ to the category of schemes over $\overline{\mathbb{F}}_2$, which is a subfunctor of the *Hilbert functor*. The Hilbert functor $\mathcal{H}(X)$ (relative to $\overline{\mathbb{F}}_2$) defined for a projective scheme X over $\overline{\mathbb{F}}_2$ parameterizes the closed subschemes of X. Let $Z \subset X$ be a closed subscheme. Let $\mathscr{F}$ be a coherent sheaf on Z. The *Hilbert polynomial of Z with respect to $\mathscr{F}$* is $P(Z, \mathscr{F})(m) := \chi(Z, \mathscr{F}(m))$, where $\mathscr{F}(m)$ is the twisting of $\mathscr{F}$ by m, and $\chi(Z, \mathscr{F})$ denotes the *Euler characteristic* of $\mathscr{F}$ given by

$$\chi(Z, \mathscr{F}) := \sum_{i=0}^{\dim Z} (-1)^i \dim_{\overline{\mathbb{F}}_2} H^i(Z, \mathscr{F}).$$

The *Hilbert polynomial of Z* is $P(Z)(m) := \chi(Z, \mathscr{O}_Z(m))$, where $\mathscr{O}_Z$ is the structure sheaf of Z. Define the subfunctor of $\mathcal{H}(X)$ induced by a polynomial $P \in \mathbb{Q}[x]$ as

$$\mathcal{H}^P(X) = \{Z \subset X \text{ is a closed subscheme} | P(Z) = P\}.$$

By the following result stated in our context, the amplifying functor is representable by a projective scheme over $\overline{\mathbb{F}}_2$.

Theorem 2.1 ([1]). *Let X be a projective scheme over $\overline{\mathbb{F}}_2$. Then for every polynomial $P \in \mathbb{Q}[x]$, there exists a projective scheme* $\mathrm{Hilb}^P(X)$ *over $\overline{\mathbb{F}}_2$, which represents the functor $\mathcal{H}^P(X)$. Furthermore, the Hilbert functor $\mathcal{H}(X)$ is represented by the* Hilbert scheme

$$\mathrm{Hilb}(X) := \coprod_{P \in \mathbb{Q}[x]} \mathrm{Hilb}^P(X).$$

We consider $X = \mathbb{P}^n_{\overline{\mathbb{F}}_2}$. In this case, the connectedness of the Hilbert scheme for a fixed Hilbert polynomial P was established by Hartshorne [2]. Note first that given a homogenized polynomial ϕ, one might consider the closed subscheme

$$\mathrm{Proj}\ \overline{\mathbb{F}}_2[x_0, x_1, \ldots, x_n]/(\phi(x_0, x_1, \ldots, x_n)),$$

so that each polynomial equation and hence a polynomial system identifies a closed subscheme of $\mathbb{P}^n_{\overline{\mathbb{F}}_2}$ via the corresponding ideal. In particular, we refer to the Hilbert polynomial of an instance.

Definition 2.2. Given an instance of Π, a non-empty subset of the equations of the corresponding polynomial system is said to form a *sub-instance* of Π.

Remark 2.3. A sub-instance has a corresponding polynomial system induced by the polynomial equations it contains.

Definition 2.4. Two sub-instances are called *disjoint* if both do not belong to a single instance of Π.

Definition 2.5. A *sub-problem* of Π is a computational problem induced by a subset of disjoint sub-instances of Π.

Definition 2.6. A sub-problem Λ of Π is called a *simple sub-problem* if the polynomial systems of Λ have the same Hilbert polynomial.

Definition 2.7. Let I_1 and I_2 be two sub-instances of Π 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.

Remark 2.8. An instance together with all of its variants define an equivalence class. We may assume without loss of generality that an algorithm is given a set of these equivalence classes. Accordingly:

Definition 2.9. Two sub-instances of Π are said to be *distinct* if they are not variants of each other.

Remark 2.10. Distinct sub-instances of Π correspond to distinct points in moduli.

Definition 2.11. A simple sub-problem Λ of Π is said to be *homogeneous* if the following two conditions hold.

- The instances of Λ are pair-wise distinct.
- None of the instances of Λ can be expressed as the Cartesian product of r sets of sub-instances of Λ defined via disjoint sets of variables, for some $r \geq 2$.

We now give a partial definition of the amplifying functor, which we denote by $\mathcal{A}$. Let Λ be a homogeneous simple sub-problem of Π consisting of a set of polynomial systems $\{P_i\}_{i=1}^{\ell}$ defined via the variables $x_{\lambda_1}, \ldots, x_{\lambda_m}$. Let ϕ_{ij} be the homogenized j-th polynomial in the polynomial system P_i. Define

$$X_i = \text{Proj } \overline{\mathbb{F}}_2[x_0, x_{\lambda_1}, \ldots, x_{\lambda_m}]/(\phi_{i1}, \ldots, \phi_{i|P_i|}),$$
$$\mathcal{A}(\Lambda) = \{X_i | i = 1, \ldots, \ell\}.$$

We define the product of computational problems 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. Given a sub-problem $\Gamma = \prod_{i=1}^{r} \Lambda_i$ of Π, for some $r \geq 1$, where Λ_i are homogeneous simple sub-problems of Π defined via disjoint sets of variables, we set

$$\mathcal{A}(\Gamma) = \prod_{i=1}^{r} \mathcal{A}(\Lambda_i),$$

and

$$\mathcal{A}(\Pi) := \coprod_{\Gamma} \mathcal{A}(\Gamma).$$

By the connectedness of the Hilbert scheme for a fixed Hilbert polynomial, we have that there is a functor $\mathcal{B}$, which is an extension of $\mathcal{A}$, such that a homogeneous simple sub-problem always maps to a connected scheme via $\mathcal{B}$. Thus, we extend $\mathcal{A}$ to $\mathcal{B}$, overriding the definition of the amplifying functor.

3 Lower Bounds via Homogeneous Simple Sub-problems

Given a homogeneous simple sub-problem Λ of Π, and a sub-problem Λ' of Λ induced by a subset of the instances of Λ, let $\tau(f) := \tau(\Lambda, \Lambda')$ denote the minimum number of *deterministic* operations required by a reduction $f : \Lambda \to \Lambda'$, possibly with an advice string, thus simulating circuits. For a general Λ, we define $\tau(\Lambda) := \tau(\Lambda, \emptyset) = \tau(\Lambda, \top)$. We denote the number of instances of Λ by $b(\Lambda)$. A sub-problem Γ of Π is called a *normal sub-problem* of Π if $\Gamma = \prod_{i=1}^{r} \Lambda_i$, for some $r \geq 1$, where Λ_i are homogeneous simple sub-problems of Π defined via disjoint sets of variables. Over all such normal sub-problems Γ of Π, let $\kappa(\Pi)$ denote the maximum value of $\sum_{i=1}^{r} b(\Lambda_i)$.

Proposition 3.1.

$$\tau(\Gamma) \geq \sum_{i=1}^{r} \tau(\Lambda_i).$$

Proof. The set of equations of a polynomial system of Γ is partitioned by those of Λ_i. Thus, in order to solve Γ one has to solve each Λ_i. □

Lemma 3.2 (Fundamental Lemma)**.**

$$\tau(\Pi) \geq \kappa(\Pi).$$

Proof. Let $\Gamma = \prod_{i=1}^{r} \Lambda_i$ be a normal subproblem of Π attaining $\kappa(\Pi)$. Since $\tau(\Pi) \geq \tau(\Gamma)$, it suffices to show $\tau(\Gamma) \geq \sum_{i=1}^{r} b(\Lambda_i)$. Furthermore, by Proposition 3.1 it suffices to show $\tau(\Lambda) \geq b(\Lambda)$, where $\Lambda = \Lambda_i$ for some i. We argue by induction on $p := b(\Lambda)$. The case $p = 1$ is clear since $\tau(\Lambda) \geq 1$. In particular, we have the following diagrams for the base case of the induction, depicting the solution of Λ in both categories.

$$\Lambda \longrightarrow \top \qquad\qquad \mathcal{B}\Lambda \longrightarrow \mathcal{B}\top$$

For $p > 1$, let $S = \{I_1, \ldots, I_p\}$ be the set of instances defining Λ, and Λ' be the sub-problem of Λ with the set of instances $S \setminus \{I_p\}$. The instances of Λ are all distinct by definition, and both $\mathcal{B}(\Lambda)$ and $\mathcal{B}(\Lambda')$ are connected. Then we have the following commutative diagrams with $\tau(\Lambda) = \tau(\alpha) + \tau(\beta) = \tau(\alpha') + \tau(\beta')$.

$$\begin{tikzcd} \Lambda \arrow[r, "\alpha"] \arrow[d, "\beta'"'] & \Lambda' \arrow[d, "\beta"] \\ \Lambda_{\mathsf{U}} \arrow[r, "\alpha'"'] & \mathsf{T} \end{tikzcd} \qquad \begin{tikzcd} \mathcal{B}\Lambda \arrow[r, "\mathcal{B}\alpha"] \arrow[d, "\mathcal{B}\beta'"'] & \mathcal{B}\Lambda' \arrow[d, "\mathcal{B}\beta"] \\ \mathcal{B}\Lambda_{\mathsf{U}} \arrow[r, "\mathcal{B}\alpha'"'] & \mathcal{B}\mathsf{T} \end{tikzcd}$$

The arrows on the left are complexity-theoretic reductions between problems, which are set-theoretic maps between instance sets. The arrows on the right are contraction morphisms. Λ_{U} is the problem defined by the single instance I_p. β' is the reduction from Λ to Λ_{U}, which is the same as β on the instances of Λ'. α is the reduction from Λ to Λ', which is the identity on the instances of Λ'. Using the induction hypothesis and the geometry of the schemes in the commutative diagram on the right, we have $\tau(\beta') \geq \tau(\beta) = \tau(\Lambda') \geq p - 1$, and $\tau(\alpha) \geq \tau(\alpha')$. If we assume $\tau(\Lambda) = \tau(\Lambda')$ so that $\tau(\alpha) = 0$, we then get the contradiction $\tau(\alpha') = 0$. Thus, $\tau(\Lambda) \geq \tau(\Lambda') + 1 \geq p$, which completes the induction and the proof. $\square$

4 The Separation of P and NP

Denote by k-$\mathsf{SAT}(n, m)$ the problem k-SAT with n variables and m clauses.

Theorem 4.1. *There exist infinitely many $n \in \mathbb{Z}^+$ such that*

$$\kappa(3\text{-}\mathsf{SAT}(n, 2n)) \geq 3^{n/4}.$$

Proof. We will construct by induction, a homogeneous simple sub-problem Λ of 3-SAT with 3^r instances, each having $4r$ variables and $8r$ clauses, for $r \geq 1$. The result follows by the definition of κ.

We use the logical representation of the instances in our tables, but note also that an instance has a solution set over $\overline{\mathbb{F}}_2$ in algebraic form. Each instance consists of r *blocks*. For $r = 1$, a block of an instance is defined on 4 variables x_1, x_2, x_3, x_4, and 8 clauses. We construct 3 instances with the solution sets over $\mathbb{F}_2$ consisting of the following points, listed for each instance in a separate column:

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

A single block for each instance is constructed with a simple algorithm best described using the truth table of the variables. Each of the 8 clauses is introduced one by one to rule out an assignment over $\mathbb{F}_2$ in the tables. For this purpose, we enumerate the rows of the tables for each instance by an indexing of these clauses in Table 2, Table 3, and Table 4. The solution sets over $\mathbb{F}_2$ are the entries left out by the introduced clauses. The solution sets over $\overline{\mathbb{F}}_2$ have the same cohomology, so that the Hilbert polynomials of the instances are the same. In particular, they are the disjoint union of two distinct closed points and an affine line as will be shown in the next paragraph.

The first 5 clauses of the instances are common. Clause 1 forces at least one of x_1, x_2 and x_3 to be 1. Given this, the following 4 clauses make $x_4 = 0$, since $x_4 \neq 0$ implies $x_1 = x_2 = x_3 = 0$ by these clauses. We now examine the last 3 clauses of the instances one by one.

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 satisfying Table 2, Table 3, and Table 4

Clause	x_1	x_2	x_3	x_4	Clause	x_1	x_2	x_3	x_4
1	0	0	0	0		1	0	0	0
1	0	0	0	1	5	1	0	0	1
	0	0	1	0	7	1	0	1	0
2	0	0	1	1	2	1	0	1	1
6	0	1	0	0		1	1	0	0
3	0	1	0	1	3	1	1	0	1
8	0	1	1	0	8	1	1	1	0
4	0	1	1	1	4	1	1	1	1

Table 2: The truth table of a block of Instance 1 with clause-indexing

Clause	x_1	x_2	x_3	x_4	Clause	x_1	x_2	x_3	x_4
1	0	0	0	0	7	1	0	0	0
1	0	0	0	1	5	1	0	0	1
	0	0	1	0		1	0	1	0
2	0	0	1	1	2	1	0	1	1
	0	1	0	0	6	1	1	0	0
3	0	1	0	1	3	1	1	0	1
8	0	1	1	0	8	1	1	1	0
4	0	1	1	1	4	1	1	1	1

Table 3: The truth table of a block of Instance 2 with clause-indexing

1. Instance 1: $x_1 = 1 \Rightarrow x_3 = 0, x_2 \in \overline{\mathbb{F}}_2$; $x_2 = 1 \Rightarrow x_1 = 1, x_3 = 0$; $x_3 = 1 \Rightarrow x_1 = 0, x_2 = 0$.

2. Instance 2: $x_1 = 1 \Rightarrow x_2 = 0, x_3 = 1$; $x_2 = 1 \Rightarrow x_1 = 0, x_3 = 0$; $x_3 = 1 \Rightarrow x_2 = 0, x_1 \in \overline{\mathbb{F}}_2$.

3. Instance 3: $x_1 = 1 \Rightarrow x_2 = 0, x_3 = 0$; $x_2 = 1 \Rightarrow x_1 = 0, x_3 \in \overline{\mathbb{F}}_2$; $x_3 = 1 \Rightarrow x_1 = 0, x_2 = 1$.

One checks that these instances are distinct. We have thus shown that they form a homogeneous simple sub-problem. Assume now the induction hypothesis for some $r \geq 1$. In the inductive step, 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

Clause	x_1	x_2	x_3	x_4	Clause	x_1	x_2	x_3	x_4
1	0	0	0	0		1	0	0	0
1	0	0	0	1	5	1	0	0	1
8	0	0	1	0	8	1	0	1	0
2	0	0	1	1	2	1	0	1	1
	0	1	0	0	6	1	1	0	0
3	0	1	0	1	3	1	1	0	1
	0	1	1	0	7	1	1	1	0
4	0	1	1	1	4	1	1	1	1

Table 4: The truth table of a block of Instance 3 with clause-indexing

3^r instances of the induction hypothesis, we obtain 3^{r+1} instances. The constructed sub-problem is a simple sub-problem. We make it into a homogeneous simple sub-problem for which a sufficient condition is as follows. Given an instance, consider the graph whose nodes are the clauses of the instance, and there is an edge between two nodes if they share a common variable. We impose that this graph be connected for all the instances, which is clearly satisfied in the base case $r = 1$. For the inductive step, replace the positive literal x_{4r}, where it appears in the last 3 clauses of its block by the positive literal x_{4r+4}. This operation does not change the solution set and hence the Hilbert polynomial, as x_{4r} is already forced to be 0 for all r by the other clauses. It also ensures connectivity of the graph across all the blocks including the newly introduced one, thus constructing a homogeneous simple sub-problem, completing the induction and the proof. □

By Theorem 4.1, Lemma 3.2, and the NP-completeness of 3-SAT [5]:

Corollary 4.2. $\mathsf{P} \neq \mathsf{NP}$.

The definition of τ also implies

Corollary 4.3. $\mathsf{NP} \not\subseteq \mathsf{P/poly}$.

Furthermore, by the specific lower bound derived for 3-SAT:

Corollary 4.4. *The exponential time hypothesis* [3] *is true against deterministic algorithms.*

Finally, this exponential lower bound implies the following by [4].

Corollary 4.5. $\mathsf{BPP} = \mathsf{P}$.

References

[1] A. Grothendieck. *Fondements de la Géométrie Algébrique [Extraits du Séminaire Bourbaki 1957-1962]*, chapter Techniques de construction et théorèmes d'existence en géométrie algébrique. IV. Les schémas de Hilbert. Secr. Math., 1962.

[2] R. Hartshorne. Connectedness of the Hilbert scheme. *Publications Mathématiques de l'IHÉS*, 29:5–48, 1966.

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

[4] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 220–229. ACM, 1997.

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