Hardness Hypotheses, Derandomization, and Circuit Complexity

John M. Hitchcock[*]
Department of Computer Science
University of Wyoming

A. Pavan[†]
Department of Computer Science
Iowa State University

Abstract

We consider hypotheses about nondeterministic computation that have been studied in different contexts and shown to have interesting consequences:

- The *measure hypothesis*: NP does not have p-measure 0.

- The *pseudo*-NP *hypothesis*: there is an NP language that can be distinguished from any DTIME(2^{n^ϵ}) language by an NP refuter.

- The NP-*machine hypothesis*: there is an NP machine accepting 0^* for which no 2^{n^ϵ}-time machine can find infinitely many accepting computations.

We show that the NP-machine hypothesis is implied by each of the first two. Previously, no relationships were known among these three hypotheses. Moreover, we unify previous work by showing that several derandomizations and circuit-size lower bounds that are known to follow from the first two hypotheses also follow from the NP-machine hypothesis. In particular, the NP-machine hypothesis becomes the weakest known uniform hardness hypothesis that derandomizes AM. We also consider UP versions of the above hypotheses as well as related immunity and scaled dimension hypotheses.

1 Introduction

The following uniform hardness hypotheses are known to imply full derandomization of Arthur-Merlin games (NP = AM):

- The *measure hypothesis*: NP does not have p-measure 0 [24].

- The *pseudo*-NP *hypothesis*: NP has a language that can be distinguished from any DTIME(2^{n^ϵ}) language by an NP refuter [32].

- NE $\cap$ coNE cannot infinitely-often be decided in $2^{2^{\epsilon n}}$ time [23].

While the hypotheses are quite different, each of these results rely on the ingenious "easy witness method" of Kabanets [26]: try to show that the hypothesis is false by searching for an easy witness, a witness that has low circuit complexity when viewed as the truth-table of a Boolean function. If

---

the hypothesis is true, we can show that this search must fail. Therefore there are no easy witnesses and we can use the mechanism in the hypothesis to nondeterministically generate witnesses of high circuit complexity that are sufficient for derandomizing AM [37, 29].

Given the similarity in the proofs, it is natural to ask how much more these hypotheses have in common. We show that all three of the above hypotheses imply the following NP-*machine hypothesis*:

> There is an NP machine accepting 0^* for which no 2^{n^ϵ}-time machine can find infinitely many accepting computations.

Roughly speaking, this hypothesis says that there is an NP search problem that cannot be solved in subexponential time. This hypothesis and several variations have been used a few times in complexity theory [16, 12, 14, 40]. The NP-machine hypothesis in this form is due to Pavan and Selman [40], who showed that it implies a separation of NP-completeness notions.

The easy witness method readily applies to show that the NP-machine hypothesis also implies NP = AM. Therefore we have the following picture:
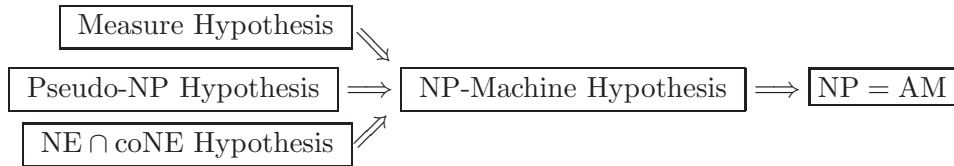
Thus the NP-machine hypothesis becomes the weakest known "uniform hardness" hypothesis that derandomizes AM.

While the NP-machine hypothesis has been used before in complexity theory, prior to our work it was never observed to have a connection with derandomization or circuit complexity. In fact, not only is the NP-machine hypothesis amenable to the easy witness method, it can be viewed as having the essential character needed to apply the method. This is because NP-machine hypothesis has an equivalent formulation in terms of Levin's Kt complexity [31] and Allender and Ronneberger [2] showed that Kt complexity has an equivalence with E-oracle circuit complexity.

The derandomization of AM is just one of many consequences of the measure hypothesis (see e.g. [35]). We show that many of the same or slightly weaker consequences also follow from the NP-machine hypothesis. For example:

- $P^{NP} = BPP^{NP}$.

- $PP = BP \cdot PP$, which implies $PH \subseteq PP$.

- E^{NP} does not have subexponential-size circuits.

- NEXP does not have polynomial-size circuits.

- P^{NP} does not have n^k-size circuits for fixed k.

- NP is not easy on average.

Furthermore, from the work of Fenner, Fortnow, Naik, and Rogers [12] on a statement they called Proposition Q, it is immediate that the NP-machine hypothesis has some additional consequences:

- There is an NP multi-valued total function that does not have a polynomial-time refinement.

2

- P $\neq$ NP $\cap$ coNP or there is an NP multi-valued total function that does not have a NP single-valued refinement.

Therefore these statements about NP functions also follow from the measure hypothesis – such a result was not known prior to this paper.

Pavan and Selman [40] showed that NP-machine hypothesis is true if NP $\cap$ coNP has a 2^{n^ϵ}-bi-immune language. Thus, if this bi-immunity assumption holds for NP $\cap$ coNP, then circuit lower bounds for E^{NP} and NEXP follow. This is particularly interesting because concepts such as bi-immunity and circuit complexity have been studied for a long time in different contexts. Our results demonstrate that they have underlying relationships.

We also consider two variants of the NP-machine hypothesis. The UP-machine hypothesis is shown to have several interesting consequences that are stronger than those of the NP-machine hypothesis. The RP-machine hypothesis is placed in the same category as some other RP-hardness assumptions and shown to be equivalent to ZPP = EXP.

Finally, we consider the hypothesis "the -3rd-order scaled dimension of NP is positive," which is weaker than the measure hypothesis but seems incomparable with the other hypotheses. However, it can be viewed as implying an advice version of the NP-machine hypothesis, from which we obtain the partial derandomization AM $\subseteq$ NP/n^ϵ for every $\epsilon > 0$. We also show that this dimension hypothesis implies NEXP $\not\subseteq$ P/poly.

This paper is organized as follows. Section 2 contains preliminaries. The hypotheses about nondeterministic computation are compared in section 3. Section 4 presents the consequences of the NP-machine hypothesis. We consider variations of the NP-machine hypothesis in section 5 and scaled dimension in section 6. We conclude in section 7 with a summary and some open problems.

2 Preliminaries

For a language L, L^n denotes the set $\{x \mid |x| = n, x \in L\}$. Given a string x, we write $L(x) = 1$ if $x \in L$, and $L(x) = 0$ if $x \notin L$. We write $L|n = L(s_1) \cdots L(s_{2^n})$, where $s_1, s_2, \cdots s_{2^n}$ are the strings of length n in lexicographic order. For a complexity class $\mathcal{C}$, the class io-$\mathcal{C}$ is

$$\{L \mid (\exists A \in \mathcal{C})(\exists^\infty n)L^n = A^n\}.$$

2.1 Circuit Complexity

An oracle circuit is a circuit that has special gates called *oracle gates*, in addition to the normal AND, OR, and NOT gates. Given an oracle A, an *A-oracle circuit* is an oracle circuit that has A as an oracle, i.e, if x is the input of an oracle gate, then the output is $A(x)$. Given a Boolean function $f : \Sigma^n \to \{0,1\}$, and an oracle A, the *A-oracle circuit complexity* of f is the size of the smallest A-oracle circuit that computes f. The class $\text{SIZE}^A(s(n))$ contains all languages B such that for all but finitely many n, the characteristic function of B^n has A-oracle circuit complexity at most $s(n)$.

2.2 Kolmogorov Complexity

Let U be an efficient universal Turing machine. For a time bound $t(n)$, the *t-time-bounded Kolmogorov complexity* of a string x is

$$K^t(x) = \min\{|p| \mid U(p) = x \text{ in at most } t(|x|) \text{ steps}\}.$$

The *Levin complexity* of a string x is

$$\mathrm{Kt}(x) = \min\{|p| + \log t \mid U(p) = x \text{ in at most } t \text{ steps}\}.$$

2.3 Immunity

Let $\mathcal{C}$ be a complexity class. A language L is $\mathcal{C}$-*immune* if L is infinite and no infinite subset of L belongs to $\mathcal{C}$. We say that L is $\mathcal{C}$-*bi-immune* if both L and $\overline{L}$ are $\mathcal{C}$-immune. We write $T(n)$-immune and $T(n)$-bi-immune for DTIME($T(n)$)-immune and DTIME($T(n)$)-bi-immune, respectively. Balcázar and Schöning [8] observed that a language L is $T(n)$-bi-immune if and only if every machine that decides L takes more than $T(n)$ time on all but finitely many strings. A set S is $t(n)$-*printable* if there exists a $t(n)$-time-bounded algorithm that on input 0^n outputs all elements of S^n.

2.4 Resource-Bounded Measure

Lutz [33] developed resource-bounded measure theory, analogous to classical Lebesgue measure, to study the quantitative structure of complexity classes. Here we briefly give the definitions; we refer to the survey papers [34, 4] for more detail.

A *martingale* is a function $d : \Sigma^* \to [0, \infty)$ with the property that, for all $w \in \Sigma^*$, $2d(w) = d(w0) + d(w1)$. A martingale d *succeeds* on a language $A \subseteq \Sigma^*$ if

$$\limsup_{n \to \infty} d(A \restriction n) = \infty,$$

where $A \restriction n$ is the length n prefix of A's characteristic sequence. A class X of languages has p-*measure zero*, written $\mu_{\mathrm{p}}(X) = 0$, if there exists a polynomial-time computable martingale that succeeds on every language in X.

2.5 Pseudo Classes

Kabanets [26] defined pseudo classes and refuters. Let A and B be any two languages and R be a nondeterministic polynomial-time machine. We assume that R prints an output along every accepting path. We can view R as a machine that computes a multi-valued function. We say R *distinguishes A from B*, if for infinitely many n, every output of $R(0^n)$ is in $(A \Delta B) \cap \Sigma^n$. Such R is called a *refuter*.

Given a class $\mathcal{C}$, [pseudo$_{\mathrm{NP}}$]-$\mathcal{C}$ is the class of all languages L such that there exists a language L' in $\mathcal{C}$ and every NP machine R, R does not distinguish L from L'. We similarly define the class [pseudo$_{\mathrm{UP}}$]-$\mathcal{C}$ where we only insist that no UP machine distinguishes L from L'.

2.6 Derandomization

Next we briefly review definitions of pseudorandom generators. We refer the reader to the recent surveys of Miltersen [38] and Kabanets [27] for more details.

Let $G = \{G_n : \{0,1\}^{r \log n} \to \{0,1\}^n\}_n$ be a family of functions, and let $C = \{C_n\}_n$ the class of SIZE(n)-circuits. Then we say G *is a pseudo-random generator* if

$$\forall n > 0, |\mathrm{Pr}_{x \in \Sigma^{r \log n}}[C_n(G_n(x))] - \mathrm{Pr}_{x \in \Sigma^n}[C_n(x)]| \leq \frac{1}{n}.$$

4

The celebrated result of Impagliazzo and Wigderson [25] states that pseudo-random generators can be constructed from any Boolean function with high circuit complexity. Klivans and van Melkebeek [29] observed that, the construction of Impagliazzo and Wigderson relativizes, i.e, for any A, given a Boolean function with high A-oracle circuit complexity, one can construct a pseudorandom generator that is secure against A-oracle circuits. More precisely,

Theorem 2.1. (Klivans and van Melkebeek [29]) *Let A be any language. There is a polynomial-time computable function $F : \Sigma^* \times \Sigma^* \to \Sigma^*$, with the following properties. For every $\epsilon > 0$, there exists $a, b \in \mathbb{N}$ such that*

$$F : \Sigma^{n^a} \times \Sigma^{b \log n} \to \Sigma^n,$$

and if r is the truth table of an $a \log n$-variable Boolean function whose A-oracle circuit complexity is bigger than $n^{a\epsilon}$, then $G_r(s) = F(r, s)$, $r \in \Sigma^{n^a}, s \in \Sigma^{b \log n}$, is a pseudo-random generator that is secure against $\mathrm{SIZE}^A(n)$ circuits. If $A = \mathrm{SAT}$, then this pseudo-random generator can be used to derandomize AM to NP and $\mathrm{BPP}^{\mathrm{NP}}$ to P^{NP}.

3 Comparing Hypotheses

We now formally state our principal hypotheses.

Measure Hypothesis. NP *does not have* p-*measure 0.*

Pseudo-NP Hypothesis. *There exists $\epsilon > 0$ such that*

$$\mathrm{NP} \not\subseteq [\text{io-pseudo}_{\mathrm{NP}}]\text{-DTIME}(2^{n^\epsilon}),$$

i.e., there exists a language L in NP such that for every language L' in $\mathrm{DTIME}(2^{n^\epsilon})$, there exists a NP refuter R such that for almost every n, $R(0^n)$ has an output, and if $R(0^n)$ outputs a string x on some path, then $|x| = n$ and $x \in L \Delta L'$.

NP-Machine Hypothesis. *There exists an NP-machine M and $\epsilon > 0$ such that M accepts 0^* and no 2^{n^ϵ}-time-bounded Turing machine computes infinitely many accepting computations of M.*

First, we show that NP-machine hypothesis is weaker than the measure hypothesis.

Theorem 3.1. *The measure hypothesis implies the NP-machine hypothesis.*

Theorem 3.1 follows immediately from Lemma 3.2 and Theorem 3.3 below.

Definition. A language L *does not have superpolynomial gaps* if there is a polynomial $p(n)$ such that for all n, there is some string x in L with $n \le |x| \le p(n)$.

Lemma 3.2. *The measure hypothesis implies that NP contains a $\mathrm{DTIME}(2^{n^\epsilon})$-bi-immune language that does not have superpolynomial gaps.*

Proof. The measure hypothesis actually yields a much stronger conclusion. Mayordomo [36] showed that the class X of all languages that are not $\mathrm{DTIME}(2^{cn})$-bi-immune has p-measure 0. It is well known that the strong law of large numbers holds for p-measure; in particular, the set

$$Y = \left\{ A \,\middle|\, \lim_{n \to \infty} \frac{|A^n|}{2^n} \ne \frac{1}{2} \right\}$$

has p-measure 0. Therefore if NP does not have p-measure 0, then NP contains a language that is not in $X \cup Y$, i.e., a language that is DTIME(2^{cn})-bi-immune and asymptotically contains half of all strings. □

Theorem 3.3. *If* NP *contains a* DTIME($2^{n^{\epsilon}}$)*-immune language that does not have superpolynomial gaps, then the* NP*-machine hypothesis holds.*

Proof. Let $L \in$ NP be a language satisfying the hypothesis and let $p(n) = n^l$ be a polynomial witnessing that L does not have superpolynomial gaps. Let $R(\cdot, \cdot)$ be a polynomial-time decidable relation that is associated with L, i.e, there exists a polynomial $q(\cdot)$ such that

$$x \in L \Leftrightarrow \exists w, |w| \leq q(n), R(x, w) = 1.$$

We now define a NP machine M that accepts 0^*. Given 0^n as input, M guesses strings x and w such that $n \leq |x| \leq p(n)$ and $|w| \leq q(|x|)$. M accepts along this path if and only if $R(x, w) = 1$. Since for every n, L has a string whose length is in between n and $p(n)$, M accepts 0^*. Let $\delta = \epsilon/2$. Assume that there exists a $2^{n^{\delta}}$ time-bounded machine N that computes infinitely many accepting computations of M.

Consider the following algorithm $\mathcal{A}$. On input 0^n, run N on $0^{n^{1/l}}, 0^{n^{1/l}+1}, \cdots 0^n$. If N outputs xw and $|x| = n$ then output x. By our assumption, for infinitely many n, N outputs an accepting computation of $M(0^n)$. Note that every accepting computation of $M(0^n)$ is of the form xw such that $n \leq |x| \leq n^l$ and $R(x, w) = 1$. Thus for infinitely many n, the above algorithm outputs a string x in $L \cap \Sigma^n$. It is clear that the running time of the this algorithm is at most $2^{n^{\epsilon}}$.

Now consider the following algorithm that decides L. Given x as input, run $\mathcal{A}(0^{|x|})$. If $\mathcal{A}(0^{|x|})$ outputs x, then accept x, else run a deterministic algorithm that decides L. Recall that $\mathcal{A}(0^{|x|})$ outputs a string from $L \cap \Sigma^n$ for infinitely many n. Thus the above algorithm correctly decides L. Since $\mathcal{A}(0^{|x|})$ runs in time $2^{n^{\epsilon}}$ time, the above algorithm takes $2^{n^{\epsilon}}$ time on infinitely many strings. This contradicts the bi-immunity of L. Thus no $2^{n^{\delta}}$ time-bounded machine can compute infinitely many accepting computations of M. Thus the NP-machine hypothesis is true. □

In addition to Theorem 3.3, the following is also known regarding immunity and the NP-machine hypothesis.

Theorem 3.4. (Pavan and Selman [40]) *If* NP$\cap$coNP *contains a* DTIME($2^{n^{\epsilon}}$)*-bi-immune language, then the* NP*-machine hypothesis holds.*

Theorem 3.1 tells us that the NP-machine hypothesis is weaker than the measure hypothesis. We would like to know how weak is the NP-machine hypothesis. For example, does NP-machine hypothesis imply the measure hypothesis? We briefly discuss this possibility.

For this, we use the predictor characterization of measure. An oracle Turing machine M is a *predictor* for a language L if for every input x, M decides whether $x \in L$ with oracle $L|x$. Here $L|x$ is the set $\{y \in L \mid y < x\}$. It is known that if the measure hypothesis is true, then NP has a language L with no superpolynomial gaps that has the following property $\mathcal{P}$: Every predictor M that correctly decides L takes more than 2^n time on all but finitely many inputs [7, 39].

Recall that by Theorem 3.3, if NP has a bi-immune language with no super-polynomial gaps, then the NP-machine hypothesis is true. Thus the difference between the measure hypothesis and the NP-machine hypothesis is at least the as much as the difference between property $\mathcal{P}$ and bi-immunity. Observe that if L has property $\mathcal{P}$, then L remains bi-immune, even when the

machine that decides L is given access to partial characteristic sequence of L. This indicates that the property of being bi-immune is much weaker than having property $\mathcal{P}$. In fact, it is easy to construct languages that are bi-immune and do not have property $\mathcal{P}$.

Thus it is reasonable to conjecture that the NP-machine hypothesis is much weaker than the measure hypothesis. There is a relativized world in which the NP-machine hypothesis is true, but the measure hypothesis does not hold [15].

The hypothesis of the following theorem was considered by Impagliazzo, Kabanets, and Wigderson [23]. They used the easy witness method to show that it implies NP = AM.

Theorem 3.5. *If* $\mathrm{NE} \cap \mathrm{coNE} \not\subseteq \mathrm{io\text{-}DTIME}(2^{2^{\epsilon n}})$ *for some* $\epsilon > 0$, *then the* NP*-machine hypothesis holds.*

Proof. Let $L \in \mathrm{NE} \cap \mathrm{coNE}$ but not in $\mathrm{io\text{-}DTIME}(2^{2^{\epsilon n}})$. Let N be a strong nondeterministic machine for L running in time 2^{cn}. Each computation path of N outputs "accept," "reject," or "?". If a string is in L, then there is at least one accepting path and no rejecting paths; if a string is not in L, then there is at least one rejecting path and no accepting paths. We define a nondeterministic machine M that does the following on input 0^n:

- Run N on each string of length $\log n$.

- If all strings of length $\log n$ have been decided, accept.

Then M is an NP machine running in time $O(n^{c+1})$. Suppose that we can compute infinitely many accepting computations of M in 2^{n^ϵ} time. Observe that behavior of M is identical on 0^m for all m where $2^n \leq m < 2^{n+1}$ (recall that log is the discrete logarithm), so we can compute infinitely many accepting computations of M when the size of the input is a power of 2. Then the following algorithm decides L on length n infinitely often. On input of x of length n:

- Compute an accepting computation of M on input 0^{2^n}.

- Examine this computation to find the decision for x.

The algorithm runs in time $O(2^{2^{\epsilon n}})$ and is correct for infinitely many lengths, contradicting our assumption about L. Therefore the NP-machine hypothesis holds. $\qquad\square$

In Theorem 3.7 we will prove that the pseudo-NP hypothesis also implies the NP-machine hypothesis. Our proof makes use of a connection with Kt complexity. The NP-machine hypothesis says that there is an NP search problem (finding an accepting computation for the machine satisfying the hypothesis) which cannot be solved in 2^{n^ϵ} time. Levin [31] showed that an asymptotically optimal algorithm for an NP search problem is to search through all witnesses in order of increasing Kt complexity. If the search problem can be solved in $O(t(n))$ time, then Levin's algorithm will finish in $O(t(n))$ time. Also, the algorithm will only examine witnesses with Kt complexity at most $O(\log t(n))$. Therefore the search problem in the NP-machine hypothesis is for strings that have Kt complexity greater than n^ϵ because Levin's algorithm cannot finish in 2^{n^ϵ} time. Also, the converse holds: if there is an NP search problem for strings with Kt complexity greater than n^ϵ, then the NP-machine hypothesis holds. The following theorem makes this precise.

Theorem 3.6. *The following are equivalent.*

(1) *The* NP*-machine hypothesis.*

(2) *There is an* NP *machine* M *accepting* 0^* *and an* $\epsilon > 0$ *such that for all sufficiently large* n, *every accepting computation path* w *of* $M(0^n)$ *has* $\mathrm{Kt}(w) > n^\epsilon$.

(3) *There is an* NP *machine* M *that has an output on* 0^n *for all* n *and there is an* $\epsilon > 0$ *such that for all sufficiently large* n, *every output* w *of* $M(0^n)$ *has* $\mathrm{Kt}(w) > n^\epsilon$.

Proof. $(1) \Rightarrow (2)$: Assume that M and ϵ satisfy the NP-machine hypothesis. Run Levin's universal search algorithm for 2^{n^ϵ} steps to try to find an accepting computation path of M on input 0^n. This search will fail, but it only considers paths that have Kt complexity at most n^ϵ. Therefore all the accepting paths have Kt complexity greater than n^ϵ, and (2) is satisfied by M.

$(2) \Rightarrow (3)$: Assume that M and ϵ satisfy (2). Define M' to run M. If M accepts, then M' outputs the accepting computation history of M. Then M' and ϵ satisfy (3).

$(3) \Rightarrow (1)$: Assume that M and ϵ satisfy (3). Define M' to run M. If M has an output on a computation path, then M' accepts. Let $\epsilon' < \epsilon$. Suppose that we can compute infinitely many accepting computations of M' in $2^{n^{\epsilon'}}$ time. Let w be the accepting computation we compute for some 0^n, and let x be output of M embedded in w. Then

$$
\begin{aligned}
\mathrm{Kt}(x) &\leq \mathrm{Kt}(w) + O(\log n) \\
&\leq \log n + \log 2^{n^{\epsilon'}} + O(\log n) \\
&< n^\epsilon
\end{aligned}
$$

if n is sufficiently large. This contradicts (3), so M' must satisfy the NP-machine hypothesis with constant ϵ'. □

Using this Kolmogorov complexity connection, we now show that the NP-machine hypothesis is implied by the pseudo-NP hypothesis.

Theorem 3.7. *The pseudo-*NP *hypothesis implies the* NP*-machine hypothesis.*

Proof. Assume that the pseudo-NP hypothesis holds and let L be a language NP that is not in $[\text{io-pseudo}_{\text{NP}}]$-$\mathrm{DTIME}(2^{n^\epsilon})$. Consider the following algorithm that tries to decide L. On input x of length n:

> Use Levin's universal search algorithm to consider every witness w with $\mathrm{Kt}(w) \leq n^\epsilon$, accepting x if a witness is found that proves $x \in L$. Otherwise reject x.

Let L' be the language decided by this algorithm. Then $L' \subseteq L$ and $L' \in \mathrm{DTIME}(2^{n^\epsilon})$.

Since L is not in $[\text{io-pseudo}_{\text{NP}}]$-$\mathrm{DTIME}(2^{n^\epsilon})$, there exists a NP refuter R such that for all but finitely many n, $R(0^n)$ has an output and every output of $R(0^n)$ is a string of length n that is in $L \Delta L'$. Since $L' \subseteq L$, every output of $R(0^n)$ is actually in $L - L'$. Also, note that if $R(0^n)$ outputs x on some path, then every witness w of x has $\mathrm{Kt}(w) > n^\epsilon$.

We now define an NP machine M that satisfies condition (3) of Theorem 3.6:

> Run $R(0^n)$. If this computation outputs a string x, guess a witness that proves $x \in L$ and output it.

Therefore the NP-machine hypothesis holds. □

4   Consequences

In this section we show that several interesting consequences of the NP-machine hypothesis.

Given a string x of length m, we can view it as boolean function $f_x : \{0,1\}^{\log m} \to \{0,1\}$. Allender and Ronneberger [2] showed that the Kt-complexity of a string x is, up to a polynomial factor, the E-oracle circuit complexity complexity of f_x.

Theorem 4.1. *(Allender and Ronneberger [2]) Let A be a complete set for* E. *For any string x:*

- *If the A-oracle circuit complexity of f_x is at most m, then* $\mathrm{Kt}(x) = O(m \log m)$.

- *If* $\mathrm{Kt}(x) \leq m$, *then the A-oracle circuit complexity of f_x is* $O(m^3)$.

Let M be a machine that satisfies the NP-machine hypothesis. By Theorem 3.6, the Kt complexity of every accepting path a of $M(0^n)$ is at least n^ϵ. Thus the E-oracle circuit complexity of f_a is $\Omega(n^{\epsilon/3})$. We obtain that the NP machine hypothesis is equivalent to being able to generate strings with high SIZE$^{\mathrm{E}}$ complexity:

Theorem 4.2. *Let M be an NP machine that accepts 0^* and let A be complete for* E *under linear-time reductions. The following are equivalent.*

1. *There exists $\epsilon > 0$ such that no 2^{n^ϵ}-time bounded Turing machine computes infinitely many accepting computations of M.*

2. *There exists $\delta > 0$ such that for all sufficiently large n, for every accepting computation a of $M(0^n)$, the A-oracle circuit complexity of f_a is at least n^δ.*

We remark the above theorem can be shown without referring to Kt-complexity. The proof that 1. implies 2. can be done as follows. Let M be a machine that satisfies the NP-machine hypothesis with constant ϵ. Consider the following machine N that attempts to find accepting computations of M. On input 0^n, N considers every A-oracle circuit C, over $k \log n$ inputs, of size n^δ and computes the string $w = C(x_1)C(x_2)\cdots C(x_{n^k})$, where $x_1, x_2, \cdots x_{n^k}$ are all strings of length $k \log n$. If w is an accepting computation of $M(0^n)$, then N outputs w. An A-oracle n^δ-size circuit can make at most n^δ queries each of size at most n^δ. Since A can be decided in time 2^{cn}, each oracle query can be answered in time 2^{cn^δ} time. Thus the total time taken to evaluate the value of the circuit on all inputs is $O(2^{2cn^\delta})$. A circuit of size n^δ can be encoded as a string of length $n^\delta \delta \log n$. Thus the machine N considers at most $2^{n^{2\delta}}$ circuits. Thus the total time taken by N is $O(2^{2n^{2\delta}})$ which is less than 2^{n^ϵ}. Since no 2^{n^ϵ}-time-bounded machine can compute infinitely many accepting computations of M, the above machine N fails to output accepting computations of $M(0^n)$ for all but finitely many n. Thus for all but finitely many n, for every accepting computation w_n of $M(0^n)$, the A-oracle circuit complexity of f_{w_n} is bigger than n^δ.

The following theorem states several consequences of the NP-machine hypothesis.

Theorem 4.3. *The* NP-*machine hypothesis implies the following for every $A \in$* E.

(1) $\mathrm{BP} \cdot \mathrm{NP}^A = \mathrm{NP}^A$. *In particular,* $\mathrm{AM} = \mathrm{NP}$.

(2) *There exists $\epsilon > 0$ such that* $\mathrm{E}^{\mathrm{NP}} \not\subseteq$ io-$\mathrm{SIZE}^A(2^{\epsilon n})$.

(3) $\mathrm{BP} \cdot \Delta_2^{\mathrm{P},A} = \Delta_2^{\mathrm{P},A}$. *In particular,* $\mathrm{BPP}^{\mathrm{NP}} = \mathrm{P}^{\mathrm{NP}}$.

(4) NEXP $\not\subseteq$ P/poly.

(5) *For every constant $k > 0$, $\mathrm{P}^{\mathrm{NP}} \not\subseteq$ io-SIZE(n^k).*

(6) *There exist $\epsilon > 0$, $\delta > 0$ such that NP $\not\subseteq$ io-DTIME$(2^{n^\delta})/n^\epsilon$.*

(7) $\mathrm{BP} \cdot \mathrm{PP} = \mathrm{PP}$. *In particular,* PH $\subseteq$ PP.

(8) NP *is not easy on average.*[1]

Proof. Since the NP machine hypothesis is true, there exist an NP machine M and $\epsilon > 0$ such that no 2^{n^ϵ}-time-bounded machine can compute infinitely many accepting computations of M. By Theorem 4.2, there is a $\delta > 0$ such that for every accepting computation a of $M(0^n)$, f_a has A-oracle circuit complexity at least n^δ. Without loss of generality assume that the length of every accepting computation of $M(0^n)$ is n^k.

(1) Let L be any language in AM. Let the randomness of Arthur is bounded by n^{rk}. Let $\epsilon' = \delta/k$. Let a and b be the constants from Theorem 2.1. The following NP machine accepts L. Given an input x of length n guess an accepting computation w of $M(0^{n^{ra}})$. Let $m = n^{kr}$. Note that $|w| = n^{kra}$ and it can be viewed as a boolean function over $a \log m$ variables. By Theorem 4.2, any accepting computation w gives a boolean function over $a \log m$ variables whose SAT-oracle circuit complexity at least $(n^{ra})^\delta = m^{a\epsilon'}$. By Theorem 2.1, from this hard Boolean function a pseudorandom generator that maps $b \log m$ bits to $m = n^{kr}$ bits can be constructed, which derandomizes the AM protocol.

(2) Let A be any language in E. Let a_n denote the maximum accepting computation of $M(0^n)$. Thus $|a_n| = n^k$. By Theorem 4.2, f_{a_n} has A-oracle circuit complexity bigger than n^δ. We now define a language L as follows: Let $L|m$ denote the characteristic sequence of L on strings of length m. Given m, let m' be the largest integer such that $m' < m$ and m' is divisible by k. Let $n = 2^{m'/k}$. We set $L|m = a_n 0^l$, where $l = 2^m - 2^{m'}$.

We claim that L is in E^{NP}. Given a string x of length m, we can compute m' and now the goal is to compute the maximum accepting computation of $M(0^{2^{m'/k}})$ which gives $L|m$. Note that the length of the maximum accepting computation of $M(0^{2^{m'/k}}) = 2^{m'} \le 2^m$. Thus we can extract the maximum accepting computation of $M(0^{2^{m'/k}})$ in linear-exponential time using an NP oracle.

Let $\delta' = \delta/2k$. Assume that $L \in$ io-SIZE$^A(2^{\delta'm})$. Thus for infinitely many m there exists a A-oracle circuit that accepts L^m. This implies that the A-oracle circuit complexity of $f_{L|m}$ is at most $2^{\delta'm}$. Recall that $L|m = a_n 0^l$, where $n = 2^{m'/k}$ and $l = 2^m - 2^{m'}$. Thus the A-oracle circuit complexity of f_{a_n} is at most $2^{\delta'm}$. However $n^\delta = 2^{m'\delta/k} > 2^{\delta'm}$ for large enough m. Thus if $L \in$ io-SIZE$^A(2^{\delta'm})$, then for infinitely many n the A-oracle circuit complexity of a_n is less than n^δ. This is a contradiction.

(3) This immediately follows from (2) and Theorem 2.1.

(4) Impagliazzo, Kabanets, and Wigderson [23] showed that if NEXP $\subseteq$ P/poly, then NEXP = MA. By item (1), the NP-machine hypothesis implies AM = NP. Thus NEXP = NP which is a contradiction of the nondeterministic time hierarchy theorem. Therefore NEXP $\not\subseteq$ P/poly.

(5) By the results of Kannan [28], Bshouty et al. [10], and Kobler and Watanabe [30], ZPP$^{\mathrm{NP}} \not\subseteq$ io-SIZE(n^k) for any $k > 0$. Since the NP-machine hypothesis implies BPP$^{\mathrm{NP}}$ = P$^{\mathrm{NP}}$, P$^{\mathrm{NP}} \not\subseteq$ io-SIZE(n^k) for every $k > 0$.

---

[1] For the definition of easy on average we refer to [11].

(6) Let $\epsilon' = \epsilon/3$ and $\delta' = \epsilon/6k$. Suppose NP $\subseteq$ io-DTIME$(2^{n^{\delta'}})/n^{\epsilon'}$. Consider the following language in NP.

$$L' = \left\{ \langle 0^n, y \rangle \,\middle|\, \begin{array}{l} |y| \le n^k, \text{ there exists } w \text{ such that } yw \\ \text{is an accepting computation of } N \text{ on } 0^n \end{array} \right\}.$$

We can use a pairing function $\langle \cdot, \cdot \rangle$ that encodes all tuples of the form $\langle 0^n, y \rangle, |y| \le n^k$, at the length n^r. Thus L' has strings at length n^r, $n \ge 1$.

We now define a language L as follows: at lengths of the form n^r, L coincides with L'. Let m be a length between n^r and $(n+1)^r$. We define $L|m = (L|n^r)0^{2^m - 2^{n^r}}$.

Observe that if there is an oracle that gives the membership of all strings of length n^r in L', then we can compute an accepting computation of $M(0^n)$. Thus if there is an oracle that gives membership of all strings of length m, for some m between n^r and n^{r+1}, in L, then also we can compute an accepting computation of $M(0^n)$. Since NP $\subseteq$ io-DTIME$(2^{n^{\delta'}})/n^{\epsilon'}$, there is a L'' in DTIME$(2^{n^{\delta'}})$ and an advice h_n such that, $|h_n| \le n^{\epsilon'}$,

$$\exists^\infty m, \forall x \in \{0,1\}^m, x \in L \Leftrightarrow \langle x, h_m \rangle \in L''.$$

Consider the following algorithm that computes infinitely many accepting computations of M. On input 0^n, it considers each length m such that $n^r \le m < n^{r+1}$. For each length m its considers all advices of size up to $m^{\epsilon'}$ and with each advice it tries to compute a witness for $M(0^n)$ witness by querying L''. For infinitely many lengths m, there is an advice h_m such that $x \in L$ if and only if $\langle x, h_m \rangle \in L''$. Thus this algorithm outputs infinitely many accepting computations of M. It can be verified that the algorithm is 2^{n^ϵ}-time bounded. This is a contradiction.

(7) Given an instance x of a BP$\cdot$PP language, guess a computation path p of the NP-machine. If p is accepting, it has high circuit complexity, so build a generator with it and derandomize the BP$\cdot$PP computation (as in [5]) for x. Let d be the length of a computation path that implements this derandomization. If p is rejecting, make a computation tree of depth d that is half accepting paths and half rejecting paths.

(8) Buhrman, Fortnow, and Pavan [11] showed that if NP is easy on average, then for any NP-machine M accepting a tally set, there is a polynomial-time algorithm M that outputs accepting computations of M. Therefore if NP is easy on average, then the NP-machine hypothesis is false. $\square$

From Theorem 3.4, we know that if NP $\cap$ coNP has 2^{n^ϵ}-bi-immune sets, then the NP-machine hypothesis is true. This gives the following corollary.

Corollary 4.4. *If* NP $\cap$ coNP *has* 2^{n^ϵ}*-bi-immune sets, then all the consequences in Theorem 4.3 follow.*

By Theorem 3.3, if NP has a 2^{n^ϵ}-bi-immune language that does not have superpolynomial gaps, then the NP-machine hypothesis is true. Thus all the consequences of Theorem 4.3 follow if NP has a 2^{n^ϵ}-bi-immune language that does not have superpolynomial gaps. We now consider the hypothesis "NP has a 2^{n^ϵ}-bi-immune language". This is weaker than both the hypotheses "NP $\cap$ coNP has a 2^{n^ϵ}-bi-immune set" and "NP contains a 2^{n^ϵ}-bi-immune language that does not have superpolynomial gaps." What consequences follow from this hypothesis?

If L is any 2^{n^ϵ}-bi-immune language in NP, then for infinitely many n, 0^n belongs to L. By using similar arguments as in Theorem 4.2, we can show that for infinitely many n, every witness of 0^n has high circuit complexity. This gives infinitely-often versions of the consequences in Theorem 4.3. For example:

Theorem 4.5. *If* NP *has a* 2^{n^ϵ}*-bi-immune language, then the following hold.*

(1) AM $\subseteq$ io-NP.

(2) *There exists* $\epsilon > 0$ *such that for every A in* E, $\mathrm{E}^{\mathrm{NP}} \not\subseteq \mathrm{SIZE}^A(2^{\epsilon n})$.

(3) NEXP $\not\subseteq$ P/poly.

Fenner, Fortnow, Naik, and Rogers [12] studied the following hypothesis **Q**:

For any NP machine M that accepts $\{0,1\}^*$, there is a polynomial-time algorithm that computes an accepting computation of M for each input.

Clearly the NP-machine hypothesis implies that **Q** is false. It is shown in [12] that **Q** has several equivalent characterizations, so we obtain from there a number of consequences of the NP-machine hypothesis. For example, while we do not know if the NP-machine hypothesis implies that P $\neq$ NP $\cap$ coNP, we have the following.

Corollary 4.6. *The* NP*-machine hypothesis implies that* P $\neq$ NP $\cap$ coNP *or there is an* NP *multi-valued total function that does not have a* NP *single-valued refinement.*

Corollary 4.7. *The* NP*-machine hypothesis implies there is a* NP *multi-valued total function that does not have a polynomial-time refinement.*

By Theorem 3.1, the measure hypothesis implies that NP-machine hypothesis. This gives the following corollary.

Corollary 4.8. *Assume that the measure hypothesis holds. The following statements hold:*

- *Either* P $\neq$ NP $\cap$ coNP *or there is an* NP *multi-valued total function that does not have a* NP*-single valued refinement.*

- *There is a* NP *multi-valued total function that does not have a polynomial-time refinement.*

We note that these consequences were not known to follow from the measure hypothesis. Our connection between the measure hypothesis and the NP-machine hypothesis makes these consequences possible.

We conclude this section with some observations about the NP-machine hypothesis and sets of strings that have high Kolmogorov complexity. Allender [3, 1] defined for any language L the function

$$\mathrm{Kt}_L(n) = \min\{\mathrm{Kt}(x) \mid x \in L_{=n}\}.$$

If L is empty at length n, then $\mathrm{Kt}_L(n)$ is undefined. In [2], the following conditions are shown equivalent:

- There is a language $L \in$ P and an $\epsilon > 0$ such that for all large n, $\mathrm{Kt}_L(n)$ is defined and $Kt_L(n) > n^\epsilon$.

- There is a language $L \in$ NP and an $\epsilon > 0$ such that for all large n, $\mathrm{Kt}_L(n)$ is defined and $Kt_L(n) > n^\epsilon$.

- There is a language $L \in \mathrm{AC}_0$ and an $\epsilon > 0$ such that for all large n, $\mathrm{Kt}_L(n)$ is defined and $Kt_L(n) > n^\epsilon$.

The following theorem states that these conditions are also equivalent to the NP-machine hypothesis. The proof is straightforward given Theorem 3.6.

Theorem 4.9. *The following are equivalent.*

(1) *The NP-machine hypothesis.*

(2) *There is a language $L \in$ NP and an $\epsilon > 0$ such that for all large n, $\mathrm{Kt}_L(n)$ is defined and $\mathrm{Kt}_L(n) > n^\epsilon$.*

These equivalences also extend to alternate definitions of time-bounded Kolmogorov complexity:

Theorem 4.10. *The following are equivalent.*

(1) *The NP-machine hypothesis.*

(2) *There is a set $A \in$ P containing a string at all but finitely many lengths and an $\epsilon > 0$ such that for all $w \in A$, $K^{2^{n^\epsilon}}(w) > n^\epsilon$.*

(3) *There is a set $A \in$ P containing a string at all but finitely many lengths and an $\epsilon > 0$ such that for all $w \in A$, $K^{2^{n^\epsilon}}(w) > \log n + O(1)$.*

Pavan and Selman [40] showed that NP-machine hypothesis implies that $\leq^{\mathrm{P}}_{\mathrm{T}}$-completeness is different from $\leq^{\mathrm{P}}_{\mathrm{m}}$-completeness for NP. An earlier result of Watanabe and Tang [41] achieved the same separation for PSPACE, under a Kolmogorov complexity hypothesis that is strikingly similar to the statements in Theorem 4.10 that are equivalent to the NP-machine hypothesis.

Theorem 4.11. (Watanabe and Tang [41]) *Suppose there is a set $A \in$ PSPACE without super-polynomial gaps and an $\epsilon > 0$ such that for every polynomial p, for all but finitely many $x \in A$, $K^{p(n)}(x) > n^\epsilon$. Then $\leq^{\mathrm{P}}_{\mathrm{T}}$-completeness differs from $\leq^{\mathrm{P}}_{\mathrm{m}}$-completeness for PSPACE.*

5 Variations of the NP-Machine Hypothesis

In this section we consider variations of our nondeterministic hypotheses: replacing NP with UP and RP. We show that the UP-machine hypothesis yields additional consequences. The RP-machine hypothesis turns out to be equivalent to ZPP = EXP.

UP-Machine Hypothesis. *There exists a UP-machine M and $\epsilon > 0$ such that M accepts 0^* and no 2^{n^ϵ}-time-bounded Turing machine computes infinitely many accepting computations of M.*

Pseudo-UP Hypothesis. *There exists $\epsilon > 0$ such that UP $\not\subseteq$ [io-pseudo$_{\mathrm{UP}}$]-DTIME(2^{n^ϵ}).*

Theorems 5.1 and 5.2 below are analogues of Theorems 3.7 and 3.5. However, we do not have a UP version of Theorem 3.1, which would state that the measure hypothesis for UP implies the UP-machine hypothesis.

Theorem 5.1. *The pseudo-UP hypothesis implies the UP-machine hypothesis.*

Proof. Similar to the proof of Theorem 3.7. □

Theorem 5.2. *The UP-machine hypothesis is equivalent to $(\exists \epsilon > 0)$UE$\cap$coUE $\not\subseteq$ io-DTIME$(2^{2^{\epsilon n}})$.*

Proof sketch. The proof from right to left is analogous to Theorem 3.5. For the other direction, let M and ϵ satisfy the UP-machine hypothesis. We define a $\mathrm{UE} \cap \mathrm{coUE}$ language A as follows. For length n, we look at the unique accepting computation path of M on input 0^N where $N = 2^{n/\epsilon}$. This path has at least 2^n bits, and we use the first 2^n bits for the characteristic string of A at length n. $\square$

The following result is analogous to Theorem 4.2.

Theorem 5.3. *Let M be a* UP *machine that accepts* 0^* *and let A be complete for* E *under linear-time reductions. The following are equivalent.*

1. *There exists $\epsilon > 0$ such that no 2^{n^ϵ}-time machine computes infinitely many accepting computations of M.*

2. *There exists $\delta > 0$ such that for all sufficiently large n, the accepting computation of $M(0^n)$ has A-oracle circuit complexity at least n^δ.*

It is obvious that all the consequences of the NP-machine hypothesis follow from the UP-machine hypothesis. In addition, we obtain the following consequences.

Theorem 5.4. *The* UP*-machine hypothesis implies the following.*

(1) *There exists $\epsilon > 0$ such that for all $A \in E$, $\mathrm{UE} \cap \mathrm{coUE} \not\subseteq \mathrm{io\text{-}SIZE}^A(2^{\epsilon n})$.*

(2) $\mathrm{PH} \subseteq \mathrm{SPP}$.

(3) *There exist $\epsilon > 0$, $\delta > 0$ such that $\mathrm{UP} \cap \mathrm{coUP} \not\subseteq \mathrm{io\text{-}DTIME}(2^{n^\delta})/n^\epsilon$.*

(4) *For every constant $k > 0$, $\mathrm{SPP} \not\subseteq \mathrm{io\text{-}SIZE}(n^k)$.*

(5) $\mathrm{BPP} \subseteq \mathrm{UP} \cap \mathrm{coUP}$.

(6) *There exists a language in* NP *for which search does not reduce to decision.*[2]

Proof. Let M and ϵ satisfy the UP-machine hypothesis. From Theorem 5.3 we know that the accepting computation of $M(0^n)$ has high A-oracle circuit complexity.

(1) The proof is similar to the proof of item 2 in Theorem 4.3. The construction of L is exactly same, except that in this case a_n is the unique accepting computation of $M(0^n)$. Since a_n is the unique accepting computation, we can place L in $\mathrm{UE} \cap \mathrm{coUE}$.

(2) By the results of Fenner, Fortnow, and Kurtz [13], $\mathrm{UE} \cap \mathrm{coUE} \subseteq \mathrm{E}^{\mathrm{SPP}}$. Thus by (1), for every k, there exists a language L in $\mathrm{E}^{\mathrm{SPP}}$ whose SAT_k-oracle circuit complexity is bigger than $2^{\epsilon n}$, where SAT_k is the set of satisfiable quantified boolean formula with k quantifiers. This implies $\mathrm{PH} \subseteq \mathrm{SPP}$ [18].

(3) The proof is similar to the proof of item 6 in Theorem 4.3. Consider the following language.

$$L = \{\langle 0^n, i \rangle \mid \text{ the } i\text{th bit of the accepting computation of } M(0^n) \text{ is } 1\}.$$

Observe that L is in $\mathrm{UP} \cap \mathrm{coUP}$. Let $\delta' = \epsilon/3$ and $\epsilon' = \epsilon/6k$. Assume L is in $\mathrm{io\text{-}DTIME}(2^{n^{\epsilon'}})/n^{\delta'}$. As in the proof of item (6) in Theorem 4.3, we can cycle through all advice strings of length $n^{\delta'}$ and

---

[2]For the definition of "search reduces to decision" we refer to [9].

compute the accepting computation of $M(0^n)$ for infinitely many n. This process takes at most 2^{n^ϵ} time. Thus $\mathrm{UP} \cap \mathrm{coUP} \not\subseteq \text{io-DTIME}(2^{n^{\epsilon'}})/n^{\delta'}$.

(4) Kannan [28] showed that for every $k > 0$, $\mathrm{PH} \not\subseteq \text{io-SIZE}(n^k)$. Combining this with item (2), we get for every $k > 0$, $\mathrm{SPP} \not\subseteq \text{io-SIZE}(n^k)$.

(5) Let $L \in \mathrm{BPP}$. On input x, the UP machine guesses the accepting computation of $M(0^m)$. This accepting computation has high circuit complexity. The UP machine uses this hard function to construct a pseudorandom generator that can derandomize BPP.

(6) Consider the following machine M'. Given 0^n, M' guesses a path of $M(0^{2^{2^n}})$ and accepts if and only if the guessed path is an accepting computation. Since M accepts 0^*, M' also accepts 0^*, and M' is an UEE machine. Since no 2^{n^ϵ}-time bounded machine can compute infinitely many accepting computations of M, no EE machine can compute infinitely many accepting computations of M'. This implies that $\mathrm{UEE} \neq \mathrm{EE}$. Bellare and Goldwasser [9] showed that if $\mathrm{NEE} \neq \mathrm{EE}$, then there exists a language in $\mathrm{NP} - \mathrm{P}$ for which search does not reduce to decision. $\square$

Finally, we consider randomized versions of the hypotheses. Because of results in derandomization we expect that $\mathrm{P} = \mathrm{RP}$, so we do not expect these hypotheses to be true. In fact, it turns out that they are all equivalent to $\mathrm{ZPP} = \mathrm{EXP}$.

Pseudo-RP **Hypothesis.** *There exists $\epsilon > 0$ such that*

$$\mathrm{RP} \not\subseteq [\text{io-pseudo}_{\mathrm{ZPP}}]\text{-DTIME}(2^{n^\epsilon}),$$

i.e., there exists a language L in RP *such that for every language L' in* $\mathrm{DTIME}(2^{n^\epsilon})$, *there exists a* ZPP *refuter R such that for almost every n, every output of $R(0^n)$ is a string of length n that is in $L \Delta L'$.*

RP-**Machine Hypothesis.** *There exists an* RP *machine M and $\epsilon > 0$ such that M accepts 0^* and no 2^{n^ϵ}-time-bounded Turing machine computes infinitely many accepting computations of M.*

Kabanets [26] showed that the pseudo-RP hypothesis is equivalent to $\mathrm{ZPP} = \mathrm{EXP}$. Impagliazzo and Moser [24] showed that $\mu_{\mathrm{p}}(\mathrm{RP}) \neq 0$ is also equivalent to $\mathrm{ZPP} = \mathrm{EXP}$. Along the same lines, we can show the same for the RP-machine hypothesis, yielding the following.

Theorem 5.5. *The following are equivalent.*

(1) *The* RP-*machine hypothesis.*

(2) RP *does not have* p-*measure 0.*

(3) *The pseudo-*RP *hypothesis.*

(4) $\mathrm{ZPP} = \mathrm{EXP}$.

6 Scaled Dimension

In addition to the measure hypothesis on NP, hypotheses on the resource-bounded dimension and scaled dimension of NP have also been considered [17, 22]. While these hypotheses are easily seen to be weaker than the measure hypothesis, they seem incomparable with the other hypotheses considered in this paper. In this section we consider a hypothesis on the scaled dimension of NP

and its consequences for derandomization of NP and circuit-complexity lower bounds for NEXP. For technical background on scaled dimension, we refer to [20, 19, 21].

In the following, we consider the hypothesis that $\dim_{\mathrm{p}}^{(-3)}(\mathrm{NP})$, the -3rd-order scaled polynomial-time dimension of NP, is positive. If $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$, then $\dim_{\mathrm{p}}^{(-3)}(\mathrm{NP}) > 0$. We first show that this scaled dimension hypothesis implies an advice version of the NP-machine hypothesis.

Theorem 6.1. *If* $\dim_{\mathrm{p}}^{(-3)}(\mathrm{NP}) > 0$, *then there is an* NP-*machine* M *such that for every rational* $\delta > 0$ *and every* 2^n-*time Turing machine* R, *the following hold for all sufficiently large* n.

(1) *There is an advice string* a_n *with* $|a_n| \leq n^\delta$ *such that* $M(0^n, a_n)$ *accepts.*

(2) $R(0^n, a_n)$ *does not output an accepting computation of* $M(0^n, a_n)$.

Proof sketch. From Theorem 5.2 in [19], we have that $\dim_{\mathrm{p}}^{(-3)}(\mathrm{NP}) > 0$ implies there is a language $A \in \mathrm{NP}$ such that for any Turing machine D that decides A, for all $\delta > 0$, D halts within 2^{2n} steps on fewer than 2^{n^δ} strings up to length n, for all but finitely many n.

Let N be an NP-machine for A. Fix a rational $\delta > 0$ and let R be any 2^n-time Turing machine. Let $m = n^\delta$ and divide $\{0,1\}^n$ into 2^m consecutive blocks of length 2^{n-m}, where each block B_b is indexed by a string $b \in \{0,1\}^m$. Define a machine D that on input $w \in \{0,1\}^n$ runs $R(0^n, b)$ for all $b \in \{0,1\}^m$. If any of these runs of R output an accepting computation of $N(w)$, then D accepts w; otherwise D uses an exhaustive search to decide whether w is in A. By the pigeonhole principle, there is some index a_n such that $D(w)$ does not find an accepting computation for any $w \in B_{a_n}$.

Consider an NP-machine M that on input $(0^n, b)$ where $|b| = n^\delta$ guesses a string $w \in B_b$ and a computation path of $N(w)$ and accepts if the computation path is accepting. (Note that M does not depend on δ or R.) The theorem follows if B_{a_n} contains a string from A for all sufficiently large n. Let X be the class of all languages L such that

$$(\exists^\infty n) \; L \text{ is empty on some block of size } 2^{n-n^\delta} \text{ at length } n.$$

If we have that $\dim_{\mathrm{p}}^{(-3)}(X) = 0$, then we can additionally assume that $A \notin X$ and therefore $B_{a_n} \cap A \neq \emptyset$ for sufficiently large n.

In fact, the stronger claim $\dim_{\mathrm{p}}^{(-1)}(X) = 0$ holds. We sketch a proof of this using the terminology and techniques of [19]. For each n, let W_n be the set of all characteristic strings of subsets of $\{0,1\}^{\leq n}$ that are empty on some block of size 2^{n-n^δ} at length n. Then $|W_n| \leq 2^{n^\delta + 2^{n+1} - 2^{n-n^\delta}}$. Define a measure ρ_n by letting

$$\rho_n(w) = \frac{|\{w' \in W_n \mid w \sqsubseteq w'\}|}{|W_n|}.$$

Let $L \in X$. Then for infinitely many n, L has a prefix $w_n \in W_n$. Then

$$-\log \rho_n(w_n) = \log |W_n| \leq 2^{n+1} - 2^{n-n^\delta} + n^\delta.$$

This is less than $g_{-1}(|w_n|, s)$ for any $s > 0$ when n is sufficiently large. Since all the ρ_n are computable within a fixed polynomial-time bound (independent of δ), $\dim_{\mathrm{p}}^{(-1)}(X) = 0$ follows Theorem 4.6 in [19]. $\qquad\square$

Using Theorem 6.1, we show that the scaled dimension hypothesis implies a derandomization of AM with a small amount of advice. This derandomization in the following theorem should be compared with the unconditional fact that $\mathrm{AM} \subseteq \mathrm{NP}/\mathrm{poly}$.

Theorem 6.2. *If* $\dim_{\mathrm{p}}^{(-3)}(\mathrm{NP}) > 0$, *then* $\mathrm{AM} \subseteq \mathrm{NP}/n^\epsilon$ *for every* $\epsilon > 0$.

Proof. Assume the hypothesis and obtain the NP-machine M from Theorem 6.1. Let $B \in \mathrm{AM}$ and $\gamma > 0$ be rational. We will show that $B \in \mathrm{NP}/n^\gamma$.

Let the randomness of Arthur be bounded by n^r. Suppose the witnesses for M on inputs of the form $(0^n, x)$, where $|x| \le n$, have length n^c. Fix $\epsilon = \frac{1}{2c}$ and obtain the constants a and b from Theorem 2.1. Let $\delta = \gamma \frac{c}{ar}$.

Consider a machine R that on input $(0^n, w)$ runs through all potential witnesses for $M(0^n, w)$ that (when viewed as Boolean functions) have SAT-oracle circuit-size complexity bounded by $n^{1/2}$ and outputs a valid witness if it finds one. Obtain the advice strings a_n from Theorem 6.1 for parameters R and δ. As $R(0^n, a_n)$ uses less than 2^n time, it fails to find an accepting computation of $M(0^n, a_n)$ for all sufficiently large n.

Let $x \in \{0,1\}^n$ and let $m = n^r$. Let $l = n^{\frac{ar}{c}}$ and guess a witness of $M(0^l, a_l)$. This witness has length $l^c = m^a$, so it can be viewed as the truth-table of a Boolean function over $a \log m$ variables. From the previous paragraph we know that this function has SAT-oracle circuit complexity at least $l^{\frac{1}{2}} = m^{\epsilon a}$. From Theorem 2.1 we obtain a pseudorandom generator providing $m = n^r$ pseudorandom bits that can be used to derandomize the AM computation. This argument works for all sufficiently large n, so we have $B \in \mathrm{NP}/n^\gamma$. $\qquad\square$

Combining Theorem 6.2 and arguments of Impagliazzo, Kabanets, and Wigderson [23], we show that the same hypothesis implies NEXP does not have polynomial-size circuits.

Theorem 6.3. *If* $\dim_{\mathrm{p}}^{(-3)}(\mathrm{NP}) > 0$, *then* $\mathrm{NEXP} \not\subseteq \mathrm{P/poly}$.

Proof. Assume that $\dim_{\mathrm{p}}^{(-3)}(\mathrm{NP}) > 0$ and $\mathrm{NEXP} \subseteq \mathrm{P/poly}$. By Theorem 6.2 we have

$$\mathrm{AM} \subseteq \mathrm{NP}/n. \tag{6.1}$$

Corollary 8 of [23] tells us that if $\mathrm{NEXP} \subseteq \mathrm{P/poly}$, then

$$\mathrm{EXP} \not\subseteq \text{io-}[\mathrm{NTIME}(2^n)/n]. \tag{6.2}$$

Also $\mathrm{EXP} \subseteq \mathrm{NEXP}$, so we have $\mathrm{EXP} \subseteq \mathrm{P/poly}$ which yields

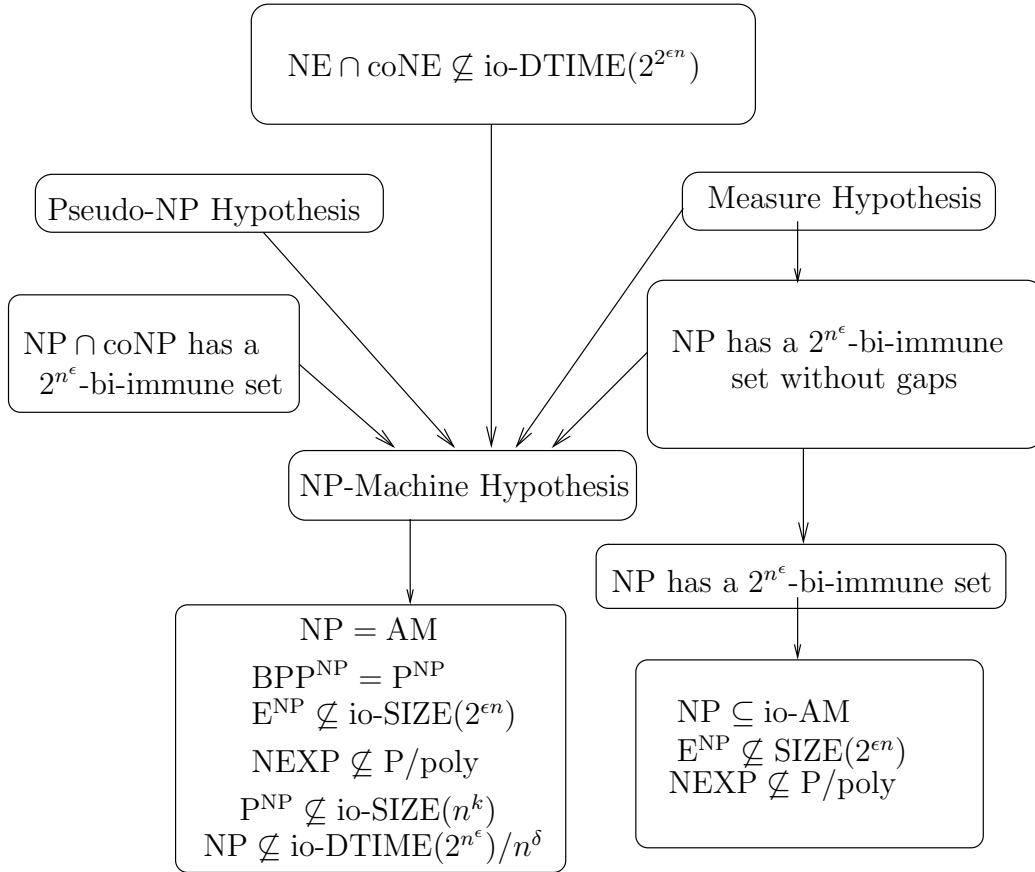$$\mathrm{EXP} \subseteq \mathrm{MA} \tag{6.3}$$

by [6]. Putting (6.1) and (6.3) together, we have

$$\mathrm{EXP} \subseteq \mathrm{NP}/n,$$

a contradiction of (6.2). $\qquad\square$

7 Conclusion

The following figure summarizes relations among several hypotheses and their consequences. It is interesting to note that the UP-machine hypothesis implies $\mathrm{P} \ne \mathrm{UP} \cap \mathrm{coUP}$, whereas a similar consequence is not known to follow from the NP-machine hypothesis. Theorem 3.1 partly explains this. Since the measure hypothesis implies the NP-machine hypothesis, if the NP-machine hypothesis implies $\mathrm{P} \ne \mathrm{NP} \cap \mathrm{coNP}$, then the measure hypothesis also implies $\mathrm{P} \ne \mathrm{NP} \cap \mathrm{coNP}$. However, it seems that the measure hypothesis does not say much about $\mathrm{NP} \cap \mathrm{coNP}$.

$$NE \cap coNE \not\subseteq \text{io-DTIME}(2^{2^{\epsilon n}})$$

Pseudo-NP Hypothesis

Measure Hypothesis

$NP \cap coNP$ has a 2^{n^ϵ}-bi-immune set

NP has a 2^{n^ϵ}-bi-immune set without gaps

NP-Machine Hypothesis

NP has a 2^{n^ϵ}-bi-immune set

$NP = AM$
$BPP^{NP} = P^{NP}$
$E^{NP} \not\subseteq \text{io-SIZE}(2^{\epsilon n})$
$NEXP \not\subseteq P/poly$
$P^{NP} \not\subseteq \text{io-SIZE}(n^k)$
$NP \not\subseteq \text{io-DTIME}(2^{n^\epsilon})/n^\delta$

$NP \subseteq \text{io-AM}$
$E^{NP} \not\subseteq \text{SIZE}(2^{\epsilon n})$
$NEXP \not\subseteq P/poly$

There are several unanswered questions. For example: does the NP-machine hypothesis imply $NEXP \not\subseteq \text{io-P}/poly$? How about $NP \not\subseteq \text{SIZE}(n^k)$? Though the NP-machine hypothesis implies derandomization of AM and BPP^{NP}, we do not know whether we can derandomize BPP. Note that relative to any oracle where $ZPP = EXP$, the NP-machine hypothesis holds and $BPP = EXP$. Perhaps we can derandomize BPP using the UP-machine hypothesis. Another interesting question is the relation between these hypotheses and the existence of cryptographic one-way functions.

Acknowledgments

References

[1] E. Allender. Applications of time-bounded Kolmogorov complexity in complexity theory. In O. Watanabe, editor, *Kolmogorov Complexity and Computational Complexity*, pages 4–22. Springer-Verlag, 1992.

[2] E. Allender. When worlds collide: Derandomization, lower bounds, and Kolmogorov complexity. In *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 1–15. Springer-Verlag, 2001.

[3] E. W. Allender. Some consequences of the existence of pseudorandom generators. *Journal of Computer and System Sciences*, 39:101–124, 1989.

[4] K. Ambos-Spies and E. Mayordomo. Resource-bounded measure and randomness. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, Lecture Notes in Pure and Applied Mathematics, pages 1–47. Marcel Dekker, New York, N.Y., 1997.

[5] V. Arvind and J. Köbler. On pseudorandomness and resource-bounded measure. *Theoretical Computer Science*, 255(1–2):205–221, 2001.

[6] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[7] J. Balcazar and E. Mayordomo. A note on genericty and bi-immunity. In *Proceedings of the Tenth Annual IEEE Conference on Computational Complexity*, pages 193–196, 9995.

[8] J. L. Balcázar and U. Schöning. Bi-immune sets for complexity classes. *Mathematical Systems Theory*, 18:1–10, 1985.

[9] M. Bellare and S. Goldwasser. The complexity of decision versus search. *SIAM Journal on Computing*, 23(1):97–119, 1994.

[10] N. H. Bshouty, R. Cleve, S. Kannan, R. Gavaldà, and C. Tamon. Oracles and queries that are sufficient for exact learning. *Journal of Computer and System Sciences*, 52(3):421–433, 1996.

[11] H. Buhrman, L. Fortnow, and A. Pavan. Some reuslts on derandomization. *Theory of Computing Systems*, 38(2):211–227, 2005.

[12] S. Fenner, L. Fortnow, A. Naik, and J. Rogers. Inverting onto functions. *Information and Computation*, 186(1):90–103, 2003.

[13] S. A. Fenner, L. Fortnow, and S. A. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

[14] C. Glaßer, A. Pavan, A. L. Selman, and S. Sengupta. Properties of NP-complete sets. In *Proceedings of the 19th IEEE Conference on Computational Complexity*, pages 184–197. IEEE Computer Society, 2004.

[15] R. C. Harkins and J. M. Hitchcock. Upward separations and weaker hypotheses in resource-bounded measure. Manuscript, 2006.

[16] L. Hemaspaandra, J. Rothe, and G. Wechsung. Easy sets and hard certificate schemes. *Acta Informatica*, 34(11):859–879, 1997.

[17] J. M. Hitchcock. MAX3SAT is exponentially hard to approximate if NP has positive dimension. *Theoretical Computer Science*, 289(1):861–869, 2002.

[18] J. M. Hitchcock. The size of SPP. *Theoretical Computer Science*, 320(2–3):495–503, 2004.

[19] J. M. Hitchcock. Small spans in scaled dimension. *SIAM Journal on Computing*, 34(1):170–194, 2004.

[20] J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. Scaled dimension and nonuniform complexity. *Journal of Computer and System Sciences*, 69(2):97–122, 2004.

[21] J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. The fractal geometry of complexity classes. *SIGACT News*, 36(3):24–38, September 2005.

[22] J. M. Hitchcock, A. Pavan, and N. V. Vinodchandran. Partial bi-immunity, scaled dimension, and NP-completeness. *Theory of Computing Systems*. To appear.

[23] R. Impagliazzo, V. Kabanets, and A. Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672–694, 2002.

[24] R. Impagliazzo and P. Moser. A zero-one law for RP. In *Proceedings of the 18th IEEE Conference on Computational Complexity*, pages 43–47. IEEE Computer Society, 2003.

[25] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Symposium on Theory of Computing*, pages 220–229, 1997.

[26] V. Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. *Journal of Computer and System Sciences*, 63(2):236–252, 2001.

[27] V. Kabanets. Derandomization: A brief overview. *Bulletin of the EATCS*, 76:88–103, 2002.

[28] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1-3):40–56, 1982.

[29] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002.

[30] J. Köbler and O. Watanabe. New collapse consequences of NP having small circuits. *SIAM Journal on Computing*, 28(1):311–324, 1998.

[31] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9:265–266, 1973.

[32] C.-J. Lu. Derandomizing Arthur-Merlin games under uniform assumptions. *Computational Complexity*, 10(3):247–259, 2001.

[33] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44(2):220–258, 1992.

[34] J. H. Lutz. The quantitative structure of exponential time. In L. A. Hemaspaandra and A. L. Selman, editors, *Complexity Theory Retrospective II*, pages 225–254. Springer-Verlag, 1997.

[35] J. H. Lutz and E. Mayordomo. Twelve problems in resource-bounded measure. *Bulletin of the European Association for Theoretical Computer Science*, 68:64–80, 1999. Also in *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 83–101, World Scientific Publishing, 2001.

[36] E. Mayordomo. Almost every set in exponential time is P-bi-immune. *Theoretical Computer Science*, 136(2):487–506, 1994.

[37] P. Miltersen and N.V. Vinodchandran. Derandomizing Arthur-Merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005.

[38] P. B. Miltersen. Derandomizing complexity classes. In *Handbook of Randomized Computing, volume II*, pages 843–934. Kluwer, 2001.

[39] A. Pavan. Comparison of reductions and completeness notions. *SIGACT News*, 34(2):27–41, June 2003.

[40] A. Pavan and A. L. Selman. Separation of NP-completeness notions. *SIAM Journal on Computing*, 31(3):906–918, 2002.

[41] O. Watanabe and S. Tang. On polynomial time Turing and many-one completeness in PSPACE. *Theoretical Computer Science*, 97(2):199–215, 1992.