NP-Hard Sets are Exponentially Dense Unless coNP ⊆ NP/poly

Harry Buhrman*        John M. Hitchcock†

Abstract

We show that hard sets S for NP must have exponential density, i.e., \(|S_n| \geq 2^{n^\epsilon}\) for some \(\epsilon > 0\) and infinitely many \(n\), unless coNP ⊆ NP/poly and the polynomial-time hierarchy collapses. This result holds for Turing reductions that make \(n^{1-\epsilon}\) queries.

In addition we study the instance complexity of NP-hard problems and show that hard sets also have an exponential amount of instances that have instance complexity \(n^\delta\) for some \(\delta > 0\). This result also holds for Turing reductions that make \(n^{1-\epsilon}\) queries.

1 Introduction

The density of NP-complete and hard sets was an early object of study in complexity theory. Assuming that P is not equal to NP, the real question is how many instances are indeed hard? In principle it could be that P ≠ NP only because of a few instances that are hard to compute, but almost all instances can be decided by an efficient algorithm. This question was formalized and investigated in a large body of work starting with that of Berman and Hartmanis [2], Meyer and Paterson [10], Fortune [5], Karp and Lipton [8], Mahaney [9], and many others.

It is problematic for this question to just focus on a fixed NPC-complete set for the following reason. Suppose that P ≠ NP, and suppose there is a machine \(M\) that runs in polynomial time on all but \(2^n\) many formulae of length \(n\). We can then solve SAT in randomized polynomial time, by simple padding. Given any formula \(\phi\) we can construct \(2^n\) many different other formulae \(\phi'\) of roughly the same length that are satisfiable if and only if \(\phi\) is satisfiable. It is easy to see that \(M\) will with high probability run in polynomial time on a randomly chosen \(\phi'\). For this reason the focus has been on the density of all NP-complete or NP-hard problems. This simple padding trick cannot work for an arbitrary NP-complete problem, since the reduction can map the equivalent formula \(\phi'_i\) back to the original \(\phi\). Therefore attention has been on the density of NP-complete and NP-hard sets under various types of reductions.

Mahaney [9] showed that if there exists a sparse many-one hard set for NP then P = NP. A set is sparse if for every length \(n\) it contains no more than \(p(n)\) strings for some polynomial \(p\). This result shows that many-one hard sets for NP are super-polynomially dense unless P = NP. Mahaney’s result has been extended to weaker notions of reductions, notably by Ogihara and Watanabe for bounded truth-table reductions [11]. But it remains an open question to show the same result for log(n)-truth-table reductions, let alone for the more general Turing reductions. Karp and Lipton [8] showed that if there exists a sparse Turing hard set for NP, or equivalently if NP ⊆ P/poly, then the polynomial-time hierarchy collapses to its second level (Σ_2^P = Π_2^P). Hence Turing hard sets for NP are also super-polynomially dense unless the polynomial-time hierarchy collapses.

In this paper we improve these results from sparse to subexponential density. Generalizations to sets with more than polynomial density had been studied before by Buhrman and Homer [3]. A set S has subexponential density if for every \(\epsilon > 0\), \(|S_m| \leq 2^{n^\epsilon}\) for almost all \(n\). We show that if there exists an NP-hard set with subexponential density then coNP ⊆ NP/poly and by a result of Yap [13] it follows that the polynomial-time hierarchy collapses to its third level (Σ_3^P = Π_3^P). Our result holds for Turing reductions that make \(n^{1-\epsilon}\) queries (any \(\epsilon > 0\)). This shows that NP-hard sets have exponential density \(2^{n^\epsilon}\) for some \(\epsilon > 0\), unless coNP ⊆ NP/poly. This is the best possible result for NP-hard sets with respect to their density, since simple padding shows that for every \(\epsilon > 0\) there exists an NP-hard set with density less than \(2^{n^\epsilon}\). Our results make use of the proof of a recent combinatorial lemma due to Fortnow and Santhanam [4].
Another way to make the notion of hard instances precise is that of \textit{instance complexity} due to Orponen et. al. \cite{Orponen}. The instance complexity of an instance \(x\) with respect to some set \(A\), \(ic(x : A)\), is the size of the smallest (polynomial-time) program \(p\) that correctly decides \(x\) and for all other instances either outputs no decision or the correct decision. It is easy to see that \(ic(x : A) \leq |x| + O(1)\). Strings with high instance complexity do not have small efficient programs that decide them. The instance complexity of NP-complete sets has been studied. The best known bound \cite{Orponen} is that if every instance of SAT (or any NP-complete problem) has logarithmic instance complexity, i.e. \(ic(\phi : SAT) \leq O(\log |\phi|)\) for all \(\phi\), then \(P = NP\). We show that if SAT has sublinear instance complexity, that is \(ic(\phi : SAT) \leq |\phi|^{1-c}\) for all \(\phi\) and some \(c > 0\), then \(coNP \subseteq NP/poly\).

\section{Preliminaries}

We shall consider decision problems for languages over the alphabet \(\Sigma = \{0, 1\}\). The length of a string \(x \in \{0, 1\}^*\) is denoted \(|x|\); \(\lambda\) denotes the empty string. Given strings \(x, y\), we denote with \(x \cdot y\) the concatenation of \(x\) and \(y\): \(xy\). We represent the pair \(<x, y>\) as the string \(\overline{x}10y\), where \(\overline{x}\) denotes \(x\) with each of its characters doubled.

For a set \(B\) and number \(n\), \(B_{\leq n} = \{x \in B \mid |x| \leq n\}\). The cardinality of a finite set \(C\) is denoted \(|C|\).

A set \(S\) has subexponential density if for every \(\varepsilon > 0\), \(|S_{\leq n}| \leq 2^{n^\varepsilon}\) for all but finitely many \(n\). We write \(\text{SUBEXP\ D}\) for the class of languages with subexponential density. A set is exponentially dense if it does not have subexponential density.

An AND-function for a set \(A\) is a polynomial-time computable function \(g\) such that for all strings \(x_1, x_2, \ldots, x_n, g(x_1, x_2, \ldots, x_n) \in A\) if \(x_i \in A\) for all \(i\). Similarly, and OR-function \(g\) satisfies \(g(x_1, x_2, \ldots, x_n) \in A\) if \(x_i \in A\) for some \(i\). We say that \(g\) has order \(s\) if \(|g(x_1, \ldots, x_n)| = O((\sum_{i=1}^{n} |x_i|)^s)\).

Observe that if \(g\) is an AND-function for \(A\), then \(g\) is also an OR-function for \(A\).

\section{Reductions}

To introduce the technique we will begin with the easier case of many-one reductions. This result has the corollary that if SAT many-one reduces to a set of subexponential density, then \(coNP \subseteq NP/poly\).

\begin{theorem}
Let \(A\) be any set that has an AND-function. If there is a set \(S\) with subexponential density such that \(A \leq^p S\) then \(A \in NP/poly\).
\end{theorem}

\begin{proof}
Let \(g(x_1, \ldots, x_n)\) be the AND-function for \(A\). Let \(f\) be the many-one reduction from \(A\) to \(S\). We say that a string \(z \in S\) is \(NP\)-proof for \(x \in A\), with \(|x| = n\), iff there exist \(x_1, \ldots, x_n\), such that for all \(i\), \(|x_i| = n\) and there exists an \(i\), with \(z = x_i\), and in addition \(f(g(x_1, \ldots, x_n)) = z\).

The idea is to show that there exists a string \(z_i \in S\) that is \(NP\)-proof for half the strings in \(A_{\leq n}\). We will then recurse on the remaining strings in \(A_{\leq n}\), for which \(z_i\) is not \(NP\)-proof, until we end up with a sequence of at most \(n\) strings \(z_1, \ldots, z_k\) such that for all \(x \in A_{\leq n}\), there is an \(i\) such that \(z_i\) is \(NP\)-proof for \(x\). These \(NP\)-proofs serve as advice to show that \(A \in NP/poly\).

First observe that if \(z\) is \(NP\)-proof for precisely \(t\) strings \(x \in A\) then

\begin{equation}
\left| \left\{ x_1, \ldots, x_n \mid |x_i| = n \quad \text{and} \quad f(g(x_1, \ldots, x_n)) = z \right\} \right| \leq t^n
\end{equation}

Assume that \(f\) and \(g\) both run in time \(n^c\) for some \(c\). Let \(m_n = n^{2c^2}\), hence \(|f(g(x_1, \ldots, x_n))| \leq m_n\). Since \(S\) has subexponential density, for large enough \(n\) it holds that \(|S_{\leq m_n}| < 2^n\).

Let \(t\) be the largest such that some \(z_i\) is \(NP\)-proof for \(t\) elements of length \(n\) in \(A\). Since for every \(n\)-tuple \(<x_1, \ldots, x_n>\) with for all \(i\), \(x_i \in A\), \(f(g(<x_1, \ldots, x_n>))\) maps to some string \(z\) in \(S_{\leq m_n}\), we now have:

\begin{equation}
t^n |S_{\leq m_n}| \geq |A_{\leq n}|^n
\end{equation}

and hence

\begin{equation}
2n^n \geq |A_{\leq n}|^n
\end{equation}

which implies that \(t \geq |A_{\leq n}|/2\), and hence \(z_i\) is \(NP\)-proof for half the elements in \(A\) of length \(n\). The proof now continues by finding a \(z_2\) that is \(NP\)-proof for half of the elements in \(A\) for which \(z_1\) is not \(NP\)-proof, resulting ultimately in the desired sequence \(z_1, \ldots, z_k\) \((k \leq n)\). The inductive generation of \(z_i\) is possible because all the strings in \(A\) for which none of the \(z_1, \ldots, z_{i-1}\) is \(NP\)-proof, let’s call them \(A’\), have the following property. For every \(y_1, \ldots, y_n \in A’\) it holds that \(f(g(y_1, \ldots, y_n)) \in S_{\leq m_n}\) \(\{z_1, \ldots, z_{i-1}\}\). Hence the counting arguments in equations (3.1), (3.2), and (3.3) still hold for \(A’\).
\end{proof}

Our main technical tool, Lemma 3.2 below, is a generalization of Theorem 3.1. Instead of a many-one reduction to a subexponentially dense set, we consider a nondeterministic disjunctive reduction to a family of sets where the density can be exponential.
Definition. Let \( B = (B_n \mid n \geq 0) \) be a family of subsets of \( \{0,1\}^* \). We say that \( A \) NP-reduces to \( B \) if there is an NPMV function \( N \) such that for all \( n \), for all \( x \in \{0,1\}^n \), \( x \in A \) iff at least one output of \( N(x) \) is in \( B_n \).

**Lemma 3.2.** Let \( A \) have an AND-function of order \( s \) and let \( \alpha < 1/s \). Let \( B = (B_n \mid n \geq 0) \) be a family of sets \( |B_n| \leq 2^{n^s} \) for sufficiently large \( n \). If \( A \) NP-reduces to \( B \), then \( A \in \text{NP/poly} \).

**Proof.** Let \( M \) compute the NPMV function for the reduction from \( A \) to \( B \). Let \( g \) be the AND-function for \( A \). For simplicity we assume that for all \( x_1, \ldots, x_n \in \{0,1\}^n \), the length of \( g(x_1, \ldots, x_n) \) is exactly \( (nm)^s \). The general case when the length is \( O((nm)^s) \) is similar.

Choose a constant \( k \) so that \( \frac{k}{k+1} \geq \alpha \). Fix an input length \( m \), let \( n = m^k \), and let \( N = (nm)^s \). Note that we have
\[
|B_n| \leq 2^{n^s} = 2^{m^k(1+1/m)} \leq 2^{m^k} = 2^n.
\]
For any \( x \in \{0,1\}^m \),
\[
x \in A \iff \text{there exist } x_1, \ldots, x_n \in \{0,1\}^m \text{ with } x_i = x \text{ for some } i \text{ such that } M \text{ on input } g(x_1, \ldots, x_n) \text{ outputs some string } z \in B_N.
\]

Call such a string \( z \) an NP-proof that \( x \in A \). As in the proof of Theorem 3.1, we claim that there exists a collection of at most \( m \) strings \( z_1, \ldots, z_t \) such that each \( x \in A_{nm} \) has an NP-proof in the collection.

Suppose that \( z_1 \) is an NP-proof for exactly \( t \) strings in \( A_{nm} \). Then
\[
|\{x \mid g(x_1, \ldots, x_n) \text{ outputs } z_1\}| \leq t^m.
\]
Let \( t \) be the maximal such that some string \( z \) is an NP-proof for \( t \) strings. Then
\[
|A_{nm}|^s \leq |B_n| \cdot t^m \leq 2^{n^s}t^m,
\]
so \( t \geq |A_{nm}|/2 \). Therefore there is a string \( z_1 \) that works for at least half of the strings in \( A_{nm} \). Repeating this argument yields a string \( z_2 \) that works for at least half of the remaining strings. After at most \( m \) repetitions we have NP-proofs for all the strings. \( \square \)

As our first application of Lemma 3.2 we extend Theorem 3.1 to disjunctive reductions.

**Theorem 3.3.** If \( A \) has an AND-function and \( A \preccurlyeq \text{p} \) \( \text{SUBEXP} \), then \( A \in \text{NP/poly} \).

**Proof.** Suppose that \( A \preccurlyeq \text{p} \) \( S \in \text{SUBEXP} \) via a reduction \( g \) in \( p(n) \) time. Define an NPMV function \( N \) that on input \( x \) guesses and outputs one of the queries in \( g(x) \). Let \( B_n = S_{\leq p(n)} \). Then \( A \) NP-reduces to the family \( (B_n \mid n \geq 0) \) via \( N \).

Let \( \alpha < 1/s \) where \( s \) is the order of the AND-function. We have \( |B_n| \leq 2^{n^s} \) for sufficiently large \( n \) because \( S \) has subexponential density. By Lemma 3.2 we have \( A \in \text{NP/poly} \). \( \square \)

We apply Theorem 3.3 with \( \text{SAT} \) to obtain the following:

**Theorem 3.4.** If \( \text{coNP} \nsubseteq \text{NP/poly} \), then every \( \preccurlyeq \text{p} \) hard set for \( \text{coNP} \) is exponentially dense.

Allender, Hemachandra, Ogihara, and Watanabe [1] showed that if \( A \preccurlyeq \text{p} \)-reduces to a sparse set, then \( A \preccurlyeq \text{p} \)-reduces to another sparse set. Part of the proof shows that the complement of any sparse set disjunctively reduces to a sparse set. This argument also applies to subexponentially dense sets. For completeness we include a proof. Here we write that \( S \) has density \( d(n) \) if \( |S_n| = d(n) \).

**Lemma 3.5.** Let \( S \) be a set with density \( d(n) \). Then there is a set \( T \) with density at most \( nd(n) \) such that \( S \preccurlyeq \text{p} \) \( T \). In particular, if \( S \in \text{SUBEXP} \), then \( S \preccurlyeq \text{p} \) \( T \) for some \( T \in \text{SUBEXP} \).

**Proof.** We isolate the part we need of the proof in [1]. Let \( T \) be the set of all \( 0^n1wb \) where \( b \) is a bit and \( w \) has an extension in \( S_m \), but \( wb \) does not have an extension in \( S_m \). If \( S_m = \emptyset \), we add \( 0^n1 \) to \( T \).

We claim that a string \( y \) is in \( S_m \) if and only if \( y \) has a prefix \( z \) such that \( 0^n1z \in T \).

- If \( y \not\in S \) and \( S_m \neq \emptyset \), then let \( z \) be the longest prefix of \( y \) that has an extension in \( S \). The string \( 0^n1z \) is in \( T \). If \( S_m = \emptyset \), then \( 0^n1 \) is in \( T \), so the claim holds for \( z = \lambda \).

- If \( y \in S \), then every prefix \( z \) of \( y \) has an extension in \( S \) and \( 0^n1z \not\in T \).

Therefore \( S \preccurlyeq \text{p} \) \( T \) via the reduction that lists the prefixes of its input.

For each length \( n \), we added at most \( (n+1)S_m + 1 \) strings to \( T \). Therefore \( |T_{\leq n}| \leq \sum_{m=0}^{n-1}(m+1)|S_m| + 1 \leq nd(n) + n \). \( \square \)

Theorem 3.3 and Lemma 3.5 yield the following for conjunctive reductions.

**Theorem 3.6.** If \( A \) has an OR-function and \( A \preccurlyeq \text{p} \) \( \text{SUBEXP} \), then \( A \in \text{coNP/poly} \).
Proof. Suppose that \( A \subseteq^p \bar{S} \in \text{SUBEXP}. \) Then \( A \subseteq^p \bar{S} \) and by Lemma 3.5 there is a \( T \in \text{SUBEXP} \) such that \( \bar{S} \subseteq^p T. \) Composing reductions yields \( A \subseteq^p T, \) so \( A \in \text{NP/poly} \) by Theorem 3.3, because the OR-function for \( A \) is an AND-function for \( \bar{A}. \) \( \square \)

**Theorem 3.7.** If \( \text{coNP} \not\subseteq \text{NP/poly}, \) then every \( \leq^p \)-hard set for \( \text{NP} \) is exponentially dense.

Our next theorem concerns query-bounded Turing reductions. In the proof we use techniques from [1, 6] to convert the Turing reduction into an NP disjunctive reduction.

**Theorem 3.8.** Let \( A \) have an AND-function of order \( s \) and let \( \alpha < 1/s. \) If \( A \leq_{n^{-\alpha}}^p \text{SUBEXP}, \) then \( A \in \text{NP/poly}. \)

**Proof.** Suppose \( A \leq_{n^{-\alpha}}^p S \in \text{SUBEXP} \) via \( M. \) Fix an input length \( n. \) For a given input \( x \in \{0,1\}^n, \) consider using each \( z \in \{0,1\}^n \) as the sequence of yes/no answers to \( M \)'s queries. Each \( z \) causes \( M \) to produce a sequence of queries \( w_0^{z,x}, \ldots, w_s^{z,x} \) and an accepting or rejecting decision. (We can assume that \( M \) always makes \( n^\alpha \) queries.) Let \( Z_x \subseteq \{0,1\}^{n^\alpha} \) be the set of all query answer sequences that cause \( M \) to accept \( x. \) Then we have \( x \in A \) if and only if

\[
(\exists z \in Z_x)(\forall 1 \leq j \leq n^\alpha) \ S[w_j^{x,z}] = z[j],
\]

which is equivalent to

\[
(\exists z \in Z_x)(\forall 1 \leq j \leq n^\alpha) \ z[j] \cdot w_j^{x,z} \in \bar{S} \oplus S,
\]

where \( \bar{S} \oplus S \) is the disjoint union \( \{0z \mid z \in S\} \cup \{1z \mid z \in \bar{S}\}. \)

By Lemma 3.5 there is a set \( T \in \text{SUBEXP} \) such that \( \bar{S} \subseteq^p T. \) Let \( U = T \oplus S. \) We then have \( \bar{S} \oplus S \leq_{\alpha}^p U \) via some reduction \( g. \) For each \( x \in Z_x, \) let

\[
H_{x,z} = \{u_1, \ldots, u_{n^\alpha}\} \text{ with } (\forall y) u_j \in g(z[j] \cdot w_j^{x,z}).
\]

Let \( r(n) \) be a polynomial bounding the run time of \( g \) on inputs of the form \( z[j] \cdot w_j^{x,z}, \) where \( |x| = n. \) Define

\[
B_n = \{u_1, \ldots, u_{n^\alpha}\} \text{ with } (\forall y) u_j \in U_{r(n)}. \]

Then we have

\[
x \in A \iff (\exists z \in Z_x)(\exists y \in H_{x,z}) y \in B_n.
\]

Define an NPMV function \( N \) that on input \( x \) chooses some \( z \in Z_x \) and takes \( y \in H_{x,z} \) and outputs \( y. \) Then \( N \) is an NP-reduction of \( A \) to the family \( (B_n \mid n \geq 0). \)

Let \( \delta = (1/s - \alpha)/2. \) Then since \( U \in \text{SUBEXP}, \)

\[
|U_{r(n)}| \leq 2^{n\delta} \quad \text{for sufficiently large } n.
\]

This implies

\[
|B_n| = |U_{r(n)}|^{n^\alpha} \leq 2^{n^{(1-s)\delta}} = 2^{n^{(1/s) - \alpha}}.
\]

Lemma 3.2 applies to show \( A \in \text{NP/poly}. \) \( \square \)

We now have the main result of this paper:

**Theorem 3.9.** If \( \text{coNP} \not\subseteq \text{NP/poly}, \) then for all \( \epsilon > 0, \) every \( \leq_{n^{-\epsilon}}^p \)-hard set for \( \text{NP} \) is exponentially dense.

**Proof.** Suppose that \( \text{SAT} \leq_{n^{-\epsilon}}^p \)-reduces to a subexponentially dense set. Then \( \text{SAT} \leq_{n^{-\epsilon}}^p \)-reduces to the same set by inverting the reduction’s answers. Moreover \( \text{SAT} \) has an AND-function of order \( s = 1. \) Theorem 3.8 applies to show \( \text{coNP} \not\subseteq \text{NP/poly}. \) \( \square \)

In fact, we can show a slightly stronger result. Theorem 3.8 still holds if the Turing reduction uses nondeterminism:

**Theorem 3.10.** Let \( A \) have an AND-function of order \( s \) and let \( \alpha < 1/s. \) If \( A \in \text{NP}^{s[n^{-\alpha}]} \) for some \( S \in \text{SUBEXP}, \) then \( A \in \text{NP/poly}. \)

**Proof.** We extend the proof of Theorem 3.8. Suppose \( A = L(M_S^{s[n^{-\alpha}]}) \) where \( M \) is an NP machine running in time \( t(n). \) For an input \( x \in \{0,1\}^n, \) we can use any pair \( <p,z> \) where \( p \in \{0,1\}^{|t(n)|} \) and \( z \in \{0,1\}^{n^\alpha} \) to run \( M \) on input \( x. \) We use \( p \) to provide the nondeterministic choices and \( z \) to provide the query answers. In this computation \( M \) produces a sequence of queries \( w_0^{p,z}, \ldots, w_s^{p,z} \) and an accepting or rejecting decision. Let \( Z_x \) be the set of all \( <p,z> \) that cause \( M \) to accept \( x. \) Then we have \( x \in A \) if and only if

\[
(\exists <p,z> \in Z_x)(\forall 1 \leq j \leq n^\alpha) \ S[w_j^{p,z}] = z[j].
\]

The remainder of the proof carries through with \( z \) replaced by \( <p,z> \) throughout. \( \square \)

We obtain an extension of Theorem 3.10 to strong nondeterministic polynomial-time reductions.

**Theorem 3.11.** If \( \text{coNP} \not\subseteq \text{NP/poly}, \) then for all \( \epsilon > 0, \) every \( \leq_{n^{-\epsilon}}^p \)-hard set for \( \text{NP} \) is exponentially dense.

**Proof.** Suppose that \( S \) has subexponential density and is \( \leq_{n^{-\epsilon}}^p \)-hard for \( \text{NP}. \) Then \( \text{SAT} \leq_{n^{-\epsilon}}^p \ S, \) so \( \text{SAT} \in \text{NP}^{s[n^{-\epsilon}]} \). Theorem 3.10 implies \( \text{SAT} \in \text{NP/poly}. \) \( \square \)

All our results to this point are conditional. For an unconditional result we go to the PH hierarchy, where \( \overline{P} \) means \( n^{O(\log n)}. \)

**Theorem 3.12.** For all \( \epsilon > 0, \) every \( \leq_{n^{-\epsilon}}^p \)-hard set for \( \Sigma^P_3 \) is exponentially dense.
Proof. First, we claim that $\Sigma^0_0 \not\subset \text{NP/poly}$. This is similar to Kannan’s proof that $\Sigma^0_2$ does not have $n^k$-size circuits [7]. We can show that there is a set $H \in \Sigma^0_1$ of $\text{NP/poly}$ by a direct counting argument.

Then we consider two cases: if $\text{coNP} \not\subset \text{NP/poly}$, the claim holds immediately because $\text{coNP} \subseteq \Sigma^0_0$. Otherwise, $\text{coNP} \subseteq \text{NP/poly}$ and we have $\text{PH} = \Sigma^0_2$ by Yap’s theorem [13]. From this padding gives $\text{PH} = \Sigma^0_2$ and therefore $H \in \Sigma^0_2$.

There is a many-one complete set $A$ for $\Sigma^0_0$ with an AND-function of order 1. Suppose that $A \triangleleft_{\text{m-or-}1} \text{coNP}$ reduces to a set $S$ of subexponential density. Theorem 3.8 implies $A \in \text{NP/poly}$, so $\Sigma^0_0 \not\subset \text{NP/poly}$, a contradiction. □

We remark that Theorem 3.12 also holds for conjunctive, disjunctive, and SNP $n^{1-\epsilon}$-Turing reductions.

4 Instance Complexity

Let $A$ be a set and let $t(n)$ be a time bound. A program $p$ is consistent with $A$ for all $x$, $p(x) \in \{0, 1, ?\}$, and whenever $p(x) \neq ?$, $p(x) = A(x)$. The $t$-instance complexity of $x$ with respect to $A$, written $\text{ic}^A_t(x : A)$, is the length of a shortest program $p$ such that

- $p$ is consistent with $A$,
- $p(x)$ halts within $t(|x|)$ steps, and
- $p(x) = A(x)$.

Formally, $p(x) = U(p, x)$ where $U$ is an efficient universal machine. See [12] for more information on instance complexity.

Theorem 4.1. Let $A$ have an AND-function of order $s$, let $\alpha < 1/s$, and let $q$ be a polynomial. If $\text{ic}^A_t(x : A) \leq n^\alpha$ for all but finitely many $x \in A$, then $A \in \text{NP/poly}$.

Proof. For each $n$, let

$$B_n = \{p \mid p \text{ is consistent with } A \text{ and } |p| \leq n^\alpha\}.$$ 

Then $|B_n| \leq 2^{n^{\alpha+1}}$. Define an NPMV function $N$ that on input $x$ guesses a program $p$ and outputs $p$ if the program accepts $x$ within $q(|x|)$ steps. Then $N$ reduces $A$ to the family $(B_n \mid n \geq 0)$, so Lemma 3.2 yields $A \in \text{NP/poly}$. □

Corollary 4.2. If $\text{NP} \not\subset \text{coNP/poly}$, then for every polynomial $q$ and $\epsilon > 0$, there exist infinitely many $\phi \in \text{SAT}$ with $\text{ic}^A_t(\phi : \text{SAT}) > |\phi|^{1-\epsilon}$. □

Corollary 4.3 should be contrasted with the result that if $P \neq \text{NP}$, then there are infinitely many $\phi$ with $\text{ic}^A_t(\phi : \text{SAT}) \geq c \log |\phi|$. With the stronger $\text{NP} \not\subset \text{coNP/poly}$ hypothesis, we get a nearly linear lower bound on the instance complexity of SAT instances. Since $\text{ic}^t(\phi : \text{SAT}) \leq |\phi| + O(1)$ for $t(n) = O(n \log n)$, this bound is fairly tight.

We can also show that the lower bound holds for a large set of SAT instances. Our next theorem is an extension of Theorem 4.1 that accounts for the density of the hard instances.

Theorem 4.3. Let $A$ have an AND-function of order $s$, let $\alpha < 1/s$, and let $q$ be a polynomial. Define $H = \{x \in A \mid \text{ic}^A_t(x : A) > |x|^\alpha\}$. If $|H_{\leq n}| \leq 2^n$ for sufficiently large $n$, then $A \in \text{NP/poly}$.

Proof. Let $P_n = \{p \mid p$ is consistent with $A$ and $|p| \leq n^\alpha\}$. We define $B_n$ as the disjoint union of $H_{\leq n}$ and $P_n$:

$$B_n = H_{\leq n} \cup P_n.$$

Then $|B_n| \leq 2^{n+2}$ for large $n$. Define an NPMV function $N$ that on input $x$ either

(i) outputs $0x$, or
(ii) guesses a program $p$ and outputs $1p$ if $p$ accepts $x$ within $q(|x|)$ steps.

Then $N$ reduces $A$ to the family $(B_n \mid n \geq 0)$ and Lemma 3.2 implies $A \in \text{NP/poly}$. □

Corollary 4.4. Suppose $\text{NP} \not\subset \text{coNP/poly}$. Then for all $\epsilon > 0$ and polynomials $q$,

$$\left\| \{\phi \in \text{SAT}_{\leq n} \mid \text{ic}^A_t(\phi : \text{SAT}) > |\phi|^{1-\epsilon}\} \right\| \geq 2^{n^{1-\epsilon}}$$

for infinitely many $n$.

Next we consider reductions to sets that have low instance complexity.

Theorem 4.5. Let $A$ have an AND-function of order $s$ and let $\alpha < 1/s$. Let $C$ be a set where for all $\delta > 0$, there is a polynomial $r$ such that $\text{ic}^A_t(x : C) < |x|^\delta$ for all but finitely many $x$. If $A \triangleleft_{\text{m-or-}1} C$, then $A \in \text{NP/poly}$.

Proof. Let $M$ compute the reduction from $A$ to $C$ in $t(n)$ time. Let $\epsilon = \left[\frac{1}{s} - \alpha\right]/2$. Choose $\delta > 0$ so that $t(n)^\delta < n^\epsilon$ for sufficiently large $n$. There is a polynomial $r$ such that $\text{ic}^A_t(x : C) < |x|^\delta$ for almost all $x$.

Let $x$ have length $n$. We can assume that $M$ makes exactly $n^\epsilon$ queries on input $x$. Define an NP machine $N$ that on input $x$ simulates $M$. When $M$ makes a query $q_i$, $N$ does the following:
(i) Guess a program \( p_i \) with \( |p_i| < |q_i|^\delta \).

(ii) Run \( p_i \) on input \( q_i \), aborting the computation if it runs for more than \( r(|q_i|) \) steps.

(iii) If \( p_i \) produces a decision, use that as the answer for query \( q_i \) in the simulation of \( M \).

(iv) If \( p_i \) was aborted or did not output a decision, \( N \) halts and outputs nothing.

If \( M \) accepts \( x \) at the end of this simulation, then \( N \) outputs the tuple \(< p_1, \ldots, p_n >\) of programs it guessed.

Each query \( q_i \) has \( |q_i| \leq t(n) \). Then for sufficiently large \( n \),

\[ ic^\delta(q_i : C) < |q_i|^\delta \leq t(n)^\delta < n^\delta. \]

Define

\[ E_n = \{ p : p \text{ is consistent with } C \text{ and } |p| < n^\delta \}. \]

and

\[ B_n = \{ < p_1, \ldots, p_n > : \text{each } p_i \in E_n \}. \]

Then \( |B_n| \leq (2^{n^\delta})^n < 2^n(1/s)^n \) and \( N \) reduces \( A \) to the family \( (B_n : n \geq 0) \). Lemma 3.2 now applies to show \( A \in \text{NP(poly)} \). \( \square \)

We can also extend Theorem 4.5 to consider the density of the hard instances.

**Theorem 4.6.** Let \( A \) have an AND-function of order \( s \) and let \( \epsilon < 1/s \). Let \( C \) be a set where for all \( \delta > 0 \), there is a polynomial \( r \) such that the collection of hard instances

\[ H^{\delta,r} = \{ x : ic^\delta(x : C) \geq n^\delta \} \]

has subexponential density. If \( A \leq^p_{n^\delta,T} C \), then \( A \in \text{NP(poly)} \).

**Proof.** Let \( M \) compute the reduction from \( A \) to \( C \) in \( t(n) \) steps. We assume that \( M \) makes exactly \( n^\delta \) queries. Let \( \epsilon = [(1/s) - \alpha]/2 \) and choose \( \delta > 0 \) such that \( t(n)^\delta < n^\delta \) for large \( n \). There is a polynomial \( r \) such that \( H^{\delta,r} \) has subexponential density.

Let \( x \) have length \( n \). Define an NP machine \( N \) that on input \( x \) simulates \( M \). When \( M \) makes a query \( q_i \), \( N \) nondeterministically chooses (I) or (II) below to answer the query:

(I) Guess a bit \( b \) and use it as the answer for query \( q_i \). Record \( z_i = < b, q_i > \).

(II) (i) Guess a program \( p_i \) with \( |p_i| < |q_i|^\delta \).

(ii) Run \( p_i \) on input \( q_i \), aborting the computation if it runs for more than \( r(|q_i|) \) steps.

(iii) If \( p_i \) was aborted or did not output a decision, \( N \) halts and outputs nothing.

(iv) If \( p_i \) produces a decision, use that as the answer for query \( q_i \). Record \( z_i = < \lambda, p_i > \).

If \( M \) accepts \( x \) at the end of the simulation, then \( N \) outputs the tuple \(< z_1, \ldots, z_n >\).

We have \( |H^{\delta,r}_{ \leq t(n)}| < 2^n \) for sufficiently large \( n \). Define

\[ E_n = \{ < \lambda, p > : p \text{ is consistent with } C \text{ and } |p| < n^\delta \}, \]

\[ D_n = \{ < 1, q > : q \in H^{\delta,r}_{ \leq t(n)} \cap C \} \]

\[ \cup \{ < 0, q > : q \in H^{\delta,r}_{ \leq t(n)} \cap \overline{C} \}, \]

and

\[ B_n = \{ < z_1, \ldots, z_n > : \text{each } z_i \in D_n \cup E_n \}. \]

Then

\[ |B_n| = (|E_n| + |H^{\delta,r}_{ \leq t(n)}|)^n < (2^n + 1)^n \approx 2^{n^{1/s}}. \]

We apply Lemma 3.2 to obtain \( A \in \text{NP(poly)} \). \( \square \)

**Corollary 4.7.** Suppose that \( \text{NP} \subseteq \text{coNP/poly} \) and let \( C \) be \( \leq^p_{n^\delta,T} \text{-hard for } \text{NP} \). There is a \( \delta > 0 \) such that for every polynomial \( r \), the set

\[ \{ x : ic^\delta(x : C) \geq |x|^\delta \} \]

has exponential density.

Just like Theorem 3.11 we can show that Corollary 4.7 also holds for strong nondeterministic polynomial-time reductions. Also, by following the line of argument in Theorem 3.12, we can obtain an absolute result for instance complexity in \( \Sigma^p_3 \)-hard sets.

**Acknowledgements.** We thank Lance Fortnow and Rahul Santhanam for sharing a preliminary version of [4], and for useful discussions. We also thank Scott Aaronson, Steve Fenner, Kolya Vereshchagin, and John Rogers for interesting discussions.

**References**


