23rd Annual IEEE Conference on Computational Complexity

NP-Hard Sets are Exponentially Dense Unless coNP $\subseteq$ 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 $\subseteq$ 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^δ *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 $\neq$ 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 NP-complete set for the following reason. Suppose that P $\neq$ NP, and suppose there is a machine M that runs in polynomial time on all but 2^{n^ϵ} many formulae of length n. We can then solve SAT in randomized polynomial time, by simple padding. Given any formula ϕ we can construct 2^n many different other formulae ϕ'_i of roughly the same length that are satisfiable if and only if ϕ is satisfiable. It is easy to see that M will with high probability run in polynomial time on a randomly chosen ϕ'_i. 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 ϕ'_i back to the original ϕ. 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 $\subseteq$ P/poly, then the polynomial-time hierarchy collapses to its second level ($\Sigma_2^p = \Pi_2^p$). 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 $\epsilon > 0$, $\|S_{=n}\| \leq 2^{n^\epsilon}$ for almost all n. We show that if there exists an NP-hard set with *subexponential* density then coNP $\subseteq$ NP/poly and by a result of Yap [13] it follows that the polynomial-time hierarchy collapses to its third level ($\Sigma_3^p = \Pi_3^p$). 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^ϵ} for some $\epsilon > 0$, unless coNP $\subseteq$ 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^ϵ}. Our results make use of the proof of a recent combinatorial lemma due to Fortnow and Santhanam [4].

*CWI Kruislaan 409, 1098 SJ Amsterdam, the Netherlands, and University of Amsterdam; e-mail:buhrman@cwi.nl. Supported by an NWO VICI grant, BRRICKS, and EU-grant QAP.

†Department of Computer Science, University of Wyoming; e-mail: jhitchco@cs.uwyo.edu. Supported in part by NSF grants 0515313 and 0652601.

1093-0159/08 $25.00 © 2008 IEEE
DOI 10.1109/CCC.2008.21

Another way to make the notion of hard instances precise is that of *instance complexity* due to Orponen et. al. [12]. The instance complexity of an instance x with respect to some set A, $\mathrm{ic}(x : A)$, 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 $\mathrm{ic}(x : A) \leq |x| + O(1)$. 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 [12] is that if every instance of SAT (or any NP-complete problem) has logarithmic instance complexity, i.e. $\mathrm{ic}(\phi : \mathrm{SAT}) \leq O(\log|\phi|)$ for all ϕ, then P = NP. We show that if SAT has sublinear instance complexity, that is $\mathrm{ic}(\phi : \mathrm{SAT}) \leq |\phi|^{1-\epsilon}$ for all ϕ and some $\epsilon > 0$, then coNP $\subseteq$ NP/poly.

2 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 $|x|$; λ 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 $\bar{x}10y$, where $\bar{x}$ denotes x with each of its characters doubled.

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

A set S has subexponential density if for every $\epsilon > 0$, $\|S_{=n}\| \leq 2^{n^\epsilon}$ for all but finitely many n. We write SUBEXPD 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 $x_1, x_2, \ldots, x_n$, $g(x_1, x_2, \ldots, x_n) \in A$ iff $x_i \in A$ for all i. Similarly, and OR-function g satisfies $g(x_1, x_2, \ldots, x_n) \in A$ iff $x_i \in A$ for some i. We say that g has order s if $|g(x_1, \ldots, x_n)| = O\big((\sum_{i=1}^n |x_i|)^s\big)$. Observe that if g is an AND-function for A, then g is also an OR-function for $\bar{A}$.

3 Reductions

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

Theorem 3.1. *Let A be any set that has an AND-function. If there is a set S with subexponential density such that $A \leq_m^p S$ then $A \in$ NP/poly.*

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 $x = x_i$, and in addition $f(g(x_1, \ldots, x_n)) = z$.

The idea is to show that there exists a string $z_1 \in S$ that is NP-proof for half the strings in $A_{=n}$. We will then recurse on the remaining strings in $A_{=n}$, for which z_1 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 $x \in A_{=n}$ 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

$$\left|\left\{ <x_1, \ldots, x_n> \;\middle|\; \begin{array}{l} |x_i| = n \text{ and} \\ f(g(x_1, \ldots, x_n)) = z \end{array} \right\}\right| \leq t^n \tag{3.1}$$

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_1 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 $S_{\leq n_m}$, we now have:

$$t^n \|S_{\leq m_n}\| \geq \|A_{=n}\|^n \tag{3.2}$$

and hence

$$t^n 2^n \geq \|A_{=n}\|^n \tag{3.3}$$

which implies that $t \geq \|A_{=n}\|/2$, and hence z_1 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 $f(g(y_1, \ldots, y_n)) \in S \setminus \{z_1, \ldots, z_{i-1}\}$. Hence the counting arguments in equations (3.1), (3.2), and (3.3) still hold for A'. □

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 $\mathcal{B} = (B_n \mid n \geq 0)$ be a family of subsets of $\{0,1\}^*$. We say that A NP-reduces to $\mathcal{B}$ 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 $\mathcal{B} = (B_n \mid n \geq 0)$ be a family of sets with $\|B_n\| \leq 2^{n^\alpha}$ for sufficiently large n. If A NP-reduces to $\mathcal{B}$, then $A \in$ NP/poly.*

Proof. Let M compute the NPMV function for the reduction from A to $\mathcal{B}$. Let g be the AND-function for A. For simplicity we assume that for all $x_1, \ldots, x_n \in \{0,1\}^m$, 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 $\frac{k}{k+1} \geq \alpha s$. Fix an input length m, let $n = m^k$, and let $N = (nm)^s$. Note that we have

$$\|B_N\| \leq 2^{N^\alpha} = 2^{m^{(k+1)s\alpha}} \leq 2^{m^k} = 2^n.$$

For any $x \in \{0,1\}^m$,

$$x \in A \iff \begin{array}{l} \text{there exist } x_1, \ldots, x_n \in \{0,1\}^m \\ \text{with } x_i = x \text{ for some } i \text{ such that} \\ M \text{ on input } g(x_1, \ldots, x_n) \text{ outputs} \\ \text{some string } z \in B_N. \end{array}$$

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 $z_1, \ldots, z_l$ such that each $x \in A_{=m}$ has an NP-proof in the collection.

Suppose that z is an NP-proof for exactly t strings in $A_{=m}$. Then

$$\|\{<x_1, \ldots, x_n> \mid M(g(x_1, \ldots, x_n)) \text{ outputs } z\}\| \leq t^n.$$

Let t be the maximal such that some string z is an NP-proof for t strings. Then

$$\|A_{=m}\|^n \leq \|B_N\| \cdot t^n \leq 2^n t^n,$$

so $t \geq \|A_{=m}\|/2$. Therefore there is a string z_1 that works for at least half of the strings in $A_{=m}$. 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. □

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 $A \leq^{\mathrm{p}}_{\mathrm{d}}$ SUBEXPD, then $A \in$ NP/poly.*

Proof. Suppose that $A \leq^{\mathrm{p}}_{\mathrm{d}} S \in$ SUBEXPD 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 $\|B_n\| \leq 2^{n^\alpha}$ for sufficiently large n because S has subexponential density. By Lemma 3.2 we have $A \in$ NP/poly. □

We apply Theorem 3.3 with $\overline{\mathrm{SAT}}$ to obtain the following:

Theorem 3.4. *If* coNP $\not\subseteq$ NP/poly, *then every $\leq^{\mathrm{p}}_{\mathrm{d}}$-hard set for* coNP *is exponentially dense.*

Allender, Hemachandra, Ogiwara, and Watanabe [1] showed that if $A \leq^{\mathrm{p}}_{\mathrm{btt}}$-reduces to a sparse set, then $A \leq^{\mathrm{p}}_{\mathrm{d}}$-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 $\|S_{\leq n}\| = d(n)$.

Lemma 3.5. *Let S be a set with density $d(n)$. Then there is a set T with density at most $nd(n) + n$ such that $\overline{S} \leq^{\mathrm{p}}_{\mathrm{d}} T$. In particular, if $S \in$ SUBEXPD, then $\overline{S} \leq^{\mathrm{p}}_{\mathrm{d}} T$ for some $T \in$ SUBEXPD.*

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 $S_{=n}$, but wb does not have an extension in $S_{=n}$. If $S_{=n} = \emptyset$, we add 0^n1 to T.

We claim that a string y is in $\overline{S}_{=n}$ if and only if y has a prefix z such that $0^n1z \in T$.

- If $y \notin S$ and $S_{=n} \neq \emptyset$, then let z be the longest prefix of y that has an extension in S. The string 0^n1z is in T. If $S_{=n} = \emptyset$, 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 $0^n1z \notin T$.

Therefore $\overline{S} \leq^{\mathrm{p}}_{\mathrm{d}} T$ via the reduction that lists the prefixes of its input.

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

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

Theorem 3.6. *If A has an OR-function and $A \leq^{\mathrm{p}}_{\mathrm{c}}$ SUBEXPD, then $A \in$ coNP/poly.*

Proof. Suppose that $A \leq_c^p S \in$ SUBEXPD. Then $\overline{A} \leq_d^p \overline{S}$ and by Lemma 3.5 there is a $T \in$ SUBEXPD such that $\overline{S} \leq_d^p T$. Composing reductions yields $\overline{A} \leq_d^p T$, so $\overline{A} \in$ NP/poly by Theorem 3.3, because the OR-function for A is an AND-function for $\overline{A}$. $\square$

Theorem 3.7. *If* coNP $\not\subseteq$ NP/poly, *then every* $\leq_c^p$*-hard set for* 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 $A \leq_{n^\alpha\text{-T}}^p$ SUBEXPD, then $A \in$* NP/poly.

Proof. Suppose $A \leq_{n^\alpha\text{-T}}^p S \in$ SUBEXPD via M. Fix an input length n. For an input $x \in \{0,1\}^n$, consider using each $z \in \{0,1\}^{n^\alpha}$ as the sequence of yes/no answers to M's queries. Each z causes M to produce a sequence of queries $w_1^{x,z}, \ldots, w_{n^\alpha}^{x,z}$ and an accepting or rejecting decision. (We can assume that M always makes n^α 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

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

where $\overline{S} \oplus S$ is the disjoint union $\{0x \mid x \in \overline{S}\} \cup \{1x \mid x \in S\}$.

By Lemma 3.5 there is a set $T \in$ SUBEXPD such that $\overline{S} \leq_d^p T$. Let $U = T \oplus S$. We then have $\overline{S} \oplus S \leq_d^p U$ via some reduction g. For each $z \in Z_x$, let

$$H_{x,z} = \{<u_1, \ldots, u_{n^\alpha}> \mid (\forall j)\ u_j \in g(z[j] \cdot w_j^{x,z})\}.$$

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

$$B_n = \{<u_1, \ldots, u_{n^\alpha}> \mid (\forall j)\ u_j \in U_{\leq r(n)}\}.$$

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 tuple $y \in H_{x,z}$ 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 $U \in$ SUBEXPD, $\|U_{\leq r(n)}\| \leq 2^{n^\delta}$ for sufficiently large n. This implies

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

Lemma 3.2 applies to show $A \in$ NP/poly. $\square$

We now have the main result of this paper:

Theorem 3.9. *If* coNP $\not\subseteq$ NP/poly, *then for all $\epsilon > 0$, every $\leq_{n^{1-\epsilon}\text{-T}}^p$-hard set for* NP *is exponentially dense.*

Proof. Suppose that SAT $\leq_{n^{1-\epsilon}\text{-T}}^p$-reduces to a subexponentially dense set. Then $\overline{\text{SAT}}$ $\leq_{n^{1-\epsilon}\text{-T}}^p$-reduces to the same set by inverting the reduction's answers. Moreover $\overline{\text{SAT}}$ has an AND-function of order $s = 1$. Theorem 3.8 applies to show coNP $\subseteq$ 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 $S \in$ SUBEXPD, then $A \in$* NP/poly.

Proof. We extend the proof of Theorem 3.8. Suppose $A = L(M^{S[n^\alpha]})$ 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 $w_0^{x,p,z}, \ldots, w_{n^\alpha}^{x,p,z}$ 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

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

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* coNP $\not\subseteq$ NP/poly, *then for all $\epsilon > 0$, every $\leq_{n^{1-\epsilon}\text{-T}}^{\text{SNP}}$-hard set for* NP *is exponentially dense.*

Proof. Suppose that S has subexponential density and is $\leq_{n^{1-\epsilon}\text{-T}}^{\text{SNP}}$-hard for NP. Then $\overline{\text{SAT}} \leq_{n^{1-\epsilon}\text{-T}}^{\text{SNP}} S$, so $\overline{\text{SAT}} \in \text{NP}^{S[n^{1-\epsilon}]}$. Theorem 3.10 implies $\overline{\text{SAT}} \in$ NP/poly. $\square$

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

Theorem 3.12. *For all $\epsilon > 0$, every $\leq_{n^{1-\epsilon}\text{-T}}^p$-hard set for $\Sigma_3^{\tilde{\text{P}}}$ is exponentially dense.*

Proof. First, we claim that $\Sigma_3^{\tilde{\mathrm{P}}} \not\subseteq \mathrm{NP/poly}$. This is similar to Kannan's proof that Σ_2^{P} does not have n^k-size circuits [7]. We can show that there is a set $H \in \Sigma_4^{\tilde{\mathrm{P}}} - \mathrm{NP/poly}$ by a direct counting argument. Then we consider two cases: if $\mathrm{coNP} \not\subseteq \mathrm{NP/poly}$, the claim holds immediately because $\mathrm{coNP} \subseteq \Sigma_3^{\tilde{\mathrm{P}}}$. Otherwise $\mathrm{coNP} \subseteq \mathrm{NP/poly}$ and we have $\mathrm{PH} = \Sigma_3^{\mathrm{P}}$ by Yap's theorem [13]. From this padding gives $\tilde{\mathrm{PH}} = \Sigma_3^{\tilde{\mathrm{P}}}$ and therefore $H \in \Sigma_3^{\tilde{\mathrm{P}}}$.

There is a many-one complete set A for $\Sigma_3^{\tilde{\mathrm{P}}}$ with an AND-function of order 1. Suppose that $A \leq_{n^{1-\epsilon}-\mathrm{T}}^{\mathrm{P}}$-reduces to a set S of subexponential density. Theorem 3.8 implies $A \in \mathrm{NP/poly}$, so $\Sigma_3^{\tilde{\mathrm{P}}} \subseteq \mathrm{NP/poly}$, 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 $\mathrm{ic}^t(x : A)$ 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 $\mathrm{ic}^q(x : A) \leq n^\alpha$ for all but finitely many $x \in A$, then $A \in \mathrm{NP/poly}$.*

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 $A \in \mathrm{NP/poly}$. □

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

Corollary 4.2 should be contrasted with the result that if $\mathrm{P} \neq \mathrm{NP}$, then there are infinitely many ϕ with $\mathrm{ic}^q(\phi : \mathrm{SAT}) \geq c \log |\phi|$. With the stronger $\mathrm{NP} \not\subseteq \mathrm{coNP/poly}$ hypothesis, we get a nearly linear lower bound on the instance complexity of SAT instances. Since $ic^{t(n)}(\phi : \mathrm{SAT}) \leq |\phi| + O(1)$ 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 $H = \{x \in A \mid ic^q(x : A) > |x|^\alpha\}$. If $\|H_{\leq n}\| \leq 2^{n^\alpha}$ for sufficiently large n, then $A \in \mathrm{NP/poly}$.*

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

$$B_n = 0H_{\leq n} \cup 1P_n.$$

Then $\|B_n\| \leq 2^{n^\alpha+2}$ 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 $A \in \mathrm{NP/poly}$. □

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

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

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 $\mathrm{ic}^r(x : C) < |x|^\delta$ for all but finitely many x. If $A \leq_{n^\alpha-\mathrm{T}}^{\mathrm{P}} C$, then $A \in \mathrm{NP/poly}$.*

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

Let x have length n. We can assume that M makes exactly n^α 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 $<p_1, \ldots, p_{n^\alpha}>$ of programs it guessed.

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

$$\mathrm{ic}^r(q_i : C) < |q_i|^\delta \leq t(n)^\delta < n^\epsilon.$$

Define

$$E_n = \{p \mid \text{ } p \text{ is consistent with } C \text{ and } |p| < n^\epsilon\}$$

and

$$B_n = \{<p_1, \ldots, p_{n^\alpha}> \mid \text{each } p_i \in E_n\}.$$

Then $\|B_n\| \leq (2^{n^\epsilon})^{n^\alpha} = 2^{n^{(1/s)-\epsilon}}$ and N reduces A to the family $(B_n \mid n \geq 0)$. Lemma 3.2 now applies to show $A \in \mathrm{NP/poly}$. □

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 $\alpha < 1/s$. Let C be a set where for all $\delta > 0$, there is a polynomial r such that the collection of hard instances*

$$H^{\delta,r} = \{x \mid \mathrm{ic}^r(x : C) \geq n^\delta\}$$

has subexponential density. If $A \leq^{\mathrm{p}}_{n^\alpha-\mathrm{T}} C$, then $A \in \mathrm{NP/poly}$.

Proof. Let M compute the reduction from A to C in $t(n)$ time. We assume that M makes exactly n^α queries. Let $\epsilon = [(1/s) - \alpha]/2$ and choose $\delta > 0$ such that $t(n)^\delta < n^\epsilon$ 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 $<z_1, \ldots, z_{n^\alpha}>$.

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

$$E_n = \{<\lambda, p> \mid \text{ } p \text{ is consistent with } C \text{ and } |p| < n^\epsilon\},$$

$$\begin{aligned} D_n &= \{<1, q> \mid q \in H^{\delta,r}_{\leq t(n)} \cap C\} \\ &\quad \cup \{<0, q> \mid q \in H^{\delta,r}_{\leq t(n)} \cap \overline{C}\}, \end{aligned}$$

and

$$B_n = \{<z_1, \ldots, z_{n^\alpha}> \mid \text{each } z_i \in D_n \cup E_n\}.$$

Then

$$\|B_n\| = \left(\|E_n\| + \|H^{\delta,r}_{\leq t(n)}\|\right)^{n^\alpha} \leq (2^{n^\epsilon+1})^{n^\alpha} \approx 2^{n^{(1/s)-\epsilon}}.$$

We apply Lemma 3.2 to obtain $A \in \mathrm{NP/poly}$. □

Corollary 4.7. *Suppose that* $\mathrm{NP} \not\subseteq \mathrm{coNP/poly}$ *and let C be $\leq^{\mathrm{p}}_{n^{1-\epsilon}-\mathrm{T}}$-hard for* NP. *There is a $\delta > 0$ such that for every polynomial r, the set*

$$\{x \,|\mathrm{ic}^r(x : C) \geq |x|^\delta\}$$

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 Σ_3^{p}-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

[1] E. Allender, L. A. Hemachandra, M. Ogiwara, and O. Watanabe. Relating equivalence and reducibility to sparse sets. *SIAM Journal on Computing*, 21(3):521–539, 1992.

[2] L. Berman and H. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing*, 6(2):305–322, 1977.

[3] H. Buhrman and S. Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In *Proceedings of the 12th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 116–127. Springer, 1992.

[4] L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *Proceedings of the 40th ACM Symposium on the Theory of Computing*, 2008. To appear.

[5] S. Fortune. A note on sparse complete sets. *SIAM Journal on Computing*, 8(3):431–433, 1979.

[6] J. M. Hitchcock. Online learning and resource-bounded dimension: Winnow yields new lower bounds for hard sets. *SIAM Journal on Computing*, 36(6):1696–1708, 2007.

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

[8] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309, 1980.

[9] S. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982.

[10] A. Meyer and M. Paterson. With what frequency are apparently intractable problems difficult? Technical Report MIT/LCS/TM-126, M.I.T., 1979.

[11] M. Ogiwara and O. Watanabe. On polynomial-time bounded truth-table reducibility of NP sets to sparse sets. *SIAM Journal on Computing*, 20(3):471–483, 1991.

[12] P. Orponen, K-I Ko, U. Schöning, and O. Watanabe. Instance complexity. *Journal of the ACM*, 41(1):96–121, 1994.

[13] C. K. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26(3):287–300, 1983.