On Average P vs. Average NP

Jie Wang * Jay Belanger

Wilkes University t

Abstract

This paper studies average-case complexity classes. Structures of polynomial-time computable distributions and polynomial-time many-one reductions on randomized decision problems are investigated. We widen the scope from the most studied class DNP (Distributional-NP) to the class ANP (Average-NP) which consists of randomized decision problems accepted by nondeterministic Turing machines in average polynomial time. Our results include: 1) $P \neq NP$ if and only if there exists a randomized decision problem $(D, \mu) \in ANP-AP$ (Average-P) such that it is hard on positive instances and $D$ is almost in $NP$ with respect to $\mu$. 2) All polynomial-time many-one complete problems for DNP are average polynomial-time many-one complete for ANP with respect to polynomial-time computable distributions. We also prove that there is a randomized decision problem which is average polynomial-time many-one complete for AP with respect to polynomial-time computable distributions. We also prove that any randomized decision problem which is average polynomial-time many-one complete for AP with respect to polynomial-time computable distributions but not contained in DNP. 3) AP, and ANP, with polynomial-time computable distributions, and DNP are not closed under polynomial-time many-one reductions. So these classes are not closed under any weaker reductions. 4) AP and ANP with respect to arbitrary distributions do not have complete problems under polynomial-time one-one reductions.

1 Introduction

NP-completeness is a worst-case concept and many NP-complete problems have been proved to be easy on average for random instances. So the average-case complexity of a problem is in many cases a more significant measure than its worst-case complexity. This motivated the recent study of average-case complexity, for example, see [Lev84, BCGL89, Gur91, and *Supported in part by NSF grant CCR-9108899.
[1]Department of Mathematics and Computer Science, Wilkes University, Wilkes-Barre, PA 18766. Internet: jwang@wilkes1.wilkes.edu and belanger@wilkes1.wilkes.edu.

[2]We may consider randomized search problems as well. Ben-David, Chor, Goldreich, and Luby [BCGL89] proved that randomized search problems are equivalent to randomized decision problems with respect to randomized polynomial-time reductions. For more information about randomized reductions of search problems, we refer the reader to Blass and Gurevich [BG91].

0-8186-2955-X/92 $03.00 © 1992 IEEE

318
unlike in the worst-case complexity, only a small number of complete randomized decision problems are known so far. Is it by nature that proving a randomized decision problem to be hard on average is difficult? To answer this question, we investigate the structures of polynomial-time computable distribution functions and the polynomial-time many-one reductions on randomized decision problems. Our results indicate that the structures of average-case complexity classes are very different from their counterparts in worst-case complexity due to the complex structures of probability distributions and distribution functions.

We first consider the relationship between average-case complexity classes and the worst-case complexity classes. It is easy to see that \( P = \text{NP} \) implies that DNP is included in AP. Ben-David et al. [BCGL89] proved that if \( \text{DTIME}(2^{O(n)}) \neq \text{NTIME}(2^{O(n)}) \), then DNP is not included in AP. Does \( P \neq \text{NP} \) imply that DNP is not included in AP? We observe that unlike worst-case complexity where NP includes P, DNP does not include AP even when distribution functions are polynomial-time computable. This makes it harder to study the relationship between DNP and AP. To resolve this problem, we consider a more general average-case complexity class. Along the definition of AP, it is both logical and natural to define a new class ANP (Average-NP) to be the class of randomized decision problems \((D, \mu)\) such that \( D \) is acceptable by a (nondeterministic) Turing machine in polynomial time on \( \mu \)-average. This has also been noted in [BCGL89] and [Gur91]. ANP is a natural analog of NP and AP includes AP. We prove that the average-case complexity classes do indeed have a very strong connection with the worst-case complexity classes. In particular, we prove that \( P \neq \text{NP} \) if and only if there is a randomized decision problem \((D, \mu) \in \text{ANP} - \text{AP} \) such that \( D \) is almost in \( \text{NP} \) with respect to \( \mu \) and \((D, \mu)\) is hard on positive instances.

In the definition of AP, there is no requirement on the computability of the probability function \( \mu \). Hence, \( \mu \) does not even have to be computable. Levin hypothesizes [Jo84] that any natural probability distribution either has a polynomial-time computable distribution function, or else is dominated by a function that does. So we consider a natural subclass \( \text{AP}_P \) of AP to be the class of randomized decision problems \((D, \mu) \) in AP such that \( \mu \) has a polynomial-time computable distribution. Similarly, we can define the natural subclass \( \text{ANP}_P \) of ANP. Ben-David et. al. [BCGL89] argued that requiring probability functions to have polynomial-time computable distributions may seem too restricting. They presented a wider family of distributions, \( \text{P-samplable} \), which consists of distributions that can be sampled by probabilistic algorithms working in time polynomial in the length of the sample generated. However, they also noted that the distributions in \( \text{P-samplable} \) seem to be too complicated. Impagliazzo and Levin [IL90] recently proved that every DNP problem complete for polynomial-time computable distributions is also complete for all \( \text{P-samplable} \) distributions. Therefore, using \( \text{P-samplable} \) distributions does not generate harder randomized decision problems than using polynomial-time computable distributions. So in this paper we will focus on polynomial-time computable distributions. We will focus on many-one reductions as well. We show that \( \text{ANP}_P \) properly includes DNP. We prove that every \( \leq_{m}^{d} \)-complete randomized decision problem for DNP is average polynomial-time complete (symbolically, \( \leq_{m}^{d} \)-complete) for \( \text{ANP}_P \). They are also \( \leq_{m}^{n} \)-complete for the class of randomized decision problems in ANP whose probability distributions are dominated by functions that have polynomial-time computable distributions. Moreover, we prove that there is a randomized decision problem which is \( \leq_{d}^{n} \)-complete for \( \text{ANP}_P \) but not contained in DNP. So this problem is not complete for DNP under any reductions. All these known many-one complete problems are in fact one-one complete.

We use the polynomial-time computability of distribution functions in proving our completeness theorems. However, one could lose some nice properties by restricting to such simple distributions. For example, we prove that \( \text{AP}_P \), DNP, and \( \text{ANP}_P \) are not closed under \( \leq_{m}^{d} \)-reductions. So these classes are not closed under any reductions weaker than \( \leq_{m}^{d} \)-reductions. On the other hand, if there is no restriction at all on the computability of distribution functions, we can prove that neither \( \text{AP} \) nor \( \text{ANP} \) contains hardest randomized decision problems with respect to polynomial-time one-one reductions.

2 Preliminaries

In the worst-case time complexity, the size of inputs is the length. In the average-case time complexity, one may allow an algorithm to run longer on "rare" inputs. So one uses \(|x| r(x)\) rather than \(|x|\) as the size of instance \( x \), where \( r(x) \) is a measure of rareness which satisfies a randomness test, i.e., its expectation \( E_{x} r(x) < 1 \). This is Levin's definition of a randomness test in Martin L"of's sense. A running
time $t(x)$ is polynomial on average if $t(x) = \left(\lfloor x r(x) \rfloor\right)^k$ for some $k > 0$ and some $r$ as above. Let $\epsilon = 1/k$, then $E_x t(x)/|x| = E_x r(x) < 1$. This is equivalent to the following definition.

Let $\mu$ be a probability distribution on $\Sigma^*$, i.e., $\mu(x) \geq 0$ for any $x \in \Sigma^*$ and $\sum_{x \in \Sigma^*} \mu(x) = 1$. In general, we only require that $\sum_{x \in E} \mu(x)$ converges. We also require that $\mu(x) > 0$ for infinitely many $x$ to avoid trivial results. The distribution function (or distribution, in short) of a probability distribution $\mu$ is defined by $\mu^*(x) = \sum_{y \leq x} \mu(y)$. So $\mu(x) = \mu^*(x) - \mu^*(x - 1)$.

**Definition 1** Let $\mu$ be a probability distribution. A function $f$ is polynomial on $\mu$-average if there is an $\epsilon > 0$ such that

$$\sum_{|x| > 0} f^\epsilon(x)/|x| \cdot \mu(x) = O(1).$$

For more information and discussion about this definition, the reader is referred to Gurevich [Gur91].

Denote by AP the class of randomized decision problems $(D, \mu)$ such that there is a deterministic Turing machine $M$ accepting $D$ in time $T$ which is polynomial on $\mu$-average. Denote by PDF the class of probability distributions $\mu$ such that its distribution function $\mu^*$ is computable in polynomial time. Let DNP denote the class of all randomized decision problems $(D, \mu)$ such that $D \in \text{NP}$ and $\mu \in \text{PDF}$; ANP the class of all $(D, \mu)$ such that $D$ is accepted by a Turing machine in time $T$ which is polynomial on $\mu$-average. So AP is a subset of ANP. Let AP denote the class of $(D, \mu) \in AP$ such that $\mu \in \text{PDF}$; and ANP the class of $(D, \mu) \in \text{ANP}$ such that $\mu \in \text{PDF}$.

A function $f$ from $\Sigma^*$ to the interval $[0,1]$ of reals is computable in polynomial time (cf. [Ko83]) if there exists a polynomial time algorithm $A(x, 1^k)$ such that, for every $\Sigma$-string $x$ and every positive integer $k$, $A(x, 1^k)$ is a binary fraction and $|f(x) - A(x, 1^k)| < 2^{-k}$.

Notice that the polynomial-time computability of a function $f$ from $\Sigma^*$ to $[0,1]$ in the above definition does not guarantee the computability of the $k$th digit of $f(x)$ [Gur91]. It is easy to see that a probability distribution $\mu$ is computable in polynomial time if the corresponding distribution $\mu^*$ is polynomial-time computable. But the converse is not true unless $P = \text{NP}$ (see [Gur91]).

We now define reductions from one randomized decision problem to another. Intuitively, such a reduction should be efficiently computable, and "preserve" the probability distribution. In other words, the reduction should not transform very likely instances of the first problem to rare instances of the second problem.

Let $\mu_1$ and $\mu_2$ be probability distributions on strings in the same alphabet $\Sigma$. $\mu_1$ dominates (resp. weakly dominates) $\mu_2$ if there is a function $f$ from $\Sigma^*$ to non-negative reals such that $\mu_1(x) \leq f(x) \mu_2(x)$ and $f$ is polynomially bounded (resp. polynomial on $\mu_1$-average). A function $f$ transforms a probability distribution $\mu_1$ into a probability function $\mu_2$ if $\mu_2(y) = \sum_{x \in f^{-1}(y)} \mu_1(x)$. A function $f$ transforms $(D_1, \mu_1)$ to $(D_2, \mu_2)$ if it transforms $\mu_1$ to $\mu_2$ and $D_1\{x : \mu_1(x) > 0\}$ is many-one reducible to $D_2$ via function $f$.

Let $\mu_1 \leq \mu_2$ (resp. $\mu_1 \leq^w \mu_2$) denote that $\mu_1$ is dominated (resp. weakly dominated) by $\mu_2$. $\mu_2$ dominates (resp. weakly dominates) $\mu_1$ with respect to a function $f$, symbolically $\mu_1 \leq f \mu_2$ (resp. $\mu_1 \leq^w f \mu_2$), if there exists some $\nu \ga \mu_1$ (resp. $\nu \ga w \mu_1$) such that $f$ transforms $\nu$ into a restriction of $\mu_2$.

**Definition 2** [Gur91]

1. $(D_1, \mu_1)$ is polynomial-time many-one reducible (symbolically, $\leq^p_m$-reducible) to $(D_2, \mu_2)$ if there is a total polynomial-time computable function $f$ such that $D_1\{x : \mu_1(x) > 0\} \leq^p_m D_2$ via $f$ and $\mu_1 \leq^f \mu_2$.

2. $(D_1, \mu_1)$ is average polynomial-time many-one reducible (symbolically, $\leq^a_m$-reducible) to $(D_2, \mu_2)$ if there is a total function $f$ which is polynomial-time computable on $\mu_1$-average such that $D_1\{x : \mu_1(x) > 0\} \leq^a_m D_2$ via $f$ and $\mu_1 \leq^a \mu_2$.

In both of these reductions, if the function $f$ is one-one, then they are called one-one reductions and they are denoted symbolically by $\leq^p_1$ and $\leq^a^p_1$ respectively.

A class $C$ is closed under a reduction $\leq_r$ if $x \leq_r y$ and $y \in C$ implies $x \in C$. Here $C$ could be a class of sets or a class of randomized decision problems. It was proved in [Gur91] that both $\leq_m^p$ and $\leq_m^a$ reductions for randomized decision problems are transitive and they are closed for AP. Clearly, if $(D, \mu)$ is $\leq_m^p$-complete for DNP, then $D$ must be $\leq_m^p$-complete for NP.

One can easily prove that every randomized decision problem in AP (resp. DNP, ANP, AP, ANP) is polynomial-time transformable to some randomized decision problem in AP (resp. DNP, ANP, AP).
ANPₚ) over the binary alphabet [Gur91]. So without loss of generality, we use the binary alphabet in this paper.

Let \( M_1, M_2, \ldots \) be a fixed standard enumeration of all (deterministic and nondeterministic) Turing machines in which \( i \) is a program which simply codes up the states, symbols, tuples, etc. of the \( i \)th Turing machine \( M_i \).

3 Average-Case and Worst-Case Complexity

We now study the relationship between average-case complexity classes and the worst-case complexity classes. We first study the inclusion relations between average-case complexity classes. It is easy to see that AP is not included in DNP because there is no restriction on the computability of probability distributions in AP. However, even when probability distributions have polynomial-time computable distributions, AP is still not included in DNP. So DNP does not include all easy problems.

**Theorem 1** APₚ is not included in DNP.

**Proof.** Let \( A \) be a set not in NP which is accepted by a deterministic Turing machine \( M \). Let \( D = 1A \cup \{0\}^* \). Then \( D \not\in NP \). Let \( \mu(x) = 1/n^2 \) if \( x = 0^n \), 0 otherwise. Clearly, \( \mu \) is a probability function and \( \mu^* \) is polynomial-time computable. We construct a deterministic Turing machine \( M' \) as follows: on any input \( x \), if \( x = 0^n \), then \( M' \) accepts; if \( x = 1y \), then \( M' \) simulates \( M \) on input \( y \); otherwise, \( M' \) rejects. Notice that \( \mu(x) = 0 \) when \( x \neq 0^n \), so clearly, \( M' \) accepts \( D \) in polynomial time on \( \mu \)-average. \( \blacksquare \)

**Remark 1** The probability distribution \( \mu \) constructed above only generates strings \( 0^n \). What is more interesting is to have a probability distribution \( \mu \) such that it is positive everywhere. We can show that if DTIME\( (2^{o(n)}) \neq \text{NP} \), then there is a set \( D \) and a probability distribution \( \mu \) such that \( \mu(x) > 0 \) for any \( x \) and \( (D, \mu) \in \text{ANP}_p - \text{DNP} \) as follows. Let \( D \in \text{DTIME}(2^{o(n)}) - \text{NP} \), then there is a deterministic Turing machine accepting \( D \) in time \( 2^{p(n)} \) for some polynomial \( p \). Let \( \mu(x) = 2^{-p(|x|)} - 1 \), then \( (D, \mu) \in \text{AP}_p - \text{DNP} \). \( \blacksquare \)

By definition, DNP is included in \( \text{ANP}_p \). We can show that DNP is properly included in \( \text{ANP}_p \).

**Theorem 2** DNP is properly included in \( \text{ANP}_p \).

**Proof.** That DNP is a subset of \( \text{ANP}_p \) follows directly from the definition. Now from the proof of Theorem 1, we know that there is a pair \( (D, \mu) \) in \( \text{AP}_p - \text{DNP} \). Since \( \text{AP}_p \subseteq \text{ANP}_p \), this completes the proof. \( \blacksquare \)

Similar to the proof of Theorem 2, it is easy to prove that there are problems \( (D, \mu) \in \text{ANP} \) such that \( D \not\subseteq \text{NP} \). Given a set \( D \) and a probability distribution \( \mu \), we define a notion of \( D \) being almost in \( \text{NP} \) with respect to \( \mu \) to indicate that hard instances of \( D \) beyond \( \text{NP} \) under the probability distribution \( \mu \) are rare.

**Definition 3** Let \( D \) be a set and \( \mu \) be a probability distribution. \( D \) is almost in \( \text{NP} \) with respect to \( \mu \) if there is a Turing machine that accepts \( D \) in time \( T \), and \( D \) has a subset \( D' \) in \( \text{NP} \) such that

\[
\sum_{x \in D - D'} \frac{\mu^*(x)}{|x|} = O(1)
\]

for some constant \( \delta > 0 \).

**Definition 4** A randomized decision problem \( (D, \mu) \) is hard on positive instances if for any deterministic Turing machine that accepts \( D \) in time \( T(x) \) and for any \( \epsilon > 0 \)

\[
\sum_{x \in D} \frac{T^\epsilon(x)}{|x|} \mu(x) = \infty.
\]

Notice that in general a randomized decision problem is hard (i.e. not in AP) does not imply that it must be hard on positive instances.

Clearly, \( P = \text{NP} \) implies that DNP is included in AP. Ben-David et al. [BCGL89] proved that if DTIME\( (2^{O(n)}) \neq \text{NTIME}(2^{2O(n)}) \), then DNP is not included in AP. Does \( P \neq \text{NP} \) imply that DNP is not included in AP? Or in a more general setting, do average-case complexity classes have a strong connection with the worst-case complexity classes? We affirmatively answer this question. In particular, we prove that \( P \neq \text{NP} \) if and only if there is a randomized problem \( (D, \mu) \) in \( \text{ANP} - \text{AP} \) such that \( D \) is almost in \( \text{NP} \) with respect to \( \mu \) and \( (D, \mu) \) is hard on positive instances. The proof uses the concept of polynomial complexity cores from worst-case complexity theory.

Let \( A \) be a set not in \( \text{P} \). A polynomial complexity core of \( A \) is a set \( C \) such that for every machine \( M \) that accepts \( A \) and every polynomial \( p \) there are at most finitely many \( x \in C \) on which the number of steps of \( M \) is bounded by \( p(|x|) \). A polynomial complexity core \( C \) is proper if \( C \subseteq A \). Du [Du85] (see
also [DB89]) proved that if $P \neq \text{NP}$, then many natural NP-complete problems contain nonsparse proper polynomial complexity cores in $\text{DTIME}(2^{O(n)})$.

**Theorem 3** $P \neq \text{NP}$ if and only if there is a pair $(D,\mu) \in \text{ANP} - \text{AP}$ such that $D$ is almost in NP w.r.t. $\mu$ and $(D,\mu)$ is hard on positive instances.

**Proof.** ($\Rightarrow$) Suppose $P \neq \text{NP}$. Then there is an NP-complete set $D$ in NP – $P$ such that $D$ has a nonsparse proper polynomial complexity core $C$ in $\text{DTIME}(2^{O(n)})$ [Du85, DB89]. Therefore, $C \subseteq D$ and for every machine $M$ that accepts $D$ and every polynomial $p$ there are at most finitely many $x \in C$ such that the number of steps of $M$ on $x$ is bounded by $p(|x|)$. Let $C^{n,n} = \{x : |x| = n \text{ and } x \in C\}$. Define

$$\mu(x) = \begin{cases} \frac{1}{|x|^3 \cdot |C|^{n,n}}, & \text{if } x \in C, \\ \frac{1}{|x|^3}, & \text{otherwise.} \end{cases}$$

Clearly, $\mu$ is a probability distribution and is $\text{DTIME}(2^{O(n)})$ computable. Since $C$ is a proper complexity core of $D$, for any deterministic Turing machine that accepts $D$ in time $T$ and any $\epsilon > 0$, there are at most finitely many $x \in C$ such that $T^*(x) \leq |x|^3$. Therefore, there is a constant $m > 0$, such that when $|x| \geq m$, $T^*(x) > |x|^3$ if $x \in C$. Hence,

$$\sum_{x \in D} \frac{T^*(x)}{|x|^3} \cdot |x|^3 \cdot |C|^{n,n} \geq \sum_{x \in C, |x| \geq m} \frac{T^*(x)}{|x|^3} \cdot |C|^{n,n} \geq \sum_{i \geq m} \frac{1}{i} = \infty.$$

So $(D,\mu)$ is hard on positive instances. It is easy to see that $D$ is almost in NP with respect to $\mu$ because $D \in \text{NP}$.

($\Leftarrow$) Let $(D,\mu)$ be a pair in $\text{ANP} - \text{AP}$ such that $D$ is almost in NP with respect to $\mu$ and $(D,\mu)$ is hard on positive instances. By definition, there is a Turing machine $M$ that accepts $D$ in polynomial time $T(x)$ on $\mu$-average, and $D$ has a subset $D' \subseteq \text{NP}$ such that

$$\sum_{x \in D' \setminus D} 2^{-\delta T^*(x)} \cdot |x|^3 \cdot |C|^{n,n} \cdot |x|^3 \cdot |C|^{n,n} < \infty$$

for some constant $\delta > 0$, and for any deterministic Turing machine that accepts $D$ in time $t(x)$ and any $\epsilon > 0$,

$$\sum_{x \in D} \frac{t^*(x)}{|x|^3} \cdot \mu(x) = \infty.$$

We shall prove that $D' \notin \text{P}$. Suppose that $D' \in \text{P}$, then there is a deterministic Turing machine $M'$ which accepts $D'$ in polynomial time $|x|^k + k$ for some $k > 0$.

Now construct a deterministic machine $\tilde{M}$ as follows. On any input $x$, $\tilde{M}$ first simulates $M'$ on $x$. $\tilde{M}$ accepts $x$ if $M'$ accepts $x$. If $M'$ rejects $x$, then $\tilde{M}$ deterministically simulates $M$ on $x$. Clearly $\tilde{M}$ accepts $D$ within time $T(x)$ which is equal to $|x|^k + k$ when $x \in D'$ and $2^{|x|^3}$ when $x \notin D'$ for some constant $c > 0$. Let $l \geq \max\{k, c\}$ such that $c/l \leq \delta$. Then

$$\sum_{x \in D} \frac{T^*(x)}{|x|^3} \cdot \mu(x) \leq \sum_{x \in D'} O(\mu(x)) + \sum_{x \in D \setminus D'} \frac{2^{(c/l)T^*(x)}}{|x|^3} \cdot \mu(x) < \infty.$$

This contradicts the assumption. \[\square\]

## 4 Complete Problems for ANP

We know that the randomized tiling problem [Lev84], the randomized halting problem [Gur91], and the randomized Post Correspondence Problem [Gur91] are $\leq_m^p$-complete for $\text{DNP}$. We will show in this section that every $\leq_m^p$-complete problem for $\text{DNP}$ is also $\leq_m^\text{ANP}$-complete for ANP. We first show that the randomized halting problem is $\leq_m^\text{ANP}$-complete for ANP.

Let $K = \{(i, x, 1^n) : M_i \text{ accepts } x \text{ within } n \text{ steps}\}$, and $\mu_K(i, x, 1^n)$ be a standard probability distribution defined by

$$\mu_K(i, x, 1^n) = \left(\frac{2^{-|i|+1}}{|i|^3}\right) \cdot \left(\frac{|x|}{|i|^3}\right) \cdot \left(\frac{1}{n^2}\right).$$

We define a version of the randomized halting problem $\text{RH}$ by $\text{RH} = (K, \mu_K)$.

**Theorem 4** $\text{RH}$ is $\leq_m^\text{ANP}$-complete for ANP.

**Proof.** The proof is a modification of [Gur91]. A simpler proof is presented here.

Let $(D,\mu) \in \text{ANP}$. Then $\mu$ has a polynomial-time computable distribution and there is a Turing machine $M$ accepting $D$ in time $T$ which is time-constructible and polynomial on $\mu$-average. It was proved in [Gur91] (Lemma 1.6) that for every probability distribution $\mu \in \text{PDF}$ there is a positive probability distribution $\mu_1 \in \text{PDF}$ such that every value of $\mu_1$ has at most $4 + 2|x|$ binary digits and $\mu(x) < 4\mu_1(x)$. So $\mu \leq \mu_1$. Therefore, without loss of generality we may assume that $\mu$ is such $\mu_1$ for the purpose of constructing a reduction.
Let $x'$ be the shortest binary string such that 
\[ \mu'(x-1) < 0.2x'1 \leq \mu'(x). \]
Then 
\[ 0.2x'1 - 2^{-|x'|} \leq \mu'(x-1) < \mu'(x) < 0.2x'1 + 2^{-|x'|} \]
because otherwise $x'$ is not the shortest. Therefore, it is easy to see that \( \mu'(x) = \mu'(x) - \mu'(x-1) < 2^{-|x'|}+1 \). So 2\(^{-|x'|} > \mu'(x). \) Such an $x'$ can be found in polynomial time of $|x|$ because $\mu'$ is polynomial-time computable and $\mu(x)$ has at most $4 + 2|x|$ binary digits. So $|x'|$ is bounded by a polynomial of $|x|$.

Now we define the desired $\leq_{m}^{P}$-reduction from 
\((D, \mu)\) to RH. Define a Turing machine $M'$ as follows. On input $w$, find $x$ (this $x$ is unique) such that 
\[ \mu'(x-1) < 0.2w1 \leq \mu'(x). \]
This can be carried out in polynomial time of $|x|$ as follows: Find the smallest $n$ such that $\mu'(0^n) \geq 0.2w1$, then binary search all strings of length $n-1$ to find such a unique $x$. $M'$ then simulates $M$ on $w = x$ and rejects otherwise.

Clearly $M'$ accepts $w$ if and only if $w = x'$ and $M$ accepts $x$. Because of the time bound on $M$, it is easy to see that $M'$ on input $x'$ is bounded in time $g(x')$ such that $g(x')$ is polynomial of $x$ on-average. Let $i$ be a program such that $M' = M_i$. Let 
\[ f(x) = (i, x, 1^{|x'|}). \]

Clearly, $f$ is computable in polynomial on-average and $z \in D$ if and only if $f(z) \in K$ by the construction. It is easy to see that $f$ is one-one. It is also easy to see that for a one-one function $f$, $\mu \preceq_{w} f \mu$ if and only if $f(x) \preceq_{w} f(f(x))$ [Gur91]. We know that $\mu_K(f(x))$ is proportional to $g(x')^{-2|x'|^{-2}-2^{-|x'|}}$ which exceeds $g(x')^{-2|x'|^{-2}-2^{-|x'|}} \mu(x)$. So $\mu \preceq_{w} \mu_K$. Hence, $(D, \mu) \leq_{m}^{P}$ RH via $f$.

**Theorem 5**
Every $\leq_{m}^{P}$-complete problem for DNP is $\leq_{m}^{P}$-complete for ANP.

**Proof.** Similar to the proof of Theorem 4, we can prove that RH is $\leq_{m}^{P}$-complete for DNP since $K \in$ NP (see also [Gur91]). Thus, every $\leq_{m}^{P}$-complete problem for DNP is $\leq_{m}^{P}$-complete for ANP.

It would be interesting to know whether there are $\leq_{m}^{P}$-complete randomized decision problems for ANP.

**Remark 2**
Theorems 4 and 5 also hold for the class of randomized decision problems $(D, \mu)$ in ANP such that $\mu$ is dominated by a probability distribution which has polynomial-time computable distribution.

We can similarly prove that AP has $\leq_{m}^{P}$-complete problems. Let $K_d = \{(i, x, 1^n): M_i$ is deterministic and accepts $x$ within $n$ steps$\}$, and $\mu_{K_d}$ be as above. Then $(K_d, \mu_{K_d})$ is $\leq_{m}^{P}$-complete for AP.

We know that $K$ defined above is in NP. So $(K, \mu_K)$ is in DNP. Does there exist a randomized decision problem in ANP?—DNP which is complete for ANP? We affirmatively answer this question.

Let $K' = \{(i, x, n): M_i$ accepts $x$ within $n$ steps$\}$, and 
\[ \mu_{K'}(i, x, n) = \left( \frac{2^{-|x'|+1}}{|x|^2} \right)^{\frac{1}{n^3}}. \]

Clearly, $\mu_{K'} \in$ PDF. It is straightforward to prove that $K'$ is $\leq_{m}^{P}$-complete for NEXP (NEXP $\leq_{m}^{P}$ NEXP) by reducibility: Let $A$ be a subset of NEXP, then there is a Turing machine $M_i$ accepting $A$ in time $2^{p(n)}$ for some polynomial $p$. Let $f(z) = (i, z, 2^{p(|z|)})$. Then $f(x)$ is polynomial-time computable. (Note that $2^{p(|z|)}$ can be written as 1 followed by $p(|x|)$ many 0's in the binary system.) Clearly, $x \in A$ if and only if $f(x) \in K'$.

**Theorem 6**
$(K', \mu_K') \in$ ANP$-$DNP and is $\leq_{m}^{P}$-complete for ANP.

**Proof.** It has been proved in Seiferas, Fischer, and Meyer [SFM78, Corollary 4.1] that NTIME($T(n)$) = U(NTIME($T(n)$) : $T(n)$ is $\leq_{m}^{P}$-complete for NEXP and NP is closed under $\leq_{m}^{P}$-reduction. So $(K', \mu_{K'}) \notin$ DNP. Let $M$ be a Turing machine that accepts $(i, z, n)$ if and only if $M_i$ accepts $x$ within $n$ steps. So the running time of $M$ is bounded by $O(|(i, z, n)| + n)$. Hence, 
\[ \sum \frac{O(|(i, z, n)| + n)}{|(i, z, n)|} \mu_{K'}(i, z, n) \leq \sum \frac{O(n)\mu_{K'}(i, z, n)}{i, z, n} < \infty. \]

Therefore, $(K', \mu_{K'}) \in$ ANP. Now we will show that $(K', \mu_{K'})$ is $\leq_{m}^{P}$-complete for ANP by reducing RH to it. Let $f((i, z, 1^n)) = (i, z, n)$. Then $f$ is polynomial-time computable and $f$ is one-one. Clearly, $(i, z, 1^n) \in K$ if and only if $f((i, z, 1^n)) \in K'$. Since $\mu_{K'}(f((i, z, 1^n))) = n^{-1} \mu_{K}(i, z, 1^n)$ and $n < |(i, z, 1^n)|$, $\mu_{K} \preceq_{w} \mu_{K'}$. Hence, RH is $\leq_{m}^{P}$ $(K', \mu_{K'})$. So $(K', \mu_{K'})$ is $\leq_{m}^{P}$-complete for ANP.

It is clear from the definitions that if $(D, \mu)$ is $\leq_{m}^{P}$-complete for DNP, then $D$ must be $\leq_{m}^{P}$-complete for NP. We also have a partial converse.
Theorem 7 If $D$ is $\leq^m_{\text{NP}}$-hard for NP, then there exists a probability function $\mu$ such that $(D, \mu)$ is $\leq^m_{\text{NP}}$-hard for DNP.

Proof. Suppose $D$ is $\leq^m_{\text{NP}}$-hard for NP. Let $(D', \mu')$ be any $\leq^m_{\text{NP}}$-complete problem for DNP. Then, since $D' \in \text{NP}$, there exists a polynomial-time computable function $f$ such that $D' \leq^m_{\text{NP}} D$ via $f$. Define $\mu(y) = \sum_{x \in f(y)} \mu'(x)$. We then have $(D', \mu') \leq^m_{\text{NP}} (D, \mu)$ via $f$, and so $(D, \mu)$ is $\leq^m_{\text{NP}}$-hard for DNP.

Notice that in the above proof, even if $D \in \text{NP}$, $(D, \mu)$ is not necessarily $\leq^m_{\text{NP}}$-complete for DNP. Since $\mu$ could not be in PDF, $(D, \mu)$ may not even be in DNP.

5 Non-Closure Properties

In this section, we will prove that by restricting to simple distributions, $\text{AP}_P$, DNP, and $\text{ANP}_P$ are not closed under $\leq^m_{\text{NP}}$-reductions. Therefore, these classes are not closed under any reductions weaker than $\leq^m_{\text{NP}}$-reductions. We first show the following lemma.

Lemma 8 There are probability distributions $\mu \notin \text{PDF}$ and $\nu \notin \text{PDF}$, and a polynomial-time computable function $f$ such that $f$ transforms $\mu$ to a restriction of $\nu$, where $\mu(x) > 0$ and $\nu(x) > 0$ for all $x \in \Sigma^*$.

Proof. We first prove that there exists a set $H = \{x_1, x_2, \ldots\} \subseteq \{0, 1\}^*$ such that $|x_n| = n$ for all $n$ and $H \notin \text{P}$. Let $P_1, P_2, \ldots$ be an enumeration of all sets in $\text{P}$. For any $n$, define $x_n$ by

$$x_n = \begin{cases} 0^n, & \text{if } 0^n \notin P_n, \\ 1^n, & \text{otherwise}. \end{cases}$$

Let $H = \{x_1, x_2, \ldots\}$. Then $H \notin \text{P}$ for any $n$. So $H \notin \text{P}$. Let $p(n) = n(n + 1)/2$ and define $\mu$ by

$$\mu(x) = \begin{cases} 2^{-p(n)}, & \text{if } x = x_n, \\ 2^{-p(n)+n}, & \text{if } |x| = n \text{ and } x \neq x_n. \end{cases}$$

Then $\sum_{|x| = n} \mu(x) = 2^{-p(n)+1} - 2^{-p(n)+n}$. Since $p(n) = p(n - 1) + n$, it is easy to show that $\sum_{|x| \leq n} \mu(x) = 1 - 2^{-p(n)+n}$. So $\mu$ is a probability distribution.

We claim that $\mu$ is not polynomial-time computable. Suppose, to the contrary, that $\mu$ is polynomial-time computable. Then there exists a polynomial-time algorithm $A(x, 1^k)$ such that for all strings $x$ and all positive integers $k$, $|A(x, 1^k) - \mu(x)| < 2^{-k}$. Let $\alpha(x) = A(x, 1^{|x|+2})$. Then $\alpha$ is a polynomial-time algorithm, and for all $x$, $|\alpha(x) - \mu(x)| < 2^{-p(|x|)+2}$. Let $n = |x|$. If $x = x_n$, then $\alpha(x) = 2^{-p(|x|)+2}$, and so $|\alpha(x) - \mu(x)| < 2^{-p(|x|)+2}$. This implies $\alpha(x) > 2^{-p(|x|)+2} > 2^{-p(|x|)+1}$. Thus, in which means that the binary expansion of $\alpha(x)$ has a 1 by the $p(|x|)+1$ position to the right of the binary point. On the other hand, if $x \neq x_n$, then $\mu(x) = 2^{-p(|x|)+2}$, and $|\alpha(x) - 2^{-p(|x|)+2}| < 2^{-p(|x|)+2}$. This implies $\alpha(x) < 2^{-p(|x|)+2} + 2^{-p(|x|)+k} = 2^{-p(|x|)+1}$ for $|x| \geq 2$, and so the binary expansion of $\alpha(x)$ does not have a 1 in the first $p(|x|)+1$ positions to the right of the binary point. Therefore,

$$H = \{ x : x = x_1 \text{ or } (|x| > 1 \text{ and the binary expansion of } \alpha(x) \text{ has a 1 by the } p(|x|) + 1 \text{ position}) \}.$$ 

Therefore, $H \notin \text{P}$. So $\mu$ cannot be polynomial-time computable. Define $f : \{0, 1\}^* \rightarrow \{0\}^*$ by $f(x) = 0^{1x}$. Define $\nu$ by

$$\nu(y) = \begin{cases} 2^{-p(n)+1} - 2^{-p(n)+n}, & \text{if } y = 0^n, \\ |y|, & \text{otherwise}. \end{cases}$$

Then $\nu^*$ is polynomial-time computable. Since for any $y \in \text{image of } f$, $\nu(y) = \sum_{y = f(x)} \mu(x)$, $f$ transforms $\mu$ to a restriction of $\nu$.

Remark 3 Lemma 8 can also be proved such that $\mu$ is not computable for any super-polynomial time complexity.

Theorem 9 $\text{AP}_P$, $\text{ANP}_P$, and DNP are not closed under $\leq^m_{\text{NP}}$-reductions.

Proof. Let $\nu$, $\mu$, and $f$ be the functions constructed in the proof of Lemma 8. Notice that $|\mu(x)| > 0$ for any $x \in \{0, 1\}^*$. Let $B \subseteq \{0\}^*$ and $B \notin \text{P}$. Let $A = \{ x : 0^n \in B \}$. Then $A \in \text{P}$ and $A \leq^m_{\text{NP}} B$ via $f$. So $(A, \mu) \leq^m_{\text{NP}} (B, \nu)$ via $f$. However, $(B, \nu)$ is in $\text{AP}_P$, $\text{ANP}_P$ and DNP, but $(A, \mu)$ is not in $\text{AP}_P$, $\text{ANP}_P$ or DNP since $\mu \notin \text{PDF}$. Hence, $\text{AP}_P$, $\text{ANP}_P$ and DNP are not closed under $\leq^m_{\text{NP}}$-reductions.

Remark 4 In fact, for each $C \in \text{NP}$, we can construct a pair of randomized decision problems witnessing that DNP is not closed under $\leq^m_{\text{NP}}$-reductions. Let $\nu$, $\mu$, and $f$ be the functions constructed in the proof of Lemma 8. For any $C \in \text{NP}$, let $A = \{ x : \ldots \}$.
\((\exists \mu)[|y| = |x| \text{ and } \mu \in \mathcal{C}])\), and \(B = \{0_i|x| : x \in \mathcal{C}\}\). Then both \(A\) and \(B\) are in NP and \(A \leq_m^p B\) via \(f\). Therefore, \((A, \mu) \leq_m^p (B, \nu)\) via \(f\). Since \((B, \nu)\) is in DNP but \((A, \mu)\) is not in DNP, \((A, \mu)\) and \((B, \nu)\) witness that DNP is not closed under \(\leq_m^p\)-reductions. 

6 Non-Completeness Results

From the proof of Theorem 4, we can see that RH is actually average polynomial-time one-one complete for ANP. In fact, all the many-one complete problems for DNP known so far [Lev84, GurS1] are one-one complete.

We use the polynomial-time computability of the distribution functions in proving completeness theorems. In general, proving completeness theorems needs certain assumptions on the distribution functions. If there is no requirement at all, then we can prove that neither AP nor ANP contains hardest randomized decision problems with respect to polynomial-time one-one reductions. Gurevich [GurS1] proved that many NP-complete problems with natural probability distributions cannot be \(\leq_m^p\)-complete for DNP unless \(\text{EXP} = \text{NEXP}\), where \(\text{EXP} = \text{DTIME}(2^{p\text{oly}})\). We start with the following lemmas.

Lemma 10 Let \(\nu\) be a probability distribution on \(\Sigma^*\), and \(f : \Sigma^* \rightarrow \Sigma^*\) any one-one, total function. Then \(\nu(f(x)) < 1/2|x|^e\) for infinitely many \(x\).

Proof. Let \(\nu\) and \(f\) be above. Suppose, to the contrary, that for some \(n\), \(\nu(f(x)) \geq 1/2|x|^e\) for all \(x\) with \(|x| \geq n\). Then

\[\infty > \sum_{|x| \geq n} \nu(y) \geq \sum_x \nu(f(x)) \geq \sum_{|x| \geq n} \nu(f(x)) \geq \sum_{|x| \geq n} \frac{1}{2|x|^e},\]

but this last sum diverges.

Lemma 11 Let \(\nu\) be a probability distribution on \(\Sigma^*\). Then there exists a probability distribution \(\mu\) on \(\Sigma^*\) such that for any one-one, total, polynomial-time computable function \(f\), \(\mu\) is not dominated by \(\nu\) with respect to \(f\).

Proof. Let \(f_0, f_1, f_2, \ldots\) be a listing of all total, one-one, polynomial-time computable functions from \(\Sigma^*\) to \(\Sigma^*\) such that each function is included in this sequence an infinite number of times. (Notice that we do not require that such a sequence to be recursively enumerable.) Choose a sequence \((x_k) \subset \Sigma^*\) such that \(|x_k| \geq 2^k\) and \(\nu(f_k(x_k)) < 1/2|x_k|^n\) for all \(k\). This is possible by Lemma 10. Define the function \(\mu\) on \(\Sigma^*\) by

\[\mu(x) = \begin{cases} \frac{|x_k|^k}{2|x_k|^n}, & \text{if } x = x_k \\ \frac{|x|^e}{|x|^e}, & \text{otherwise}. \end{cases}\]

Then

\[\sum_x \mu(x) < \sum_x \frac{1}{|x|^2|x|^e} + \sum_x \mu(x_k) = \sum_n \frac{1}{2^n} + \sum_k \frac{|x_k|^k}{2|x_k|^n}.\]

Since \(x_k^k/2^k\) is a decreasing function of \(x\) for \(x > k/\ln 2\) and \(|x_k| \geq 2^k\), we have

\[\frac{|x_k|^k}{2|x_k|^n} \leq \frac{2^k}{2^{2^n}} = \frac{2^k}{2^{2^n}},\]

and so

\[\sum_x \mu(x) \leq \sum_n \frac{1}{2^n} + \sum_k \frac{2^k}{2^{2^n}} < \infty.\]

Hence, \(\mu\) is a probability distribution on \(\Sigma^*\). Suppose \(\mu \not\leq f \nu\) for some total, one-one, polynomial-time computable function \(f\). This means that for some probability distribution \(\mu_1\) on \(\Sigma^*\), \(\mu \leq \mu_1\) and \(f\) transforms \(\mu_1\) to \(\nu\). Since \(\mu \leq \mu_1\), for some polynomially bounded function \(g\), \(\mu(x) \leq g(x)\mu_1(x)\) for all \(x\). This implies that there exists an \(n\) such that \(\mu(x) \leq |x|^n\mu_1(x)\), and so \(\mu_1(x) \geq \mu(x)/|x|^n\) for all \(x\). Since \(f\) transforms \(\mu_1\) to \(\nu\), we have

\[\nu(y) = \sum_{x | y = f(x)} \mu_1(x).\]

By the definition of sequence \(f_0, f_1, \ldots\), we know that \(f = f_k\) for infinitely many \(k\), and for these values of \(k\),

\[\nu(f_k(x_k)) = \sum_{x_k | f_k(x_k) = x_k} \mu_1(x_k) \geq \mu(x_k) \geq \frac{|x_k|^k}{2|x_k|^n} \frac{1}{|x_k|^n} = \frac{|x_k|^k-n}{2|x_k|^n}.\]

Since \(\nu(f_k(x_k)) < 1/2|x_k|^n\), this cannot happen for arbitrarily large \(k\). So \(\mu\) cannot be dominated by \(\nu\) with respect to any total, one-one, polynomial-time computable function.
Theorem 12 AP (resp. ANP) does not contain polynomial-time one-one complete problems.

Proof. Suppose that $(E, \nu)$ is $\leq_{f}^p$-complete for AP. Let $D \in P$, and let $\mu$ be as in the proof of Lemma 11. Then $(D, \mu) \in AP$, but since $\mu$ cannot be dominated by $\nu$ with respect to any total, one-one, polynomial-time computable function, $(D, \mu)$ cannot be $\leq_{f}^p$-reducible to $(E, \nu)$. This is a contradiction.

The proof for ANP is similar.

It is easy to see that Theorem 12 also holds for functions $f$ if $\{y : f(y) = f(z)\}$ is bounded by a fixed constant for any $x$. It is open, however, whether Theorem 12 is true for other polynomial-time reductions.

Acknowledgments. We are grateful to Yuri Gurevich and R. Venkatesan for their comments.

References


