Rankable Distributions Do Not Provide Harder Instances Than Uniform Distributions

Jay Belanger
Div. of Math & Computer Science
Northeast Missouri State Univ.
Kirksville, MO 63501

Jie Wang *
Dept. of Mathematical Sciences
Univ. of North Carolina at Greensboro
Greensboro, NC 27412
E-mail: *wang@uncg.edu*

Abstract. We show that polynomially rankable distributions do not provide harder instances than uniform distributions for NP problems. In particular, we show that if Levin's randomized tiling problem is solvable in polynomial time on average, then every NP problem under any p-rankable distribution is solvable in average polynomial time with respect to rankability. We then present a reasonably tight hierarchy result for average-case complexity classes under uniform distributions.

1 Introduction

When finding a solution to an NP-complete problem one would be satisfied if one could find an algorithm to solve the problem in expected polynomial time with respect to the underlying distribution on instances. Instance distributions are an important factor affecting average-case behaviors of computational problems. There is strong evidence to believe, as hypothesized by Levin (see [Joh84]), that any natural probability distribution used in practice either has a polynomial-time computable distribution function or else is dominated by a probability distribution that does. [1] Such distributions are referred to as p-time computable distributions. Several NP-complete problems have been proved to be solvable in average polynomial time with respect to their underlying natural distributions [Joh84, Wil84, GS87]. On the other hand, there are NP-complete problems under p-time computable distributions which cannot have polynomial time on average algorithms unless every NP-complete problem under any p-time computable distribution has one [Lev86, Gur91, VL88, BG94, VR92, WB, Wan95]. The theory of average-case completeness, initiated by Levin [Lev86], studies how likely hard instances may be in a problem.

Polynomial-time computable distributions are simple and natural. But they may seem somewhat restrictive or not precise enough in some situations as noted in [BCGL92, RS93]. Other types of instance distributions have thus been proposed recently in the theory of average-case completeness from different aspects. Among them are polynomial-time samplable distributions [BCGL92] and polynomially-rankable distributions [RS93]. Polynomial-time samplable distributions are a natural generalization of p-time computable distributions. A distribution $\mu(x)$ is p-samplable if there is a polynomial-time bounded probabilistic Turing machine that starts with no input and outputs x with probability $\mu(x)$. All p-computable distributions are p-samplable. But the inverse is not true if there exists a p-time computable function which is hard to invert on most instances [BCGL92]. P-samplable distributions define a new type of average-case NP-complete problems. But these

*Supported in part by NSF under grant CCR-9396331.

[1] This property holds for all commonly used distributions, e.g., in [JK69, Joh84, Bol85, GS87, Jai91].

problems are no harder than average-case NP-complete problems with p-time computable distributions. In particular, Impagliazzo and Levin [IL90] proved that every NP search problem complete for p-time computable distributions is also complete for all p-samplable distributions. So p-samplable distributions do not generate harder instances than p-time computable distributions. Similar investigation for randomized NP decision problems is currently undertaken by Blass and Gurevich [BG].

We investigate rankable distributions in this paper. Rankable distributions were introduced in [RS93] due to the consideration that all distributions with the same rankability should be treated in the same way. In other words, only the ranking of the inputs by decreasing weights matters. Two distributions μ and ν are said to have the same rank if for all x and y, $\mu(x) \leq \mu(y)$ iff $\nu(x) \leq \nu(y)$. The notion of average polynomial time is now with respect to all distributions with the same ranking. In so doing, a new type of average-case complexity class is defined and a tight hierarchy result is obtained for these classes. A new type of average-case NP-complete problems with respect to polynomial-time rankable (p-rankable, in short) distribution is also constructed. A distribution μ is p-rankable if its ranking function $\lambda x.|\{y : \mu(y) \geq \mu(x)\}|$ is one-to-one, and polynomial-time computable.

There are two major concerns regarding its "naturalness" for any new type of distributions. First, one would like to know whether there are NP-complete problems that can be solved in average polynomial time under distributions in the new type which occur naturally. Second, one would like to know whether the new type of distributions can provide harder instances of a computationally difficult problem than p-time computable distributions on the average case.

P-rankable distributions are not directly comparable to p-time computable distributions or p-samplable distributions. Regarding the first concern, it is not known whether there are natural NP-complete problems which are solvable in average polynomial time with respect to the rank of a practical distribution.

The second concern regarding rankable distributions is whether rankable distributions can provide harder instances than p-time computable distributions. If this were true, then rankable distributions would be useful in studying the average-case hardness of computational problems. However, we show that it is not the case. In particular, we show that if Levin's randomized tiling problem is solvable in polynomial time on average, then every NP problem under any p-rankable distribution is solvable in average polynomial time with respect to rankability. This result holds for both decision and search problems in NP. So p-rankable distributions do not provide harder instances than p-time computable distributions. Randomizing reductions are employed to prove these results.

Finally, we present a reasonably tight hierarchy result for standard average-case complexity classes under p-time computable distributions. In particular, we show that if t_1 and t_2 are time-constructible and $t_1(n \log^\epsilon n) \log t_1(n \log^\epsilon n) = o(t_2(n))$ for some $\epsilon > 0$, then $\mathrm{AvDTime}(t_1(n), \text{P-comp})$ is properly included in $\mathrm{AvDTime}(t_2(n), \text{P-comp})$, where $\mathrm{AvDTime}(t(n), \text{P-comp})$ is the class of randomized decision problems decidable in time t on average with respect to p-time computable distributions.

So the notion of p-time computable distributions is robust in that it provides the hardest instances of computationally difficult problems on the average case compared to p-samplable and p-rankable distributions, and it provides reasonably tight hierarchy results for average-case complexity classes. Moreover, the notion of p-time computable distributions is simple and all the commonly used distributions are p-time computable.

This paper is organized as follows. Some basic definitions and results are reviewed in Section 2. Randomizing reductions with respect to rankability are defined in Section 3. The main theorems are proved in Section 4. Search problems are discussed in Section 5, and the hierarchy results are shown in Section 6. Finally, some open questions are listed in Section 7.

2 Rankable Distributions

We use $\Sigma = \{0,1\}$ as the alphabet for languages and use $|x|$ to denote the length of x. Let A be a set and μ_A be a probability distribution on random instances. An instance x can be positive, meaning $x \in A$, or negative, meaning $x \notin A$. A randomized (or distributional) decision problem is a pair (A, μ_A). If $A \in$ NP, then (A, μ_A) is called a randomized NP decision problem. In the average-case complexity, one may allow an algorithm to run longer time on less frequent inputs with respect to a given distribution μ. So $|x|r(x)$ rather than $|x|$ is used as the size of instance x, where $r(x)$ is a measure of rareness satisfying a randomness test, i.e., its expectation $\sum_x r(x)\mu(x) = O(1)$. A running time (function) $f(x)$ is polynomial on average with respect to μ (in short, polynomial on μ-average) if $f(x) = (|x|r(x))^k$ for some fixed $k > 0$, i.e., $\sum_x \frac{f^{1/k}(x)}{|x|}\mu(x) = O(1)$. A function f is T-average on μ if

$$\sum_x \frac{T^{-1}(f(x))}{|x|}\mu(x) = O(1),$$

where $T^{-1}(n) = \min\{m : T(m) \geq n\}$. This definition is due to Levin [Lev86], which overcomes inappropriate consequences of other more obvious definitions of the concept of polynomial time on average (see [Gur91] for more details).

Probability distributions on instances are important toward learning about average-case completeness. Let μ be a probability distribution (distribution, in short). The (cumulative) distribution function of μ is defined by $\mu^*(x) = \sum_{y \leq x} \mu(y)$, where $\leq$ is the standard lexicographical order on Σ^*. μ^* is p-time computable if there exists a deterministic algorithm $\mathcal{A}$ such that for every string x and every positive integer k, $\mathcal{A}$ outputs a finite binary fraction y with $|\mu^*(x) - y| \leq 2^{-k}$, and the running time of $\mathcal{A}$ is polynomially bounded on $|x|$ and k. A distribution μ is dominated by a distribution ν if $\mu(x) \leq f(x)\nu(x)$ and $f(x) \leq p(|x|)$ for some fixed polynomial p. A more liberal notion of domination is for f to be polynomial on μ-average. Denote by P-comp the class of all probability distributions which have p-time computable distribution function or else is dominated by a probability distribution that does. These probability distributions are called p-time computable distributions for simplicity. Most of the average-case complexity papers are built on p-time computable distributions. Denote by DNP the class of all randomized decision problems (A, μ_A), where $A \in$ NP and $\mu_A \in$ P-comp. A DNP problem is average-case NP-complete if any other DNP problem is reducible to it. So if an average-case NP-complete problem is solvable in average polynomial time, then so is every NP problem under any p-time computable distributions.

In their effort in studying hierarchies of average-case complexity classes, Reischuk and Schindelhauer [RS93] introduced a new type of distributions that provides precise measurement for average-case complexity classes in the sense that all distributions with the same rankability are treated in the same way. In other words, they believed that only the ranking of the inputs by decreasing weights matters. Recall that two distributions μ and ν are said to have the same rankability if for all x and y, $\mu(x) \leq \mu(y)$ iff $\nu(x) \leq \nu(y)$.

Define $\text{rank}_\mu(x)$ to be $|\{z \in \Sigma^* : \mu(z) \geq \mu(x)\}|$. A function f is T-average with respect to ranking function rank_μ if for all real-valued monotone function m with $\sum_x m(\mu(x)) \leq 1$,

$$\sum_x \frac{T^{-1}(f(x))}{|x|} m(\mu(x)) = O(1).$$

This condition depends only on rank_μ and not on μ itself. That is, it depends on all probability distributions that have the same ranking function as μ. f is polynomial on μ-average with respect to rankability if there is a polynomial p such that f is p-average with respect to rank_μ. A randomized NP problem (A, μ_A) is solvable in average polynomial time with respect to rankability if there exists a deterministic Turing machine that computes A in time polynomial on

μ_A-average with respect to rankability.

Polynomially rankable distributions are used to define average-case NP-completeness with respect to rankability [RS93]. Let *p-rankable* denote the set of all probability distributions μ such that rank_μ is one-to-one, and p-time computable. The injectivity of ranking functions provides a unique rank for distributions. By a slight perturbation of the probability distributions, this can always be achieved.

Studying average polynomial time with respect to rankability directly from definition is difficult due to the fact that arbitrary real-valued function m is involved. This obstacle is overcome by the following lemma due to [RS93].

Lemma 1 ([RS93]) *A function f is T-average with respect to ranking function* rank_μ *if and only if*

$$\forall l: \sum_{\mathrm{rank}_\mu(x)\leq l} \frac{T^{-1}(f(x))}{|x|} \leq l.$$

3 Randomizing Reductions

Let μ be a p-rankable distribution. Let $r(x) = |\{y : \mu(y) < \mu(x)\}|$. Then r transforms the distribution of inputs into a monotone distribution on the outputs. Under monotone distributions no sets have probability greater (by a super-polynomial factor) than under uniform distribution. However, while r may be p-time computable, r may not transform an NP problem into an NP problem if r is not p-honest. Also, while any fixed monotone distribution is bounded by the uniform distribution with a super-polynomial factor, the same factor may not work with every monotone distribution, and we have to take care of all of them at the same time.

To prove that p-rankable distributions do not provide harder instances than uniform distributions, we need to construct a hardest problem in (NP, p-rankable) with respect to rankability such that its ranking function is p-honest.[2] Here (NP, p-rankable) is the class of all randomized NP decision problems with p-rankable distributions. A function f is p-honest if there is a polynomial p such that for all x, $p(|f(x)|) \geq |x|$ when $f(x)$ is defined. The idea is to associate coin flips into deterministic reductions.

Deterministic reductions are defined in [RS93] for randomized decision problems with respect to rankable distributions with a restriction that the reductions are required to be injective. It does not pose a real restriction for natural NP-complete problems as they are all complete under injective reductions, so we will follow this restriction in defining reductions with respect to rankability. Notice that a ranking function has small values for likely instances and have large values for rare instances. This is used in defining the notion of domination with respect to rankability. The following definition of reduction is due to [RS93], which is transitive.

(A, μ_1) is p-time reducible to (B, μ_2) with respect to rankability if $A \leq^p_m B$ via a one-to-one reduction f and satisfies the domination property: $r_2(f(x)) \leq p(|x|)r_1(x)$ for some fixed polynomial p, where $r_i(x) = \mathrm{rank}_{\mu_i}(x)$ for $i = 1, 2$.

A randomized NP decision problem is *rankably complete* if its distribution is p-rankable and any other randomized NP decision problem with p-rankable distribution is p-time reducible to it. Rankably complete randomized NP decision problems were constructed in [RS93]. It is easy to see that if a rankably complete randomized NP decision problem is solvable in average polynomial time with respect to rankability, then so is every NP decision problem under any p-rankable distribution.

Notice that in the average case measure with respect to rankability, (X, μ) and (X, rank_μ) denote the same randomized problems, where X is either a language or a binary predicate (for search problems discussed in Section 5).

Randomizing reductions for randomized NP problems with p-time computable distributions were first defined in [VL88] and were further studied in [BG93, BG94], which have been ap-

[2] Note that the ranking functions of the rankably complete problems constructed in [RS93] are not p-honest.

plied successfully to obtain a number of average-case NP complete problems under flat distributions. [3] We will follow the same idea to define randomizing reductions for problems in (NP, p-rankable) with respect to rankability by allowing coin flips in algorithms, called randomizing algorithms.

We assume that a randomizing algorithm does not flip a coin unless the computation requires another random bit. For simplicity, coins are assumed to be unbiased. So a randomizing algorithm on A with probability distribution μ_A can be viewed as a deterministic algorithm on inputs x and a sequence of coin flips s, which form a dilation Δ with probability distribution μ_Δ such that the following conditions are satisfied [Gur91].

1. Δ is a subset of $\Sigma^* \times \Sigma^*$ with the following property. For every x with $\mu_A(x) \neq 0$: $\Delta(x) \neq \emptyset$, and no string in $\Delta(x)$ is a prefix of a different string in $\Delta(x)$, where $\Delta(x) = \{s : (x, s) \in \Delta\}$.

2. For all $(x, s) \in \Delta$, the length of (x, s) is defined as the length of x.

3. For all x and s, $\mu_\Delta(x, s)$ is defined as $\mu_A(x)2^{-|s|}r_\Delta(x)$ if $s \in \Delta(x)$ and 0 otherwise, where $r_\Delta(x) = 1/\Sigma_{t\in\Delta(x)}2^{-|t|}$ is called the rarity function of Δ.

Yet for rankability an extra condition is required to make sure that the ranking of a dilation will solely depend on the ranking of the distribution. Otherwise, different distributions with the same rank may result in dilations with different ranks. This condition is formulated as below. This condition is not needed if one can live with dilations with different ranks generated by different distributions with the same rank.

4. Let $r_A = \text{rank}_A$. If $r_A(x') \leq r_A(x)$ and $(x, s), (x', s') \in \Delta$, then $|s'| \leq |s|$.

(Δ, μ_Δ) is called a dilation of (A, μ_A). We will only need the simplest randomizing algorithms and dilations in this paper, namely, the rarity function of the underlying dilation is always equal to 1. Such a dilation is called an "almost total" dilation.

A randomized decision problem (A, μ_A) is considered solvable efficiently with respect to rankability if there is a randomizing algorithm that decides A in average polynomial time with respect to ranking function rank_{μ_A}.

Definition 1 A randomized decision problem (A, μ_A) is solvable in average polynomial time with respect to rankability if there is an almost total dilation (Δ, μ_Δ) of (A, μ_A) and a deterministic Turing machine on Δ which decides A in average polynomial time with respect to ranking function rank_{μ_Δ}.

Notice that a deterministic algorithm can be thought of as a special case of randomizing algorithm with the dilation containing only the empty string as a coin toss for any input x. A similar notion can be defined for solvability in T-time on average with respect to rankability. More liberal notions of solvability on average with respect to rankability (namely, the rarity function may not always equal to 1) can be similarly defined following [BG93, BG94] and all our results presented in this section are still true.

Definition 2 (A, μ_A) is p-time randomizing reducible to (B, μ_B) with respect to rankability if there is an almost total dilation (Δ, μ_Δ) of (A, μ_A) and a p-time computable, one-to-one function f such that

1. For each $(x, s) \in \Delta$: $x \in A$ iff $f(x, s) \in B$.

2. $\text{rank}_{\mu_B}(f(x, s)) \leq p(|x|) \cdot \text{rank}_{\mu_\Delta}(x, s)$ for some fixed polynomial p.

It can be shown that if (A, μ_A) is p-time randomizing reducible to (B, μ_B) and (B, μ_B) is solvable in average polynomial time with respect to rankability, then so is (A, μ_A). We can similarly define a completeness notion for randomized NP decision problems under randomizing reductions with respect to rankability.

[3] A distribution μ is flat if there exists an $\epsilon > 0$ such that for all x, $\mu(x) \leq 2^{-|x|^\epsilon}$. Randomized NP problems with flat distributions cannot be complete under deterministic many-one reductions unless nondeterministic exponential time collapses to deterministic exponential time. See [Gur91] for more details.

We consider a bounded version of a randomized halting problem with respect to p-rankable distribution, where the ranking function is p-honest.

Let $\mathcal{N} = \{0, 1, 2, ...\}$ be the set of all natural numbers. Let $\langle\cdot,\cdot\rangle$ be a standard pairing function from $\Sigma^* \times \Sigma^*$ to Σ^* in lexicographical order which is both p-time computable and invertible. We can recursively define $\langle\cdot,\cdot,\cdot\rangle$. Let f be a function and we write $f(\cdot,\cdot)$ for $f(\langle\cdot,\cdot\rangle)$. Let β be a standard function that maps all binary strings to all binary numbers in $\mathcal{N}$ in lexicographical order, and β is both linear-time computable and invertible. Let $M_0, M_1, M_2, ...$ be a fixed enumeration of all (deterministic/nondeterministic) Turing machines.

When a ranking function r is defined for a particular problem, we assume that $r(x) = \infty$ for x not being an instance of the problem. Let

$$K = \{\langle i, x, 1^n\rangle : M_i \text{ accepts } x \text{ within } n \text{ steps}\},$$

$$\rho(i, x, 1^n) = \beta(i, x, n).$$

Let

$$K' = \{\langle i, x, w\rangle : M_i \text{ accepts } x \text{ within } |w| \text{ steps}\},$$

$$\rho'(i, x, w) = \beta(i, x, w).$$

It is easy to see that K' is NP-complete and ρ' is p-time computable, p-honest, and p-time invertible. It was shown in [RS93] that (K, ρ) is rankably complete for (NP, p-rankable). But ρ is not p-honest. We will show that (K, ρ) is p-time randomizing reducible to (K', ρ') with respect to rankability. [4]

Theorem 2 (K', ρ') *is rankably complete for* (NP, p-rankable) *under p-time randomizing reductions.*

Proof. Let μ be a p-rankable distribution with ranking function ρ such that $\mu(y) = 0$ if y is not in the form $\langle i, x, 1^n\rangle$, and $\mu(i', x', 1^{n'}) \geq \mu(i, x, 1^n)$ if $i' \leq i$, $x' \leq x$, and $n' \leq n$ while maintaining injectivity.

Clearly, K is nondeterministic linear time computable. Let M be a nondeterministic Turing machine that accepts K in linear time. Let M' be a nondeterministic Turing machine such that M' accepts input z iff there is an $y = \langle i, x, 1^n\rangle$ such that $\rho(y) = \beta(z)$ and M accepts y. It is easy to see that there is a linear polynomial p such that M' accepts z iff there is a computation of M' that accepts z in $p(|y|)$ steps. So $y \in K$ iff M' accepts $\beta^{-1}(r(y))$ in time $p(|y|)$. Let j be an index such that $M_j = M'$.

Define a dilation (Γ, μ_Γ) of (K, μ) by

$$\Gamma = \{(y, s) : \mu(y) > 0 \text{ and } |s| = p(|y|)\}.$$

Clearly, Γ is polynomial-time computable and $\sum_{t\in\Gamma(y)} 2^{-t} = 1$ for all y with $\mu(y) \neq 0$. In particular, condition 4 is satisfied by noticing that $\rho(y') \leq \rho(y)$ iff $y' \leq y$, and $y' \leq y$ implies that $|y'| \leq |y|$, and so $p(|y'|) \leq p(|y|)$.

Define a reduction $f : \Gamma \to K'$ as follows. For all y and s with $\mu(y) > 0$:

$$f(y, s) = \langle j, \beta^{-1}(\rho(y)), s\rangle.$$

It is easy to see that f is one-to-one and polynomial-time computable since both β^{-1} and ρ are one-to-one and polynomial-time computable. Clearly, for all $(y, s) \in \Gamma$: $y \in K$ iff $f(y, s) \in K'$.

Now we check the domination property. Let r_Γ denote the ranking function of μ_Γ, where $\mu_\Gamma(y, s) = \mu(y)2^{-|s|}$ for $(y, s) \in \Gamma$, and 0 otherwise. For $(y, s) \in \Gamma$, write $y = \langle i, x, 1^n\rangle$. Notice that p is a linear polynomial, we have

$$\begin{aligned}
& r_\Gamma(y, s) \\
= \; & |\{(y', s') \in \Gamma : \mu(y')2^{-|s'|} \geq \mu(y)2^{-|s|}\}| \\
> \; & |\{(y', s') : |i'| = |i| - 1, |x'| = |x| - 1, \\
& n' = n, \text{ and } |s'| = p(|\langle i', x', 1^{n'}\rangle|)\}| \\
& (\text{where } y' = \langle i', x', 1^{n'}\rangle) \\
\geq \; & O(2^{|i|+|x|+|s|}) \\
\geq \; & O(\beta(i, x, n)2^{|s|}/|\langle i, x, 1^n\rangle|) \\
= \; & O(\rho(y)2^{|s|}/|y|).
\end{aligned}$$

[4] It can also be shown that $(K', \mu_{K'})$ is average-case NP-complete in Levin's sense under randomizing reductions, where $\mu_{K'}(i, x, w)$ is flat, and is defined as $c \cdot \frac{2^{-(|i|+|x|+|w|)}}{(|i||x||w|)^2}$ for an appropriate constant c.

We know that $\rho'(f(y,s)) = \beta(f(y,s))$. By construction, $\beta(f(y,s)) = \beta(j, \beta^{-1}(\rho(y)), s) = O(\beta(\beta^{-1}(\rho(y))2^{|s|})) = O(\rho(y)2^{|s|}) \leq O(|y| r_{\Gamma}(y,s))$. This completes the proof. ∎

4 Rankable Instances Are Not Harder

We prove in this section that rankable distributions do not provide harder instances than uniform distributions for NP decision problems.

The standard uniform probability distribution μ is given by $\mu(x) = \frac{2^{-|x|}}{|x|(|x|+1)}$ or $\mu(x) = \frac{6}{\pi|x|^2}2^{-|x|}$, although this is often replaced by $\mu(x) = \frac{c}{|x|^k}2^{-|x|}$ for some $k > 1$ and appropriate c, or even $\mu(x) = \frac{c}{|x|\log^{1+\epsilon}|x|}2^{-|x|}$ for some $\epsilon > 0$, where $\log^e n$ denotes $(\log n)^e$. For notational convenience, we simply use $\frac{2^{-|x|}}{|x|^2}$ as the default uniform probability distribution of binary strings.

We first show that there is p-rankable distribution μ and an NP-complete set S such that (S, μ) is average-case NP-complete with respect to rankability, where the ranking function r of μ satisfies $|r(x)|^3 \leq |x|$ and r is p-honest.

Lemma 3 *There is an* NP*-complete set S and a p-rankable distribution μ such that (K', ρ') is p-time reducible to (S, μ) with respect to rankability, where the ranking function r of μ satisfies $|r(x)|^3 \leq |x|$, and r is p-honest. Moreover, r is one-to-one.*

Proof. We use an easy fact that K' and SAT are p-isomorphic [BH77], meaning that there is a p-time computable and invertible bijection f such that K' is reducible to SAT via f. Pad the boolean formula generated by $f(i,x,w)$ such that the length of it is greater than the cubic root of the length of $\rho'(i,x,w)$. Let g denote this new reduction, which is one-to-one, p-time computable, and p-time invertible. Let $S = g(K')$ and define r as follows: For all instances F of S (positive or negative), let $r(F) = \rho'(i,x,w)$, where $g^{-1}(F) = \langle i,x,w \rangle$. This completes the proof. ∎

Lemma 4 *Let (S, μ) and r be from Lemma 3. Let $L = r(S)$ and $\nu(y)$ be the uniform distribution $2^{-|y|}/|y|^2$. If (L, ν) can be solved in T time on ν-average, then (S, μ) from Lemma 3 can be solved in average $O(T + p)$ time with respect to rankability for some polynomial p.*

Proof. Assume that (L, ν) can be solved in time T on ν-average. This means that L can be solved by a deterministic algorithm with running time t and the following is satisfied:

$$\sum_y \frac{(T)^{-1}(t(y))}{|y|}\nu(y) = O(1).$$

Let

$$M = \sum_y \frac{(T)^{-1}(t(y))}{|y|}\nu(y).$$

So S can be solved by a deterministic algorithm with running time $(t \circ r) + p$, where p is a polynomial time bound for computing r. We will show that $t \circ r$ is T-average with respect to r. For any natural number ℓ, let $R_\ell = \{y : y = r(x) \leq \ell\}$. We know that $2^{|r(x)|} \leq r(x) < 2 \cdot 2^{|r(x)|}$. We get

$$\begin{aligned} M &\geq \sum_{y \in R_\ell} \frac{(T)^{-1}(t(y))}{|y|}\nu(y) \\ &= \sum_{y \in R_\ell} \frac{(T)^{-1}(t(y))}{2^{|y|}|y|^3} \\ &= \sum_{r(x) \leq \ell} \frac{(T')^{-1}(t(r(x)))}{2^{|r(x)|}|r(x)|^3} \\ &\geq \sum_{r(x) \leq \ell} \frac{(T)^{-1}(t \circ r(x))}{r(x)|x|} \\ &\geq \sum_{r(x) \leq \ell} \frac{(T)^{-1}(t \circ r(x))}{\ell|x|}. \end{aligned}$$

So $t \circ r$ is $M \cdot T$-average with respect to r from Lemma 1. This completes the proof. ∎

As a corollary, we get

Corollary 5 *If an average-case* NP*-complete decision problem can be solved in T time on average, then (S, μ) from Lemma 3 can be solved in average $O(T \circ q)$ time with respect to rankability for some polynomial q.*

Proof. Let r be from Lemma 3 and (L, ν) be from Lemma 4. Since r is p-time computable and p-honest, $L \in$ NP. So $(L, \nu) \in$ DNP. If a DNP-complete problem (in Levin's sense), say the tiling problem [Lev86], is solvable in average T-time, then (L, ν) is solvable in $O(T \circ q)$ time on ν-average for some polynomial q. We can also assume that q bounds the running time for computing r. This completes the proof by Lemma 4. ∎

We therefore obtain the following theorem.

Theorem 6 *If the randomized Tiling problem can be solved in average polynomial time, then any* NP *decision problem under any p-rankable distribution is solvable in average polynomial time with respect to rankability.*

5 Search Problems

We can similarly define randomized NP search problems with respect to rankability. An NP search problem is specified by a p-time computable binary predicate R. For a given input x, the search problem is to find w (a witness) such that $|w|$ is polynomially bounded and $R(x, w)$ is satisfied. A randomized NP search problem is a pair (R, μ) of a p-time computable binary predicate and a probability distribution. A randomized NP search problem (R, μ) is solvable in average polynomial time with respect to rankability if there is a deterministic Turing machine that solves the search problem in time T that is p-average with respect to rank_μ for some polynomial p.

We define a reduction for randomized search problems with respect to rankability following the one for randomized decision problems.

Definition 3 (R_1, μ_1) is p-time reducible to (R_2, μ_2) with respect to rankability if there is a pair of p-time computable functions f and g with the following conditions. Let $D_i = \{x : \mu_i(x) > 0\}$, $Y_i = \{x : x \in D_i \text{ and } \exists w{:}R_i(x, w)\}$, $r_i(x) = \text{rank}_{\mu_i}(x)$, $i = 1, 2$. Function f is one-to-one and maps D_1 to D_2.

1. (The solvability) For any $x \in D_1$: $x \in Y_1$ iff $f(x) \in Y_2$
2. (The witnesses) For any $x \in D_1$ and any w: if $R_2(f(x), w)$, then $R_1(x, g(w))$.
3. (The domination) For any $x \in D_1$: $r_2(f(x)) \le p(|x|) r_1(x)$ for some fixed polynomial p.

A randomized NP search problem is *rankably complete* if its probability distribution is p-rankable and any other NP search problem under any p-rankable distribution is p-time reducible to it with rankability. Clearly, if a rankably complete NP search problem is solvable in average polynomial time with rankability, then so is every NP search problem under any p-rankable distribution.

We can similarly define the search version of the randomizing reductions with respect to rankability following Definitions 3 and 2.

Let $R_{K'}$ be a binary predicate for K': $R_{K'}(\langle i, x, w\rangle, z)$ is true iff $\langle i, x, w\rangle \in K'$ and z is a computation path witnessing that M_i accepts x in $|w|$ steps.

Theorem 7 $(R_{K'}, \rho')$ *is complete for randomized* NP *search problems with p-rankable distributions under p-time randomizing reductions with respect to rankability.*

Theorem 6 is also true for randomized NP search problems. We leave the proof to the reader.

Theorem 8 *If an average-case* NP*-complete search problem can be solved in average polynomial time with respect to p-time distributions, then any* NP *search problem under any p-rankable distribution is solvable in average polynomial time with respect to rankability.*

6 Average-case Hierarchies

The study of hierarchies among complexity classes is a fruitful area in complexity theory, yet surprisingly little has been done to investigate hierarchies among average-case complexity classes. Studying natural distributions that can provide hard instances of problems and finding suitable reductions to identify more naturally occurred NP problems to be average-case complete have been the major concerns in the theory of average-case complexity. Nevertheless, there is an interest to investigate hierarchy results among interesting average-case complexity classes.

Let t be a time-constructible function. We assume in this section that a Turing machine will read all of its input before accepting or rejecting. So, if t is a time bound for a Turing machine, it must be the case that $t(n) \geq n+1$ for all n. We will then follow Hopcroft and Ullman [HU79] and assume that a time-complexity function t is implicitly replaced by $\max\{n+1, t(n)\}$. Denote by $\mathrm{AvDTime}(t(n))$ the class of randomized decision problems which can be decided by a Turing machine whose running time is t on average (in Levin's sense).

It can be seen that a problem which requires n^2 time to solve on inputs of length n will be, with the uniform distribution, in $\mathrm{AvDTime}(n^{1+\epsilon})$ for every $\epsilon > 0$. This may seem not precise enough and to prevent this from happening, Reischuk and Schindelhauer [RS93] proposed the notions of rankable distributions and average time with respect to rankability as defined in Section 2. Although they were able to establish a rather tight hierarchy for their average time complexity classes with respect to rankability, there is a doubt that the definition of average time with respect to rankability is a natural one as shown in previous sections. Thus, it would be much more desirable to have hierarchy results using the standard notions.

It is known that under the universal distribution, the average-case complexity of a problem is the same as the worst-case complexity [LV92],[5] and so if no restrictions are put on the distributions, any hierarchy results for $\mathrm{DTime}(t(n))$ apply to $\mathrm{AvDTime}(t(n))$. Ben-David, Chor, Goldreich and Luby [BCGL92] have a similar result, using a non-standard definition of worst-case complexity. However, the distributions used in these results require super-polynomial time to compute, and we would like to restrict ourselves to distributions which can be computed in polynomial time. With this restriction, we obtain the following hierarchy result.

Theorem 9 *Let t and T be time-constructible functions such that*

$$\lim_{n\to\infty} \frac{t(n\log^\epsilon n)\log t(n\log^\epsilon n)}{T(n)} = 0$$

for some $\epsilon > 0$. Then there is a randomized decision problem $(L,\mu) \in \mathrm{AvDTime}(T(n)) - \mathrm{AvDTime}(t(n))$, where μ is a uniform distribution.

Proof. Let t and T be as above. We immediately have $\mathrm{AvDTime}(t(n))$ is included in $\mathrm{AvDTime}(T(n))$. Define U by $U(n) = t(n\log^\epsilon n)$, so

$$\lim_{n\to\infty} \frac{U(n)\log U(n)}{T(n)} = 0.$$

It was shown by Goldmann, Grape and Håstad [GGH94] that there exists a language L in $\mathrm{DTIME}(T(n))$ such that if T_M is the running time of a Turing machine M which decides L, then for sufficiently large n, say $n \geq N$, $T_M(x) \geq U(|x|) = t(|x|\log^\epsilon |x|)$ for a constant fraction c_M of instances x of length n. For these x, we have $t^{-1}(T_M(x)) \geq |x|\log^\epsilon |x|$. Letting $\mu(x) = \frac{c}{|x|\log^{1+\epsilon}|x|}2^{-|x|}$ for the appropriate c, we get

$$\begin{aligned}
& \sum_x \frac{t^{-1}(T_M(x))}{|x|}\mu(x) \\
= & \sum_x \frac{t^{-1}(T_M(x))}{|x|}\frac{c}{|x|\log^{1+\epsilon}|x|}2^{-|x|} \\
\geq & \sum_{|x|\geq N} \frac{t^{-1}(T_M(x))}{|x|}\frac{c}{|x|\log^{1+\epsilon}|x|}2^{-|x|} \\
\geq & \sum_{n=N}^{\infty} \frac{c_M 2^n n\log^\epsilon n}{n}\frac{c}{n\log^{1+\epsilon} n}2^{-n} \\
= & \sum_{n=N}^{\infty} \frac{cc_M}{n\log n},
\end{aligned}$$

[5] Actually, this is shown for a different notion of average time, but it will imply it for our notion of average time for a large class of time-complexity functions.

which diverges.

So, (L, μ) cannot be in AvDTime($t(n)$). Since L is in DTIME($T(n)$), (L, μ) will be in AvDTime($T(n)$), and so AvDTime($t(n)$) is properly included in AvDTime($T(n)$). ∎

It is often useful to restrict our attention to a smaller class of distributions in order to obtain completeness results (e.g., see [WB93]). Let t be a time-constructible function and $\mathcal{F}$ a class of distributions. Then AvDTime($t(n)$,$\mathcal{F}$) is the class of languages which can be solved in average t time with respect to a distribution in $\mathcal{F}$.

Similar to the proof of Theorem 9, we can show the following hierarchy result.

Theorem 10 *Let $\mathcal{F}$ be a class of distributions containing $\mu(x) = \frac{c}{|x|^k} 2^{-|x|}$ for some k and suitable c, and let t and T be time-constructible functions such that*

$$\lim_{n \to \infty} \frac{t(n^k) \log t(n^k)}{T(n)} = 0.$$

Then AvDTime($t(n), \mathcal{F}$) *is properly included in* AvDTime($T(n), \mathcal{F}$).

There is a weaker result shown in [SY92]. Under the same assumption of t and $\mathcal{F}$ as in Theorem 10, Schuler and Yamakami [SY92] were able to show that if $T(n) = \omega(t(4cn^{k+1}) + n)$, then AvDTime($t(n), \mathcal{F}$) is properly included in AvDTime($T(n) \log T(n), \mathcal{F}$).

Our result presented here is tighter. As a corollary, we get

Corollary 11 *Let t and T be time-constructible functions such that, for some $\epsilon > 0$,*

$$\lim_{n \to \infty} \frac{t(n \log^\epsilon n) \log t(n \log^\epsilon n)}{T(n)} = 0.$$

Then AvDTime($t(n)$, P-comp) *is properly included in* AvDTime($T(n)$, P-comp).

7 Open Questions

We list some open questions here for future study.

1. Does there exist a natural NP-complete problem D (or a natural problem not in P) and a practical distribution μ on D such that (D, μ) is solvable in average polynomial time with respect to rankability?

There are many questions regarding average-case hierarchies which remain unanswered. We list some of them below.

2. Is it possible to get hierarchy results for AvDTime($t(n)$) that are as good or better than the known results for DTIME($t(n)$) under p-time computable distributions?

3. What is the best possible hierarchy for AvDTime($t(n)$, P-comp)?

We can similarly define average-case complexity classes for non-deterministic time as well as space.

4. What hierarchy results are possible for these classes?

Acknowledgement. The second author thanks Leonid Levin for his constructive comments.

References

[BCGL92] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average case complexity. *J. Comp. Sys. Sci.*, 44:193–219, 1992.

[BG] A. Blass and Y. Gurevich. Randomizing reductions of decision problems (tentative title). Personal communication.

[BG93] A. Blass and Y. Gurevich. Randomizing reductions of search problems. *SIAM J. Comput.*, 22:949–975, 1993.

[BG94] A. Blass and Y. Gurevich. Matrix decomposition is complete for the average case. *SIAM J. Comput.*, 1994. to appear.

[BH77] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.*, 6:305–321, 1977.

[Bol85] B. Bollobás. *Random Graphs.* Academic Press, 1985.

[GGH94] M. Goldmann, P. Grape, and J. Hastad. On average time hierarchies. *Inf. Proc. Lett.*, 49:15–20, 1994.

[GS87] Y. Gurevich and S. Shelah. Expected computation time for hamiltonian path problem. *SIAM J. Comput.*, 16:486–502, 1987.

[Gur91] Y. Gurevich. Average case completeness. *J. Comp. Sys. Sci.*, 42:346–398, 1991.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison–Wesley, Reading, MA, 1979.

[IL90] R. Impagliazzo and L. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proc. 31st FOCS*, pages 812–821, 1990.

[Jai91] R. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[JK69] N. Johnson and S. Kotz. *Distributions in Statistics–Discrete Distributions.* John Wiley & Sons, 1969.

[Joh84] D. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 5:284–299, 1984.

[Lev86] L. Levin. Average case complete problems. *SIAM J. Comput.*, 15:285–286, 1986.

[LV92] M. Li and P. Vitányi. Average case complexity under the universal distribution equals worst-case complexity. *Inf. Proc. Lett.*, 42:145–149, 1992.

[RS93] R. Reischuk and C. Schindelhauer. Precise average case complexity. In *Proc. 10th Annual Symposium on Theoretical Aspects of Computer Science*, volume 665 of *Lect. Notes in Comp. Sci.*, pages 650–661. Springer Verlag, 1993.

[SY92] R. Schuler and T. Yamakami. Structural average case complexity. In *Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 652 of *Lect. Notes in Comp. Sci.*, pages 128–139. Springer Verlag, 1992.

[VL88] R. Venkatesan and L. Levin. Random instances of a graph coloring problem are hard. In *Proc. 20th STOC*, pages 217–222, 1988.

[VR92] R. Venkatesan and S. Rajagopalan. Average case intractability of diophantine and matrix problems. In *Proc. of STOC*, pages 632–642, 1992.

[Wan95] J. Wang. Average-case completeness of a word problem for groups. In *Proc. of STOC*, 1995. To appear.

[WB93] J. Wang and J. Belanger. On average-P vs. average-NP.
In K. Ambos-Spies, S. Homer, and U. Schönings, editors, *Complexity Theory—Current Research*, pages 47–67. Cambridge University Press, 1993.

[WB] J. Wang and J. Belanger. On the NP-isomorphism problem with respect to random instances. *J. Comp. Sys. Sci.*. To appear.

[Wil84] H. Wilf. An O(1) expected time algorithm for the graph coloring problem. *Inf. Proc. Lett.*, 18:119–122, 1984.