Chicago Journal of Theoretical Computer Science

The MIT Press

Volume 1999, Article 2
24 February 1999

ISSN 1073–0486. MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142-1493 USA; (617)253-2889; *journals-orders@mit.edu, journals-info@mit.edu*. Published one article at a time in LaTeX source form on the Internet. Pagination varies from copy to copy. For more information and other articles see:

- *http://mitpress.mit.edu/CJTCS/*
- *http://www.cs.uchicago.edu/publications/cjtcs/*
- *ftp://mitpress.mit.edu/pub/CJTCS*
- *ftp://cs.uchicago.edu/pub/publications/cjtcs*

The *Chicago Journal of Theoretical Computer Science* is abstracted or indexed in *Research Alert,*® *SciSearch,*® *Current Contents*®*/Engineering Computing & Technology*, and *CompuMath Citation Index.*®

©1999 The Massachusetts Institute of Technology. Subscribers are licensed to use journal articles in a variety of ways, limited only as required to ensure fair attribution to authors and the journal, and to prohibit use in a competing commercial product. See the journal's World Wide Web site for further details. Address inquiries to the Subsidiary Rights Manager, MIT Press Journals; (617)253-2864; *journals-rights@mit.edu*.

The *Chicago Journal of Theoretical Computer Science* is a peer-reviewed scholarly journal in theoretical computer science. The journal is committed to providing a forum for significant results on theoretical aspects of all topics in computer science.

Editor-in-Chief: Janos Simon

Consulting Editors: Joseph Halpern, Stuart A. Kurtz, Raimund Seidel

Editors:

Martin Abadi	Greg Frederickson	John Mitchell
Pankaj Agarwal	Andrew Goldberg	Ketan Mulmuley
Eric Allender	Georg Gottlob	Gil Neiger
Tetsuo Asano	Vassos Hadzilacos	David Peleg
Laszló Babai	Juris Hartmanis	Andrew Pitts
Eric Bach	Maurice Herlihy	James Royer
Stephen Brookes	Ted Herman	Alan Selman
Jin-Yi Cai	Stephen Homer	Nir Shavit
Anne Condon	Neil Immerman	Eva Tardos
Cynthia Dwork	Howard Karloff	Sam Toueg
David Eppstein	Philip Klein	Moshe Vardi
Ronald Fagin	Phokion Kolaitis	Jennifer Welch
Lance Fortnow	Stephen Mahaney	Pierre Wolper
Steven Fortune	Michael Merritt	

Managing Editor: Michael J. O'Donnell

Electronic Mail: chicago-journal@cs.uchicago.edu

This article is included in the *Special Issue on Computational Complexity* containing articles based on selected presentations made at the Dagstuhl-Seminar *Structure and Complexity*, held September 30—October 4, 1996, at Schloß Dagstuhl, Germany. This special issue was edited by Eric Allender.

Randomized Reductions and Isomorphisms

Jie Wang

24 February, 1999

Abstract

Randomizing reductions have provided new techniques for tackling average-case complexity problems. For example, although some NP-complete problems with uniform distributions on instances cannot be complete under deterministic one-one reductions [WB95], they are complete under randomized reductions [VL88]. We study randomized reductions in this paper on reductions that are one-one and honest mappings over certain input domains. These are reasonable assumptions since all the randomized reductions in the literature that are used in proving average-case completeness results possess this property. We consider whether randomized reductions can be inverted efficiently. This gives rise to the issue of randomized isomorphisms. By generalizing the notion of isomorphisms under deterministic reductions, we define what it means for two distributional problems to be isomorphic under randomized reductions. We then show a randomized version of the Cantor-Bernstein-Myhill theorem, which provides a sufficient condition for two distributional problems to be isomorphic under randomized reductions. Based on that condition we show that all the known average-case NP-complete problems (including those that are complete under deterministic reductions) are indeed isomorphic to each other under randomized reductions.

1 Introduction

1-1 Average-case complexity has attracted increasing attention recently. A major objective of this study is to identify (standard, worst-case) NP-complete problems that are difficult, on average, from those that are easy, on average.

Various notions of reductions for distributional problems are useful tools for studying average-case complexity. A distributional problem is a pair consisting of a problem A and a probability distribution μ on instances of A. If A is an NP problem, then we call (A, μ) a distributional NP problem. Deterministic reductions are simpler and easier to use; they have been used to prove that some NP-complete problems under plausible distributions are average-case complete. However, deterministic reductions have certain limitations. For example, under the assumption that EXP $\neq$ NEXP, distributional NP problems with flat distributions cannot be complete under deterministic reductions [Gur91]; and (without any assumption) distributional NP problems with flat distributions cannot be complete under deterministic one-one reductions [WB95]. Randomized reductions were introduced to overcome these limitations [VL88]. Some NP problems with flat distributions have been shown to be complete under randomized reductions [VL88, Gur91, BG95]. Randomized reductions have also been used to obtain several other interesting results. For example, using randomized reductions, Impagliazzo and Levin [IL90] showed that any NP search problem under any polynomial-time samplable distribution is reducible to some NP search problem under a uniform distribution. This result is also true for NP decision problems under randomized truth-table reductions [BDCGL92, Wan97]. Also, using randomized reductions, Belanger and Wang [BW97] showed that no NP problems over ranking of distributions are harder than over uniform distributions.

1-2 Gurevich [Gur91] and Blass and Gurevich [BG93, BG95] have conducted intensive studies on randomized reductions. Through their work, we have gained deeper understandings on randomized reductions. We carry on this study a step further by considering reductions that are one-one and honest mappings over certain input domains. (A mapping f is honest if for any input x, the length of its image $f(x)$ cannot be too short; namely, there is a polynomial p such that for any input x, $p(|f(x)|) \geq |x|$.) These are reasonable assumptions since all the randomized reductions in the literature that are used in proving average-case completeness results possess this property. Moreover, we consider whether these reductions can be "inverted" efficiently. This gives rise naturally to the issue of randomized isomorphisms of average-case NP-complete problems.

1-3 Berman and Hartmanis [BH77] were the first to study isomorphisms of complete sets for resource-bounded worst-case complexity classes. They defined a notion of isomorphism under deterministic polynomial-time reductions. They showed that all the known NP-complete sets are polynomially

isomorphic, and they conjectured that it should also be true for *all* NP-complete sets. This conjecture implies that P is different from NP. Although this conjecture has not been solved, it has stimulated a host of papers in structural complexity theory. Wang and Belanger [WB95] observed that some standard NP-complete problems, while isomorphic on the worst case, are not isomorphic on the average case under commonly used distributions. On the other hand, all the average-case NP-complete problems may still be isomorphic. To extend the isomorphism theory of Berman and Hartmanis from worst case to average case, Wang and Belanger [WB95] defined a notion of isomorphisms for distributional problems under deterministic reductions — a natural generalization of Berman and Hartmanis' notion of isomorphisms. They showed that all the known average-case NP-complete problems that are complete under deterministic reductions are indeed isomorphic.

1-4 We consider whether the notion of isomorphisms can be further extended to include randomized reductions, and we devote this paper to answering this question. We provide an affirmative answer: By generalizing the notion of isomorphisms under deterministic reductions, we define in a natural way what it means for two distributional problems to be isomorphic under randomized reductions. We then show a randomized version of the Cantor-Bernstein-Myhill theorem, which provides a sufficient condition for two distributional problems to be isomorphic under randomized reductions. Based on that condition we show that all the known average-case NP-complete problems (including those that are complete under deterministic reductions) are indeed isomorphic to each other under randomized reductions.

2 Average Time and Instance Distributions

2-1 We provide in this section basic definitions and results of average-case complexity. For a recent survey of average-case complexity, its motivations, traps and escapes, the reader is referred to [Gur91, Wan97].

2-2 Let $\Sigma = \{0, 1\}$. We assume that all languages are subsets of Σ^*. Let μ denote a probability distribution (distribution, in short) over Σ^*; i.e., μ is a real-valued function from Σ^* to [0,1] such that $\sum_x \mu(x) = 1$. The probability distributions we consider are on instances of computational problems. If a binary string does not encode an instance of the underlying problem, then that string has zero probability. The *distribution function* of μ, denoted by μ^*, is defined by $\mu^*(x) = \sum_{y \leq x} \mu(y)$, where $\leq$ is the standard lexicographical

order on Σ^*. For a function f, we use $f^\varepsilon(x)$ to denote $(f(x))^\varepsilon$ for $\varepsilon > 0$ and we use f^{-1} to denote the inverse of f. We use $\mathbb{N}$ to denote the natural numbers. We will consider decision problems in this paper; we will omit the word "decision" when there is no confusion.

2-3 Levin [Lev86] suggested to use *multi-median* time to measure average computation time, and he gave the following definition.

Definition 1 ([Lev86]) *A function $f : \Sigma^+ \to \mathbb{N}$ is* polynomial on μ-average *if there is an $\varepsilon > 0$ such that $\sum_x f^\varepsilon(x)|x|^{-1}\mu(x) < \infty$.*

2-4 This notion of average time is robust and machine independent.

2-5 Let AP denote the class of all distributional problems (A, μ) such that A can be solved by a deterministic algorithm whose running time is polynomial on μ-average. AP is an average-case analogue of P.

2-6 In what follows, we will use "p-time" to denote "polynomial-time," and "ap-time" to denote "average-polynomial-time."

2-7 Let μ and ν be two distributions; then μ is *dominated by* ν, denoted by $\mu \preceq^p \nu$, if there is a polynomial p such that for all x, $\mu(x) \leq p(|x|)\nu(x)$.

2-8 A real-valued function $r : \Sigma^* \to [0, 1]$ 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 such that $|r(x) - y| \leq 2^{-k}$ and the running time of $\mathcal{A}$ is polynomially bounded in $|x|$ and k [Ko83]. Clearly, if μ^* is p-time computable, then so is μ; but Blass showed that the converse is not true unless P = NP (see [Gur91]). With this fact in mind, we assume throughout that when we say that μ is p-time computable, both μ and μ^* are p-time computable.

2-9 Levin (see [Joh84]) hypothesized that any natural distribution μ is either p-time computable or is dominated by a distribution that is. Strong evidence that supports this hypothesis is the fact that all the commonly used discrete distributions do satisfy this hypothesis. It is often natural to consider uniform distributions. We say that a distribution μ is *uniform* if μ is p-time computable and, for all x, $\mu(x) = \rho(|x|)2^{-|x|}$, where $\sum_n \rho(n) = 1$ and there exists a polynomial p such that for all but finitely many n, $\rho(n) \geq 1/p(n)$. Levin [Lev86] used n^{-2} for $\rho(n)$ for notational convenience (normalized by dividing by $\sum_n n^{-2} = \pi^2/6$), and $|x|^{-2}2^{-|x|}$ is often referred to as the *default* uniform distribution. If an instance X consists of several parameters $(x_1, x_2, \ldots, x_k)$, then by uniform distribution of X we mean that each parameter x_i is selected independently and uniformly with respect to the default uniform distribution on x_i.

2-10 Let DistNP denote the class of distributional problems (A, μ) such that $A \in$ NP and μ is either p-time computable or is dominated by a distribution that is. By Levin's hypothesis, DistNP includes all natural distributional NP problems. DistNP is a distributional analogue of NP.

3 Deterministic and Randomized Reductions

3-1 Several NP-complete problems under uniform distributions have been shown to be in AP (e.g., see [Joh84, GS87]). To find out whether there are complete problems for DistNP, Levin [Lev86] defined and used the notion of polynomial-time many–one reducibility. Gurevich [Gur91] conducted a thorough investigation on this notion.

3.1 Deterministic Reductions

3-1.1 Let f be a function from Σ^* to Σ^*. Write $f(\nu)(y)$ to denote $\sum_{f(x)=y} \nu(x)$. Then f induces a distribution $f(\nu)$ on Σ^* for the outputs of f. We say that μ is *dominated by ν with respect to f*, denoted by $\mu \preceq_f^p \nu$, if there exists a distribution μ_1 such that μ is dominated by μ_1 and, for all $y \in \text{range}(f)$, $\nu(y) = f(\mu_1)(y)$.

Definition 2 ([Lev86, Gur91]) *Let (A, μ_A) and (B, μ_B) be two distributional problems. Then (A, μ_A) is* p-time many–one reducible to *(B, μ_B), denoted by $(A, \mu_A) \leq_m^p (B, \mu_B)$, if there exists a p-time computable reduction f such that A is many–one reducible to B via f and $\mu_A \preceq_f^p \mu_B$.*

3.1-2 Polynomial-time many–one reductions have the desired properties [Gur91]: If $(A, \mu_A) \leq_m^p (B, \mu_B)$ and $(B, \mu_B) \in$ AP, then $(A, \mu_A) \in$ AP; polynomial-time many–one reductions are transitive.

3.1-3 The requirements of p-time many–one reductions can be weakened in two ways without losing the two desired properties [Lev86, Gur91].

3.1-4 First, we may require only that the reduction be computable in time polynomial on μ_A-average.

3.1-5 Second, we may use the following weaker domination condition. Distribution μ is *weakly dominated by* μ_1, denoted by $\mu \preceq^{ap} \mu_1$, if for all x, $\mu(x) \leq h(x)\mu_1(x)$, where h is polynomial on μ-average. Distribution μ is

weakly dominated by ν with respect to f, written as $\mu \preceq_f^{ap} \nu$, if there exists a distribution μ_1 such that μ is weakly dominated by μ_1 and $\nu(y) = f(\mu_1)(y)$ for all $y \in \text{range}(f)$. Reductions defined under these two weaker requirements are called *ap-time many–one reductions*.

3.1-6 A distributional problem is *many–one complete* for DistNP (or simply average-case NP-complete) if it is in DistNP and every other problem in DistNP is $\leq_m^p$-reducible to it. Several distributional NP problems under uniform distributions have been shown to be many–one complete for DistNP [Lev86, Gur91, WB95, Wan95].

3.2 Randomized Reductions

3.2-1 The notion of many-one reductions has certain limitations. In particular, it is not suitable for studying completeness of problems when "flat" distributions are presented. A distribution μ is *flat* [Gur91] if there exists an $\epsilon > 0$ such that for all x, $\mu(x) \leq 2^{-|x|^\epsilon}$. Uniform distributions for graph problems often turn out to be flat [Gur91, WB95]. Gurevich [Gur87, Gur91] showed that, unless nondeterministic exponential time collapses to deterministic exponential time, no distributional problems with flat distributions can be complete under ap-time many–one reductions. Wang and Belanger [WB95] further showed that, without any assumption, a distributional problem with a flat distribution cannot be complete under ap-time, one-one, and p-honest reductions.

3.2-2 We explain why deterministic reductions would fail: When instances of the target problem are under a flat distribution, each instance of approximately the same length will have approximately the same weight, and this weight is too small to dominate the input distribution. So no matter how an instance of the target problem is selected by an honest deterministic reduction, the domination property will always be violated.

3.2-3 Venkatesan and Levin [VL88] showed that by using randomized reductions, one can overcome the difficulties caused by flat distributions. A randomized reduction is a standard randomized (probabilistic) algorithm that transforms one string to another. In [VL88], the purpose of using a randomized reduction is to generate instances of the target problem with sufficiently large probability. The idea is to supply a random source S_x for a reduction f from (A, μ_A) to (B, μ_B) on each instance x of A, such that different random strings $s \in S_x$ will produce different instances $f(x, s)$, and for all $s \in S_x$, $x \in A$ if and only if $f(x, s) \in B$. Thus, although for each individual

instance $f(x,s)$ of B, $\mu_B(f(x,s))$ may be too small to dominate $\mu_A(x)$, the summation $\sum_s \mu_B(f(x,s))$ may be large enough to meet the domination requirement for distributions. To make this point clearer, let us consider the following special case. Let $S = \{\, (x,s) : s \in S_x \,\}$ and for all $s \in S_x$, $|s| \geq |x|$. Assume that the reduction f is one-one on S and, for all x, $\sum_{s\in S_x} 2^{-|s|} = 1$. For a particular $s \in S_x$, $\mu_B(f(x,s))$ may be too small to dominate $\mu_A(x)$, but $\sum_{s\in S_x} \mu_B(f(x,s)) = \mu_B(f(x,s))2^{|s|}$ may well be large enough to dominate $\mu_A(x)$. Thus, if $\mu_A(x)2^{-|s|}$ is dominated by $\mu_B(f(x,s))$, then we have $\mu_A(x) = \sum_{s\in S_x} \mu_A(x)2^{-|s|}$, which is dominated by $\sum_{s\in S_x} \mu_B(f(x,s))$, a property we desire.

3.2-4 Following [VL88, BG93, BG95], we formulate the notion of randomized reductions as follows. For simplicity, we assume that only unbiased random coins are used in randomizations. Let $\mathcal{A}$ be a randomized algorithm and x be an input. We are only interested in sequences s of random bits such that $\mathcal{A}(x)$ halts using s. Such a random sequence must be finite.

3.2-5 We allow a randomized algorithm (for solving a problem) to produce incorrect outputs on some sequences of random bits. For example, suppose that a randomized algorithm $\mathcal{A}$ computes a reduction from A to B, and suppose that $\mathcal{A}$ on input x, with s being the random sequence, produces an output string y. Then the output y is incorrect if $x \in A$ but $y \notin B$. If $\mathcal{A}$ halts on input x with random sequence s and produces a correct output, then (x,s) is called a *good* input of $\mathcal{A}$.

3.2-6 From now on, we will be interested only in good inputs. Let $\mathcal{A}$ be a randomized algorithm and μ an input distribution of $\mathcal{A}$. A *good-input domain* of $\mathcal{A}$ (with respect to μ) is the set of all pairs (x,s) such that $\mu(x) > 0$, $\mathcal{A}$ on input x halts and produces a correct output, and s is the random sequence it generates during the computation. Clearly, $\mathcal{A}$ is deterministic on (x,s) and we may view (x,s) as the input of $\mathcal{A}$. If $\mathcal{A}$ computes a function f, then we can view f as a deterministic function on (x,s). If $\mathcal{A}$ is deterministic on x, then (x,e) is in the good-input domain of $\mathcal{A}$, where e denotes the empty string.

3.2-7 Let Γ be a good-input domain of $\mathcal{A}$. Let $\Gamma(x) = \{\, s \colon (x,s) \in \Gamma \,\}$. We can see that no string in $\Gamma(x)$ is a prefix of a different string in $\Gamma(x)$; otherwise, the longer string cannot be in $\Gamma(x)$ as the algorithm stops before it can be generated. We say that $\Gamma(x)$ is *non-rare* [BG93] if the rarity function of Γ, defined by $U_\Gamma(x) = 1/\sum_{s\in\Gamma(x)} 2^{-|s|}$ if $\mu(x) > 0$ and $U_\Gamma(x) = 1$ otherwise, is

polynomial on μ-average.[1] A good-input domain Γ is called *non-rare* if for all x, $\Gamma(x)$ is non-rare.

3.2-8 For all $(x, s) \in \Gamma$, let

$$\mu_\Gamma(x, s) = \mu(x)2^{-|s|}U_\Gamma(x).$$

Here the factor $U_\Gamma(x)$ is used for normalization (see Definition 2.8 in [BG95] on page 955).

Definition 3 ([BG93, BG95]) *Let $\mathcal{A}$ be a randomized algorithm with input distribution μ. Then $\mathcal{A}$ runs in time* polynomial on μ-average *if there is a good-input domain Γ of $\mathcal{A}$ and an $\varepsilon > 0$ such that Γ is non-rare, and*

$$\sum_{(x,s)\in\Gamma} t^\varepsilon(x, s)|x|^{-1}\mu_\Gamma(x, s) < \infty,$$

where $t(x, s)$ is the running time of $\mathcal{A}$ on input x with random sequence s.

3.2-9 For simplicity, when there is no confusion about the input distribution μ, we call a randomized algorithm that runs in time polynomial on μ-average a *randomized ap-time algorithm*.

3.2-10 The probability that a randomized ap-time algorithm $\mathcal{A}$ on input x produces a correct output is $1/U_\Gamma(x)$, a value that could be small. Blass and Gurevich [BG93, BG95] showed that the algorithm can be iterated to obtain a correct solution with probability 1 in ap-time, provided that the correct output can be verified in ap-time. For the purpose of iteration, we would like to require that the good-input domain of the algorithm be decidable in ap-time with respect to μ_Γ. In [BG95], such a good-input domain is called *certifiable*.

3.2-11 Let (A, μ) be a distributional problem. Let $\mathcal{D}_A$ be the set of all (positive and negative) instances of A. When (A, μ) is solvable by a randomized algorithm, we may view a good-input domain of the algorithm as a "randomized" input domain of (A, μ). Sometimes we would like to refer to a good-input domain without specifying a randomized algorithm, and we will call it a *randomized input domain of* (A, μ).

3.2-12 Let RAP be the class of distributional problems (A, μ) for which there is a randomized ap-time algorithm $\mathcal{A}$ with a good-input domain Γ such that

[1] If for all x, $U_\Gamma(x) = 1$, then the randomized algorithm produces a correct output with probability 1. For our purpose, we only need to require that the value of $U_\Gamma(x)$ be "reasonable" in the sense that U_Γ is polynomial on μ-average.

Γ is non-rare, certifiable, and for all $(x, s) \in \Gamma$, $x \in A$ if and only if $\mathcal{A}(x, s)$ accepts. RAP is an average-case analogue of ZPP. [2]

Definition 4-1 **Definition 4 ([BG95])** *Let (A, μ_A) and (B, μ_B) be distributional problems. Then (A, μ_A) is* ap-time randomly reducible to *(B, μ_B), denoted by $(A, \mu_A) \leq_r^{ap} (B, \mu_B)$, if there is a reduction f such that the following conditions are satisfied.*

1. *The reduction f is computable by a randomized algorithm in time polynomial on μ_A-average with a good-input domain Γ_A.*
2. *Γ_A is non-rare and certifiable.*
3. *For all $(x, s) \in \Gamma_A$, $x \in A$ if and only if $f(x, s) \in B$.*
4. *$\mu_{\Gamma_A} \preceq_f^{ap} \mu_B$.*

Definition 4-2 *If the reduction f can be computed by a randomized p-time algorithm, then f is called a* randomized p-time reduction.

3.2-13 Clearly, deterministic reductions are a special case of randomized reductions with good inputs (x, e), where e is the empty string.

3.2-14 The following lemma is straightforward.

Lemma 1 *If $f : \Gamma_A \to B$ is one-one over Γ_A, then $\mu_{\Gamma_A} \preceq_f^{ap} \mu_B$ if and only if $\mu_{\Gamma_A} \preceq^{ap} \mu_B \circ f$.*

3.2-15 We will often use the phrase "via (f, Γ_A)" to emphasize the randomized reduction f and its non-rare good-input domain Γ_A.

Lemma 2 ([BG95]) *(1) If $(A, \mu_A) \leq_r^{ap} (B, \mu_B)$ and (B, μ_B) is in* RAP*, then so is (A, μ_A). (2) Randomized ap-time reductions are transitive.*

[2] Clearly, for every set $A \in$ ZPP and every distribution μ, $(A, \mu) \in$ RAP. We might as well use AZPP to denote RAP. We may also define ABPP as analogous to BPP, and use ABPP as a notion of *easiness*.

4 Randomized Isomorphisms

4-1 Berman and Hartmanis defined two problems A and B to be p-isomorphic if there exists a p-time computable and invertible bijection $\phi : \mathcal{D}_A \to \mathcal{D}_B$ such that $A \leq_m^p B$ via ϕ and $B \leq_m^p A$ via ϕ^{-1}. For distributional problems (A, μ_A) and (B, μ_B), Wang and Belanger [WB95] defined that (A, μ_A) and (B, μ_B) are p-isomorphic if there exists a p-time computable and invertible bijection $\phi : \mathcal{D}_A \to \mathcal{D}_B$ such that $(A, \mu_A) \leq_m^p (B, \mu_B)$ via ϕ and $(B, \mu_B) \leq_m^p (A, \mu_A)$ via ϕ^{-1}. Polynomial isomorphisms on distributional problems are transitive.

4-2 Let μ and ν be two distributions. Write $\mu \approx^p \nu$ if μ is dominated by ν and ν is dominated by μ. Wang and Belanger [WB95] showed the following p-time equivalent of the Cantor-Bernstein-Myhill theorem for distributional problems.

Theorem 1 ([WB95]) *Let $(A, \mu_A) \leq_m^p (B, \mu_B)$ via f, and $(B, \mu_B) \leq_m^p (A, \mu_A)$ via g, then (A, μ_A) is p-isomorphic to (B, μ_B) if both f and g are one-one, length-increasing, p-time computable, and p-time invertible, and $\mu_A \approx^p \mu_B \circ f$ (here $\mu_B \circ f(x) = \mu_B(f(x))$) and $\mu_B \approx^p \mu_A \circ g$.*

4-3 Note that if f is one-one, then $\mu \preceq_f^p \nu$ if and only if $\mu \preceq \nu \circ f$ [Gur91]. Using Theorem 1, Wang and Belanger [WB95] showed that all the known distributional NP problems that are complete under p-time many–one reductions are p-isomorphic.

4-4 Let π_1 and π_2 be functions defined on tuples of strings such that π_1 returns the first element and π_2 returns the second element.

4-5 Next, we consider how to generalize the notion of isomorphisms to the setting of randomized reductions. Let (A, μ_A) and (B, μ_B) be two distributional problems. Recall that the notion of isomorphisms under deterministic reductions is defined to be a bijection between input domains of the underlying problems. Following this framework, we consider bijections Ψ between a randomized input domain of (A, μ_A) and a randomized input domain of (B, μ_B). Note that a randomized reduction takes a random string as input and does not output one. This leads us to consider $\pi_1 \circ \Psi$ and $\pi_1 \circ \Psi^{-1}$ as possible reductions; namely, we would like to transform (encode) each instance of A to a unique instance of B via $\pi_1 \circ \Psi$, and transform each instance of B back to a unique instance of A via $\pi_1 \circ \Psi^{-1}$. This gives rise naturally to the following definition of isomorphism under randomized reductions.

Definition 5 (A, μ_A) *is* randomly isomorphic to (B, μ_B), *denoted by* $(A, \mu_A) \equiv_r (B, \mu_B)$, *if the following conditions hold.*

1. *There exist randomized input domains* Γ_A *of* (A, μ_A) *and* Γ_B *of* (B, μ_B)*, and a bijection* Ψ *between* Γ_A *and* Γ_B *such that* Ψ *is ap-time computable on* Γ_A *and* Ψ^{-1} *is ap-time computable on* Γ_B*.*

4-6

2. *Let* $f = \pi_1 \circ \Psi$ *and* $g = \pi_1 \circ \Psi^{-1}$*. Then* $(A, \mu_A) \leq_r^{ap} (B, \mu_B)$ *via* (f, Γ_A)*, and* $(B, \mu_B) \leq_r^{ap} (A, \mu_A)$ *via* (g, Γ_B)*.*

4-7 Clearly, randomized isomorphisms are transitive.

4-8 Next, we show a randomized ap-time equivalent of the Cantor-Bernstein-Myhill theorem for distributional problems as a natural generalization of the p-time equivalent of the Cantor-Bernstein-Myhill theorem.

4-9 A (partial) function $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ (or $f : \Sigma^* \times \Sigma^* \to \Sigma^* \times \Sigma^*$) is said to be *length-increasing* if $|f(x, y)| > |x| + |y|$ whenever $f(x, y)$ is defined.

4-10 Let (A, μ_A) and (B, μ_B) be two distributional problems. Assume that $(A, \mu_A) \leq_r^{ap} (B, \mu_B)$ via (f, Γ_A) and $(B, \mu_B) \leq_r^{ap} (A, \mu_A)$ via (g, Γ_B). In view of Theorem 1, we would first require that f and g be one-one, length-increasing, and invertible. We then would like μ_A and μ_B to have certain relations under f and g. If f and g are deterministic many-one reductions, we have required, as stated in Theorem 1, that $\mu_A \approx^p \mu_B \circ f$ and $\mu_B \approx^p \mu_A \circ g$. What is needed there is actually the following two requirements: $\mu_B \circ f \preceq^p \mu_A$ and $\mu_A \circ g \preceq^p \mu_B$; the other parts, namely, $\mu_A \preceq^p \mu_B \circ f$ and $\mu_B \preceq^p \mu_A \circ g$, are guaranteed by the one-one reductions. In the setting of randomization, these two requirements may be written as follows: (1) for all $(x, s) \in \Gamma_B$ and for all $s' \in \Gamma_A(g(x, s))$: $\mu_{\Gamma_A}(g(x, s), s') \preceq^{ap} \mu_B(x)$; (2) for all $(x, s) \in \Gamma_A$ and for all $s' \in \Gamma_B(f(x, s))$: $\mu_{\Gamma_B}(f(x, s), s') \preceq^{ap} \mu_A(x)$. We can obtain, as definitions, equivalents of these two statements without using s and s'. Since these two statements are symmetric, we present the equivalent of the first statement below. (The equivalent of the second statement can be obtained similarly.)

4-11 Assume that $(A, \mu_A) \leq_r^{ap} (B, \mu_B)$ via (f, Γ_A). For all $x \in \mathcal{D}_A$, let $l_A(x) = \min \{ |s| : s \in \Gamma_A(x) \}$. Then μ_{Γ_A} *on* g *is weakly dominated by* μ_B if for all $x \in \text{range}(g)$, $\mu_A(x) 2^{-l_A(x)} \preceq^{ap} \mu_B(\pi_1(g^{-1}(x)))$.

4-12 To prove Theorem 2, we would also like to acquire the following property.

Definition 6 *Let* Γ *be a randomized input domain of* (A, μ_A)*. Then* Γ *is* selectable *if there is a p-time computable function* $\xi : \mathcal{D}_A \to \Sigma^*$ *such that for every* $x \in \mathcal{D}_A$*,* $(x, \xi(x)) \in \Gamma$*. The function* ξ *is called a* selection function of Γ*.*

Theorem 2 *Suppose* $(A, \mu_A) \leq_r^{ap} (B, \mu_B)$ *via* (f, Γ_A) *and* $(B, \mu_B) \leq_r^{ap} (A, \mu_A)$ *via* (g, Γ_B). *Then* $(A, \mu_A) \equiv_r (B, \mu_B)$ *if the following conditions hold.*

1. *Both* Γ_A *and* Γ_B *are certifiable and selectable.*
2. *Both* $f : \Gamma_A \to B$ *and* $g : \Gamma_B \to A$ *are one-one, length-increasing, and ap-time invertible.*
3. *The distribution* μ_{Γ_A} *on* g *is weakly dominated by* μ_B, *and* μ_{Γ_B} *on* f *is weakly dominated by* μ_A.

Proof of Theorem 2-1 **Proof of Theorem 2** By assumption of the reductions, for all $(x, s) \in \Gamma_A$, $f(x, s)$ is defined; and, for all $(x, s) \in \Gamma_B$, $g(x, s)$ is defined. Without loss of generality, assume that, for $(x, s) \notin \Gamma_A$, $f(x, s)$ is not defined, and for $(x, s) \notin \Gamma_B$, $g(x, s)$ is not defined. Hence, f is a deterministic function on input domain Γ_A, and g is a deterministic function on input domain Γ_B.

Proof of Theorem 2-2 By assumption, both Γ_A and Γ_B are selectable. Let ξ_A and ξ_B be the p-time computable selection functions of Γ_A and Γ_B, respectively.

Proof of Theorem 2-3 Define functions $F : \Gamma_A \to \Gamma_B$ and $G : \Gamma_B \to \Gamma_A$ as follows.

$$\forall (x, s) \in \Gamma_A : F(x, s) = (f(x, s), \xi_B(f(x, s))) \quad (1)$$

$$\forall (x, s) \in \Gamma_B : G(x, s) = (g(x, s), \xi_A(g(x, s))) \quad (2)$$

Proof of Theorem 2-4 Since both f and g are one-one and length-increasing, both F and G are one-one and ap-time computable.

Proof of Theorem 2-5 We have

$$F^{-1}(x, \xi_B(x)) = (\pi_1(f^{-1}(x)), \pi_2(f^{-1}(x)))$$
$$G^{-1}(x, \xi_A(x)) = (\pi_1(g^{-1}(x)), \pi_2(g^{-1}(x))).$$

Proof of Theorem 2-6 It is easy to see that both F^{-1} and G^{-1} are ap-time computable. They are also length-decreasing; namely, $|F^{-1}(x, s)| < |x| + |s|$ and $|G^{-1}(x, s)| < |x| + |s|$. Hence, the sets R_1, R_2, S_1, and S_2 defined below are all ap-time decidable.

$$R_1 = \left\{ (G \circ F)^k(x, s) : k \geq 0, (x, s) \notin G(\Gamma_B), \text{ and } (x, s) \in \Gamma_A \text{ if } k = 0 \right\},$$
$$R_2 = \left\{ G \circ (F \circ G)^k(x, s) : k \geq 0, \text{ and } (x, s) \notin F(\Gamma_A) \right\},$$
$$S_1 = \left\{ (F \circ G)^k(x, s) : k \geq 0, (x, s) \notin F(\Gamma_A), \text{ and } (x, s) \in \Gamma_B \text{ if } k = 0 \right\},$$
$$S_2 = \left\{ F \circ (G \circ F)^k(x, s) : k \geq 0, \text{ and } (x, s) \notin G(\Gamma_B) \right\}.$$

Proof of Theorem 2-7 Following the proof given in [BH77], we can show that $\Gamma_A = R_1 \cup R_2$, $R_1 \cap R_2 = \emptyset$, and $\Gamma_B = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$. Also, for every $(x,s) \in \Gamma_A$, $(x,s) \in R_1$ if and only if $F(x,s) \in S_2$, and $(x,s) \in R_2$ if and only if $F(x,s) \in S_1$. Similarly, for every $(x,s) \in \Gamma_B$, $(x,s) \in S_1$ if and only if $G(x,s) \in R_2$, and $(x,s) \in S_2$ if and only if $G(x,s) \in R_1$.

Proof of Theorem 2-8 Let, for all $(x,s) \in \Gamma_A$,

$$\Psi(x,s) = \begin{cases} F(x,s), & \text{if } (x,s) \in R_1, \\ G^{-1}(x,s), & \text{if } (x,s) \in R_2. \end{cases}$$

Then the inverse of Ψ is given by, for all $(x,s) \in \Gamma_B$,

$$\Psi^{-1}(x,s) = \begin{cases} G(x,s), & \text{if } (x,s) \in S_1, \\ F^{-1}(x,s), & \text{if } (x,s) \in S_2. \end{cases}$$

The function Ψ is the desired bijection between Γ_A and Γ_B.

Proof of Theorem 2-9 Next, we show that $(A,\mu_A) \leq_r^{ap} (B,\mu_B)$ via $(\pi_1 \circ \Psi, \Gamma_A)$. (We can similarly show that $(B,\mu_B) \leq_r^{ap} (A,\mu_A)$ via $(\pi_1 \circ \Psi^{-1}, \Gamma_B)$.)

Proof of Theorem 2-10 Let $(x,s) \in \Gamma_A$.

Proof of Theorem 2-11 *Case 1:* $(x,s) \in R_1$. Then $\pi_1 \circ \Psi(x,s) = f(x,s)$. Since by assumption, $(A,\mu_A) \leq_r^{ap} (B,\mu_B)$ via (f,Γ_A), we have $x \in A$ if and only if $f(x,s) \in B$ and $\mu_{\Gamma_A} \preceq_f^{ap} \mu_B$. Since f is one-one over Γ_A, it follows from Lemma 1 that $\mu_{\Gamma_A} \preceq^{ap} \mu_B \circ f$. By the definition of F, we have $\pi_1 \circ \Psi(x,s) = f(x,s)$. Thus, we have $x \in A$ if and only if $\pi_1 \circ \Psi(x,s) \in B$, and $\mu_{\Gamma_A} \preceq^{ap} \mu_B \circ (\pi_1 \circ \Psi)$. Since $\pi_1 \circ \Psi$ is one-one, we have that μ_{Γ_A} is weakly dominated by μ_B with respect to $\pi_1 \circ \Psi$.

Proof of Theorem 2-12 *Case 2:* $(x,s) \in R_2$. Then $g^{-1}(x)$ is defined and it follows from Equation (2) that $\pi_1 \circ \Psi(x,s) = \pi_1 \circ G^{-1}(x,s) = \pi_1(g^{-1}(x))$. Since $g^{-1}(x)$ is defined, $(\pi_1 \circ g^{-1}(x), \pi_2 \circ g^{-1}(x)) \in \Gamma_B$. So we have $\pi_1 \circ g^{-1}(x) \in B$ if and only if $g(\pi_1 \circ g^{-1}(x), \pi_2 \circ g^{-1}(x)) \in A$. Since $g(\pi_1 \circ g^{-1}(x), \pi_2 \circ g^{-1}(x)) = x$, this implies that $x \in A$ if and only if $\pi_1 \circ \Psi(x,s) \in B$. By Condition 3 in the statement of the theorem, $\mu_A(x) 2^{-l_A} \preceq^{ap} \mu_B(\pi_1(g^{-1}(x)))$, where $l_A = \min\{\, |s| : s \in \Gamma_A(x) \,\}$. Since $|s| \geq l_A$, we have $\mu_{\Gamma_A}(x,s) = \mu_A(x) 2^{-|s|} \leq \mu_A(x) 2^{-l_A}$, which implies that μ_{Γ_A} is weakly dominated by $\mu_B \circ (\pi_1 \circ \Psi)$. Again, since $\pi_1 \circ \Psi$ is one-one, we have that μ_{Γ_A} is weakly dominated by μ_B with respect to $\pi_1 \circ \Psi$. This completes the proof.

Proof of Theorem 2 □

4-13 Built on rich structures of complete sets, Berman and Hartmanis [BH77] obtained a sufficient condition under which reductions are guaranteed to be p-time invertible. In particular, let A be a set for which two p-time computable functions $S_A(\cdot,\cdot)$ and $D_A(\cdot)$ exist with the following properties: (1) $(\forall x, y)[S_A(x,y) \in A$ if and only if $x \in A]$, and (2) $(\forall x, y)[D_A(S_A(x,y)) = y]$. Then if f is any p-time reduction of some set C to A, the map $f'(x) = S_A(f(x), x)$ is one-one and invertible in p-time and reduces C to A. An easy proof shows that all the known, standard NP-complete sets do satisfy these two properties [BH77]. It follows from the p-time equivalent of the Cantor-Bernstein-Myhill theorem for (worst-case) decision problems that all the known standard NP-complete problems are p-isomorphic.

4-14 For distributional problems, however, the reductions need to be easily invertible *and* preserve the probability distributions. Unfortunately, no functions such as S_A and D_A above are known that preserve the probabilities in a useful way. For $w \in A$, we would need the probability of $D_A(w)$ to depend on the probability function associated to the arbitrary set C, and hence to an undetermined probability function. So to use Theorem 2, we will investigate individual reduction used in each completeness proof. Fortunately, this task can often be accomplished by using the existing completeness proofs, with some minor modifications if necessary. Using Theorem 2, we can show that all the known average-case NP-complete problems are indeed randomly isomorphic.

5 Isomorphism Proofs

5-1 Levin [Lev86] (see also [Gur91]) showed that any p-time computable distribution on string x is dominated by a uniform distribution on strings whose length is polynomially related to $|x|$. Based on that, Wang and Belanger proved the following Distribution Controlling Lemma [WB95].

Lemma 3 (Distribution Controlling Lemma) *Let μ be a p-time computable distribution. If there exists a polynomial p such that for all x, $\mu(x) > 2^{-p(|x|)}$, then there is a total, one-one, p-time computable and p-time invertible function $\alpha\colon \Sigma^* \to \Sigma^*$ such that for all x, $4 \cdot 2^{-|\alpha(x)|} \leq \mu(x) < 20 \cdot 2^{-|\alpha(x)|}$.*

5-2 We first use distributional halting problems as an example of obtaining an isomorphism proof.

5.1 Distributional Halting

5.1-1 Let $M_1, M_2, \ldots$ be a fixed enumeration of nondeterministic Turing machines in which the index i is a binary integer that codes up the symbols, states, and transition table of the i-th Turing machine M_i.

5.1-2 Distributional Halting (Version 1)
Instance. Binary strings i, x, and a unary notation 1^n representing positive integer n, where i is a positive integer in binary form.
5.1-3 *Question.* Does M_i accept x within n steps?
5.1-4 *Distribution.* Uniform; namely, the distribution on instance $(i, x, 1^n)$ is proportional to $2^{-(l+m)}l^{-2}m^{-2}n^{-2}$, where $l = |i|$ and $m = |x|$.

5.1-5 Distributional Halting (Version 2)
Instance. Binary strings i, x, and t, where i is a positive integer.
5.1-6 *Question.* Does M_i accept x within $|t|$ steps?
5.1-7 *Distribution.* Uniform; namely, the distribution on instance (i, x, t) is proportional to $2^{-(l+m+n)}l^{-2}m^{-2}n^{-2}$, where $l = |i|$, $m = |x|$, and $n = |t|$.

5.1-8 Let (K, μ_K) and $(K', \mu_{K'})$ denote the halting problem version 1 and version 2, respectively. (K, μ_K) is is complete for DistNP under p-time many–one reductions [Gur91, BDCGL92, WB95]. $(K', \mu_{K'})$ (note that $\mu_{K'}$ is flat) is complete for DistNP under randomized p-time reductions [Gur91, Wan97].

Theorem 3 $(K, \mu_K) \equiv_r (K', \mu_{K'})$.

Proof of Theorem 3-1 **Proof of Theorem 3** Let $\Gamma_K = \{\, (y, s) : y = (i, x, 1^n)$ and $|s| = n \,\}$. Then Γ_K is certifiable, non-rare (the rarity function $U_\Gamma(x) = 1$), and selectable. For a given string $y = (i, x, 1^n)$, let h be a function that pads the program i such that h is p-time computable, p-time invertible, $|h(i)| > |i| + O(1)$, and $M_{h(i)} = M_i$. Let $h(i) = j$; then $|h^{-1}(j)| = |j| - O(1)$. For all $(y, s) \in \Gamma_K$, let $f(y, s) = (h(\pi_1(y)), \pi_2(y), s)$. Then f is one-one, length-increasing, p-time computable, and p-time invertible. Hence, for all $(y, s) \in \Gamma_K$, we have $y \in K$ if and only if $(\pi_1(y), \pi_2(y), s) \in K'$ if and only if $(h(\pi_1(y)), \pi_2(y), s) \in K'$. This implies that $y \in K$ if and only if $f(y, s) \in K'$. By definition, $\mu_{\Gamma_K}(y, s) = \mu_K(y)2^{-|s|}$, which is proportional to $\mu_{K'}(f(y, s))$. Hence, $\mu_{\Gamma_K}(y, s)$ is dominated by $\mu_{K'}(f(y, s))$. Thus, $(K, \mu_K) \leq_r^{ap} (K', \mu_{K'})$ via (f, Γ_K).

Proof of Theorem 3-2 Next, we show that $(K', \mu_{K'})$ is polynomially reducible to (K, μ_K) via a length-increasing, one-one, deterministic reduction. We can see that

$(K', \mu_{K'}) \leq_m^p (K, \mu_K)$ by a reduction that maps (i, x, s) to $(h(i), x, 1^{|s|})$. But this reduction is not one-one and so cannot be used. We will construct a different reduction using a standard technique as shown in [WB95]. Note that $\mu_{K'}$ satisfies the hypothesis of the Distribution Controlling Lemma. So there is a total, one-one, p-time computable, and p-time invertible function α such that for all $y \in \mathcal{D}_{K'}$, $4 \cdot 2^{-|\alpha(y)|} < \mu_{K'}(y) < 20 \cdot 2^{-|\alpha(y)|}$. Let M be a nondeterministic Turing machine M that accepts K' in polynomial time. Define a Turing machine M' as follows: On binary input w, if $\alpha^{-1}(w)$ is defined, then M' simulates M on $\alpha^{-1}(w)$ and rejects otherwise. So for all $y \in \mathcal{D}_{K'}$, M accepts y if and only if M' accepts $\alpha(y)$. It is easy to see that M' on input $\alpha(y)$ is bounded in polynomial time, and we call it $p(|y|)$.

Proof of Theorem 3-3 Let i be a program such that $M' = M_i$. Let $g(y) = (i, \alpha(y), 1^{p(|y|)})$. Then g is one-one, length-increasing, p-time computable, and p-time invertible. By construction, $y \in K'$ if and only if $g(y) \in K$. Moreover, $\mu_K \circ g(y) \approx^p 2^{-|\alpha(y)|} \approx^p \mu_{K'}(y)$. Thus, we have $(K', \mu_{K'}) \leq_m^p (K, \mu_K)$ via g.

Proof of Theorem 3-4 Conditions 1 and 2 of Theorem 2 regarding $\Gamma_{K'}$ are obviously satisfied. To complete this direction, we need only show that Condition 3 of Theorem 2 holds. For this purpose, we view function g as a function defined on $\Gamma_{K'} = \{\, (y, e) : y \in \mathcal{D}_{K'} \,\}$ by $g(y, e) = g(y)$. Let $z \in \text{range}(g)$. Then $z = g(y)$ for some $y \in \mathcal{D}_{K'}$. Hence, we have $\mu_K(z) \approx^p \mu_{K'} \circ (g^{-1}(z))$. Let $l_K = \min\{\, |s| : s \in \Gamma_K(z) \,\}$. Then $\mu_K(z) 2^{-l_K} < \mu_K(z)$. This implies that $\mu_K(z) 2^{-l_K}$ is dominated by $\mu_{K'}(\pi_1(g^{-1}(z)))$. The first statement of Condition 3 of Theorem 2 is therefore satisfied. We now show that the second statement of Condition 3 is also satisfied. Let $y = (i, x, s) \in \text{range}(f)$. Then $f^{-1}(y) = (y', s)$, where $y' = (h^{-1}(i), x, 1^{|s|})$. Hence, $\pi_1(f^{-1}(y)) = (h^{-1}(i), x, 1^{|s|})$. Since $|h^{-1}(i)| = |i| - O(1)$, this implies that $\mu_{\Gamma_{K'}}(y)$ is dominated by $\mu_K(\pi_1(f^{-1}(y)))$.

Proof of Theorem 3-5 We have therefore verified that all the conditions of Theorem 2 are satisfied, and so $(K, \mu_K) \equiv_r (K', \mu_{K'})$.

Proof of Theorem 3 □

5.2 Distributional Tiling and Graph Spot Coloring

5.2-1 A tile is a square with a symbol on each corner. Tiles may not be rotated or turned over. We assume that there are infinitely many copies of each type of tile. By a tiling of an $n \times n$ square we mean an arrangement of n^2 tiles to cover the entire square so that the symbols on the touching corners of

adjacent tiles are the same.

5.2-2 DISTRIBUTIONAL TILING

Instance. A finite set of tiles T, a unary notation 1^n for a positive integer n, and a sequence $S = s_1 s_2 \dots s_k$ ($k \leq n$) of tiles that match each other; that is, the symbols on the touching corners of s_i and s_{i+1} are the same.

5.2-3 *Question*. Can S be extended to tile an $n \times n$ square using tiles from T?

5.2-4 *Distribution*. Uniform; namely, the distribution is proportional to $\Pr[T]n^{-2}\Pr[S]$, where $\Pr[T]$ is the probability of choosing T,[3] and $\Pr[S]$ is the probability of choosing S. S is chosen by first choosing k at random with probability $1/n$, then choosing the first tile s_1 at random from T, and choosing the s_i ($i > 1$) sequentially and uniformly at random from those tiles in T that match s_{i-1}.

5.2-5 Levin [Lev86] showed that the distributional tiling problem is average-case NP-complete under a deterministic reduction. Gurevich [Gur91] provided a detailed proof for tiles with marked edges, where each edge of a tile is marked with a symbol; in the tiling of a square, symbols on the touching edge of adjacent tiles are the same. Belanger and Wang [BW93, WB95] presented a simpler proof. Distributional tiling with marked corners and with marked edges are polynomially isomorphic.

5.2-6 We are interested in the following variant of distributional tiling on tiles with marked corners. First, the set of tiles T is fixed; so T is no longer a component of an instance. Second, in the S component of an instance, the two corners at the same side of a tile have the same binary digit. The first tile in S has a special symbol on its lower-left corner. It is easy to see that there exists T, denote it by $\mathcal{T}$, such that this variant of distributional tiling is p-isomorphic to the standard distributional tiling. (For example, we can construct $\mathcal{T}$ based on a fixed Turing machine that accepts K.) Denote by $(\mathcal{T}, \mu_{\mathcal{T}})$ this average-case NP-complete variant.

5.2-7 Venkatesan and Levin [VL88] studied the following edge-coloring problem for directed graphs (digraphs, in short), where nodes are labeled and may have self-loops. For convenience, we assume that self-loops with single direction and self-loops with double directions are distinct. (This assumption can be used to simplify proofs.) Let G be such a digraph. An edge of G may be colored or left blank (i.e., uncolored), with a constant number of colors. A *spot* in a colored digraph is a 3-node subgraph with induced colored edges

[3] One can use one's favorite distribution to select T or simply select it uniformly at random among binary strings, since T is coded in binary.

(including self-loops if there are any) and the nodes unlabeled. It is easy to see that there are only a constant number of different spots. The *coloration* $C(G)$ of G consists of the set of all spots induced from the colored graph G and the number of blank edges.

5.2-8 Write $f(n) \sim g(n)$ if $\lim_{n\to\infty} f(n)/g(n) = 1$.

5.2-9 Distributional Graph Spot Coloring

Instance. A digraph G of n nodes, a set C of spots, and a unary notation 1^k for a positive integer k, where $k < \binom{n}{2}$.

5.2-10 *Question*. Can G be colored such that $C(G) = (C', k)$ and $C' \subseteq C$? (If so, we then say that G is colorable.)

5.2-11 *Distribution*. Uniform: First choose a positive integer n with probability $\Theta(n^{-2})$, then randomly and independently choose a directed graph of n nodes with uniform probability $4^{-\binom{n}{2}}3^{-n} \sim 2^{-n^2}$, a set of spots with probability $\Theta(1)$, the number of blank edges k with probability $\Theta(n^{-2})$. Hence, the probability distribution is proportional to $n^{-4}2^{-n^2}$, which is flat.

5.2-12 Let (E, μ_E) denote the distributional graph spot-coloring problem. Venkatesan and Levin [VL88] (a slightly different proof was given in [Ven91]) showed that (E, μ_E) is average-case NP-complete under a randomized reduction.[4]

Theorem 4 $(\mathcal{T}, \mu_{\mathcal{T}}) \equiv_r (E, \mu_E)$.

Proof of Theorem 4-1 **Proof of Theorem 4** For a large part, we follow a proof given in [VL88], showing that distributional graph spot-coloring is average-case NP-complete. To arrive quickly to the point showing why a randomized isomorphism exists, we will omit proofs to certain lemmas; all of the missing proofs can be found in [VL88, Ven91].

Proof of Theorem 4-2 A *tournament* is an acyclic (except for self-loops) complete digraph. Any tournament contains a Hamiltonian path; there is a deterministic p-time algorithm that starts from the node with smallest label and finds a Hamiltonian path uniquely (see, e.g., [Liu68]). Let $k = |T|$. Let $t_1, t_2, \ldots, t_k$ be a Hamiltonian path of a tournament T found in this way. Define the code for a node u to be the binary string of length $2|T|$ of the form

[4] What would be the minimum number of colors that can make the distributional graph spot coloring problem complete for DistNP? That is an interesting question. It was claimed that 20 colors [VL88], or even much fewer colors [Ven91], are sufficient.

$c(t_1,u)c(u,t_1)\ldots c(t_k,u)c(u,t_k)$, where $c(u,v)=1$ if $u \to v$ and 0 otherwise [BES80]. So T determines the code for u uniquely.

Proof of Theorem 4-3 We now construct a randomized reduction f from $(\mathcal{T},\mu_{\mathcal{T}})$ to (E,μ_E). Let $X=(1^n,S)$ ($|S|<n$) be an instance of $(\mathcal{T},\mu_{\mathcal{T}})$.

Proof of Theorem 4-4 The reduction f first partitions $\{1,\ldots,2n^2+k-1\}$ at random into disjoint sets $T,\ L,\ U$, with $|T|=k=\lceil 2\log n\rceil+1$, $|L|=n^2$, and then randomly adds edges to generate a random graph of $2n^2+k-1$ nodes. Let $r(n)$ be the number of random bits required in this random process. (In [VL88], it indicates that $r(n)=5n^4$ is sufficient.) Since there are $2^{(2n^2+k-1)^2}$ many graphs and we need extra random bits to generate a partition, $r(n)>(2n^2+k-1)^2$. Clearly, a partition and a graph can be uniquely constructed in time polynomial of n when $r(n)$ random bits are given.

Proof of Theorem 4-5 Among the random graphs so generated, we are particularly interested in the ones that satisfy the following conditions.

1. T is a tournament.
2. T is the set of all nodes with double-direction self-loops and L is the set of all nodes with single-direction self-loops. (Self-loops are used to enforce the graph structure and the color pattern.)
3. Every node in $L\cup U$ is connected to every node in T.
4. All unlooped nodes (i.e., nodes in U) have distinct codes with respect to T.
5. Let $v_1,v_2,\ldots,v_{|U|}$ be the nodes in U in the decreasing order of codes with respect to T. If the symbol of the right side of the i-th tile of S is 0, then $v_i \to v_{i+1}$ is the only edge between v_i and v_{i+1}; if the symbol is 1, then $v_i \leftarrow v_{i+1}$ is the only edge. If $i \geq |S|$, then there are no edges between v_i and v_{i+1}.

Proof of Theorem 4-6 Note that the last condition above gives an encoding of S in the graph. Also, in [VL88], it was required that T be the unique k-node tournament. In our construction, all nodes in T are with double-direction self-loops, and no other nodes have such self-loops. This makes T unique. Let G denote the resultant graph; we call it a *tiling graph*. Venkatesan and Levin [VL88] showed that there are sufficiently many tiling graphs.

Lemma 4 ([VL88]) *The probability that G is a tiling graph is at least $\Omega(n^{-d})$ for some constant $d>1$.*

Proof of Theorem 4-7 Let $\Gamma_{\mathcal{T}}(X) = \{\, s : |s| = r(n)$ and s produces a tiling graph G from $X \,\}$. Then $\Gamma_{\mathcal{T}} = \{\, (X, s) : s \in \Gamma_{\mathcal{T}}(X) \,\}$ is a good-input domain of f. It follows from Lemma 4 that $U_{\Gamma_{\mathcal{T}}}(X) = \Omega(n^{-d})$ and so $\Gamma_{\mathcal{T}}$ is non-rare. Also, given a random sequence s, it is easy (in p-time) to check whether the graph generated by s is a tiling graph.

Proof of Theorem 4-8 Based on the structures of G, Venkatesan and Levin [VL88] (see also [Ven91]) constructed a color specification (C, b) in time polynomial of n, where C is a finite set of spots and $b = O(\sqrt{n})$ such that the following lemma holds true.

Lemma 5 ([VL88]) *For every random string $s \in \Gamma_{\mathcal{T}}(X)$, X is a positive instance of distributional tiling $\mathcal{T}$ if and only if the graph G produced by $f(X, s)$ is colorable under the color specification (C, b). Moreover, the tiling can be constructed in polynomial time from a coloring.*

Proof of Theorem 4-9 The desired reduction $f(X, s)$, for $(X, s) \in \Gamma_{\mathcal{T}}$, is the digraph G as described above, plus the color specification (C, b).

Proof of Theorem 4-10 It is easy to see that f is one-one, length-increasing, and p-time computable on input $(X, s) \in \Gamma_{\mathcal{T}}$. To see that $\mu_{\Gamma_{\mathcal{T}}}(X, s)$ is dominated by $\mu_E(f(X, s))$, we note that $\mu_E(f(X, s))$ is proportional to $(2n^2 + k - 1)^{-4} 2^{-(2n^2+k-1)^2}$, and $\mu_{\Gamma_{\mathcal{T}}}(X, s) = \Pr[X] 2^{-|s|} \leq 2^{-|s|} = 2^{-r(n)}$, where $\Pr[X]$ is the probability distribution of X. Since $-r(n) < -(2n^2 + k - 1)^2$, $\mu_{\Gamma_{\mathcal{T}}}(X, s)$ is dominated by $\mu_E(f(X, s))$.

Proof of Theorem 4-11 We now show that f is p-time invertible. Given a tiling graph G, we can easily identify T, L, and U by checking whether a node has a double-direction self-loop, a single-direction self-loop, or no self-loop. Our task is therefore to find S that is embedded in G. From T, we can obtain distinct codes for nodes in U and so S can be identified. The partition T, L, and U, and the edges of G reveal the random string s that generates the graph. Clearly, this algorithm can be carried out in time polynomial of n. This algorithm also shows that $\Gamma_{\mathcal{T}}$ is selectable.

Proof of Theorem 4-12 Next, we consider the other direction. Since μ_E satisfies the hypothesis of the Distribution Controlling Lemma, it has been shown in [WB95] that $(E, \mu_E) \leq_m^p (\mathcal{T}, \mu_{\mathcal{T}})$ via a deterministic reduction g that is one-one, p-time computable, and p-time invertible. Moreover, $\mu_E \approx^p \mu_{\mathcal{T}} \circ g$. So for every $X \in$ range(g), where $X = (1^n, S) \in \mathcal{D}_{\mathcal{T}}$, and for every $s \in \Gamma_{\mathcal{T}}(X)$, $\mu_{\Gamma_{\mathcal{T}}}(X, s) < \mu_{\mathcal{T}}(X)$, which is dominated by $\mu_E(g^{-1}(X))$. This shows that the first part of Condition 3 of Theorem 2 is satisfied.

Proof of Theorem 4-13 We verify that the second part of Condition 3 of Theorem 2 is also satisfied. First, we may view g as a function defined on $\Gamma_E = \{ (Y, e) : Y \in \mathcal{D}_E \}$ by $g(Y, e) = g(Y)$. Let $Y \in \text{range}(f)$, then $Y = (G, (C, b)) = f((1^n, S), s)$ for a unique $((1^n, S), s) \in \Gamma_{\mathcal{T}}$, where $|S| < n$, and $|G| = 2n^2 + k - 1$, where $|G|$ denotes the number of nodes in G. Hence, $\mu_{\Gamma_E}(Y, e) = \mu_E(Y) \approx^p |G|^{-4} 2^{-|G|^2}$. Since $\mu_{\mathcal{T}}(1^n, S)$ is proportional to $n^{-2} c^{-|S|}$ for some constant c ($c \leq |\mathcal{T}|$) and $|S| < n$, we have that $\mu_{\Gamma_E}(Y, e)$ is dominated by $\mu_{\mathcal{T}}(1^n, S) = \mu_{\mathcal{T}}(\pi_1(f^{-1}(Y)))$.

Proof of Theorem 4-14 From Theorem 2, this completes the proof.

Proof of Theorem 4 □

5.3 Distributional Matrix Transformation

5.3-1 A square matrix X is called *unimodular* if all entries in X are integers and its determinant $\det(X) = 1$. Let $\text{SL}_2(\mathbb{Z})$ denote the set of 2×2 unimodular matrices. Define the size of a unimodular matrix X, denoted by $|X|$, to be the length of the binary representation of the maximal absolute value of its entries.

5.3-2 The distributional matrix transformation problem deals with linear transformations on 2×2 unimodular matrices. A *linear transformation* of $\text{SL}_2(\mathbb{Z})$ is a function $T : \text{SL}_2(\mathbb{Z}) \to \text{SL}_2(\mathbb{Z})$ such that $T(\sum X_i) = \sum T(X_i)$ whenever all the X_i and $\sum X_i$ are unimodular matrices. (Note that in general, $\text{SL}_2(\mathbb{Z})$ is not closed under addition.) A linear transformation of $\text{SL}_2(\mathbb{Z})$ can be represented by a 4×4 integer matrix, and it is decidable in polynomial time whether a given 4×4 integer matrix represents a linear transformation of $\text{SL}_2(\mathbb{Z})$ [Gur91, BG95].

5.3-3 Let T be a linear transformation and let $M(T)$ be its 4×4 integer matrix representation. Define the *size* of T to be the length of the largest absolute value (in the binary notation) of entries in $M(T)$. Gurevich [Gur91] (see also [BG95]) showed that the uniform distribution of T among all linear transformations of size l is $\Theta(l^{-1} 2^{-2l})$.

5.3-4 Distributional Matrix Transformation

Instance. A unimodular matrix X, a finite set S of linear transformations of unimodular matrices, and a unary notation 1^n for a positive integer n.

5.3-5 *Question*. Does a linear transformation T exist, where $T = T_1 \circ T_2 \circ \ldots \circ T_k$, $k \leq n$, $T_i \in S$, such that $T(X)$ is the identity matrix?

5.3-6 *Distribution*. The three components are chosen randomly and independently. The integer component n is chosen with respect to the default uniform

distribution $1/n^2$. The unimodular component X is chosen with probability $|X|^{-2}2^{-2|X|}$. Linear transformations are chosen with respect to the uniform distribution on transformations of the same size. Finally, the probability of S is proportional to the product of the probabilities of the members in S.

5.3-7 Let (T, μ_T) denote the distributional matrix transformation problem. Blass and Gurevich [BG95] showed that (T, μ_T) is average-case NP-complete under a randomized reduction. Based on that we can show the following result, and we leave the proof to the reader.

Theorem 5 $(K, \mu_K) \equiv_r (T, \mu_T)$.

5.3-8 We can also show that the distributional matrix representability problem with flat distribution [VR92] is randomly isomorphic to (K, μ_K). The reader is referred to [VR92] for a definition of the problem, and we again leave the isomorphism proof to the reader.

Acknowledgments

I am grateful to Jay Belanger for several interesting discussions and for proofreading an early draft of this paper. I thank the two anonymous referees for several useful suggestions.

Acknowledgment of support: Supported in part by NSF under grant CCR-9424164 and by a research grant from the University of North Carolina at Greensboro.

References

[BDCGL92] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average case complexity. *Journal of Computer and System Sciences*, 44:193–219, 1992.

[BES80] L. Babai, P. Erdos, and M. Selkow. Random graph isomorphism. *SIAM Journal on Computing*, 9:628–635, 1980.

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

[BG95] A. Blass and Y. Gurevich. Matrix transformation is complete for the average case. *SIAM Journal on Computing*, 24:3–29, 1995.

[BH77] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing*, 2:305–322, 1977.

[BW93] J. Belanger and J. Wang. Isomorphisms of NP-complete problems on random instances. In *Proceedings of the 8th Annual Conference on Structure in Complexity Theory*, pages 65–74, Los Alamitos, CA, 1993. IEEE Computer Society Press.

[BW97] J. Belanger and J. Wang. No NP problems over ranking of distributions are harder. *Theoretical Computer Science*, 181:229–245, 1997.

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

[Gur87] Y. Gurevich. Complete and incomplete randomized NP problems. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pages 111–117, Los Alamitos, CA, 1987. IEEE Computer Society Press.

[Gur91] Y. Gurevich. Average case completeness. *Journal of Computer and System Sciences*, 42:346–398, 1991.

[IL90] R. Impagliazzo and L. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31th Annual Symposium on Foundations of Computer Science*, pages 812–821, Los Alamitos, CA, 1990. IEEE Computer Society Press.

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

[Ko83] K. Ko. On the definition of some complexity classes of real numbers. *Mathematical Systems Theory*, 16:95–109, 1983.

[Lev86] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15:285–286, 1986.

[Liu68] C. L. Liu. *Introduction to Combinatorial Mathematics*. McGraw-Hill, New York, 1968.

[Ven91] R. Venkatesan. *Average-Case Intractability*. PhD thesis, Boston University, 1991.

[VL88] R. Venkatesan and L. Levin. Random instances of a graph coloring problem are hard. In *Proceedings of the 20th Annual Symposium on Theory of Computing*, pages 217–222, New York, 1988. ACM Press.

[VR92] R. Venkatesan and S. Rajagopalan. Average case intractability of diophantine and matrix problems. In *Proceedings of the 24th Annual Symposium on Theory of Computing*, pages 632–642, New York, 1992. ACM Press.

[Wan95] J. Wang. Average-case completeness of a word problem for groups. In *Proceedings of the 27th Annual Symposium on Theory of Computing*, pages 25–334, New York, 1995. ACM Press.

[Wan97] J. Wang. Average-case computational complexity theory. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 295–328. Springer-Verlag, Berlin, 1997.

[WB95] J. Wang and J. Belanger. On the NP-isomorphism problem with respect to random instances. *Journal of Computer and System Sciences*, 50:151–164, 1995.