On Nonadaptive Reductions to the Set of Random Strings and Its Dense Subsets

Shuichi Hirahara[1] and Osamu Watanabe[2(✉)]

[1] National Institute of Informatics, Tokyo, Japan
s_hirahara@nii.ac.jp
[2] Tokyo Institute of Technology, Tokyo, Japan
watanabe@c.titech.ac.jp

Abstract. We explain our recent results [21] on the computational power of an arbitrary distinguisher for (not necessarily computable) hitting set generators. This work is motivated by the desire of showing the limits of black-box reductions to some distributional NP problem. We show that a black-box nonadaptive randomized reduction to any distinguisher for (not only polynomial-time but also) exponential-time computable hitting set generators can be simulated in $\mathsf{AM} \cap \mathsf{coAM}$; we also show an upper bound of $\mathsf{S}_2^{\mathsf{NP}}$ even if there is no computational bound on a hitting set generator. These results provide additional evidence that the recent worst-case to average-case reductions within NP shown by Hirahara (2018, FOCS) are inherently non-black-box. (We omit all detailed arguments and proofs, which can be found in [21].)

Dedication to Ker-I from Osamu

I, Osamu Watanabe, (with my co-author, Shuichi Hirahara) dedicate this article to my senior colleague and good friend Ker-I Ko. I met Ker-I in 1985 when I visited University of California, Santa Barbara (UCSB) for participating in a small work shop organized by Ron, Professor Ronald V. Book. We then met again when I was a Key Fan visiting professor at Department of Mathematics, UCSB from 1987 to 1988. He was visiting Ron around that time. We discussed a lot on various things almost every day with me sitting in his office for many hours. I still recall him saying "Osamu, you know what?", which was usually followed by an interesting episode of famous researchers, politicians, among other things. This period was very important for me to develop my career as a computer scientist, in particular, in theoretical computer science. Certainly, I learnt a lot from Ker-I. I am also proud of having the following sentence in the acknowledgement of his paper [25]:

> The author would like to thank Ronald Book and Osamu Watanabe. Without their *help*, this work would never be finished *in polynomial time*.

During that time, we discussed a lot on the structure of complexity classes such as reducibilities, relativiations, sparse sets, approximability, etc. For example, we spent a lot of time trying to improve Mahaney's theorem: For any NP-complete set L, if L is polynomial-time many-one reducible to a sparse set, then L

D.-Z. Du and J. Wang (Eds.): Ko Festschrift, LNCS 12000, pp. 67–79, 2020.
https://doi.org/10.1007/978-3-030-41672-0_6

is indeed in P; that is, it is polynomial-time computable. Since then, the complexity theory has been developed (not so rapidly but) steadily. Several important notions have been introduced, and many powerful computational/mathematical tools have been developed for analyzing computability of various types. In this article, we are glad to explain our result that is much stronger (in several aspects emphasized below with underlined comments) than Mahaney's theorem. One of the results stated in Theorem 1 here can be interpreted as follows: For any set L (for which no complexity class assumption is needed) if L is randomized polynomial-time nonadaptively and "robustly" reducible (which is much more general than the one considered in Mahaney's theorem) to a relatively small density set (that could be much larger than sparse sets), then L is indeed in $\mathsf{S}_2^{\mathsf{NP}}$. Another interesting and exciting point of our results is that it is motivated from a question in a quite different context, the average-case vs. the worst-case complexity in NP, which was also one of the topics that I discussed with Ker-I with no idea at all of how to attack it at that time. Hope Ker-I would like these results and the following explanation.

1 Introduction

We explain our recent investigation on what can be reduced to the set of random strings, and its dense subset, which is related to several lines of research of complexity theory – including average-case complexity and black-box reductions, hitting set generators, the Minimum Circuit Size Problem, and the computational power of the set of random strings.

The underlying theme that unifies these research lines is Kolmogorov complexity. *Kolmogorov complexity* enables us to quantify how a finite string looks "random" in terms of compressibility. For a string $x \in \{0,1\}^*$, its Kolmogorov complexity is the length of the shortest program d such that running d will print x. More specifically, we fix an arbitrary universal Turing machine U, and the Kolmogorov complexity of x is defined as $\mathrm{K}_U(x) := \min\{\, |d| \mid U(d) = x \,\}$. A string x is called *random* (with threshold s) if $\mathrm{K}_U(x) \geq s$, i.e., x cannot be compressed into a short program. While Kolmogorov complexity is not computable, by either imposing a time constraint on U or taking another "decoder" U, we are led to several important concepts of complexity theory mentioned above. Below, we review these concepts through the lens of Kolmogorov complexity.

An important motivation for this work is the case when a decoder U is defined as a circuit interpreter G^{int}: Let G^{int} denote the function that takes a description of a Boolean circuit C, and outputs the truth table of the function computed by C. Here a *truth table* of a function $f\colon \{0,1\}^n \to \{0,1\}$ is the string of length 2^n that can be obtained by concatenating $f(x)$ for every input $x \in \{0,1\}^n$, and we often identify a function with its truth table. Taking $U = G^{\mathsf{int}}$, the Kolmogorov complexity $\mathrm{K}_{G^{\mathsf{int}}}(f)$ is approximately equal to the minimum circuit size for computing f. Therefore, a circuit lower bound question can be seen as a question of finding a random string f with respect to $\mathrm{K}_{G^{\mathsf{int}}}$. For example, one of the central open questions in complexity theory, $\mathsf{E} \not\subset \mathsf{SIZE}(2^{\epsilon n})$ for some

constant $\epsilon > 0$, can be equivalently rephrased as the question whether there exists a polynomial-time algorithm that, on input 1^N, finds a "random" string f of length N such that $\mathrm{K}_{G^{\mathsf{int}}}(f) = N^{\Omega(1)}$ for infinitely many N. The problem of computing $\mathrm{K}_{G^{\mathsf{int}}}(f)$ on input f is called the Minimum Circuit Size Problem (MCSP) [24], which is intensively studied recently.

A dense subset of random strings (with respect to $\mathrm{K}_{G^{\mathsf{int}}}$) is also one of the important concepts in complexity theory, which was called a natural property by Razborov and Rudich [30]. In their influential work, Razborov and Rudich introduced the notion of natural proof, and explained the limits of current proof techniques for showing circuit lower bounds. A *natural property* $R \subset \{0,1\}^*$ is a polynomial-time computable $1/\mathsf{poly}(\ell)$-dense subset of random strings with respect to $\mathrm{K}_{G^{\mathsf{int}}}$. Here, a set is called γ-*dense* if $\Pr_{x \in_R \{0,1\}^\ell}[x \in R] \geq \gamma(\ell)$ for every $\ell \in \mathbb{N}$. It is known that a natural property is equivalent to an errorless average-case algorithm for MCSP [19].

More generally, a dense subset of random strings with respect to K_G can be seen as an adversary for a hitting set generator G. We consider a family of functions $G = \{G_\ell : \{0,1\}^{s(\ell)} \to \{0,1\}^\ell\}_{\ell \in \mathbb{N}}$. A *hitting set generator* (HSG) is the notion that is used to derandomize one-sided-error randomized algorithms. For a set $R \subset \{0,1\}^*$, we say that G is a hitting set generator (with parameter γ) for R if $\Pr_{r \in_R \{0,1\}^\ell}[r \in R] \geq \gamma(\ell)$ implies $R \cap \mathrm{Im}(G_\ell) \neq \varnothing$, for every $\ell \in \mathbb{N}$. Conversely, R is said to γ-*avoid* G if G is not a hitting set generator for R, that is, (1) $\Pr_{r \in_R \{0,1\}^\ell}[r \in R] \geq \gamma(\ell)$ for all $\ell \in \mathbb{N}$ (i.e., R is γ-dense), and (2) $R \cap \mathrm{Im}(G_\ell) = \varnothing$ (i.e., R does not intersect with the image $\mathrm{Im}(G_\ell)$ of G_ℓ). Since $\mathrm{Im}(G_\ell)$ contains all the non-random strings with respect to K_{G_ℓ}, this definition means that R is a γ-dense subset of random strings with respect to K_G.

Next, we proceed to reviewing each research line. We start with average-case complexity and black-box reductions.

2 Reducing from the Worst-Case to the Average-Case: Limits of Black-Box Reductions

The security of modern cryptography is based on average-case hardness of some computational problems in NP. It is, however, a challenging question to find a problem in NP that is hard with respect to a random input generated efficiently. The fundamental question of average-case complexity is to find a problem in NP whose average-case hardness is based on the worst-case complexity of an NP-complete problem.

A line of work was devoted to understanding why resolving this question is so difficult. Given our limited understanding of unconditional lower bounds, the most prevailing proof technique in complexity theory for showing intractability of a problem is by means of reductions. Moreover, almost all reduction techniques are *black-box* in the sense that, given two computational problems A and B, a reduction R solves A given any oracle (i.e., a black-box algorithm) solving B. The technique of reductions led to the discovery of a large number of NP-complete problems computationally equivalent to each other—in the *worst-case*

sense. On the other hand, it turned out that the power of black-box reductions is limited for the purpose of showing intractability of average-case problems based on worst-case problems.

Building on the work of Feigenbaum and Fortnow [11], Bogdanov and Trevisan [9] showed that if a worst-case problem L is reducible to some average-case problem in NP via a nonadaptive black-box randomized polynomial-time reduction, then L must be in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. This in particular shows that the hardness of any average-case problem in NP cannot be based on the worst-case hardness of an NP-complete problem via such a reduction technique (unless the polynomial-time hierarchy collapses [34]). Akavia, Goldreich, Goldwasser and Moshkovitz [1,2] showed that, in the special case of a nonadaptive reduction to the task of inverting a one-way function, the upper bound of [9] can be improved to $\mathsf{AM} \cap \mathsf{coAM}$, thereby removing the advice "/poly". Bogdanov and Brzuska [8] showed that even a general (i.e. adaptive) reduction to the task of inverting a size-verifiable one-way function cannot be used for any problem outside $\mathsf{AM} \cap \mathsf{coAM}$. Applebaum, Barak, and Xiao [7] studied black-box reductions to PAC learning, and observed that the technique of [1] can be applied to (some restricted type of) a black-box reduction to the task of inverting an auxiliary-input one-way function.

3 A Motivation for Investigating Non-black-box Reductions Further

It was very recent that the first worst-case to average-case reductions from worst-case problems conjectured to be outside coNP to some average-case problems in NP were found: Hirahara [18] showed that approximation versions of the minimum time-bounded Kolmogorov complexity problem (MINKT [26]) and MCSP admit worst-case to average-case reductions. These problems ask, given a string x and a threshold s, whether x can be compressed by certain types of algorithms of size s. For example, MCSP asks whether x can be compressed as a truth table of a circuit of size at most s. For a constant $\epsilon > 0$, its approximation version $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ is the problem of approximating the minimum circuit size for a function $f\colon \{0,1\}^n \to \{0,1\}$ (represented as its truth table) within a factor of $2^{(1-\epsilon)n}$. Specifically, the YES instances of $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ consists of (f, s) such that $\mathsf{size}(f) \leq s$, and the NO instances of $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ consists of (f, s) such that $\mathsf{size}(f) > 2^{(1-\epsilon)n}s$. MCSP can be defined as $\mathrm{Gap}_1\mathrm{MCSP}$. It is easy to see that $\mathrm{MCSP} \in \mathsf{NP}$ and $\mathrm{MINKT} \in \mathsf{NP}$, but these are important examples of problems for which there is currently neither a proof of NP-completeness nor evidence against NP-completeness. Allender and Das [4] showed that MCSP is SZK-hard, but this hardness result is unlikely to be improved to NP-hardness using "oracle-independent" reduction techniques: Hirahara and Watanabe [20] showed that a one-query randomized polynomial-time reduction to MCSP^A for every oracle A can be simulated in $\mathsf{AM} \cap \mathsf{coAM}$. Nonetheless, MCSP and MINKT are (indirectly) conjectured to be outside $\mathsf{coNP/poly}$ by Rudich [31] based on some assumptions of average-case complexity: He conjectured that there exists a (certain type

of) hitting set generator secure even against nondeterministic polynomial-size circuits. We also mention that the approximation version of MINKT is harder than Random 3SAT, which is conjectured by Ryan O'Donnell (cf. [19]) to not be solvable by coNP algorithms.

The work of Hirahara motivates us to study black-box reductions further. We ask whether the technique used in [18] is inherently non-black-box or not. As mentioned above, there are several results and techniques developed in order to simulate black-box reductions by AM∩coAM algorithms. Why can't we combine these techniques with the (seemingly non-black-box) reductions of [18] in order to prove $\mathrm{Gap}_\epsilon\mathrm{MCSP} \in \mathsf{coAM}$ and refute Rudich's conjecture? Note that refuting Rudich's conjecture would significantly change our common belief about average-case complexity and the power of nondeterministic algorithms. We emphasize that while the proof of [18] seems to yield only non-black-box reductions, it does not necessarily mean that there is no alternative proof that yields a black-box reduction.

In order to address the question, we aim at improving our understanding of the limits of black-box reductions. We summarize a landscape around average-case complexity in Fig. 1.

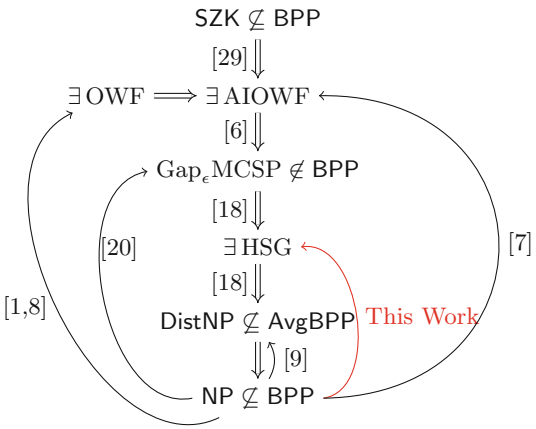

Fig. 1. Average-case complexity and limits of black-box reductions. "$A \to B$" means that there is no black-box (or oracle-independent) reduction technique showing "$A \Rightarrow B$" under reasonable complexity theoretic assumptions. The security of all cryptographic primitives is with respect to an almost-everywhere polynomial-time randomized adversary.

A couple of remarks about implications written in Fig. 1 are in order: First, the implication from the existence of an auxiliary-input one-way function (AIOWF) to $\mathrm{Gap}_\epsilon\mathrm{MCSP} \notin \mathsf{BPP}$ was implicitly proved in [3] and explicitly in [6], based on [13,17,30]. The implication from $\mathsf{SZK} \not\subset \mathsf{BPP}$ to the existence of an auxiliary-input one-way function is due to Ostrovsky [29] (see also [33]). Second,

building on [10,19], it was shown in [18, Theorem VI.5] that $\mathrm{Gap}_\epsilon\mathrm{MCSP} \notin \mathsf{BPP}$ implies the nonexistence of natural properties, which yields a hitting set generator $G^{\mathsf{int}} = \{G_{2^n} : \{0,1\}^{\widetilde{O}(2^{\epsilon' n})} \to \{0,1\}^{2^n}\}_{n\in\mathbb{N}}$ defined as a "circuit interpreter": a function that takes a description of a circuit of size $2^{\epsilon' n}$ and outputs its truth table (cf. [18, Definition V.3]). The existence of a hitting set generator naturally induces a hard problem in DistNP with respect to AvgBPP algorithms (cf. [18, Lemma VI.4]). Therefore, the reduction of [18] can be regarded as a non-black-box (in fact, nonadaptive) reduction to a distinguisher for the hitting set generator G^{int}.

We thus continue the study of the limits of black-box reductions to a distinguisher for a hitting set generator, initiated by Gutfreund and Vadhan [15]. Motivated by the question on whether derandomization is possible under uniform assumptions (cf. [32]), they investigated what can be reduced to any oracle avoiding a hitting set generator in a black-box way.[1] They showed that any polynomial-time randomized nonadaptive black-box reductions to any oracle avoiding an exponential-time computable hitting set generator G can be simulated in $\mathsf{BPP}^{\mathsf{NP}}$, which is a trivial upper bound when G is polynomial-time computable.

4 Our Results

We significantly improve the above $\mathsf{BPP}^{\mathsf{NP}}$ upper bound to $\mathsf{AM} \cap \mathsf{coAM}$, thereby putting the study of hitting set generators into the landscape of black-box reductions within NP (Fig. 1). We also show a uniform upper bound of $\mathsf{S}_2^{\mathsf{NP}}$ even if G is not computable.

Theorem 1. *Let $G = \{G_\ell : \{0,1\}^{s(\ell)} \to \{0,1\}^\ell\}_{\ell\in\mathbb{N}}$ be any (not necessarily computable) hitting set generator such that $s(\ell) \le (1-\Omega(1))\ell$ for all large $\ell \in \mathbb{N}$. Let $\mathsf{BPP}_{\|}^R$ denote the class of languages solvable by a randomized polynomial-time nonadaptive machine with oracle access to R. (The subscript $\|$ stands for parallel queries.) Then,*

$$\bigcap_R \mathsf{BPP}_{\|}^R \subset \mathsf{NP/poly} \cap \mathsf{coNP/poly} \cap \mathsf{S}_2^{\mathsf{NP}},$$

where the intersection is taken over all oracles R that $(1 - 1/\mathsf{poly}(\ell))$-avoid G. Moreover, if G_ℓ is computable in $2^{O(\ell)}$, then we also have

$$\bigcap_R \mathsf{BPP}_{\|}^R \subset \mathsf{AM} \cap \mathsf{coAM}.$$

[1] As a *black-box* reduction to any distinguisher for G, it is required in [15] that there exists a *single* machine that computes a reduction to every oracle avoiding G. On the other hand, as stated in Theorem 1, we allow reductions to depend on oracles, which makes our results stronger.

Compared to the line of work showing limits of black-box reductions within NP, a surprising aspect of Theorem 1 is that it generalizes to any function G that may not be computable. Indeed, almost all the previous results [1,7,9,11] crucially exploit the fact that a verifier can check the correctness of a certificate for an NP problem; thus a dishonest prover can cheat the verifier only for one direction, by not providing a certificate for a YES instance. In our situation, a verifier cannot compute G and thus cannot prevent dishonest provers from cheating in this way. At a high level, our technical contributions are to overcome this difficulty by combining the ideas of Gutfreund and Vadhan [15] with the techniques developed in [9,11].

Moreover, we present a new $\mathsf{S}_2^{\mathrm{p}}$-type algorithm for simulating reductions to an oracle R avoiding G. Indeed, at the core of Theorem 1 is the following two types of algorithms simulating reductions: One is an $\mathsf{S}_2^{\mathrm{p}}$ algorithm that simulates any query $q \stackrel{?}{\in} R$ of length at most $\Theta(\log n)$, and the other is an $\mathsf{AM} \cap \mathsf{coAM}$ algorithm that simulates any query $q \stackrel{?}{\in} R$ of length at least $\Theta(\log n)$. In particular, when G is exponential-time computable, the $\mathsf{S}_2^{\mathrm{p}}$ algorithm can be replaced with a polynomial-time algorithm and obtain the $\mathsf{AM} \cap \mathsf{coAM}$ upper bound.

We remark that Theorem 1 improves all the previous results mentioned before in some sense. Compared to [9], our results show that the advice "/poly" is not required in order to simulate black-box reductions to any oracle avoiding an exponential-time computable hitting set generator. Compared to [1,7], our results "conceptually" improve their results because the existence of one-way functions imply the existence of hitting set generators; on the other hand, since the implication goes through the *adaptive* reduction (from the task of inverting a one-way function to a distinguisher for a PRG) of [17], technically speaking, our results are incomparable with their results.[2] Similarly, our results conceptually improve the result of [20], but these are technically incomparable, mainly because the implication goes through the non-black-box reduction of [18].

5 Why Are the Reductions of [18] Non-black-box?

Based on Theorem 1, we now argue that the reductions of [18] are inherently non-black-box in a certain formal sense, without relying on any unproven assumptions: The reason is that the idea of [18] can be applied to not only time-bounded Kolmogorov complexity but also any other types of Kolmogorov complexity, including resource-unbounded Kolmogorov complexity. Therefore, if this generalized reduction could be made black-box, then (as outlined below) by Theorem 1

[2] We emphasize that we are concerned the nonadaptivity of reductions used in the security proof of pseudorandom generators. Several simplified constructions of pseudorandom generators G^f from one-way functions f (e.g., [16,23]) are nonadaptive in the sense that G^f can be efficiently computed with nonadaptive oracle access to f; however, the security reductions of these constructions are adaptive because of the use of Holenstein's uniform hardcore lemma [22]. Similarly, the reduction of [17, Lemma 6.5] is adaptive. (We note that, in the special case when the degeneracy of a one-way function is efficiently computable, the reduction of [17] is nonadaptive.).

we would obtain a finite algorithm $\mathsf{S}_2^{\mathsf{NP}}$ that approximates resource-unbounded Kolmogorov complexity, which is a contradiction, *unconditionally*.

To give one specific example, we briefly outline how the reductions of [18] can be generalized to the case of Levin's Kt-complexity [27]: Fix any efficient universal Turing machine U, and the Kt-complexity of a string x is defined as

$$\mathrm{Kt}(x) := \min\{|d| + \log t \mid U(d) \text{ outputs } x \text{ within } t \text{ steps } \}.$$

We define a hitting set generator $G = \{G_\ell : \{0,1\}^{\ell/2} \to \{0,1\}^\ell\}_{\ell \in \mathbb{N}}$ as $G_\ell(d,t) := U(d)$ for $(d,t) \in \{0,1\}^{\ell/2}$ when $|U(d)| = \ell$ and $U(d)$ halts within t steps, which is computable in exponential time. Note that $\mathrm{Im}(G)$ contains all strings with low Kt-complexity. Given an efficient algorithm D that γ-avoids G, we can approximate $\mathrm{Kt}(x)$ by the following algorithm: Fix any input x. Take any list-decodable code Enc, and let $\mathrm{NW}^{\mathrm{Enc}(x)}(z)$ denote the Nisan-Wigderson generator [28] instantiated with $\mathrm{Enc}(x)$ as the truth table of a hard function, where z is a seed of the generator. Then check whether the distinguishing probability $|\mathbb{E}_{z,w}[D(\mathrm{NW}^{\mathrm{Enc}(x)}(z)) - D(w)]|$ is large or small by sampling, whose outcome tells us whether $\mathrm{Kt}(x)$ is small or large, respectively. Indeed, if the distinguishing probability is large, then by using the security proof of the Nisan-Wigderson generator, we obtain a short description (with oracle access to D) for x. Conversely, if $\mathrm{Kt}(x)$ is small, then since D γ-avoids G, the distinguishing probability is at least γ. Now, if we could make this analysis work for any oracle that γ-avoids G, then by Theorem 1 we would put a problem of approximating $\mathrm{Kt}(x)$ in AM, which is not possible unless $\mathsf{EXP} = \mathsf{PH}$. (Note that the minimization problem of Kt is EXP-complete under NP reductions [3].)

6 Our Techniques

We outline our proof strategy for Theorem 1 below. Suppose that we have some reduction from L to any oracle R that avoids a hitting set generator G. Let $\mathcal{Q}$ denote the query distribution that a reduction makes. We focus on the case when the length of each query is larger than $\Theta(\log n)$, and explain the ideas of the $\mathsf{AM} \cap \mathsf{coAM}$ simulation algorithms.

As a warm-up, consider the case when the support $\mathsf{supp}(\mathcal{Q})$ of $\mathcal{Q}$ is small (i.e., $|\mathsf{supp}(\mathcal{Q}) \cap \{0,1\}^\ell| \ll 2^\ell$ for any length $\ell \in \mathbb{N}$). In this case, we can define an oracle R_1 so that $R_1 := \{0,1\}^* \setminus \mathsf{supp}(\mathcal{Q}) \setminus \mathrm{Im}(G)$; this is a dense subset and avoids the hitting set generator G. Therefore, we can simulate the reduction by simply answering all the queries by saying "No"; hence such a reduction can be simulated in BPP.

In general, we cannot hope that $\mathsf{supp}(\mathcal{Q})$ is small enough. To generalize the observation above, let us recall the notion of α-heaviness [9]: We say that a query q is *α-heavy* (with respect to $\mathcal{Q}$) if the query q is α times more likely to be sampled under $\mathcal{Q}$ than the uniform distribution on $\{0,1\}^{|q|}$; that is, $\Pr_{w \sim \mathcal{Q}}[w = q] \geq \alpha 2^{-|q|}$. Now we define our new oracle $R_2 := \{0,1\}^* \setminus \{\, q \in \{0,1\}^* \mid q\colon \alpha\text{-heavy} \setminus \mathrm{Im}(G) \,\}$, which can be again shown to avoid G because the fraction of α-heavy queries is at most $1/\alpha$ ($\ll 1$).

The problem now is that it is difficult to simulate the new oracle R_2; it appears that, given a query q, we need to test whether $q \stackrel{?}{\in} \mathrm{Im}(G)$, which is not possible in $\mathsf{AM} \cap \mathsf{coAM}$. However, it turns out that we do not need to test it, as we explain next: Observe that the size of $\mathrm{Im}(G)$ is very small; it is at most $2^{s(\ell)}$ $\left(\ll 2^{\ell}\right)$. Thus, the probability that a query q is in $\mathrm{Im}(G)$ and q is not α-heavy (i.e., q is rarely queried) is at most $\alpha \cdot 2^{s(\ell)-\ell}$, where ℓ is the length of q. As a consequence, the reduction cannot "distinguish" the oracle R_2 and a new oracle $R_3 := \{0,1\}^* \setminus \{\, q \in \{0,1\}^* \mid q\colon \alpha\text{-heavy} \,\}$; hence we can simulate the reduction if, given a query q, we are able to decide whether $q \stackrel{?}{\in} R_3$ in $\mathsf{AM} \cap \mathsf{coAM}$.

This task, however, still appears to be difficult for $\mathsf{AM} \cap \mathsf{coAM}$; indeed, at this point, Gutfreund and Vadhan [15] used the fact that the approximate counting is possible in $\mathsf{BPP}^{\mathsf{NP}}$, and thereby simulated the oracle R_3 by $\mathsf{BPP}^{\mathsf{NP}}$.

Our main technical contribution is to develop a way of simulating the reduction to R_3. First, note that the lower bound protocol of Goldwasser and Sipser [14] enables us to give an AM certificate for α-heaviness; we can check, given a query q, whether q is $\alpha(1+\epsilon)$-heavy or α-light for any small error parameter $\epsilon > 0$. Thus, we have an AM protocol for $\{0,1\}^* \setminus R_3$ for every query q (except for $\alpha(1 \pm \epsilon)$-heavy and light queries).

If, in addition, we had an AM protocol for R_3, then we would be done; unfortunately, it does not seem possible in general. The upper bound protocol of Fortnow [12] does a similar task, but the protocol can be applied only for a limited purpose: we need to keep the randomness used to generate a query $q \sim \mathcal{Q}$ from being revealed to the prover. When the number of queries of the reduction is limited to 1, we may use the upper bound protocol in order to give an AM certificate for R_3; on the other hand, if the reduction makes two queries $(q_1, q_2) \sim \mathcal{Q}$, we cannot simultaneously provide AM certificates of the upper bound protocol for *both* of q_1 and q_2, because the fact that q_1 and q_2 are sampled *together* may reveal some information about the private randomness. To summarize, the upper bound protocol works only for the *marginal* distribution of each query, but does not work for the *joint* distribution of several queries.

That is, what we can obtain by using the upper bound protocol is information about *each* query. For example, the heavy-sample protocol of Bogdanov and Trevisan [9] (which combines the lower and upper bound protocol and sampling) estimates, in $\mathsf{AM} \cap \mathsf{coAM}$, the probability that a query q sampled from $\mathcal{Q}$ is α-heavy.

Our idea is to overcome the difficulty above by generalizing the Feigenbaum-Fortnow protocol [11]. Feigenbaum and Fortnow developed an $\mathsf{AM} \cap \mathsf{coAM}$ protocol that simulates a nonadaptive reduction to an NP oracle R, given as advice the probability that a query is a positive instance of R. We generalize the protocol in the case when the oracle $\{0,1\}^* \setminus R_3$ is solvable by AM on average (which can be done by the lower bound protocol [14]), and given as advice the probability that a query q is in $\{0,1\}^* \setminus R_3$ (which can be estimated by the heavy-sample protocol [9]):

Theorem 2 (Generalized Feigenbaum-Fortnow Protocol; Informal). *Suppose that M is a randomized polynomial-time nonadaptive reduction to oracle R whose queries are distributed according to $\mathcal{Q}$, and that R is solvable by* AM *on average (that is, there exists an* AM *protocol Π_R such that, with probability $1 - 1/\mathsf{poly}(n)$ over the choice of $q \sim \mathcal{Q}$, the protocol Π_R computes R on input q). Then, there exists an* $\mathsf{AM} \cap \mathsf{coAM}$ *protocol Π_M such that, given a probability $p^* \approx \Pr_{q\sim\mathcal{Q}}[q \in R]$ as advice, the protocol Π_M simulates the reduction M with probability at least $1 - 1/\mathsf{poly}(n)$.*

On the Case of Adaptive Reductions. We mention that Theorem 1 cannot be extended to the case of *adaptive* reductions. Indeed, Trevisan and Vadhan [32] constructed an exponential-time computable pseudorandom generator based on the intractability of some PSPACE-complete problem, and its security reduction is black-box in the sense of Theorem 1 and adaptive. If Theorem 1 could be extended to the case of adaptive reductions, we would obtain $\mathsf{PSPACE} = \mathsf{AM}$, which is unlikely to be true.

7 Some Evidence for the Tightness of Our Upper Bounds

Theorem 1 leads us to the natural question whether the upper bound is tight. We present evidence that our two types of simulation algorithms are nearly tight.

First consider the $\mathsf{AM} \cap \mathsf{coAM}$-type simulation algorithms. In [21] we observe that the SZK-hardness of MCSP [4] also holds for an average-case version of MCSP:

Theorem 3. *Let $\epsilon > 0$ be any constant, and R be any oracle $\frac{1}{2}$-avoiding $G^{\mathsf{int}} = \{G_n^{\mathsf{int}} : \{0,1\}^{n^\epsilon} \to \{0,1\}^n\}_{n\in\mathbb{N}}$. Then, $\mathsf{SZK} \subset \mathsf{BPP}^R$.*

The reduction of Theorem 3 is adaptive because of the use of [17]. We conjecture that $\mathsf{SZK} \subset \bigcap_R \mathsf{BPP}_{\|}^R$, which implies that the $\mathsf{AM} \cap \mathsf{coAM}$ upper bound of Theorem 1 cannot be significantly improved.

Next consider our $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm. This is in fact completely tight in a certain setting. Let G be a universal Turing machine. We consider an exponential-time analogue of Theorem 1 when the reduction can make only short queries. Specifically, for an oracle R, denote by $\mathsf{EXP}^{R^{\leq \mathsf{poly}}}$ the class of languages that can be computed by a $2^{n^{O(1)}}$-time algorithm that can query $q \stackrel{?}{\in} R$ of length $\leq n^{O(1)}$, on inputs of length n. Then by an exponential-time analogue of Theorem 1 (more specifically, by using the $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm), we can show the following upper bound on the computational power of $\mathsf{EXP}^{R^{\leq \mathsf{poly}}}$ where R is an arbitrary dense subset of Kolmogorov-random strings, i.e., R is a set avoiding the outputs of a universal Turing machine U on short inputs. (We note that all the queries of polynomial length can be asked by an exponential-time reduction, and thus the adaptivity does not matter here.)

Theorem 4. *Fix any universal Turing machine U. Then we have*

$$\bigcap_{R:\ \frac{1}{2}\text{-}avoids\ U} \mathsf{EXP}^{R^{\leq \mathsf{poly}}} \subseteq \bigcap_{R:\ \frac{1}{2}\text{-}avoids U} \mathsf{BPEXP}^{R^{\leq \mathsf{poly}}} \subseteq \mathsf{S}_2^{\mathrm{exp}}.$$

Here $R^{\leq \mathsf{poly}}$ means that the length of queries is restricted to be at most a polynomial in the input length. We also have $\mathsf{EXP}^{\mathsf{NP}} \subset \bigcap_R \mathsf{S}_2^R \subset \mathsf{S}_2^{\mathrm{exp}}$.

Previously, Allender, Friedman and Gasarch [5] showed that black-box BPP reductions to any avoiding oracle can be simulated in $\mathsf{EXPSPACE}$. Theorem 4 significantly improves their upper bound to $\mathsf{S}_2^{\mathrm{exp}}$. What is interesting here is that we can also show [21] the same lower bound, that is,

$$\bigcap_{R:\ \frac{1}{2}\text{-avoids}U} \mathsf{EXP}^{R^{\leq \mathsf{poly}}} \supseteq \mathsf{S}_2^{\mathrm{exp}}$$

Thus, a complexity class, i.e., the exponential-time analogue of $\mathsf{S}_2^{\mathrm{p}}$, is exactly characterized by using Kolmogorov-random strings. The above lower bound also shows the tightness of the exponential-time analogue of the $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm.

References

1. Akavia, A., Goldreich, O., Goldwasser, S., Moshkovitz, D.: On basing one-way functions on NP-hardness. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 701–710 (2006)
2. Akavia, A., Goldreich, O., Goldwasser, S., Moshkovitz, D.: Erratum for: on basing one-way functions on NP-hardness. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 795–796 (2010)
3. Allender, E., Buhrman, H., Koucký, M., van Melkebeek, D., Ronneburger, D.: Power from random strings. SIAM J. Comput. **35**(6), 1467–1493 (2006)
4. Allender, E., Das, B.: Zero knowledge and circuit minimization. Inf. Comput. **256**, 2–8 (2017)
5. Allender, E., Friedman, L., Gasarch, W.I.: Limits on the computational power of random strings. Inf. Comput. **222**, 80–92 (2013)
6. Allender, E., Hirahara, S.: New insights on the (non-) hardness of circuit minimization and related problems. In: Proceedings of the International Symposium on Mathematical Foundations of Computer Science (MFCS), pp. 54:1–54:14 (2017)
7. Applebaum, B., Barak, B., Xiao, D.: On basing lower-bounds for learning on worst-case assumptions. In: Proceedings of the Symposium on Foundations of Computer Science (FOCS), pp. 211–220 (2008)
8. Bogdanov, A., Brzuska, C.: On basing size-verifiable one-way functions on NP-hardness. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 1–6. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_1
9. Bogdanov, A., Trevisan, L.: On worst-case to average-case reductions for NP problems. SIAM J. Comput. **36**(4), 1119–1159 (2006)
10. Carmosino, M.L., Impagliazzo, R., Kabanets, V., Kolokolova, A.: Learning algorithms from natural proofs. In: Proceedings of the Conference on Computational Complexity (CCC), pp. 10:1–10:24 (2016)

11. Feigenbaum, J., Fortnow, L.: Random-self-reducibility of complete sets. SIAM J. Comput. **22**(5), 994–1005 (1993)
12. Fortnow, L.: The complexity of perfect zero-knowledge. Adv. Comput. Res. **5**, 327–343 (1989)
13. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
14. Goldwasser, S., Sipser, M.: Private coins versus public coins in interactive proof systems. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 59–68 (1986)
15. Gutfreund, D., Vadhan, S.: Limitations of hardness vs. randomness under uniform reductions. In: Goel, A., Jansen, K., Rolim, J.D.P., Rubinfeld, R. (eds.) APPROX/RANDOM -2008. LNCS, vol. 5171, pp. 469–482. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85363-3_37
16. Haitner, I., Reingold, O., Vadhan, S.P.: Efficiency improvements in constructing pseudorandom generators from one-way functions. SIAM J. Comput. **42**(3), 1405–1430 (2013)
17. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
18. Hirahara, S.: Non-black-box worst-case to average-case reductions within NP. In: Proceedings of the Symposium on Foundations of Computer Science (FOCS), pp. 247–258 (2018)
19. Hirahara, S., Santhanam, R.: On the average-case complexity of MCSP and its variants. In: Proceedings of the Computational Complexity Conference (CCC), pp. 7:1–7:20 (2017)
20. Hirahara, S., Watanabe, O.: Limits of minimum circuit size problem as oracle. In: Proceedings of the Conference on Computational Complexity (CCC), pp. 18:1–18:20 (2016)
21. Hirahara, S., Watanabe, O.: On nonadaptive reductions to the set of random strings and its dense subsets. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 26, p. 25 (2019)
22. Holenstein, T.: Key agreement from weak bit agreement. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 664–673 (2005)
23. Holenstein, T.: Pseudorandom generators from one-way functions: a simple construction for any hardness. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 443–461. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_23
24. Kabanets, V., Cai, J.: Circuit minimization problem. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 73–79 (2000)
25. Ko, K.: On helping by robust oracle machines. Theor. Comput. Sci. **52**, 15–36 (1987)
26. Ko, K.: On the complexity of learning minimum time-bounded turing machines. SIAM J. Comput. **20**(5), 962–986 (1991)
27. Levin, L.A.: Randomness conservation inequalities; information and independence in mathematical theories. Inf. Control **61**(1), 15–37 (1984)
28. Nisan, N., Wigderson, A.: Hardness vs Randomness. J. Comput. Syst. Sci. **49**(2), 149–167 (1994)
29. Ostrovsky, R.: One-way functions, hard on average problems, and statistical zero-knowledge proofs. In: Proceedings of the Structure in Complexity Theory Conference, pp. 133–138 (1991)
30. Razborov, A.A., Rudich, S.: Natural proofs. J. Comput. Syst. Sci. **55**(1), 24–35 (1997)

31. Rudich, S.: Super-bits, demi-bits, and *NP/qpoly*-natural proofs. In: Rolim, J. (ed.) RANDOM 1997. LNCS, vol. 1269, pp. 85–93. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63248-4_8
32. Trevisan, L., Vadhan, S.P.: Pseudorandomness and average-case complexity via uniform reductions. Comput. Complex. **16**(4), 331–364 (2007)
33. Vadhan, S.P.: An unconditional study of computational zero knowledge. SIAM J. Comput. **36**(4), 1160–1214 (2006)
34. Yap, C.: Some consequences of non-uniform conditions on uniform classes. Theor. Comput. Sci. **26**, 287–300 (1983)