# On Nonadaptive Security Reductions of Hitting Set Generators

**Shuichi Hirahara**
National Institute of Informatics, Tokyo, Japan
s_hirahara@nii.ac.jp

**Osamu Watanabe**
Tokyo Institute of Technology, Japan
watanabe@c.titech.ac.jp

## Abstract

One of the central open questions in the theory of average-case complexity is to establish the equivalence between the worst-case and average-case complexity of the Polynomial-time Hierarchy (PH). One general approach is to show that there exists a PH-computable hitting set generator whose security is based on some NP-hard problem. We present the limits of such an approach, by showing that there exists no exponential-time-computable hitting set generator whose security can be proved by using a nonadaptive randomized polynomial-time reduction from any problem outside $\mathsf{AM} \cap \mathsf{coAM}$, which significantly improves the previous upper bound $\mathsf{BPP}^{\mathsf{NP}}$ of Gutfreund and Vadhan (RANDOM/APPROX 2008 [14]). In particular, any security proof of a hitting set generator based on some NP-hard problem must use either an adaptive or non-black-box reduction (unless the polynomial-time hierarchy collapses). To the best of our knowledge, this is the first result that shows limits of black-box reductions from an NP-hard problem to some form of a distributional problem in DistPH.

Based on our results, we argue that the recent worst-case to average-case reduction of Hirahara (FOCS 2018 [18]) is inherently non-black-box, without relying on any unproven assumptions. On the other hand, combining the non-black-box reduction with our simulation technique of black-box reductions, we exhibit the existence of a "non-black-box selector" for GapMCSP, i.e., an efficient algorithm that solves GapMCSP given as advice two circuits one of which is guaranteed to compute GapMCSP.

**2012 ACM Subject Classification** Theory of computation → Problems, reductions and completeness; Theory of computation → Pseudorandomness and derandomization

**Keywords and phrases** hitting set generator, black-box reduction, average-case complexity

**Digital Object Identifier** 10.4230/LIPIcs.APPROX/RANDOM.2020.15

**Category** RANDOM

**Related Version** A full version of the paper is available at `https://eccc.weizmann.ac.il/report/2019/025/`.

**Funding** Shuichi Hirahara and Osamu Watanabe: JSPS KAKENHI Grant Number 18H04090 JP
*Shuichi Hirahara*: Supported by ACT-I, JST.

## 1 Introduction

The technique of reductions is one of central tools in complexity theory. In order to show that a computational task $A$ is easier than another computational task $B$, it suffices to design a (black-box) reduction, i.e., the algorithm that solves $A$ given oracle access to $B$. Most reductions of complexity theory are *black-box*. That is, the correctness of a reduction can be established without assuming any computational efficiency of the oracle. Black-box reductions are quite powerful and led us to, for instance, the discovery of thousands of NP-complete problems computationally equivalent to each other. However, a line of work

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020).
Editors: Jarosław Byrka and Raghu Meka; Article No. 15; pp. 15:1–15:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

has exhibited limits of black-box reductions: Black-box reductions are too general to resolve several important open questions. We herein continue the study of black-box reductions especially in the context of the construction of a hitting set generator.

A *hitting set generator* $\gamma$-secure against a class $\mathcal{C}$ is a family of functions $G = \{G_\ell : \{0,1\}^{s(\ell)} \to \{0,1\}^\ell\}_{\ell \in \mathbb{N}}$ such that no $\mathcal{C}$-algorithm can $\gamma$-avoid $G$; here we say that an algorithm $R$ *$\gamma$-avoids* $G$ if $R$ rejects every string in the image of $G$, and $R$ accepts at least a $\gamma$-fraction of all inputs of length $\ell$ for every $\ell \in \mathbb{N}$. By default, we assume $\gamma := 1/4$ and we say that $R$ *avoids* $G$ if $R$ $(1/4)$-avoids $G$. A typical approach for constructing a hitting set generator is to design a black-box reduction that reduces some computationally hard task to the task of avoiding a hitting set generator.

In fact, there have been already several known proof techniques that are not black-box. Impagliazzo and Wigderson [27] constructed a hitting set generator based on the uniform hardness assumption that $\mathsf{EXP} \neq \mathsf{BPP}$. Their security proof of the hitting set generator is not a (black-box) reduction from the task of solving $\mathsf{EXP}$ to the task of avoiding the hitting set generator; they crucially exploited the fact that there exists an efficient algorithm that avoids the hitting set generator. Trevisan and Vadhan [36] and Gutfreund and Vadhan [14] showed that the security reduction of [27] is inherently non-black-box in some senses. More recently, building on [10, 22], Hirahara [18] applied the proof techniques for constructing a hitting set generator to the context of average-case complexity, and presented the first non-black-box worst-case to average-case reduction within $\mathsf{NP}$.

Given the fact that there are already non-black-box proof techniques, why should we study the limits of black-box reductions? We highlight several points:

1. Black-box reductions are *more general* and *useful* than non-black-box reductions. Therefore, it is desirable to have a black-box reduction when it is possible; studying limits of black-box reductions enables us to identify when one can hope to construct a black-box reduction.
   For example, Impagliazzo and Wigderson [27] showed that $\mathsf{EXP} \not\subseteq \mathsf{BPP}$ implies that $\mathsf{BPP}$ can be derandomized in *sub-exponential* time (on most inputs, for infinitely many input lengths). This is shown by a non-black-box reduction, and it is not known whether the result can be generalized to a "high-end" result: does $\mathsf{EXP} \not\subseteq \mathsf{BPSUBEXP}$ imply that $\mathsf{BPP}$ can be derandomized in *quasi-polynomial* time? On one hand, Trevisan and Vadhan [36] used a black-box reduction and provided a positive answer to this question when $\mathsf{EXP}$ is replaced with $\mathsf{PSPACE}$. On the other hand, Gutfreund and Vadhan [14] showed that a (mildly adaptive) black-box reduction cannot be used to prove the "high-end" result for $\mathsf{EXP}$.
2. Studying limits of black-box reductions can inspire new black-box reductions. Inspired by this work, Hirahara [21, 20] subsequently presented new constructions of black-box reductions to Kolmogorov complexity, which were previously conjectured to be impossible.
3. Surprisingly, in some cases, the proof techniques for showing limits of black-box reductions can be combined with non-black-box reductions. We will show that one of our new algorithms for simulating black-box reductions can be combined with a non-black-box reduction of [18] (under some assumptions), and present a new structural property of an approximation version of the Minimum Circuit Size Problem (MCSP [28]).

As a main result of this paper, we show that any security proof of a hitting set generator based on some $\mathsf{NP}$-hard problem must use either an adaptive or non-black-box reduction. This is the first limit of black-box worst-case to average-case reductions from $\mathsf{NP}$-hard problems to some form of a distributional problem in $\mathsf{DistPH}$.

Due to the connection to several research areas such as average-case complexity, black-box reductions, and derandomization, it is not easy to describe the literature in few words; we thus review the literature in the subsequent two sections. In Section 2, we review the theory of average-case complexity and state our main results. In Section 3, we review the non-black-box reduction of [18], present some applications of our results, and describe our proof techniques. Due to the space limitation, details are omitted in this version; see the full version of the paper.

# 2 Average-Case Complexity

## 2.1 Background

One of the central open questions in the theory of average-case complexity [29] is to establish the equivalence between the worst-case and average-case complexity of $\mathsf{NP}$.

▶ **Open Question 1.** *Does* $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ *imply* $\mathsf{NP} = \mathsf{P}$*?*

Here $\mathsf{DistNP}$ is the class of distributional problems $(L, \mathcal{D})$ (i.e., a pair of a problem and its input distribution) such that $L \in \mathsf{NP}$ and $\mathcal{D}$ is an efficiently samplable distribution. $\mathsf{AvgP}$ is the class of distributional problems that admit an errorless heuristic polynomial-time scheme [8] (also known as an "average-case polynomial-time algorithm"). Here, for $L \subseteq \{0,1\}^*$ and $\mathcal{D} = \{\mathcal{D}_m\}_{m \in \mathbb{N}}$, a distributional problem $(L, \mathcal{D})$ is said to be in $\mathsf{AvgP}$ if there exists an algorithm $M$ such that, for every $m \in \mathbb{N}$, given an input $x$ in the support of $D_m$, and a parameter $\delta > 0$,

1. $M(x, \delta)$ halts in time $\mathsf{poly}(m, 1/\delta)$,
2. $M(x, \delta)$ outputs either the correct answer $L(x)$ or $\perp$ ("I don't know"), and
3. the probability that $M(x, \delta)$ outputs $\perp$ over a choice of $x \sim \mathcal{D}_m$ is at most $\delta$.

Open Question 1 is of particular importance from the perspective of cryptography: Average-case hardness of $\mathsf{NP}$ is a prerequisite for constructing secure complexity-theoretic cryptographic primitives such as one-way functions (OWFs). Thus resolving Open Question 1 is an important step towards building cryptographic primitives whose security is based on more plausible assumptions (e.g., the worst-case hardness of $\mathsf{NP}$).

There has been a line of work showing that Open Question 1 cannot be resolved by using either relativizing proof techniques [26], black-box worst-case-to-average-case reductions [11, 9, 2, 7, 6], or error-correcting codes [37].

For large enough complexity classes such as $\mathsf{PSPACE}$ and $\mathsf{EXP}$, there is a general technique for converting any worst-case hard function $f$ to some two-sided-error average-case hard function $\mathrm{Enc}(f)$ based on error-correcting codes [35, 36]. Here, the encoded function $\mathrm{Enc}(f)$ is computable in $\mathsf{EXP}$ or $\mathsf{PSPACE}$ given oracle access to $f$; thus, the worst-case and average-case complexity of such large complexity classes are known to be equivalent. Viola [37] showed limits of such an approach: $\mathrm{Enc}(f)$ cannot be computed in $\mathsf{PH}^f$; thus, the proof technique of using error-correcting codes is not sufficient to resolve Open Question 1 as well as the following weaker open question:

▶ **Open Question 2.** *Does* $\mathsf{DistPH} \subseteq \mathsf{AvgP}$ *imply* $\mathsf{PH} = \mathsf{P}$ *(or, equivalently,* $\mathsf{NP} = \mathsf{P}$*)?* [1]

[1] We mention in passing that Pavan, Santhanam, and Vinodchandran [32] made some progress, by proving that $\mathsf{DistP}^{\mathsf{NP}} \subseteq \mathsf{AvgP}$ implies $\mathsf{NP} = \mathsf{P}$, under the implausible assumption that $\mathsf{NP} \subseteq \mathsf{P/poly}$.

Note that Open Question 2 is an easier question than Open Question 1, since $\mathsf{PH} = \mathsf{P}$ is known to be equivalent to $\mathsf{NP} = \mathsf{P}$. In fact, this well-known equivalence between $\mathsf{PH} = \mathsf{P}$ and $\mathsf{NP} = \mathsf{P}$ is shown by using a non-black-box reduction technique[2]; as we will explain later, this is one reason why all the previous limits of black-box reductions [11, 9, 2, 7, 6] fail to explain the difficulty of resolving Open Question 2. In this work, we present the first limit of black-box reduction techniques for resolving Open Question 2, thereby clarifying what kind of proof techniques are useful. We emphasize that, while Viola's result [37] excludes the construction of error-correcting codes within $\mathsf{PH}$, it does not show limits of black-box worst-case to average-case reduction techniques such as Ajtai's reduction [1].

One general approach for constructing an (errorless) average-case hard function is to make use of a hitting set generator. Indeed, a hitting set generator $G$ secure against polynomial-time algorithms naturally induces a hard distributional problem in $\mathsf{DistNP}^G$: Consider the distributional problem $(\mathrm{Im}(G), \mathcal{U})$, i.e., the distributional problem of checking whether an input $x$ is in the image of $G$, where $x$ is randomly chosen from the uniform distribution $\mathcal{U}$. Since the number of Yes instances of $\mathrm{Im}(G)$ is small under the uniform distribution, any errorless heuristic algorithm must reject a large fraction of No instances, which gives rise to an algorithm that avoids $G$. To summarize:

▶ **Fact 3** (Implicit in [18]). *Suppose there exists a hitting set generator $G := \{G_\ell \colon \{0,1\}^{\ell-1} \to \{0,1\}^\ell\}_{\ell\in\mathbb{N}}$ that is $1/4$-secure against polynomial-time algorithms. Then,* $\mathsf{DistNP}^G \not\subseteq \mathsf{AvgP}$*. In particular, when $G$ is computable in* $\mathsf{PH}$*, we obtain* $\mathsf{DistPH} \not\subseteq \mathsf{AvgP}$*.*

Fact 3 suggests an approach for resolving Open Question 2: Try to construct a $\mathsf{PH}$-computable hitting set generator whose security is based on the worst-case hardness of $\mathsf{NP}$. How do we compare this approach with the technique based on error-correcting codes [37]? Our approach is *more general*, because, given a two-sided-error average-case hard function $\mathrm{Enc}(f)$, one can construct a pseudorandom generator $G = \{G_\ell \colon \{0,1\}^{\ell-1} \to \{0,1\}^\ell\}_{\ell\in\mathbb{N}}$ defined as $G_\ell(z) := (z, \mathrm{Enc}(f)(z))$ for a seed $z \in \{0,1\}^{\ell-1}$ [38].

In order to construct a secure hitting set generator based on the hardness of a problem $L$, we need to argue that, if there exists an efficient algorithm that avoids $G$, then $L$ can be solved efficiently. A typical way to establish such an implication is to design *black-box reductions* from $L$ to a distinguisher for a hitting set generator. Specifically, for a candidate hitting set generator $G$, a reduction $M$ is said to be a black-box reduction from $L$ to any $\gamma$-avoiding oracle $R$ for $G$ if, for every input $x$ and any oracle $R$ that $\gamma$-avoids $G$, $M$ computes $L$ on input $x$ under the oracle $R$.

Gutfreund and Vadhan [14] initiated the study of limits of such a black-box reduction, motivated by the question on whether derandomization is possible under uniform assumptions (e.g., [27, 36]). 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 $\mathsf{BPP}^{\mathsf{NP}}$. Unfortunately, their upper bound is too weak to deduce any limit of the approach on Open Question 2 since $\mathsf{NP} \subseteq \mathsf{BPP}^{\mathsf{NP}}$. Similarly, it is impossible to deduce any limit of the approach on Open Question 1, because the upper bound becomes trivial when $G$ is polynomial-time-computable.

[2] Indeed, if the reduction is black-box, we should have $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{NP}}$, which means that $\mathsf{PH}$ collapses.

## 2.2 Our Results: Limits of Security Proof of Hitting Set Generators

We significantly improve the upper bound of [14] to $\mathsf{AM} \cap \mathsf{coAM}$. We also show upper bounds of $\mathsf{NP/poly} \cap \mathsf{coNP/poly} \cap \mathsf{S}_2^{\mathsf{NP}}$ even if $G$ is not computable.

To state our results formally, let $\mathsf{BPP}_{\parallel}^{R}$ denote the class of languages solvable by a randomized polynomial-time machine with nonadaptive oracle access to $R$.[3] In the definition of a black-box reduction $M$ to any $\gamma$-avoiding oracle $R$, the reduction $M$ is not allowed to depend on $R$. However, we will show that the existence of a randomized nonadaptive black-box reduction from $L$ to any $\gamma$-avoding oracle $R$ is equivalent to saying that $L \in \mathsf{BPP}_{\parallel}^{R}$ for every oracle $R$ that $\gamma$-avoids $R$.[4] In light of this, the result of Gutfreund and Vadhan [14] can be stated as $\bigcap_R \mathsf{BPP}_{\parallel}^{R} \subseteq \mathsf{BPP}^{\mathsf{NP}}$, where the intersection is taken over all oracles $R$ that $\gamma$-avoid an exponential-time computable function $G$. Our main result improves $\mathsf{BPP}^{\mathsf{NP}}$ to $\mathsf{AM} \cap \mathsf{coAM}$:

▶ **Theorem 4** (Main). *Let $G = \{G_\ell : \{0,1\}^{s(\ell)} \to \{0,1\}^{\ell}\}_{\ell\in\mathbb{N}}$ be any (not necessarily computable) family of functions and $\gamma\colon \mathbb{N} \to [0,1)$ be a parameter such that*

- *there exists a constant $\epsilon > 0$ such that $s(\ell) \le (1-\epsilon)\ell$ for all large $\ell \in \mathbb{N}$, and*
- *there exists a constant $c > 0$ such that $\gamma(\ell) \le 1 - \ell^{-c}$ for all large $\ell \in \mathbb{N}$.*

*Then,*

$$\bigcap_R \mathsf{BPP}_{\parallel}^{R} \quad \subseteq \quad \mathsf{NP/poly} \cap \mathsf{coNP/poly} \cap \mathsf{S}_2^{\mathsf{NP}},$$

*where the intersection is taken over all oracles $R$ that $\gamma$-avoids $G$. Moreover, if $G$ can be computed in time $2^{O(\ell)}$, then we also have*

$$\bigcap_R \mathsf{BPP}_{\parallel}^{R} \quad \subseteq \quad \mathsf{AM} \cap \mathsf{coAM}.$$

At the core of Theorem 4 is the following two types of algorithms simulating black-box reductions: One is an $\mathsf{S}_2^{\mathsf{p}}$-type algorithm that simulates any query $q \overset{?}{\in} R$ of length at most $\Theta(\log n)$, and the other is an $\mathsf{AM} \cap \mathsf{coAM}$-type algorithm that simulates any query $q \overset{?}{\in} R$ of length at least $\Theta(\log n)$. In particular, if $G$ is exponential-time computable, the $\mathsf{S}_2^{\mathsf{p}}$-type algorithm can be replaced with a polynomial-time algorithm and obtain the $\mathsf{AM} \cap \mathsf{coAM}$ upper bound.

Theorem 4 shows that there exists no hitting set generator whose security can be based on the hardness of some $\mathsf{NP}$-hard problem via a nonadaptive reduction (unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$). In particular, the approach for Open Question 2 by constructing a $\mathsf{PH}$-computable hitting set generator based on an $\mathsf{NP}$-hard problem must use either an adaptive or non-black-box reduction.

It is worthy of note that Theorem 4 is almost tight from several perspectives: First, it is impossible to extend Theorem 4 to the case of *adaptive* reductions (unless $\mathsf{PSPACE} = \mathsf{AM}$). Indeed, Trevisan and Vadhan [36] constructed an exponential-time-computable pseudorandom generator based on the intractability of some $\mathsf{PSPACE}$-complete problem, and its security reduction is adaptive and black-box in the sense of Theorem 4. Second, our $\mathsf{S}_2^{\mathsf{p}}$-type algorithm for simulating short queries is completely tight when $G$ is a universal Turing machine. Third, it is possible to construct a hitting set generator based on the hardness of $\mathsf{SZK}$ (Statistical

[3] The subscript $\parallel$ stands for parallel queries.

[4] We state our results in the latter way because this makes our impossibility results stronger. The proof of the equivalence can be found in the full version of the paper.

Zero Knowledge), which is one of the best lower bound on $\mathsf{AM} \cap \mathsf{coAM}$; thus, the $\mathsf{AM} \cap \mathsf{coAM}$ upper bound of Theorem 4 cannot be significantly improved. (The details can be found in the full version of the paper.)

## 2.3 Related Work: Limits of Worst-case to Average-case Reductions within NP

To the best of our knowledge, Theorem 4 is *the first result* that shows limits of black-box reductions from an NP-hard problem to (some form of) a distributional problem in DistPH. In order to explain this in more detail, we review the previous work on limits of worst-case to average-case reductions within NP.

A natural approach for establishing the equivalence between the worst-case and average-case complexity of NP is by means of black-box reductions. That is, it is sufficient for resolving Open Question 1 to design a reduction that solves some NP-hard problem $L$, using oracle access to an errorless heuristic algorithm $M$ that solves some distributional problem in DistNP. A line of work has been devoted to explaining why such a black-box reduction technique is too general to establish a worst-case to average-case connection for an NP-complete problem.

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 distributional problem in DistNP 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 average-case hardness of NP cannot be based on the worst-case hardness of an NP-complete problem using such a reduction technique (unless the polynomial-time hierarchy collapses [39]). Akavia, Goldreich, Goldwasser and Moshkovitz [2, 3] 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 [7] showed that even an 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 [6] studied black-box reductions to PAC learning, and observed that the technique of [2] can be applied to (some restricted type of) a black-box reduction to the task of inverting an auxiliary-input one-way function (AIOWF), which is a weaker primitive than a one-way function. We summarize the limits of black-box reductions (depicted by $\not\to$) as well as known implications (depicted by $\Longrightarrow$) in Figure 1.

Compared to the previous results on the limits of black-box worst-case-to-average-case reductions within NP, a surprising aspect of Theorem 4 is that it generalizes to any function $G$ that may not be computable (and this is a key property for obtaining the limits of the approach on Open Question 2). Indeed, almost all the previous results [11, 9, 2, 6] 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 in one direction by not providing a certificate for a Yes instance. In our simulation algorithms, 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 [14] with the techniques developed in [11, 9].

Is it possible to directly deduce some limits of an approach on Open Question 2 from the previous results [11, 9]? No! Recall that, in order to resolve Open Question 2, it suffices to establish a reduction from an NP-complete problem to DistPH (using the non-black-box equivalence between $\mathsf{P} = \mathsf{NP}$ and $\mathsf{P} = \mathsf{PH}$). The results of [11, 9] crucially rely on the fact that a Yes instance of DistNP is verifiable in polynomial time. If we would like to simulate

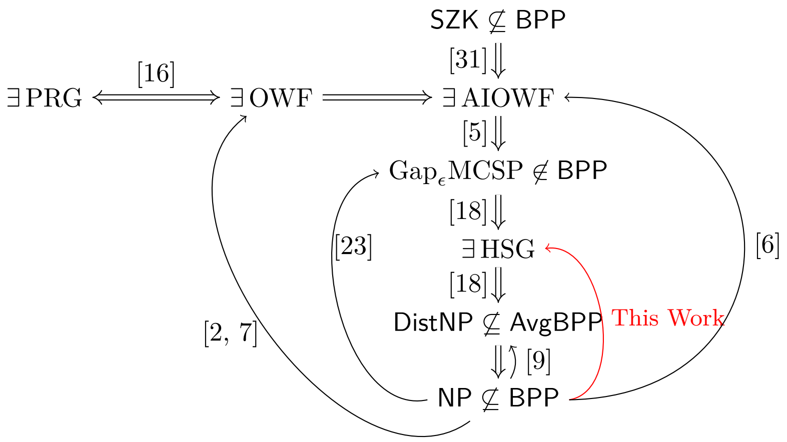

**Figure 1** Average-case complexity and limits of black-box reductions. "$A \to B$" means that there is no black-box (or oracle-independent) reduction technique showing "$A \Rightarrow B$" under reasonable complexity theoretic assumptions. The security of all cryptographic primitives is with respect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

a black-box reduction to $\mathsf{DistNP}^A$ for some oracle $A$, the simulation protocol of Feigenbaum and Fortnow [11] runs in $\mathsf{NP}^A/\mathsf{poly} \cap \mathsf{coNP}^A/\mathsf{poly}$. Thus, in order to simulate a reduction to $\mathsf{Dist}\Sigma_2^{\mathrm{p}} \subseteq \mathsf{DistPH}$, the upper bound becomes $\mathsf{NP}^{\mathsf{NP}}/\mathsf{poly} \cap \mathsf{coNP}^{\mathsf{NP}}/\mathsf{poly}$, which trivially contains $\mathsf{NP}$.

It is also worthy of note that Theorem 4 improves some aspects of all the previous results about limits of black-box reductions within $\mathsf{NP}$. Compared to [9], our results show that the advice "/poly" is not required to simulate black-box reductions to any oracle avoiding an exponential-time-computable hitting set generator. Compared to [2, 6], our results are "improvement" on their results in the sense that the existence of auxiliary-input one-way functions implies 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 [16], technically speaking, our results are incomparable with their results.[5] Similarly, our results conceptually improve the result of [23], but these are technically incomparable, mainly because the implication goes through the non-black-box reduction of [18].

## 3 In Search of Inherently Non-Black-Box Reduction Techniques

Hirahara [18] presented the first non-black-box worst-case to average-case reduction within $\mathsf{NP}$, which is the motivation for this work. Building on [10, 22], Hirahara [18] presented a (nonadaptive) reduction from $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ to a distinguisher for a polynomial-time-computable hitting set generator $G^{\mathsf{int}} = \{G^{\mathsf{int}}_{2^n} \colon \{0,1\}^{\widetilde{O}(2^{\delta n})} \to \{0,1\}^{2^n}\}_{n\in\mathbb{N}}$. Here, $G^{\mathsf{int}}$ is a "circuit

[5] We emphasize that we concern 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., [25, 15]) 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 [24]. Similarly, the reduction of [16, 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 [16] is nonadaptive.)

interpreter": a function that takes a description of a circuit of size $2^{\delta n}$ and outputs its truth table. For a constant $\epsilon > 0$, $\text{Gap}_\epsilon\text{MCSP}$ denotes the problem of approximating the minimum circuit size of a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ within a factor of $2^{(1-\epsilon)n}$, given the truth table of $f$. Rudich [33] conjectured that $\text{Gap}_\epsilon\text{MCSP}$ cannot be solved in $\mathsf{coNP/poly}$ (even in the sense of average-case complexity). Therefore, the reduction of [18] is indeed non-black-box under Rudich's conjecture, as otherwise it contradicts the limits of black-box reductions (such as Theorem 4 and [9]).

Here we pose the following question:

> Are the reductions of [18] *inherently* non-black-box? Or should we regard it as an approach for refuting Rudich's conjecture?

On one hand, the proofs of [18] seem to yield only non-black-box reductions, in the sense that the efficiency of an oracle is crucially exploited. On the other hand, if the reduction from $\text{Gap}_\epsilon\text{MCSP}$ to $\mathsf{DistNP}$ could be made black-box, by using our $\mathsf{coAM}$ simulation protocol of black-box reductions (i.e., Theorem 4), we would obtain $\text{Gap}_\epsilon\text{MCSP} \in \mathsf{coAM} \subseteq \mathsf{coNP/poly}$, which refutes Rudich's conjecture.

In order to answer the question, it is desirable to clarify a fundamental obstacle to applying the simulation techniques of black-box reductions to the reductions of [18], without relying on any unproven assumption.

## 3.1 Hirahara's Reduction is Unconditionally Non-Black-Box

Based on Theorem 4, we can argue that the reductions of [18] are inherently non-black-box in a certain formal sense. 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 4 we would obtain a finite-running-time algorithm $\mathsf{S}_2^{\mathsf{NP}}$ that approximates resource-unbounded Kolmogorov complexity, which is a contradiction *unconditionally.*

More specifically, fix any universal Turing machine $U$, and regard it as a hitting set generator $U = \{U_\ell : \{0,1\}^{\ell/2} \to \{0,1\}^\ell\}_{\ell\in\mathbb{N}}$. That is, $U_\ell$ takes an input $(M, x)$ of length $\ell/2$, simulates the Turing machine $M$ on input $x$, and outputs $M(x)$ if the length of the output $M(x)$ is exactly $\ell$; otherwise, $U_\ell$ outputs $1^\ell$.

▷ Claim 5. Suppose that there exists a computable oracle $R$ that avoids $U$. Then, there exists a randomized polynomial-time nonadaptive $R$-oracle algorithm that approximates $\mathrm{K}_U(x)$.

Proof Sketch. The idea of the non-black-box reduction of [18] is as follows: Given an input $x \in \{0,1\}^n$, take any construction of a pseudorandom generator $G^x\colon \{0,1\}^{\ell/4} \to \{0,1\}^\ell$ based on a worst-case hard function $x\colon \{0,1\}^{\log n} \to \{0,1\}$.[6] For example, we can use the Nisan-Wigderson generator [30] combined with some error-correcting codes. The reduction estimates $p := \mathbb{E}_z[R(G^x(z))]$ by sampling, and accepts if and only if $p$ is small.

The correctness is proved as follows: If $\mathrm{K}_U(x) \le \ell/5$, then $\mathrm{K}_U(G^x(z)) \le |z| + \mathrm{K}_U(x) \ll \ell/2$ for a large enough $\ell$; thus $p = 0$. Conversely, if $p \approx 0$, then by using the security proof of $G^x$, we obtain a small $R$-oracle Turing machine that outputs $x$; thus $\mathrm{K}_U^R(x) \le \mathsf{poly}(\ell, \log n)$; in particular, *by using the assumption that $R$ is computable*, we obtain $\mathrm{K}_U(x) \le \mathsf{poly}(\ell, \log n)$. Therefore, the reduction distinguishes the YES instances $x$ such that $\mathrm{K}_U(x) \le \ell/5$ and the NO instances $x$ such that $\mathrm{K}_U(x) > \mathsf{poly}(\ell, \log n)$. ◁

[6] Here we identify a function with its truth table.

Observe that, in the proof above, we crucially used the assumption that $R$ is computable. Can we avoid the assumption and generalize Claim 5 for any $R$ that avoids $U$? In other words, is there a black-box reduction from approximating $\mathrm{K}_U(x)$ to the task of avoiding $U$? If it is the case, Theorem 4 implies that approximating $\mathrm{K}_U(x)$ can be done in $\mathsf{S}_2^{\mathsf{NP}}$, which contradicts the undecidability of Kolmogorov complexity. Therefore, we conclude that the reduction of Claim 5 is inherently non-black-box.

## 3.2 Applications: Non-Black-Box Selector for GapMCSP

As explained in the previous subsection, the non-black-box reductions of [18] cannot be combined with Theorem 4 *unconditionally*. However, we show that our simulation protocol of black-box reductions can be combined with the non-black-box reductions *conditionally*, which constitutes a new structural property of GapMCSP – the existence of a "non-black-box selector."

▶ **Theorem 6** (GapMCSP Has a "Non-Black-Box Selector"). *For any constant $\epsilon > 0$, there exist some constant $\delta > 0$ and a randomized polynomial-time algorithm that takes as advice two circuits one of which is guaranteed to solve* $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ *and solves* $\mathrm{Gap}_\delta\mathrm{MCSP}$ *with high probability.*

A *selector* for a problem $L$ is an efficient algorithm that solves $L$ given oracle access to two oracles one of which is guaranteed to solve; thus, it "selects" the correct answer from the two oracles. The notion of selector exactly characterizes the class of languages for which advice of logarithmic length can be removed [17]. The selector of Theorem 6 is non-black-box in the sense that it requires to take as advice two polynomial-size circuits instead of black-box access to two oracles.

The main building block of the non-black-box selector is our $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm of Theorem 4. Recall that $\mathsf{S}_2^{\mathrm{p}}$ is a proof system where two competing provers, one of which is guaranteed to be honest, try to convince a polynomial-time verifier. In our $\mathsf{S}_2^{\mathrm{p}}$ simulation algorithm of black-box reductions, for each $i \in \{0,1\}$, the $i$th prover sends a set $R_i$; the honest prover sends a set $R_i$ that avoids a hitting set generator $G$. Then a verifier obtains an oracle $R_0 \cap R_1$ that avoids $G$, to which the reduction is guaranteed to work.

Theorem 6 is proved by combining this $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm with the non-black-box reductions of [10, 18].[7] The reason why we can combine the non-black-box reductions with our $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm is that the non-black-box reduction of [18] is, in fact, a *size-restricted black-box reduction* [14]. This is a black-box reduction which works correctly when an oracle can be computed by a polynomial-size circuit. Our $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm can simulate the size-restricted black-box reduction under the assumption that there exists a polynomial-size circuit that avoids a hitting set generator.

In contrast, we were not able to combine our $\mathsf{AM} \cap \mathsf{coAM}$ algorithm of Theorem 4 with the non-black-box reductions under similar conditions. We leave it as an interesting open question, which could have an application to fixed-polynomial circuit lower bounds (e.g., [34]).

▶ **Open Question 7** ("Non-Black-Box Instance Checkability" of GapMCSP). *Prove that* $\mathrm{MCSP} \in \mathsf{P/poly}$ *(or* $\mathsf{NP} \subseteq \mathsf{P/poly}$*) implies* $\mathrm{Gap}_\epsilon\mathrm{MCSP} \in \mathsf{coAM}$ *for some constant $\epsilon > 0$.*

[7] There is an alternative proof based on the search-to-decision reduction of GapMCSP given by Carmosino, Impagliazzo, Kabanets, and Kolokolova [10]. However, we choose to present the proof by combining the $\mathsf{S}_2^{\mathrm{p}}$-type simulation algorithm with the non-black-box reductions in order to highlight the difference between Theorem 6 and Open Question 7.

## 3.3 Our Techniques

We outline our proof strategy for Theorem 4 below. Suppose that we have some reduction $M$ from $L$ to any oracle $R$ that avoids a hitting set generator $G$. Fix any input $x \in \{0,1\}^*$, and let $\mathcal{Q}_x$ denote the query distribution that a reduction makes on input $x$. We focus on the case when the length of each query is larger than $\Theta(\log n)$, and explain the proof ideas for showing $L \in \mathsf{AM} \cap \mathsf{coAM}$.

As a warm-up, consider the case when the support $\mathsf{supp}(\mathcal{Q}_x)$ of $\mathcal{Q}_x$ is small (i.e., $|\mathsf{supp}(\mathcal{Q}_x) \cap \{0,1\}^\ell| \ll 2^\ell$ for all large $\ell \in \mathbb{N}$). In this case, we can define an oracle $R_1$ so that $R_1 := \{0,1\}^* \setminus \mathsf{supp}(\mathcal{Q}_x) \setminus \mathrm{Im}(G)$; this avoids the hitting generator $G$ because $R_1 \cap \mathrm{Im}(G) = \varnothing$ and the size of $R_1 \cap \{0,1\}^\ell$ is at least $2^\ell - |\mathsf{supp}(\mathcal{Q}_x)| - |\mathrm{Im}(G_\ell)| \gg 2^{\ell-1}$ for all large $\ell \in \mathbb{N}$. Therefore, it is guaranteed that the reduction $M$ computes $L$ correctly under the oracle $R_1$; we can simulate the reduction by simply answering all the queries by saying "No" (since $q \notin R_1$ for every $q \in \mathcal{Q}_x$); hence $L \in \mathsf{BPP}$.

In general, we cannot hope that $\mathsf{supp}(\mathcal{Q}_x)$ is small enough. To generalize the observation above, let us recall the notion of $\alpha$-heaviness [9]: We say that a query $q$ is $\alpha$*-heavy* (with respect to $\mathcal{Q}_x$) if the query $q$ is $\alpha$ times more likely to be sampled under $\mathcal{Q}_x$ than the uniform distribution on $\{0,1\}^{|q|}$; that is, $\Pr_{w \sim \mathcal{Q}_x}[w = q] \geq \alpha 2^{-|q|}$. 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 again be shown to avoid $G$ because the fraction of $\alpha$-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 $q \overset{?}{\in} \mathrm{Im}(G)$, which is not possible in $\mathsf{AM} \cap \mathsf{coAM}$. However, it turns out that it is not necessary 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 $\alpha$-heavy (i.e., $q$ is rarely queried) is at most $\alpha \cdot 2^{s(\ell)-\ell}$, where $\ell$ denotes 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 can decide whether $q \overset{?}{\in} R_3$ 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 [14] used the fact that the approximate counting is possible in $\mathsf{BPP}^{\mathsf{NP}}$, and thereby simulated the oracle $R_3$ by a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm.

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 [13] enables us to give an $\mathsf{AM}$ certificate for $\alpha$-heaviness; we can check, given a query $q$, whether $q$ is $\alpha(1+\epsilon)$-heavy or $\alpha$-light for any small error parameter $\epsilon > 0$. Thus, we have an $\mathsf{AM}$ protocol for $\{0,1\}^* \setminus R_3$ for every query $q$ (except for the queries that are $\alpha$-heavy and $\alpha(1+\epsilon)$-light).

If, in addition, we had an $\mathsf{AM}$ protocol for $R_3$, then we would be done; however, it does not seem possible in general. The upper bound protocol of Fortnow [12] performs 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 $q \sim \mathcal{Q}_x$ from being revealed to the prover. When the number of queries of the reduction is limited to 1, we can use the upper bound protocol in order to give an $\mathsf{AM}$ certificate for $R_3$; on the other hand, if the reduction makes two queries $(q_1, q_2) \sim \mathcal{Q}_x$, we cannot simultaneously provide $\mathsf{AM}$ certificates of the upper bound protocol for *both* $q_1$ 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.

Still, the upper bound protocol is useful for extracting some 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 $\mathcal{Q}_x$ is $\alpha$-heavy. This protocol enables us to estimate the probability that $q \in R_3$ over the choice of $q \sim \mathcal{Q}_x$.

The probability that $q \in R_3$ is useful for simulating the reduction $M$. Feigenbaum and Fortnow [11] developed an $\mathsf{AM} \cap \mathsf{coAM}$ protocol that simulates a nonadaptive reduction to an $\mathsf{NP}$ oracle $R$, given as advice the probability that a query $q$ is in $R$. We generalize this protocol for the case when the oracle $R$ is solvable by $\mathsf{AM}$ on average:

▶ **Theorem 8** (Generalized Feigenbaum–Fortnow Protocol; informal). *Suppose that $M$ is a randomized polynomial-time nonadaptive reduction to an oracle $R$ whose queries are distributed according to $\mathcal{Q}_x$ on input $x \in \{0,1\}^n$, and that $R$ is solvable by $\mathsf{AM}$ on average (i.e., there exists an $\mathsf{AM}$ protocol $\Pi_R$ such that, with probability $1 - 1/\mathsf{poly}(n)$ over the choice of $q \sim \mathcal{Q}_x$, the protocol $\Pi_R$ computes $R$ on input $q$). Then, there exists an $\mathsf{AM} \cap \mathsf{coAM}$ protocol $\Pi_M$ such that, given a probability $p^* \approx \Pr_{q \sim \mathcal{Q}_x}[q \in R]$ as advice, the protocol $\Pi_M$ simulates the reduction $M$ with probability at least $1 - 1/\mathsf{poly}(n)$.*

Let $R$ denote the complement of $R_3$, i.e., $R := \{\, q \in \{0,1\}^* \mid q \colon \alpha\text{-heavy} \,\}$. Using the generalized Feigenbaum–Fortnow protocol, we simulate the reduction $M$ to $R$ as follows. Firstly, we use the heavy-sample protocol of [9] in order to estimate $p^* \approx \Pr_{q \sim \mathcal{Q}_x}[q \colon \alpha\text{-heavy}]$. Secondly, using the lower bound protocol of [13], we argue that $R$ can be solved by some $\mathsf{AM}$-protocol $\Pi_R$ on average. Lastly, we use the protocol of Theorem 8 to simulate $M$. The details can be found in the full version of the paper.

We mention in passing the difficulty of Open Question 7, i.e., the reason why we were not able to combine our $\mathsf{AM} \cap \mathsf{coAM}$-type simulation algorithm with the non-black-box reduction *even conditionally*: The non-black-box reduction outlined in Subsection 3.1 reduces the promise problem whose Yes instance consists of $\mathrm{K}_U(x) \le \ell/5$ and No instance consists of $\mathrm{K}_U^R(x) > \mathsf{poly}(\ell, \log n)$ to an oracle $R$. In order to make sure that the promise problem is non-trivial, it is important that $R$ does not depend on $x$. On the other hand, in our simulation algorithm, we need to choose an oracle $R_x$ depending on the input $x$, which potentially makes the promise problem trivial. (For example, $\mathrm{K}_U^{R_x}(x)$ may be always close to 0.)

## 3.4 Subsequent Work

Inspired by this work, Allender's conjectures [4] were refuted under the plausible assumptions about the exponential-time hierarchy [21, 20]. Moreover, it turned out that the stretch of a hitting set generator construction is important. In [20], it was shown that there exists a function $G = \{G : \{0,1\}^{n - O(\log n)} \to \{0,1\}^n\}_{n \in \mathbb{N}}$ such that $\mathsf{NEXP} \cup \mathsf{coNEXP} \subseteq \mathsf{BPP}^R_{\parallel}$ for any oracle $R$ that avoids $G$. This result bypasses our limits of black-box reductions (Theorem 4) because $G$ extends its seed by a small amount of $O(\log n)$ whereas Theorem 4 requires that $G$ extends its seed by a constant factor. In [19], the approximation quality of non-black-box reductions of [18] is improved. Moreover, based on the improvement, it is shown that, under the assumption that $\mathsf{DistPH} \subseteq \mathsf{AvgP}$, the time-bounded $\mathsf{SAT}$-oracle Kolmogorov complexity of a string $x$ is equal to the time-bounded Kolmogorov complexity of $x$ up to an additive term of $O(\log n)$, for any string $x \in \{0,1\}^n$.

## References

1 Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *Proceedings of the Symposium on the Theory of Computing (STOC)*, pages 99–108, 1996. doi:10.1145/237814.237838.

2 Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on NP-hardness. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 701–710, 2006. doi:10.1145/1132516.1132614.

3 Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. Erratum for: on basing one-way functions on NP-hardness. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 795–796, 2010. doi:10.1145/1806689.1806797.

4 Eric Allender. Curiouser and Curiouser: The Link between Incompressibility and Complexity. In *Proceedings of the 8th Conference on Computability in Europe (CiE)*, pages 11–16, 2012. doi:10.1007/978-3-642-30870-3_2.

5 Eric Allender and Shuichi Hirahara. 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)*, pages 54:1–54:14, 2017. doi:10.4230/LIPIcs.MFCS.2017.54.

6 Benny Applebaum, Boaz Barak, and David Xiao. On Basing Lower-Bounds for Learning on Worst-Case Assumptions. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 211–220, 2008. doi:10.1109/FOCS.2008.35.

7 Andrej Bogdanov and Christina Brzuska. On Basing Size-Verifiable One-Way Functions on NP-Hardness. In *Proceedings of the Theory of Cryptography Conference (TCC)*, pages 1–6, 2015. doi:10.1007/978-3-662-46494-6_1.

8 Andrej Bogdanov and Luca Trevisan. Average-Case Complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1), 2006. doi:10.1561/0400000004.

9 Andrej Bogdanov and Luca Trevisan. On Worst-Case to Average-Case Reductions for NP Problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006. doi:10.1137/S0097539705446974.

10 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning Algorithms from Natural Proofs. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 10:1–10:24, 2016. doi:10.4230/LIPIcs.CCC.2016.10.

11 Joan Feigenbaum and Lance Fortnow. Random-Self-Reducibility of Complete Sets. *SIAM J. Comput.*, 22(5):994–1005, 1993. doi:10.1137/0222061.

12 Lance Fortnow. The Complexity of Perfect Zero-Knowledge. *Advances in Computing Research*, 5:327–343, 1989.

13 Shafi Goldwasser and Michael Sipser. Private Coins versus Public Coins in Interactive Proof Systems. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 59–68, 1986. doi:10.1145/12130.12137.

14 Dan Gutfreund and Salil P. Vadhan. Limitations of Hardness vs. Randomness under Uniform Reductions. In *Proceedings of the Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 469–482, 2008. doi:10.1007/978-3-540-85363-3_37.

15 Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency Improvements in Constructing Pseudorandom Generators from One-Way Functions. *SIAM J. Comput.*, 42(3):1405–1430, 2013. doi:10.1137/100814421.

16 Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A Pseudorandom Generator from any One-way Function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. doi:10.1137/S0097539793244708.

17 Shuichi Hirahara. Identifying an Honest $\mathsf{EXP}^{\mathsf{NP}}$ Oracle Among Many. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 244–263, 2015. doi:10.4230/LIPIcs.CCC.2015.244.

18 Shuichi Hirahara. Non-black-box Worst-case to Average-case Reductions within NP. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 247–258, 2018.

19 Shuichi Hirahara. Non-Disjoint Promise Problems from Meta-Computational View of Pseudorandom Generator Constructions. To appear in CCC'20, 2020.

20 Shuichi Hirahara. Unexpected Hardness Results for Kolmogorov Complexity Under Uniform Reductions. *To appear in STOC'20*, 2020.

21 Shuichi Hirahara. Unexpected Power of Random Strings. In *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*, pages 41:1–41:13, 2020. doi:10.4230/LIPIcs.ITCS.2020.41.

22 Shuichi Hirahara and Rahul Santhanam. On the Average-Case Complexity of MCSP and Its Variants. In *Proceedings of the Computational Complexity Conference (CCC)*, pages 7:1–7:20, 2017. doi:10.4230/LIPIcs.CCC.2017.7.

23 Shuichi Hirahara and Osamu Watanabe. Limits of Minimum Circuit Size Problem as Oracle. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 18:1–18:20, 2016. doi:10.4230/LIPIcs.CCC.2016.18.

24 Thomas Holenstein. Key agreement from weak bit agreement. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 664–673, 2005. doi:10.1145/1060590.1060689.

25 Thomas Holenstein. Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness. In *Proceedings of the Theory of Cryptography Conference (TCC)*, pages 443–461, 2006. doi:10.1007/11681878_23.

26 Russell Impagliazzo. Relativized Separations of Worst-Case and Average-Case Complexities for NP. In *Proceedings of the Conference on Computational Complexity (CCC)*, pages 104–114, 2011. doi:10.1109/CCC.2011.34.

27 Russell Impagliazzo and Avi Wigderson. Randomness vs Time: Derandomization under a Uniform Assumption. *J. Comput. Syst. Sci.*, 63(4):672–688, 2001. doi:10.1006/jcss.2001.1780.

28 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 73–79, 2000. doi:10.1145/335305.335314.

29 Leonid A. Levin. Average Case Complete Problems. *SIAM J. Comput.*, 15(1):285–286, 1986. doi:10.1137/0215020.

30 Noam Nisan and Avi Wigderson. Hardness vs Randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

31 Rafail Ostrovsky. One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs. In *Proceedings of the Structure in Complexity Theory Conference*, pages 133–138, 1991. doi:10.1109/SCT.1991.160253.

32 Aduri Pavan, Rahul Santhanam, and N. V. Vinodchandran. Some Results on Average-Case Hardness Within the Polynomial Hierarchy. In *Proceedings of the Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 188–199, 2006. doi:10.1007/11944836_19.

33 Steven Rudich. Super-bits, Demi-bits, and NP/qpoly-natural Proofs. In *Proceedings of the Randomization and Approximation Techniques in Computer Science (RANDOM/APPROX)*, pages 85–93, 1997. doi:10.1007/3-540-63248-4_8.

34 Rahul Santhanam. Circuit Lower Bounds for Merlin–Arthur Classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009. doi:10.1137/070702680.

35 Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom Generators without the XOR Lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001. doi:10.1006/jcss.2000.1730.

36 Luca Trevisan and Salil P. Vadhan. Pseudorandomness and Average-Case Complexity Via Uniform Reductions. *Computational Complexity*, 16(4):331–364, 2007. doi:10.1007/s00037-007-0233-x.

37 Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2005. doi:10.1007/s00037-004-0187-1.

38 Andrew Chi-Chih Yao. Theory and Applications of Trapdoor Functions (Extended Abstract). In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 80–91, 1982. doi:10.1109/SFCS.1982.45.

39 Chee-Keng Yap. Some Consequences of Non-Uniform Conditions on Uniform Classes. *Theor. Comput. Sci.*, 26:287–300, 1983. doi:10.1016/0304-3975(83)90020-8.