Electronic Colloquium on Computational Complexity, Report No. 9 (2021)

One-way Functions and Partial MCSP

Eric Allender[*] Mahdi Cheraghchi[†] Dimitrios Myrisiotis[‡] Harsha Tirumala[§]
Ilya Volkovich[¶]

February 1, 2021

Abstract

One-way functions (OWFs) are central objects of study in cryptography and computational complexity theory. In a seminal work, Liu and Pass (FOCS 2020) proved that the average-case hardness of computing time-bounded Kolmogorov complexity is *equivalent* to the existence of OWFs. It remained an open problem to establish such an equivalence for the average-case hardness of some NP-complete problem. In this paper, we make progress on this question by studying a polynomially-sparse variant of Partial Minimum Circuit Size Problem (Partial MCSP), which we call *Sparse Partial MCSP*, as follows.

1. First, we prove that if Sparse Partial MCSP is zero-error average-case hard on a polynomial fraction of its instances, then there exist OWFs.
2. Then, we observe that Sparse Partial MCSP is NP-complete under polynomial-time deterministic reductions. That is, there *are* NP-complete problems whose average-case hardness implies the existence of OWFs.
3. Finally, we prove that the existence of OWFs implies the nontrivial zero-error average-case hardness of Sparse Partial MCSP.

Thus the existence of OWFs is inextricably linked to the average-case hardness of this NP-complete problem.

Keywords. Minimum Circuit Size Problem, MCSP, Partial MCSP, one-way functions, average-case hardness, pseudorandom generators, pseudorandom functions, distinguishers, learning algorithms, reductions

1 Introduction

One-way functions (OWFs) —that is, functions that are easy to compute but hard to invert— are objects of great importance in cryptography and computational complexity. For example, it is known that OWFs exist if and only if pseudorandom generators exist [HILL99] and, moreover, if OWFs exist, then $\mathsf{P} \neq \mathsf{NP}$.

In this paper, we ask the following question: Can the existence of OWFs be based on the average-case hardness of some NP-complete problem? We take concrete steps toward giving an affirmative answer to this question, by presenting a candidate problem.

Of course, if there is any problem in NP that is hard on average, it follows that there is an NP-complete problem that also shares this property; see Proposition A.1. Thus, our main contribution is to present an NP-complete problem whose average-case complexity is tightly linked to the existence of OWFs.

[*] Department of Computer Science, Rutgers University, Piscataway, NJ, USA; E-mail: allender@cs.rutgers.edu.
[†] Department of EECS, University of Michigan, Ann Arbor, MI, USA; E-mail: mahdich@umich.edu.
[‡] Department of Computing, Imperial College London, London, UK; E-mail: d.myrisiotis17@ic.ac.uk.
[§] Department of Computer Science, Rutgers University, Piscataway, NJ, USA; E-mail: hs675@scarletmail.rutgers.edu.
[¶] Computer Science Department, Boston College, MA, USA; E-mail: ilya.volkovich@bc.edu.

ISSN 1433-8092

1.1 Prior work

An early goal in cryptographic research was to base the existence of cryptographically-secure one-way functions on the worst-case complexity of some NP-complete problem. This goal remains elusive; it was shown in [AGGM06] that no black-box argument of this sort can proceed based on non-adaptive reductions. Non-adaptive worst-case-to-average-case reductions were also studied by Bogdanov and Trevisan [BT06b], who showed that such reductions to sets in NP exist only for problems in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. Recent work by Nanashima [Nan21] holds open the possibility that the security of OWFs can be based on an *adaptive* black-box reduction, by first establishing a non-adaptive black-box reduction basing the existence of *auxiliary input one-way functions* on the worst-case complexity of an NP-complete problem, although this would also require non-relativizing techniques. Instead of worst-case hardness, the focus of our work is on average-case hardness assumptions. A nice survey on this area, that lays out many of the issues about one-way functions and average-case complexity, is the one by Bogdanov and Trevisan [BT06a].

Recently, Liu and Pass showed that OWFs exist if and only if computing the time-bounded Kolmogorov complexity of strings is average-case hard [LP20]. This is an important result, but it falls short of basing the existence of OWFs on an NP-complete problem, since

1. computing the time-bounded Kolmogorov complexity is not known to be NP-hard, and

2. the computation of this function can be done in polynomial time with an NP-oracle, but it does not translate directly to the average-case complexity of a language *in* NP.

Santhanam [San20] showed that a restricted type of hitting-set generator exists if and only if the Minimum Circuit Size Problem (MCSP) is zero-error average-case hard. Hirahara also proved similar results connecting the worst-case and the zero-error average-case complexity of problems related to MCSP and Kolmogorov complexity [Hir18].

More recently, Brzuska and Couteau [BC20] discuss basing OWFs on average-case hardness, noting that it remains an open question to do this for the general notion of average-case hardness; they do not consider zero-error average-case hardness as studied by Hirahara and Santhanam [HS17]. They present some negative results, indicating the difficulty of establishing the existence of fine-grained one-way functions, based on the existence of average-case hardness, via black-box reductions.

1.2 Our results

We connect the existence of OWFs to the average-case hardness of computing an NP-complete variant of MCSP.

Initially, we prove that the zero-error average-case hardness of a polynomially-sparse variant of Partial MCSP, which we term *Sparse Partial MCSP*, implies the existence of OWFs.

Theorem 1.1 (Informal)**.** *OWFs exist if Sparse Partial MCSP is zero-error hard-on-average on a polynomial fraction of its instances.*

Theorem 1.1 should be contrasted to the recent work by Liu and Pass [LP20], where they prove that the average-case hardness of computing the time-bounded Kolmogorov complexity K^t of a string is equivalent to the existence of OWFs. Moreover, Theorem 1.1 is complemented by proving that Sparse Partial MCSP is NP-complete.

Theorem 1.2 (Informal)**.** *Sparse Partial MCSP is* NP*-complete under deterministic polynomial-time reductions.*

Theorem 1.1 and Theorem 1.2 together answer in the affirmative the open question of whether there exists some NP-complete problem whose average-case hardness implies the existence of OWFs.

Moreover, Theorem 1.1 suggests an approach for excluding Impagliazzo's *Pessiland* [Imp95], that is, a version of our world where there are average-case hard problems in NP *and* there are no OWFs. This approach is based on the following observation. If Sparse Partial MCSP is NP-hard under average-case reductions, then by Theorem 1.1 the existence of an average-case hard problem in NP would imply the existence of OWFs. Therefore proving that Sparse Partial MCSP is NP-hard under average-case reductions excludes Pessiland.

Finally, we prove a *weak* converse of Theorem 1.1.

Theorem 1.3 (Informal)**.** *OWFs exist only if Sparse Partial MCSP is zero-error hard-on-average on an* exponential *fraction of its instances.*

Theorem 1.3 gives us some evidence that Theorem 1.1 is *not* a vacuous implication. Moreover, Theorem 1.1, Theorem 1.2, and Theorem 1.3 *almost* establish an equivalence between the existence of OWFs and the average-case hardness of an NP-complete problem.

1.3 Our techniques

Our main results are Theorem 1.1, Theorem 1.2, and Theorem 1.3. Below we provide some intuition regarding their proofs.

1. Theorem 1.1 is proved by

   (a) giving a zero-error average-case decision-to-search reduction for Sparse Partial MCSP (see Lemma 4.3) and

   (b) observing that a recent result by Liu and Pass [LP20], whereby they prove that the average-case hardness of a search variant of time-bounded Kolmogorov complexity K^t yields OWFs, can be adjusted to the case of Sparse Partial MCSP as well (see Lemma 4.4).

   The only two properties of time-bounded Kolmogorov complexity that are used in the paper by Liu and Pass, are as follows.

      i. One can create a string of low time-bounded Kolmogorov complexity in polynomial time. This can be done by running a universal Turing machine U on some string, for polynomially-many steps, and subsequently recording the output of U.

      ii. For any string x, the possible values of its K^t complexity are polynomially-many in $|x|$. In fact, there is a $c > 0$ such that, for any function $t : \mathbb{N} \to \mathbb{N}$ such that $t(n) \geq n$ for all $n \in \mathbb{N}$, and any string x, the possible values of $K^t(x)$ are at most $|x| + c$.

   As it turns out, both of these properties are satisfied even when one considers Partial MCSP with polynomially-many samples; we shall call this feature *polynomial sparsity.* This is true for the following reasons.

      i. A partial truth table of polynomial sparsity can be created in polynomial time by evaluating a polynomial-size circuit on polynomially-many inputs.

      ii. For every partial truth table of polynomial sparsity, the possible values of its circuit complexity are at most polynomial in its number of samples. This is so, as one may hardcode polynomially-many input-output pairs in a polynomial-size circuit, so that the resulting circuit agrees with all of these pairs.

2. Theorem 1.2 is proved by

   (a) noting that Sparse Partial MCSP is in NP (see Lemma 2.11) and

   (b) showing the NP-hardness of Sparse Partial MCSP (see Corollary 3.5) by outlining a deterministic polynomial-time reduction from Partial MCSP, which is NP-hard under deterministic polynomial-time reductions (see Theorem 2.12), to Sparse Partial MCSP; see Theorem 3.4. This reduction illustrates a simple padding argument.

3. Theorem 1.3 is proved by giving a proof of its contrapositive statement, as explained by the numbered items below.

   (a) Assume that Sparse Partial MCSP is easy on average under the uniform distribution.

   (b) By a corollary of Ilango, Loff, and Oliveira (see Corollary 2.24), for all $k > 0$, there exists a learning algorithm for $\mathsf{SIZE}[n^k]$ that works for infinitely many $n \in \mathbb{N}$; see Lemma 2.25.

   (c) By a learner-to-distinguisher reduction (see Lemma 2.33), for every polynomial-time computable Boolean function family $\{f_y\}_{y\in\{0,1\}^*}$, there is a distinguisher for $\{f_y\}_{y\in\{0,1\}^*}$.

   (d) By the correctness of the works by Håstad, Implagliazzo, Levin, and Luby [HILL99], and Goldreich, Goldwasser, and Micali [GGM86], there are no OWFs.

1.4 Open problems

Perhaps the most prominent question that this work leaves open, is the following:

> Can we prove that OWFs exist only if Sparse Partial MCSP is zero-error hard-on-average on a *polynomial* fraction of its instances?

That would give an *equivalence* between the existence of OWFs and the zero-error average-case hardness of Sparse Partial MCSP.

1.5 Paper organization

In Section 2 we give some background knowledge, and useful facts. In Section 3, we prove Theorem 1.2. Finally, we prove Theorem 1.1 in Section 4 and Theorem 1.3 in Section 5.

2 Preliminaries

2.1 Sets, strings, and projections

We denote the natural numbers by $\mathbb{N}$ and the positive reals by $\mathbb{R}_{>0}$. For any $n \in \mathbb{N}$, we denote the set $\{1, \dots, n\}$ by $[n]$. Let $x = (x_1, \dots, x_n) \in \{0,1\}^n$ be a string of length n; we write $|x| := n$. Let $S = \{i_1, \dots, i_k\}$ be a subset of $[n]$ of $1 \leq k \leq n$ elements, and x as before. Then, the *S-projection of x*, denoted $x|_S$, is the string $x_{i_1} \cdots x_{i_k} \in \{0,1\}^k$.

2.2 Boolean functions and their truth tables

We denote by $\mathcal{F}_n$ the class of all Boolean functions on n variables. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function from $\mathcal{F}_n$; we denote by $\texttt{tt}(f)$ its truth table $f(0^n)\, f\big(0^{n-1}1\big) \cdots f(1^n)$ of length 2^n.

Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$ from $\mathcal{F}_n$, a *partial truth table P of f* is a *multiset* of pairs $\{(x_i, f(x_i))\}_{i=1}^m$ in $\{0,1\}^n \times \{0,1\}$ for $1 \leq m \leq 2^n$. In this case, we say that the *dimension of P* is n, and that the *sparsity of P* is m. The partial truth table is defined as a multiset, instead of as a set of pairs, in order to simplify the analysis of average-case complexity. Let $m : \mathbb{N} \to \mathbb{N}$ be a function. If we use m to upper bound the sparsity of some partial truth table, then we shall call m a *sparsity function.*

We identify infinite Boolean functions $f : \{0,1\}^* \to \{0,1\}$ with collections $\{f_n\}_{n \in \mathbb{N}}$, whereby $f_n : \{0,1\}^n \to \{0,1\}$ for all $n \in \mathbb{N}$.

2.3 Probability

We will use the following useful facts from probability theory.

Lemma 2.1 (Averaging argument)**.** *If $X \in [0,1]$ is a random variable with $\mathbf{E}[X] = \mu$, then for all $0 < c < 1$ it is the case that*

$$\mathbf{Pr}[X \geq c\mu] \geq (1-c)\,\mu.$$

Lemma 2.2 (Markov's inequality)**.** *If X is a non-negative random variable with $\mathbf{E}[X] = \mu$, then for all $k > 0$ it is the case that*

$$\mathbf{Pr}[X \geq k\mu] \leq \frac{1}{k}.$$

Lemma 2.3 (Chernoff bound)**.** *Let $n \in \mathbb{N}$ and $X_1, \dots, X_n$ be Boolean random variables that are independent and identically distributed. Let $X := \sum_{i=1}^n X_i$ and $\mu := \mathbf{E}[X]$. Then, for all $0 \leq \delta \leq 1$, it is the case that*

$$\mathbf{Pr}[X \geq (1+\delta)\,\mu] \leq e^{-\frac{\delta^2 \mu}{3}}.$$

2.4 Circuit complexity

We consider Boolean circuits over the bounded fan-in $\{\wedge_2, \vee_2, \neg, 0, 1\}$ basis. Given a circuit, its *size* is the number of its gates. Let $s : \mathbb{N} \to \mathbb{N}$ be a function. If we use s to upper bound the size of some circuit, then we shall call s a *size function.*

The following result bounds from above the number of small circuits.

Lemma 2.4. *Let $s : \mathbb{N} \to \mathbb{N}$ be a size function such that $s(n) \geq n$ for all $n \in \mathbb{N}$. Then, for all $n \in \mathbb{N}$, the number of circuits with n inputs which are of size at most $s(n)$ is at most $2^{3s(n)\log s(n)}$.*

Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the *circuit complexity of f*, denoted $\mathrm{CC}(f)$, is the size of a minimum size circuit that computes f. For a size function $s : \mathbb{N} \to \mathbb{N}$, we denote by $\mathsf{SIZE}[s(n)]$ the class of Boolean functions $f = \{f_n\}_{n \in \mathbb{N}}$, whereby $f_n : \{0,1\}^n \to \{0,1\}$ for all $n \in \mathbb{N}$, such that $\mathrm{CC}(f_n) \leq s(n)$ for all $n \in \mathbb{N}$.

Let $n \in \mathbb{N}$ and $1 \leq m \leq 2^n$. Let $P := \{(x_i, b_i)\}_{i=1}^m$, where $(x_i, b_i) \in \{0,1\}^n \times \{0,1\}$ for all $1 \leq i \leq m$, be a partial truth table of dimension n and sparsity m, and C a circuit with n inputs. We say that C *agrees with* P if, for all $1 \leq i \leq m$, it is the case that $C(x_i) = b_j$, where j is the minimum element of $\{k \in \{1, \ldots, m\} \mid x_k = x_i\}$. This guarantees that every partial truth table P agrees with *some* circuit, because it is possible that there exists some $x \in \{0,1\}^n$ such that both $(x, 0)$ and $(x, 1)$ are elements of P. If P is a partial truth table, then the *circuit complexity of P*, denoted $\mathrm{CC}(P)$, is the size of a minimum-size circuit that agrees with P.

The following lemma asserts that *almost all* partial truth tables of polynomial sparsity have high circuit complexity.

Lemma 2.5. *Let $n \in \mathbb{N}$ be sufficiently large. Let $c > 0$ be a constant, $s := n^c$, $\ell := 100c$, and $m := n^\ell$. Then, it is the case that*

$$\Pr_{\{(x_i,b_i)\}_{i\in[m]}}\Big[\mathrm{CC}\Big(\{(x_i, b_i)\}_{i\in[m]}\Big) \leq s\Big] \leq \frac{1}{10}$$

where $x_1, \ldots, x_m \in \{0,1\}^n$ and $b_1, \ldots, b_m \in \{0,1\}$ are independent and uniformly random.

Proof. Let $N := n^\ell(n+1)$ be the representation size of $\{(x_i, b_i)\}_{i\in[m]}$. By Lemma 2.4, we have that

$$\Pr_{\{(x_i,b_i)\}_{i\in[m]}}\Big[\mathrm{CC}\Big(\{(x_i, b_i)\}_{i\in[m]}\Big) \leq s\Big] \leq \frac{(2^n)^{n^\ell} \cdot 2^{3n^c \log n^c}}{2^N},$$

where the factor $(2^n)^{n^\ell}$ comes from the fact that each circuit may agree with at most $(2^n)^{n^\ell}$ partial truth tables of dimension n and sparsity n^ℓ. Continuing, we have

$$\begin{aligned}\Pr_{\{(x_i,b_i)\}_{i\in[m]}}\Big[\mathrm{CC}\Big(\{(x_i, b_i)\}_{i\in[m]}\Big) \leq s\Big] &\leq \frac{2^{n^{\ell+1}} \cdot 2^{O(n^c \log n)}}{2^{n^\ell(n+1)}}\\ &= \frac{2^{O(n^c \log n)}}{2^{n^\ell}}\\ &= \frac{2^{O(n^c \log n)}}{2^{n^{100c}}}\\ &\leq \frac{1}{2^{n^{50c}}}\\ &\leq \frac{1}{10}.\end{aligned}$$

□

We also require the following helpful lemmas about Boolean circuits.

Lemma 2.6. *For all $n \in \mathbb{N}$, there exists a circuit of size $O(n)$ that on input $x, y \in \{0,1\}^n$ outputs 1 if and only if $x = y$.*

Lemma 2.7. *For all $n \in \mathbb{N}$, there exists a circuit of size $O(n)$ that on input $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ outputs $\bigvee_{i=1}^n x_i$.*

Lemma 2.8. *For all large $n \in \mathbb{N}$, if $P = \{(x_i, b_i)\}_{i=1}^m$ is a multiset of $1 \le m \le 2^n$ pairs from $\{0,1\}^n \times \{0,1\}$, then, there exists some circuit with n inputs and size at most $m \cdot n^2$ that agrees with P.*

Proof. Let E_n be a circuit of size $O(n)$ that on inputs $x, y \in \{0,1\}^n$ outputs 1 if and only if $x = y$. Such a circuit exists by Lemma 2.6. For all $1 \le i \le m$, let $z_i : \{0,1\}^n \to \{0,1\}$ be a function such that $z_i(y) := E_n(y, x_i) \wedge_2 b_i$ for all $y \in \{0,1\}^n$. Note that $z_i(y) = 1$ if and only if $y = x_i$ and $b_i = 1$, and z_i can be computed by a circuit of size $O(n)$. Let D_m be a circuit of size $O(m)$ that on input $z = (z_1, \ldots, z_m) \in \{0,1\}^m$ outputs $\bigvee_{i=1}^m z_i$. Such a circuit exists by Lemma 2.7. Define now a circuit C to be such that

$$C(y) := D_m(z_1(y), \ldots, z_m(y)) = \bigvee_{i=1}^{m} z_i(y)\,,$$

for all $y \in \{0,1\}^n$. Then, C agrees with P and has size $m \cdot O(n) + O(m) \le m \cdot n^2$. □

2.5 Uniform complexity

The following lemma upper bounds the circuit complexity of uniform computations.

Lemma 2.9 ([PF79]). *Let $T : \mathbb{N} \to \mathbb{N}$ be time-constructible. Let $n \in \mathbb{N}$ and $f : \{0,1\}^n \to \{0,1\}$ be a function computable by a Turing machine in time $O(T(n))$. Then, there exists a circuit of size $O(T(n) \log T(n))$ that computes f.*

In this work, we do not distinguish between Turing machines and algorithms. We say that an algorithm A is a *PPT algorithm* if A is a probabilistic polynomial-time algorithm. If A is a PPT algorithm that runs in time $p(n)$ for a polynomial p, then we denote by $A(x; r)$ the output of A on input $x \in \{0,1\}^*$ using random bits $r \in \{0,1\}^{p(|x|)}$. We say that an algorithm A is a *PPT oracle algorithm* if A is a PPT algorithm that has access to some oracle. If A is a PPT oracle algorithm that runs in time $p(n)$ for a polynomial p and has access to an oracle for a language $L \subseteq \{0,1\}^*$, then we denote by $A^L(x; r)$ the output of A^L on input $x \in \{0,1\}^*$ using random bits $r \in \{0,1\}^{p(|x|)}$.

2.6 Partial MCSP, and variants

Below, we provide the definitions of the main problems that we will encounter in this work.

2.6.1 Decision problems

Definition 2.10. *Let $m : \mathbb{N} \to \mathbb{N}$ be a sparsity function such that $1 \le m(n) \le 2^n$ for all $n \in \mathbb{N}$. The* $m(n)$-Sparse Partial Minimum Circuit Size Problem of dimension n (m-Sparse Partial MCSP of dimension n) *is defined as follows.*

- *Input: A partial truth table $P = \{(x_i, b_i)\}_{i=1}^{m(n)}$, where $(x_i, b_i) \in \{0,1\}^n \times \{0,1\}$ for all $1 \le i \le m(n)$, and a size parameter $0 \le k \le m(n) \cdot n^2 - 1$ in binary.*
- *Question: Is there a circuit of size at most k that agrees with P?*

When choosing the size parameter k of Definition 2.10 to be such that $0 \le k \le m(n) \cdot n^2 - 1$, we had in mind Lemma 2.8.

The following result is a standard observation [KC00].

Lemma 2.11. *For all sparsity functions $m : \mathbb{N} \to \mathbb{N}$, it is the case that m-Sparse Partial MCSP is in* NP.

Moreover, there is a choice for the sparsity function $m : \mathbb{N} \to \mathbb{N}$ that makes m-Sparse Partial MCSP NP-hard.

Theorem 2.12 ([HJLT96, ABF+08]; see also Ilango, Loff, and Oliveira [ILO20]). *There exists a sparsity function $m : \mathbb{N} \to \mathbb{N}$ such that m-Sparse Partial MCSP is* NP*-hard under deterministic polynomial-time reductions.*

2.6.2 Search problems

We shall also require the following *search* variant of m-Sparse Partial MCSP.

Definition 2.13. *Let $m : \mathbb{N} \to \mathbb{N}$ be a sparsity function such that $1 \le m(n) \le 2^n$ for all $n \in \mathbb{N}$. The* search variant of $m(n)$-Sparse Partial Minimum Circuit Size Problem of dimension n (Search m-Sparse Partial MCSP of dimension n) *is defined as follows.*

- *Input: A partial truth table $P = \{(x_i, b_i)\}_{i=1}^{m(n)}$, where $(x_i, b_i) \in \{0,1\}^n \times \{0,1\}$ for all $1 \le i \le m(n)$.*
- *Output: A description of a minimum-size circuit that agrees with P.*

2.7 One-way functions

In the following, a function μ is said to be *negligible* if for every polynomial p there exists a $n_0 \in \mathbb{N}$ such that for all naturals $n > n_0$ it is the case that $\mu(n) \le 1/p(n)$.

Definition 2.14. *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. We say that f is a* one-way function (OWF) *if for every PPT algorithm A there exists a negligible function μ such that for all $n \in \mathbb{N}$ it is the case that*

$$\Pr_{x \sim \{0,1\}^n, r}\left[A(1^n, f(x)\,; r) \in f^{-1}(f(x))\right] < \mu(n)$$

where the size of r is equal to the running time of A.

We will also employ the following weaker notion of OWFs.

Definition 2.15. *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. We say that f is an* α-weak one-way function (α-weak OWF) *if for every PPT algorithm A and all sufficiently large $n \in \mathbb{N}$ it is the case that*

$$\Pr_{x \sim \{0,1\}^n, r}\left[A(1^n, f(x)\,; r) \in f^{-1}(f(x))\right] < 1 - \alpha(n)$$

where the size of r is equal to the running time of A. We say that f is a weak one-way function (weak OWF) *if there exists some polynomial $q > 0$ such that f is a $(1/q)$-weak OWF.*

Yao [Yao82] proved that the existence of weak OWFs implies the existence of OWFs.

Theorem 2.16 ([Yao82])**.** *Assume that there exists a weak one-way function. Then there exists a one-way function.*

2.8 Average-case hardness/easiness

2.8.1 Decision problems

A *heuristic H* is a PPT algorithm that, on input any $x \in \{0,1\}^n$, outputs a value in $\{0,1\}$ along each computation path.

Definition 2.17 (Average-case hardness)**.** *Let $\alpha : \mathbb{N} \to [0,1]$ be a failure parameter function. We say that a function $f : \{0,1\}^n \to \{0,1\}$ is* α-hard-on-average (α-HoA) *if for all heuristics H and all sufficiently large $n \in \mathbb{N}$ it is the case that*

$$\Pr_{x \sim \{0,1\}^n, r}[H(x; r) = f(x)] \le 1 - \alpha(n)$$

where the size of r is equal to the running time of H.

Definition 2.18 (Average-case easiness)**.** *Let $\alpha : \mathbb{N} \to [0,1]$ be a success parameter function. We say that a function $f : \{0,1\}^n \to \{0,1\}$ is* α-easy-on-average (α-EoA) *if f is not $(1-\alpha)$-hard-on-average.*

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. A *zero-error heuristic H for f* is a PPT algorithm that on input any $x \in \{0,1\}^n$ outputs a value in $\{0,1\} \cup \{?\}$, along each computation path, that is either equal to $f(x)$ or ?.

Definition 2.19 (Zero-error average-case hardness; following Hirahara and Santhanam [HS17]). *Let $\alpha : \mathbb{N} \to [0,1]$ be a failure parameter function. We say that a function $f : \{0,1\}^n \to \{0,1\}$ is* zero-error α-hard-on-average (zero-error α-HoA) *if for all zero-error heuristics H for f and all sufficiently large $n \in \mathbb{N}$ it is the case that*

$$\Pr_{x \sim \{0,1\}^n, r}[H(x;r) = ?] \geq \alpha(n)$$

where the size of r is equal to the running time of H.

Definition 2.20 (Zero-error average-case easiness; following Hirahara and Santhanam [HS17]). *Let $\alpha : \mathbb{N} \to [0,1]$ be a success parameter function. We say that a function $f : \{0,1\}^n \to \{0,1\}$ is* zero-error α-easy-on-average (zero-error α-EoA) *if f is not zero-error $(1-\alpha)$-hard-on-average.*

2.8.2 Search problems

Let $R \subseteq \{0,1\}^n \times \{0,1\}^*$ be a search problem. A *heuristic H* is a PPT algorithm that, on input any $x \in \{0,1\}^n$, outputs a value in $\{0,1\}^*$ along each computation path.

The notions of average-case hardness and easiness for search problems are defined in a fashion similar to that of decision problems; see Definition 2.17 and Definition 2.18.

2.8.3 Average-case easiness of Sparse Partial MCSP, with advantage

We require the following definition by Ilango, Loff, and Oliveira [ILO20].

Definition 2.21 ([ILO20]). *A randomized algorithm B* solves m-Sparse Partial MCSP with advantage γ for a size parameter *s if, for every $f : \{0,1\}^n \to \{0,1\}$ with $\mathrm{CC}(f) \leq s$, it is the case that*

$$\left| \Pr_{\{x_i\}_{i \in [m]}, r}[B(1^n, \{(x_1, f(x_1)), \ldots, (x_m, f(x_m))\}; r) = 1] \right.$$
$$\left. - \Pr_{\{(x_i, b_i)\}_{i \in [m]}, r}[B(1^n, \{(x_1, b_1), \ldots, (x_m, b_m)\}; r) = 1] \right| \geq \gamma(n)$$

where $x_1, \ldots, x_m \in \{0,1\}^n$ and $b_1, \ldots, b_m \in \{0,1\}$ are independent and uniformly random, and r has size equal to the running time of B. In this case, we may also say that Sparse Partial MCSP is easy-on-average with advantage γ (EoA with advantage γ)*.*

2.9 Learning algorithms and applications

2.9.1 Learning algorithms

For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, an *example oracle for f*, denoted $\mathrm{EX}(f)$, is a procedure that when invoked returns a pair $(x, f(x))$ where $x \sim \{0,1\}^n$.

Let $0 < \varepsilon < 1$. We say that a circuit C with n inputs is *ε-close* to a function $f : \{0,1\}^n \to \{0,1\}$ if

$$\Pr_{x \sim \{0,1\}^n}[C(x) \neq f(x)] \leq \varepsilon.$$

We recall the following intuitive notion of learning by Valiant [Val84].

Definition 2.22 ([Val84]). *A randomized algorithm* learns *a class of Boolean functions $\mathcal{F}$ with accuracy error ε and confidence error δ if, for all $f \in \mathcal{F}$ of the form $f = \{f_n\}_{n \in \mathbb{N}}$, whereby $f_n : \{0,1\}^n \to \{0,1\}$ for all $n \in \mathbb{N}$, when A is given access to example oracle $\mathrm{EX}(f_n)$ it is the case that*

$$\Pr_{\mathrm{EX}(f_n), r}\left[A^{\mathrm{EX}(f_n)}(1^n; r) \text{ outputs a description of a circuit that is } \varepsilon(n)\text{-close to } f_n\right] \geq 1 - \delta(n),$$

for all $n \in \mathbb{N}$, where the size of r is equal to the running time of A.

We will use the following result by Ilango, Loff, and Oliveira [ILO20].

Theorem 2.23 ([ILO20, Theorem 36, Item (2)])**.** *If for every $c \in \mathbb{N}$ there exists $\ell \in \mathbb{N}$ such that for all n the problem n^ℓ-Sparse Partial MCSP of dimension n can be solved on average in polynomial time with advantage $1/10$ and size parameter s, then for every $a \in \mathbb{N}$ and all $n \in \mathbb{N}$ the class $\mathsf{SIZE}[n^a]$ can be learned in polynomial time with accuracy error $1/n$ and confidence error $1/n$.*

By examining the proof of Theorem 2.23, we get the following corollary.

Corollary 2.24. *If for every $c \in \mathbb{N}$ there exists $\ell \in \mathbb{N}$ such that for infinitely many n the problem n^ℓ-Sparse Partial MCSP of dimension n can be solved on average in polynomial time with advantage $1/10$ and size parameter s, then for every $a > 0$ and infinitely many $n \in \mathbb{N}$ the class $\mathsf{SIZE}[n^a]$ can be learned in polynomial time with accuracy error $1/n$ and confidence error $1/n$.*

2.9.2 Applications

We will require the following result, which asserts that heuristics with good average-case performance guarantees can be used to design learning algorithms.

Lemma 2.25. *If for all $\ell > 0$ n^ℓ-Sparse Partial MCSP is zero-error $(1-1/h)$-EoA, for a function $h : \mathbb{N} \to \mathbb{R}_{>0}$ such that $h(N) := 2^{N/100}$ for all $N \in \mathbb{N}$, then for every $a > 0$ and infinitely many $n \in \mathbb{N}$ the class $\mathsf{SIZE}[n^a]$ can be learned in polynomial time with accuracy error $\varepsilon = 1/n$ and confidence error $\delta = 1/n$.*

Proof. We will apply Corollary 2.24, by proving that if, for all $\ell > 0$, n^ℓ-Sparse Partial MCSP is zero-error $(1-1/h)$-EoA, then, for all $\ell > 0$, n^ℓ-Sparse Partial MCSP is EoA with advantage $1/10$, according to Definition 2.21. The desired result will then follow from Corollary 2.24.

To this end, assume that for all $\ell > 0$, n^ℓ-Sparse Partial MCSP is zero-error $(1-1/h)$-EoA. In what follows, let $N := n^\ell\,(n+1) + (\ell+2)\log n$ be the size of the n^ℓ-Sparse Partial MCSP instances of dimension n; see Definition 2.10.

Let $c > 0$ be arbitrary, and $\ell := 100c$. Let H be the heuristic that witnesses the fact that n^ℓ-Sparse Partial MCSP is zero-error $(1-1/h)$-EoA, and assume that H runs in time $q(N)$ for some polynomial q. Let $n \in \mathbb{N}$ be sufficiently large and such that H satisfies its average-case performance guarantees on inputs of size n; see Definition 2.18. Let $s := n^c$. Let H^* be a probabilistic algorithm such that, for all partial truth tables P of dimension n and sparsity n^ℓ and all random strings $r \in \{0,1\}^{q(N)}$, $H^*(1^n, P; r) := H(P, s; r)$.

Let $f : \{0,1\}^n \to \{0,1\}$ be such that $\mathrm{CC}(f) \le s$. Then, by the definition of H^*, we have that the first term of the LHS of the inequality of Definition 2.21 is

$$\begin{aligned}&\Pr_{\{x_i\}_{i\in[m]},r}\left[H^*(1^n, \{(x_1, f(x_1)), \ldots, (x_m, f(x_m))\}; r) = 1\right] \\ &\quad= \Pr_{\{x_i\}_{i\in[m]},r}\left[H(\{(x_1, f(x_1)), \ldots, (x_m, f(x_m))\}, s; r) = 1\right].\end{aligned}$$

We claim that this probability is sufficiently large.

Claim 2.26. *It is the case that*

$$\Pr_{\{x_i\}_{i\in[m]},r}\left[H(\{(x_1, f(x_1)), \ldots, (x_m, f(x_m))\}, s; r) = 1\right] \ge \frac{1}{5}.$$

Proof. Towards a contradiction, assume that

$$\Pr_{\{x_i\}_{i\in[m]},r}\left[H(\{(x_1, f(x_1)), \ldots, (x_m, f(x_m))\}, s; r) = 1\right] < \frac{1}{5}.$$

Then, the number of inputs to H that make H output “?” is greater than

$$K := 2^{n^\ell \cdot n} \cdot 2^{q(N)} \cdot \left(1 - \frac{1}{5}\right) = 2^{n^\ell \cdot n + q(N)} \cdot \frac{4}{5}.$$

However, $K < 2^{N+q(N)}/h(N)$, by our assumption on the average-case performance of H. So

$$2^{n^\ell \cdot n + q(N)} \cdot \frac{4}{5} < \frac{1}{h(N)} \cdot 2^{N+q(N)} = \frac{1}{2^{N/100}} \cdot 2^{n^\ell \cdot (n+1) + (\ell+2)\log n + q(N)}$$

or

$$\frac{4}{5} < \frac{1}{2^{N/100}} \cdot 2^{n^{\ell}+(\ell+2)\log n} \leq \frac{1}{2^{n^{\ell+1}/100}} \cdot 2^{100n^{\ell}} \leq \frac{1}{2^{200n^{\ell}}} \cdot 2^{100n^{\ell}} = \frac{1}{2^{100n^{\ell}}} < \frac{4}{500};$$

this yields a contradiction. (Claim 2.26) □

We now turn to the second term of the LHS of the inequality of Definition 2.21. To this end, we have

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\{(x_i,b_i)\}_{i\in[m]},r}\left[H^*(1^n, \{(x_1,b_1),\ldots,(x_m,b_m)\}; r) = 1\right] \\
&\quad = \mathop{\mathbf{Pr}}_{\{(x_i,b_i)\}_{i\in[m]},r}\left[H(\{(x_1,b_1),\ldots,(x_m,b_m)\}, s; r) = 1\right] \\
&\quad \leq \mathop{\mathbf{Pr}}_{\{(x_i,b_i)\}_{i\in[m]},r}\left[H(\{(x_1,b_1),\ldots,(x_m,b_m)\}, s; r) = 1 \mid \mathrm{CC}\Big(\{(x_i,b_i)\}_{i\in[m]}\Big) > s\right] \\
&\qquad + \mathop{\mathbf{Pr}}_{\{(x_i,b_i)\}_{i\in[m]}}\left[\mathrm{CC}\Big(\{(x_i,b_i)\}_{i\in[m]}\Big) \leq s\right] \\
&\quad = 0 + \mathop{\mathbf{Pr}}_{\{(x_i,b_i)\}_{i\in[m]}}\left[\mathrm{CC}\Big(\{(x_i,b_i)\}_{i\in[m]}\Big) \leq s\right],
\end{aligned}$$

as H is zero-error, or

$$\leq \frac{1}{10},$$

by Lemma 2.5.

Therefore, by Claim 2.26 and the discussion above,

$$\begin{aligned}
&\Bigg|\mathop{\mathbf{Pr}}_{\{x_i\}_{i\in[m]},r}\left[H^*(1^n, \{(x_1,f(x_1)),\ldots,(x_m,f(x_m))\}; r) = 1\right] \\
&\qquad - \mathop{\mathbf{Pr}}_{\{(x_i,b_i)\}_{i\in[m]},r}\left[H^*(1^n, \{(x_1,b_1),\ldots,(x_m,b_m)\}; r) = 1\right]\Bigg| \\
&\qquad\quad \geq \frac{1}{5} - \frac{1}{10} \\
&\qquad\quad = \frac{1}{10},
\end{aligned}$$

as desired. □

2.10 Pseudorandom generators

We recount the notion of fooling a PPT algorithm.

Definition 2.27. *Let $\ell, n \in \mathbb{N}$, such that $\ell < n$, and $G : \{0,1\}^{\ell} \to \{0,1\}^{n}$ be a function. Let A be a PPT algorithm, and $0 < \varepsilon < 1$. We say that G* is a function that ε-fools A*, if*

$$\left|\mathop{\mathbf{Pr}}_{x\sim\{0,1\}^n}[A(x)=1] - \mathop{\mathbf{Pr}}_{y\sim\{0,1\}^{\ell}}[A(G(y))=1]\right| < \varepsilon.$$

The notion of fooling is used in the definition of a *pseudorandom generator (PRG)*. We will make use of PRGs $\{G_n\}_{n\in\mathbb{N}}$ for which there exists a function $\ell : \mathbb{N} \to \mathbb{N}$ that satisfies $\ell(n) < n$ for all $n \in \mathbb{N}$, such that $G_n : \{0,1\}^{\ell(n)} \to \{0,1\}^{n}$ for all $n \in \mathbb{N}$ and the following holds: For every PPT algorithm A there exists a function $\varepsilon : \mathbb{N} \to [0,1]$ such that for all $n \in \mathbb{N}$ it is the case that G_n is a function that $\varepsilon(n)$-fools A.

Håstad, Impagliazzo, Levin, Luby [HILL99] have shown that the existence of OWFs implies the existence of PRGs.

Theorem 2.28 (OWFs imply PRGs; see Håstad, Impagliazzo, Levin, Luby [HILL99])**.** *If there exists a OWF, then for every $c > 0$ there exists a function $\{G_n\}_{n\in\mathbb{N}}$, whereby $G_n : \{0,1\}^{n^{1/c}} \to \{0,1\}^{n}$ for all $n \in \mathbb{N}$, such that for every PPT algorithm A there is a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that for all $n \in \mathbb{N}$ it is the case that G_n is a function that $\varepsilon(n)$-fools A.*

2.11 Pseudorandom functions and applications

2.11.1 Pseudorandom functions

We will also require the notions of pseudorandom function families and distinguishers for function families.

Definition 2.29. *Let $\{f_y\}_{y\in\{0,1\}^*}$ be a family of functions such that $f_y : \{0,1\}^{|y|} \to \{0,1\}$ for all $y \in \{0,1\}^*$, and there is a polynomial-time algorithm that computes $f_y(x)$ given y and $x \in \{0,1\}^{|y|}$. We say that the function family $\{f_y\}_{y\in\{0,1\}^*}$ is a* pseudorandom function family (PRF family) *if for all PPT oracle algorithms A there is a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that for all sufficiently large $n \in \mathbb{N}$ it is the case that*

$$\left|\mathop{\mathbf{Pr}}_{y\sim\{0,1\}^n,r}\left[A^{f_y}(1^n;r) = 1\right] - \mathop{\mathbf{Pr}}_{g\sim\mathcal{F}_n,r}\left[A^g(1^n;r) = 1\right]\right| < \varepsilon(n)$$

where the size of r is equal to the running time of A.

Definition 2.30. *Let $\{f_y\}_{y\in\{0,1\}^*}$ be a family of functions such that $f_y : \{0,1\}^{|y|} \to \{0,1\}$ for all $y \in \{0,1\}^*$, and there is a polynomial-time algorithm that computes $f_y(x)$ given y and $x \in \{0,1\}^{|y|}$. We say that a PPT oracle algorithm A is a* distinguisher for $\{f_y\}_{y\in\{0,1\}^*}$ *if for all negligible functions $\varepsilon : \mathbb{N} \to [0,1]$ and infinitely many $n \in \mathbb{N}$ it is the case that*

$$\left|\mathop{\mathbf{Pr}}_{y\sim\{0,1\}^n,r}\left[A^{f_y}(1^n;r) = 1\right] - \mathop{\mathbf{Pr}}_{g\sim\mathcal{F}_n,r}\left[A^g(1^n;r) = 1\right]\right| \geq \varepsilon(n)$$

where the size of r is equal to the running time of A.

Note how a distinguisher violates the guarantees of a PRF family. Below, we present a result by Goldreich, Goldwasser, and Micali [GGM86], where they proved that the existence of PRGs implies the existence of PRFs.

Theorem 2.31 (PRGs imply PRFs; see Goldreich, Goldwasser, and Micali [GGM86])**.** *If there exists a function $\{G_n\}_{n\in\mathbb{N}}$, whereby $G_n : \{0,1\}^{n/2} \to \{0,1\}^n$ for all $n \in \mathbb{N}$, such that for every PPT algorithm A there is a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that for all $n \in \mathbb{N}$ it is the case that G_n is a function that $\varepsilon(n)$-fools A, then there exists a PRF family.*

Theorem 2.28 and Theorem 2.31 yield the following corollary.

Corollary 2.32 (OWFs imply PRFs)**.** *If there exists a OWF, then there exists a PRF family.*

Proof. Assume that there exists a OWF. Then, by Theorem 2.28 and $c := 2$, there exists a function $\{\Gamma_n\}_{n\in\mathbb{N}}$, whereby $\Gamma_n : \{0,1\}^{n^{1/2}} \to \{0,1\}^n$ for all $n \in \mathbb{N}$, such that for every PPT algorithm A there is a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that for all $n \in \mathbb{N}$ it is the case that Γ_n is a function that $\varepsilon(n)$-fools A. For all $n \in \mathbb{N}$, let $G_n : \{0,1\}^{n/2} \to \{0,1\}^n$ be such that for all $x \in \{0,1\}^{n/2}$ it is the case that

$$G_n(x) := \Gamma_n\Big(x|_{[n^{1/2}]}\Big) .$$

We claim that $\{G_n\}_{n\in\mathbb{N}}$ is such that for every PPT algorithm A there is a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that for all $n \in \mathbb{N}$ it is the case that G_n is a function that $\varepsilon(n)$-fools A.

To this end, fix an arbitrary PPT algorithm A. Then, for all $n \in \mathbb{N}$, it is the case that

$$\begin{aligned}
&\left|\mathop{\mathbf{Pr}}_{x\sim\{0,1\}^n}[A(x) = 1] - \mathop{\mathbf{Pr}}_{y\sim\{0,1\}^{n/2}}[A(G_n(y)) = 1]\right| \\
&\quad = \left|\mathop{\mathbf{Pr}}_{x\sim\{0,1\}^n}[A(x) = 1] - \mathop{\mathbf{Pr}}_{y\sim\{0,1\}^{n/2}}\left[A\Big(\Gamma_n\Big(y|_{[n^{1/2}]}\Big)\Big) = 1\right]\right| \\
&\quad = \left|\mathop{\mathbf{Pr}}_{x\sim\{0,1\}^n}[A(x) = 1] - \frac{\left|\left\{y \in \{0,1\}^{n/2} \mid A\Big(\Gamma_n\Big(y|_{[n^{1/2}]}\Big)\Big) = 1\right\}\right|}{2^{n/2}}\right|
\end{aligned}$$

$$
\begin{aligned}
&= \left| \Pr_{x\sim\{0,1\}^n}[A(x)=1] - \frac{2^{n/2-n^{1/2}}\left|\left\{z\in\{0,1\}^{n^{1/2}} \mid A(\Gamma_n(z))=1\right\}\right|}{2^{n/2}} \right| \\
&= \left| \Pr_{x\sim\{0,1\}^n}[A(x)=1] - \Pr_{z\sim\{0,1\}^{n^{1/2}}}[A(\Gamma_n(z))=1] \right| \\
&< \varepsilon(n)\,,
\end{aligned}
$$

by the correctness of Γ_n. The result now follows by Theorem 2.31. □

2.11.2 Applications

An important observation is that a learning algorithm for polynomial-size circuits may be used to create distinguishers for polynomial-time computable function families.

Lemma 2.33 (See also Oliveira and Santhanam [OS17, Theorem 8]). *Assume that for every $a > 0$ and infinitely many $n \in \mathbb{N}$ the class $\mathsf{SIZE}[n^a]$ can be learned in polynomial time with accuracy error $\varepsilon = 1/n$ and confidence error $\delta = 1/n$. Then, for all function families $\{f_y\}_{y\in\{0,1\}^*}$ such that $f_y : \{0,1\}^{|y|} \to \{0,1\}$ for all $y \in \{0,1\}^*$, and there is a polynomial-time algorithm that computes $f_y(x)$ given y and $x \in \{0,1\}^{|y|}$, there is a distinguisher for $\{f_y\}_{y\in\{0,1\}^*}$.*

Proof. Let $\{f_y\}_{y\in\{0,1\}^*}$ be a function family such that $f_y : \{0,1\}^{|y|} \to \{0,1\}$ for all $y \in \{0,1\}^*$, and there is a polynomial-time algorithm that computes $f_y(x)$ given y and $x \in \{0,1\}^{|y|}$. In particular, for all $y \in \{0,1\}^*$ it is the case that $f_y : \{0,1\}^{|y|} \to \{0,1\}$ is computable in time $|y|^{k/2}$ for some $k > 0$. By Lemma 2.9, for all $y \in \{0,1\}^*$ it is the case that $f_y : \{0,1\}^{|y|} \to \{0,1\}$ is computable by some circuit of size

$$O\Big(|y|^{k/2}\log|y|\Big) \le |y|^k\,.$$

Let A be the PPT learning algorithm for $\mathsf{SIZE}\big[n^k\big]$ that works for infinitely many $n \in \mathbb{N}$, and that runs in time $q(n)$ for some polynomial q, as guaranteed to exist by our assumption. In particular, let $n \in \mathbb{N}$ be sufficiently large and such that A satisfies its learning guarantees on inputs of size n. Now let $t := n^{100}$ and define D to be a probabilistic oracle algorithm as follows.

> On input 1^n, random bits $r := (r', r'') \in \{0,1\}^{O(q(n))} \times \{0,1\}^{t\cdot n}$, and given oracle access to some function $g : \{0,1\}^n \to \{0,1\}$, the algorithm D runs A on 1^n using random bits $r' \in \{0,1\}^{O(q(n))}$ to get a hypothesis h for g, whereby simulating calls to EX(g) by using the oracle for g. Then, D samples t strings $x_1, \ldots, x_t$ from $\{0,1\}^n$ using random bits $r'' \in \{0,1\}^{t\cdot n}$ and uses the oracle for g to compute
>
> $$\alpha := \frac{|\{i \in [t] \mid h(x_i) = g(x_i)\}|}{t}.$$
>
> Finally, if $\alpha \ge 2/3$, then D outputs 1; else, D outputs 0.

Note that D runs in time polynomial in n. We will now prove that D is a distinguisher for $\{f_y\}_{y\in\{0,1\}^*}$. To this end, we will show that D satisfies Definition 2.30.

The first term of the LHS of the inequality of Definition 2.30 is

$$
\begin{aligned}
\Pr_{y\sim\{0,1\}^n, r}\big[D^{f_y}(1^n; r) = 1\big] &= \Pr_{y,r}\left[\alpha \ge \frac{2}{3}\right] \\
&= \Pr_{y,r}\left[\frac{|\{i\in[t] \mid h(x_i) = f_y(x_i)\}|}{t} \ge \frac{2}{3}\right] \\
&\ge \Pr_{y,r}\left[\frac{|\{i\in[t] \mid h(x_i) = f_y(x_i)\}|}{t} \ge \frac{3}{4}\left(1 - \frac{1}{n}\right)\right] \\
&= \Pr_{y,r}\left[\frac{|\{i\in[t] \mid h(x_i) = f_y(x_i)\}|}{t} \ge \frac{3}{4}\left(1 - \varepsilon\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&\geq \Pr_{y,r}\left[\frac{|\{i \in [t] \mid h(x_i) = f_y(x_i)\}|}{t} \geq \frac{3}{4}(1-\varepsilon) \mid h \text{ is } \varepsilon\text{-close to } f_y\right] \\
&\quad \cdot \Pr_{y,r}[h \text{ is } \varepsilon\text{-close to } f_y]\,.
\end{aligned}
$$

For all $1 \leq i \leq t$, let X_i be a Boolean variable such that $X_i := 1$ if and only if $h(x_i) = f_y(x_i)$. Then,

$$
\begin{aligned}
\Pr_{y,r}\big[D^{f_y}(1^n;r)=1\big] &\geq \Pr_{y,r}\left[\frac{\sum_{i=1}^t X_i}{t} \geq \frac{3}{4}(1-\varepsilon) \mid h \text{ is } \varepsilon\text{-close to } f_y\right] \cdot \Pr_{y,r}[h \text{ is } \varepsilon\text{-close to } f_y] \\
&\geq \Pr_{y,r}\left[\frac{\sum_{i=1}^t X_i}{t} \geq \frac{3}{4}\mathop{\mathbf{E}}_{y,r}\left[\frac{\sum_{i=1}^t X_i}{t}\right]\right](1-\delta)\,,
\end{aligned}
$$

as $\mathrm{CC}(f_y) \leq |y|^k = n^k$, or

$$
\geq \frac{1}{4}\mathop{\mathbf{E}}_{y,r}\left[\frac{\sum_{i=1}^t X_i}{t}\right](1-\delta)\,,
$$

by Lemma 2.1, or

$$
\begin{aligned}
&\geq \frac{1}{4}(1-\varepsilon)(1-\delta) \\
&= \frac{1}{4}\left(1-\frac{1}{n}\right)\left(1-\frac{1}{n}\right) \\
&\geq \frac{1}{4} - \frac{1}{8} \\
&= \frac{1}{8}.
\end{aligned}
$$

We now turn to the second term of the LHS of the inequality of Definition 2.30. To this end, we have

$$
\begin{aligned}
\Pr_{g\sim\mathcal{F}_n,r}[D^g(1^n;r)=1] &= \Pr_{g,r}\left[\alpha \geq \frac{2}{3}\right] \\
&= \Pr_{g,r}\left[\frac{|\{i \in [t] \mid h(x_i) = g(x_i)\}|}{t} \geq \frac{2}{3}\right] \\
&\leq \Pr_{g,r}\left[\frac{|\{i \in [t] \mid h(x_i) = g(x_i)\}|}{t} \geq \frac{2}{3} \mid \{x_i\}_{i=1}^t \text{ are distinct}\right] \\
&\quad + \Pr\left[\{x_i\}_{i=1}^t \text{ are not distinct}\right] \\
&= \Pr_{b,r}\left[|\{i \in [t] \mid h(x_i) = b_i\}| \geq \frac{2t}{3} \mid \{x_i\}_{i=1}^t \text{ are distinct}\right] \\
&\quad + \Pr\left[\{x_i\}_{i=1}^t \text{ are not distinct}\right],
\end{aligned}
$$

where $b_1, \ldots, b_t \in \{0,1\}$ are independent and uniformly random, and $b := (b_1, \ldots, b_t)$.

Similarly as above, for all $1 \leq i \leq t$, let X_i be a Boolean variable such that $X_i := 1$ if and only if $h(x_i) = b_i$. Let Y be the event that all of the x_i are distinct. Then,

$$
\begin{aligned}
\Pr_{g\sim\mathcal{F}_n,r}[D^g(1^n;r)=1] &\leq \Pr_{b,r}\left[\sum_{i=1} X_i \geq \frac{2t}{3} \mid Y\right] + \Pr\left[\{x_i\}_{i=1}^t \text{ are not distinct}\right] \\
&= \Pr_{b,r}\left[\sum_{i=1} X_i \geq \frac{4}{3}\cdot\frac{t}{2} \mid Y\right] + \Pr[\exists i,j \in [t] : i \neq j \text{ and } x_i = x_j] \\
&\leq \Pr_{b,r}\left[\sum_{i=1} X_i \geq \frac{4}{3}\cdot\frac{t}{2} \mid Y\right] + \binom{t}{2}2^{-n},
\end{aligned}
$$

by a union bound, or

$$\begin{aligned}
&= \Pr_{b,r}\left[\sum_{i=1}^{t} X_i \geq \left(1+\frac{1}{3}\right)\mathop{\mathbf{E}}_{b,r}\left[\sum_{i=1}^{t} X_i \mid Y\right] \mid Y\right] + \binom{n^{100}}{2}2^{-n} \\
&\leq \left(e^{-\frac{1/9}{3}}\right)^{\mathbf{E}_{b,r}\left[\sum_{i=1}^{t} X_i \mid Y\right]} + \frac{n^{200}}{2^n},
\end{aligned}$$

by Lemma 2.3, or

$$\begin{aligned}
&\leq \left(e^{-1/27}\right)^{t/2} + \frac{1}{32} \\
&= e^{-t/54} + \frac{1}{32} \\
&= e^{-n^{100}/54} + \frac{1}{32} \\
&\leq \frac{1}{32} + \frac{1}{32} \\
&= \frac{1}{16}.
\end{aligned}$$

Therefore,

$$\left|\Pr_{y\sim\{0,1\}^n,r}\left[D^{f_y}(1^n;r)=1\right] - \Pr_{g\sim\mathcal{F}_n,r}\left[D^{g}(1^n;r)=1\right]\right| \geq \frac{1}{8} - \frac{1}{16} = \frac{1}{16} = \Omega(1)$$

and as every negligible function $\varepsilon : \mathbb{N} \to [0,1]$ is such that $\varepsilon(n) = o(1) < \Omega(1)$, the desired result follows. □

3 Sparse Partial MCSP is NP-complete

In this section, we prove Theorem 1.2. By Theorem 2.12, the NP-hardness of Sparse Partial MCSP would follow by giving a reduction from m-Sparse Partial MCSP to n^ℓ-Sparse Partial MCSP for all $\ell > 0$. This is what we do in Lemma 3.2 and Lemma 3.3 below, by taking cases depending on whether m is superpolynomial in n or not. A proof of Theorem 1.2 would then follow by Lemma 2.11.

Remark 3.1. *We acknowledge that a proof of the* NP*-hardness of Sparse Partial MCSP is implicit in the work by Ilango, Loff, and Oliveira [ILO20]. However, the reduction they provide is* randomized*, while ours is* deterministic*.*

Continuing, we first consider the case where m is superpolynomial in n.

Lemma 3.2. *For all superpolynomial sparsity functions $m : \mathbb{N} \to \mathbb{N}$, all $\ell > 0$, and all $n \in \mathbb{N}$, it is the case that m-Sparse Partial MCSP of dimension n is polynomial-time reducible to $(n')^\ell$-Sparse Partial MCSP of dimension n', where $n' := m(n)^{1/\ell}$.*

Proof. Fix an arbitrary superpolynomial sparsity function $m : \mathbb{N} \to \mathbb{N}$, a $n \in \mathbb{N}$, and let $m := m(n)$. The claimed reduction R works as follows. On input a list of pairs $P = \{(x_i, b_i)\}_{i=1}^m$, where $(x_i, b_i) \in \{0,1\}^n \times \{0,1\}$ for all $1 \leq i \leq m$, and a number $0 \leq k \leq m \cdot n^2 - 1$ in binary the function R outputs $P' := \{(x'_i, b_i)\}_{i=1}^m$, where $x'_i := x_i 0^{n'-n} \in \{0,1\}^{n'}$ for all $1 \leq i \leq m$, and k. Note that $n' \geq n$ for all sufficiently large $n \in \mathbb{N}$, as m is a superpolynomial function on n.

Let A denote m-Sparse Partial MCSP of dimension n and B denote $(n')^\ell$-Sparse Partial MCSP of dimension n'. We will now prove the correctness of R.

Consider a YES instance of A, namely (P, k); we will prove that $R(P, k) = (P', k)$ is a YES instance of B. By our assumption, there is a circuit C of size k that agrees with P. Let C' be a circuit on $n' \geq n$ inputs such that $C'(x') := C\left(x'|_{[n]}\right)$ for all $x' \in \{0,1\}^{n'}$. Then, for all $1 \leq i \leq m$,

$$C'(x'_i) = C'\left(x_i 0^{n'-n}\right) = C\left(x_i 0^{n'-n}\Big|_{[n]}\right) = C(x_i) = b_i;$$

that is, C' agrees with P'. As the size of C' is at most $|C| \leq k$, we get that $R(P,k)$ is a YES instance of B, as desired.

Consider now a NO instance of A, namely (P,k); we will prove that $R(P,k) = (P',k)$ is a NO instance of B. Towards a contradiction, assume that (P',k) is a YES instance of B. By our assumption, there is a circuit C' of size k that agrees with P'. Let C be a circuit on $n \leq n'$ inputs such that $C(x) := C'\left(x0^{n'-n}\right)$ for all $x \in \{0,1\}^n$. Then, for all $1 \leq i \leq m$,

$$C(x_i) = C'\left(x_i 0^{n'-n}\right) = C'(x_i') = b_i;$$

that is, C agrees with P. As the size of C is at most $|C'| \leq k$, we get a contradiction and therefore $R(P,k)$ is a NO instance of B, as desired. This concludes the proof of the correctness of R.

What is left is to upper bound the running time of R by some polynomial in its input size. By inspecting R, we see that an upper bound on its running time is

$$m \cdot O((n'-n)) + \log\left(m \cdot n^2\right) \leq O(m \cdot n') = O\left(m^{1+1/\ell}\right) \leq O\left(m^2\right) \leq O\left(\left(m \cdot (n+1) + \log\left(m \cdot n^2\right)\right)^2\right).$$

As the input size of R is $m \cdot (n+1) + \log\left(m \cdot n^2\right)$, we get that the running time of R is polynomial in its input size, as desired. □

We now turn to the case where m is polynomial in n.

Lemma 3.3. *For all sparsity functions $m : \mathbb{N} \to \mathbb{N}$ such that $m(n) < n^c$ for some $c > 0$ and all $n \in \mathbb{N}$, all $\ell > 0$, and all $n \in \mathbb{N}$, it is the case that m-Sparse Partial MCSP of dimension n is polynomial-time reducible to $(n')^{\ell}$-Sparse Partial MCSP of dimension n', where $n' := n^{c/\ell}$ if $\ell < c$ and $n' := n$ if $\ell \geq c$.*

Proof. Fix an arbitrary sparsity function $m : \mathbb{N} \to \mathbb{N}$ such that $m(n) < n^c$ for some $c > 0$ and all $n \in \mathbb{N}$, and a $n \in \mathbb{N}$. We first claim that m-Sparse Partial MCSP of dimension n is polynomial-time reducible to n^{ℓ}-Sparse Partial MCSP of dimension n for all $\ell \geq c$. The reduction R that establishes this fact maps $(P,k) = \left(\{(x_i,b_i)\}_{i=1}^{m(n)}, k\right)$, where $(x_i,b_i) \in \{0,1\}^n \times \{0,1\}$ for all $1 \leq i \leq m(n)$ and $0 \leq k \leq m(n) \cdot n^2 - 1$, to

$$R(P,k) := (P',k) \qquad \text{where} \qquad P' := \{(x_i,b_i)\}_{i=1}^{m(n)} \cup \{(x_1,y_1)\}_{i=m(n)+1}^{n^{\ell}}.$$

The correctness of R follows from the fact that P and P' have the *same* circuit complexity and the same dimension. The running time of R is $O\left(n^{\ell}(n+1)\right) + \log\left(m(n) \cdot n^2\right) \leq \left(m(n)(n+1) + \log\left(m(n) \cdot n^2\right)\right)^{50\ell}$, that is, polynomial in its input size.

What is left is to show a reduction from n^c-Sparse Partial MCSP of dimension n to $(n')^{\ell}$-Sparse Partial MCSP of dimension $n' := (n^c)^{1/\ell}$ for all $\ell < c$. By the discussion above, this is sufficient in order to reduce m-Sparse Partial MCSP of dimension n to $(n')^{\ell}$-Sparse Partial MCSP of dimension n' for all $\ell < c$. However, a proof of this fact follows from the proof of Lemma 3.2, as in this case $n' \geq n$ for all sufficiently large $n \in \mathbb{N}$. □

Lemma 3.2 and Lemma 3.3 yield the following theorem.

Theorem 3.4. *For all sparsity functions $m : \mathbb{N} \to \mathbb{N}$, all $\ell > 0$, and all $n \in \mathbb{N}$, there exists a $n' \in \mathbb{N}$ such that m-Sparse Partial MCSP of dimension n is polynomial-time reducible to $(n')^{\ell}$-Sparse Partial MCSP of dimension n'.*

Theorem 2.12 and Theorem 3.4 yield the following corollary, because the function m from Theorem 2.12 is either superpolynomial or polynomial in n.

Corollary 3.5. *For all $\ell > 0$, n^{ℓ}-Sparse Partial MCSP is* NP*-hard under deterministic polynomial-time reductions.*

Finally, by combining Lemma 2.11 and Corollary 3.5 we get a proof of Theorem 1.2.

Corollary 3.6 (Theorem 1.2, restated)**.** *For all $\ell > 0$, n^{ℓ}-Sparse Partial MCSP is* NP*-complete under deterministic polynomial-time reductions.*

4 OWFs from zero-error average-case hardness of Sparse Partial MCSP

In this section, we prove the following result.

Theorem 4.1. *Assume that, for some $\ell > 0$, n^{ℓ}-Sparse Partial MCSP is zero-error $(1/p)$-HoA for some polynomial p. Then, there exists some weak OWF.*

By Theorem 4.1 and Theorem 2.16, we get the following corollary.

Corollary 4.2 (Theorem 1.1, restated)**.** *Assume that, for some $\ell > 0$, n^{ℓ}-Sparse Partial MCSP is zero-error $(1/p)$-HoA for some polynomial p. Then, there exists some OWF.*

4.1 Proof of Theorem 4.1

We will first require the following two lemmas.

Lemma 4.3. *For all sparsity functions $m : \mathbb{N} \to \mathbb{N}$, if m-Sparse Partial MCSP is zero-error $(1/p)$-HoA for some polynomial p, then Search m-Sparse Partial MCSP is $\big(1/p^2\big)$-HoA.*

Proof. We will prove the contrapositive. That is, we will prove that if Search m-Sparse Partial MCSP is $\big(1 - 1/p^2\big)$-EoA, then m-Sparse Partial MCSP is zero-error $(1 - 1/p)$-EoA.

Let $N' := |P| = m(n)\,(n+1)$ be the input size of Search m-Sparse Partial MCSP of dimension n. Assume that Search m-Sparse Partial MCSP is $\big(1 - 1/p^2\big)$-EoA. That is, assume that there exists some heuristic H' that on input a random partial truth table P outputs with probability $1 - 1/p(N')^2$ a description of a minimum-size circuit that agrees with P.

Given H', a zero-error heuristic H for m-Sparse Partial MCSP of dimension n and input size $N := |(P, k)| = m(n)\,(n+1) + \log\big(m(n) \cdot n^2\big)$ works as follows:

> On input a partial truth table P and a size parameter k, run H' on P to get a circuit C. If C agrees with P and the size of C is at most k, then return YES. Otherwise, return ?.

The success probability of H over a random instance (P, k) and random bits r is

$$\begin{aligned}
&\Pr_{P,k,r}[H(P,k;r) = m\text{-Sparse\_Partial\_MCSP}(P,k)] \\
&\geq \Pr_{P,k,r}[H(P,k;r) = m\text{-Sparse\_Partial MCSP}(P,k) \mid H'(P;r) = \text{Search\_}m\text{-Sparse\_Partial\_MCSP}(P)] \\
&\quad \cdot \Pr_{P,r}[H'(P;r) = \text{Search\_}m\text{-Sparse\_Partial\_MCSP}(P)] \\
&> 1 \cdot \left(1 - \frac{1}{p(N')^2}\right) \\
&= 1 - \frac{1}{p(N - \log(m(n) \cdot n^2))^2} \\
&\geq 1 - \frac{1}{p(N)},
\end{aligned}$$

since $p\big(N - \log\big(m(n) \cdot n^2\big)\big)^2 \geq p(N)$ for all sufficiently large $n \in \mathbb{N}$, as desired.

The running time of H is polynomial in N, for reasons outlined below. The running time of H' is polynomial in $N' \leq N$, any circuit C of size $|C| = |H'(P;r)| = \text{poly}(N') \leq \text{poly}(N)$ may be checked for agreement with P in time $O(m(n)\,|C|) \leq O(N \cdot \text{poly}(N)) \leq \text{poly}(N)$, computing the size of C may be done in time polynomial in $|H'(P;r)| \leq \text{poly}(N)$ and comparing $|C|$ and k can be done in time $O(\max\{\log|C|, \log k\}) \leq O(\max\{\log \text{poly}(N), n\}) \leq O(\max\{n, n\}) \leq O(n) \leq O(N)$.

Therefore, m-Sparse Partial MCSP is zero-error $(1 - 1/p)$-EoA as witnessed by H. □

The following is an elaboration on the seminal work by Liu and Pass [LP20].

Lemma 4.4 (Following Liu and Pass [LP20])**.** *Assume that, for some $\ell > 0$, n^{ℓ}-Search Sparse Partial MCSP is $(1/p)$-HoA for some polynomial p. Then, there exists some weak OWF.*

Proof. Consider the function $f : \{0,1\}^* \to \{0,1\}^*$ defined by the mapping rule

$$(s, x_1, \dots, x_m, C) \mapsto (s, (x_1, C(x_1)), \dots, (x_m, C(x_m))),$$

where C is a circuit with n inputs, s is the size of C, $m := n^\ell$, and $x_1, \dots, x_m \in \{0,1\}^n$. Note that C may *always* be replaced by a circuit of size at most n^c for some $c = c(\ell) > 0$, by Lemma 2.8, in the sense that for some $c > 0$ size n^c is enough for a circuit to agree with a partial truth table of sparsity m. For that matter, f is a function from $\{0,1\}^{O(\log n + mn + n^c \log n)}$ to $\{0,1\}^{O(\log n + mn + m)}$ and is computable in polynomial time, as one can evaluate a polynomial-size circuit on polynomially-many inputs in polynomial time.

Let N be the number of inputs of f; observe that $N \geq m \cdot n = n^{\ell+1} \geq n$. Observe also that the function f is only defined over infinitely many input lengths. However, by a padding trick, f can be transformed into another function f' that is defined over all input lengths, and such that f' is a weak one-way function, given that f is [LP20].

We now claim that if Search n^ℓ-Sparse Partial MCSP is $(1/p)$-HoA, then f is a $(1/q)$-weak OWF, where q is a polynomial such that $q(n) := 4n^c p(n)^3$ for all $n \in \mathbb{N}$. Towards a contradiction, assume that there exists a PPT algorithm A that inverts f with probability at least $1 - 1/q(N) \geq 1 - 1/q(n)$.

Except for a fraction $1/(2p(n))$ of random tapes r for A, the deterministic machine A_r, given by $A_r(z) := A(z; r)$ for all $z \in \{0,1\}^*$, fails to invert f with probability at most $2p(n)/q(n)$ over a uniformly random input z. This is so, as

$$\begin{aligned}\mathop{\mathbf{Pr}}_{r}\left[\mathop{\mathbf{Pr}}_{z}[A_r(z) \text{ fails}] > \frac{2p(n)}{q(n)}\right] &\leq \mathop{\mathbf{Pr}}_{r}\left[\mathop{\mathbf{Pr}}_{z}[A_r(z) \text{ fails}] \geq 2p(n) \cdot \mathop{\mathbf{Pr}}_{z,r}[A_r(z) \text{ fails}]\right] \\ &= \mathop{\mathbf{Pr}}_{r}\left[\mathop{\mathbf{Pr}}_{z}[A(z;r) \text{ fails}] \geq 2p(n) \cdot \mathop{\mathbf{E}}_{r}\left[\mathop{\mathbf{Pr}}_{z}[A(z;r) \text{ fails}]\right]\right] \\ &\leq \frac{1}{2p(n)},\end{aligned}$$

by Lemma 2.2. Henceforth, we will call such a random tape *good*; otherwise, we will call a random tape *bad*. Therefore, we have

$$\mathop{\mathbf{Pr}}_{i,P,r}[A(i,P;r) \text{ fails} \mid r \text{ is good}] = \mathop{\mathbf{Pr}}_{i,P,r}[A_r(i,P) \text{ fails} \mid r \text{ is good}] \leq \frac{2p(n)}{q(n)}.$$

We propose the following heuristic H for Search n^ℓ-Sparse Partial MCSP:

> On input P of dimension n and sparsity n^ℓ, and using random bits $r \in \{0,1\}^{O(\log n + mn + m)}$, the algorithm H runs $A(i,P;r)$ for all $i \in [n^c]$. For each $i \in [n^c]$, the algorithm $A(i,P;r)$ returns a tuple $(i, x_1, \dots, x_m, C_i)$. Then, $H(P;r)$ returns a minimum-size circuit from $\{C_i\}_{i \in [n^c]}$ that agrees with P.

We will now analyze the average-case performance of H. The probability that $A(i,P;r)$ fails, given that the random tape r is good, is

$$\begin{aligned}&\mathop{\mathbf{Pr}}_{i \sim [n^c],P,r}[A(i,P;r) \text{ fails} \mid r \text{ is good}] \\ &\quad \geq \mathop{\mathbf{Pr}}_{i,P,r}[A(i,P;r) \text{ fails} \mid H(P;r) \text{ fails and } r \text{ is good}] \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}] \\ &\quad = \mathop{\mathbf{Pr}}_{i,P,r}[A(i,P;r) \text{ fails} \mid A(\mathrm{CC}(P),P;r) \text{ fails and } r \text{ is good}] \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}],\end{aligned}$$

since $H(P;r)$ fails if and only if $A(\mathrm{CC}(P),P;r)$ fails, by the definition of H. Continuing, we have that

$$\begin{aligned}&\mathop{\mathbf{Pr}}_{i,P,r}[A(i,P;r) \text{ fails} \mid r \text{ is good}] \\ &\quad \geq \mathop{\mathbf{Pr}}_{i,P,r}[A(i,P;r) \text{ fails} \mid A(\mathrm{CC}(P),P;r) \text{ fails and } r \text{ is good}] \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}] \\ &\quad \geq \mathop{\mathbf{Pr}}_{i,P,r}[(i,P) = (\mathrm{CC}(P),P) \mid A(\mathrm{CC}(P),P;r) \text{ fails and } r \text{ is good}] \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}] \\ &\quad = \mathop{\mathbf{Pr}}_{i,P}[i = \mathrm{CC}(P)] \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}]\end{aligned}$$

$$
\begin{aligned}
&= \left(\sum_{j\in[n^c]} \mathop{\mathbf{Pr}}_{i,P}[i = j \text{ and } \mathrm{CC}(P) = j]\right) \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}]\\
&= \left(\sum_{j\in[n^c]} \mathop{\mathbf{Pr}}_{i}[i = j] \cdot \mathop{\mathbf{Pr}}_{P}[\mathrm{CC}(P) = j]\right) \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}]\\
&= \frac{1}{n^c} \cdot \left(\sum_{j\in[n^c]} \mathop{\mathbf{Pr}}_{P}[\mathrm{CC}(P) = j]\right) \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}]\\
&= \frac{1}{n^c} \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}]\,,
\end{aligned}
$$

by Lemma 2.8, the sparsity of P, and the choice of c, or

$$\frac{1}{n^c} \cdot \mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}] \leq \mathop{\mathbf{Pr}}_{i,P,r}[A(i,P;r) \text{ fails} \mid r \text{ is good}] \leq \frac{2p(n)}{q(n)},$$

by the discussion above, or

$$\mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}] \leq \frac{2n^c p(n)}{q(n)} = \frac{2n^c p(n)}{4n^c p(n)^3} = \frac{1}{2p(n)^2} < \frac{1}{2p(n)}.$$

Therefore, the heuristic H fails with probability at most

$$\mathop{\mathbf{Pr}}_{P,r}[H(P;r) \text{ fails} \mid r \text{ is good}] + \mathop{\mathbf{Pr}}_{r}[r \text{ is bad}] < \frac{1}{2p(n)} + \frac{1}{2p(n)} = \frac{1}{p(n)};$$

this yields a contradiction. □

We now turn to the proof of Theorem 4.1.

Proof of Theorem 4.1. Immediate; by Lemma 4.3 and Lemma 4.4, since if p is a polynomial, then p^2 is a polynomial too. □

5 Zero-error average-case hardness of Sparse Partial MCSP from OWFs

In this section, we prove the following result, which is a *partial* converse to Theorem 4.1.

Theorem 5.1 (Theorem 1.3, restated)**.** *Assume that there exists a OWF. Then, there exists some $\ell > 0$ such that n^ℓ-Sparse Partial MCSP is zero-error $(1/h)$-HoA, for a function $h : \mathbb{N} \to \mathbb{R}_{>0}$ such that $h(N) := 2^{N/100}$ for all $N \in \mathbb{N}$.*

Proof. We will prove the contrapositive. To this end, assume that for all $\ell > 0$, n^ℓ-Sparse Partial MCSP is $(1 - 1/h)$-EoA, for a function $h : \mathbb{N} \to \mathbb{R}_{>0}$ such that $h(N) := 2^{N/100}$ for all $N \in \mathbb{N}$. By Lemma 2.25, for all $k > 0$, we get a learning algorithm for $\mathsf{SIZE}\left[n^k\right]$ that works for infinitely many $n \in \mathbb{N}$. By Lemma 2.33, for every function family $\{f_y\}_{y\in\{0,1\}^*}$ such that $f_y : \{0,1\}^{|y|} \to \{0,1\}$ for all $y \in \{0,1\}^*$, and there is a polynomial-time algorithm that computes $f_y(x)$ given y and $x \in \{0,1\}^{|y|}$, there is a distinguisher for $\{f_y\}_{y\in\{0,1\}^*}$. By Corollary 2.32, there are no OWFs. □

Acknowledgements

We would like to thank Russell Impagliazzo for explaining his work [IL90] to us, and Ninad Rajgopal for illuminating discussions.

This work was partly carried out during a visit of Dimitrios Myrisiotis to DIMACS, with support from the Special Focus on Lower Bounds in Computational Complexity program.

Eric Allender and Harsha Tirumala were partially supported by NSF Grant CCF-1909216; Harsha Tirumala was also partially supported by the Simons Collaboration on Algorithms and Geometry. Mahdi Cheraghchi's research was partially supported by the National Science Foundation under Grant No. CCF-2006455.

References

[ABF$^+$08] Michael Alekhnovich, Mark Braverman, Vitaly Feldman, Adam R. Klivans, and Toniann Pitassi. The complexity of properly learning simple concept classes. *J. Comput. Syst. Sci.*, 74(1):16–34, 2008.

[AGGM06] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on NP-hardness. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 701–710. ACM, 2006. See also [AGGM10].

[AGGM10] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. Erratum for: On basing one-way functions on NP-hardness. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 795–796. ACM, 2010.

[BC20] Chris Brzuska and Geoffroy Couteau. Towards fine-grained one-way functions from strong average-case hardness. *IACR Cryptol. ePrint Arch.*, 2020:1326, 2020.

[BT06a] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Found. Trends Theor. Comput. Sci.*, 2(1), 2006.

[BT06b] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[HILL99] 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.

[Hir18] Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–258. IEEE Computer Society, 2018.

[HJLT96] Thomas R. Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bounds on learning decision lists and trees. *Inf. Comput.*, 126(2):114–122, 1996.

[HS17] Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 7:1–7:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[IL90] Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume II*, pages 812–821. IEEE Computer Society, 1990.

[ILO20] Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. NP-hardness of circuit minimization for multi-output functions. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 22:1–22:36. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147. IEEE Computer Society, 1995.

[KC00] Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79. ACM, 2000.

[LP20] Yanyi Liu and Rafael Pass. On one-way functions and Kolmogorov complexity. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1243–1254. IEEE, 2020.

[Nan21] Mikito Nanashima. On basing auxiliary-input cryptography on NP-hardness via nonadaptive black-box reductions. In *12th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 185 of *LIPIcs*, pages 29:1–29:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[OS17] Igor Carboni Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 18:1–18:49. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[PF79] Nicholas Pippenger and Michael J. Fischer. Relations among complexity measures. *J. ACM*, 26(2):361–381, 1979.

[San20] Rahul Santhanam. Pseudorandomness and the minimum circuit size problem. In *11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151 of *LIPIcs*, pages 68:1–68:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Val84] Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91. IEEE Computer Society, 1982.

A Hard-on-average problems in NP

We first introduce some useful notations. For a language $L \subseteq \{0,1\}^*$ we define its *characteristic function*, namely $f_L : \{0,1\}^* \to \{0,1\}$, to be a function given by

$$f_L(x) := \begin{cases} 1, & \text{if } x \in L, \\ 0, & \text{otherwise} \end{cases}$$

for all $x \in \{0,1\}^*$.

For sets $K, L \subseteq \{0,1\}^*$, the *disjoint union of K and L*, denoted $K \uplus L$, is the set $\{0x \mid x \in K\} \cup \{1x \mid x \in L\}$.

For a failure parameter function $\alpha : \mathbb{N} \to [0,1]$, we say that a language L is *α-hard-on-average (α-HoA)* if its characteristic function f_L is α-HoA. Similarly we define average-case easiness for languages.

We prove the following.

Proposition A.1. *Let L be a language in* NP *that is α-HoA for some failure parameter function $\alpha : \mathbb{N} \to [0,1]$. Then, the language $L^* := L \uplus \mathrm{SAT}$ is* NP*-complete and α^*-HoA, where $\alpha^* : \mathbb{N} \to [0,1]$ is a failure parameter function such that $\alpha^*(n) := \alpha(n-1) - 1/2$ for all naturals $n \geq 2$.*

Before we prove Proposition A.1, we recount the following basic observation.

Lemma A.2. NP *is closed under disjoint union.*

We now turn to the proof of Proposition A.1.

Proof of Proposition A.1. By Lemma A.2, the language L^* is in NP since L^* is the disjoint union of $L \in \mathsf{NP}$ and SAT $\in \mathsf{NP}$.

We will now show that L^* is NP-hard, by giving a polynomial-time reduction R from SAT to L^*. For all $x \in \{0,1\}^*$, let $R(x) := 1x \in \{0,1\}^*$. We see that R is polynomial-time computable. Moreover, if $x \in \text{SAT}$, then $R(x) = 1x \in L^*$, and if $R(x) \in L^*$, then $1x \in L^*$ and so $x \in \text{SAT}$.

What is left is to prove that L^* is α^*-HoA, where $\alpha^* : \mathbb{N} \to [0,1]$ is such that $\alpha^*(n) := \alpha(n-1) - 1/2$ for all naturals $n \geq 2$. Towards a contradiction, assume that L^* is $(1-\alpha^*)$-EoA and let H^* be a heuristic that witnesses this phenomenon. We will give a heuristic H that witnesses the fact that L is $(1-\alpha)$-EoA, whereby establishing the desired contradiction. To this end, let

$$H(x) := H^*(0x)$$

for all $x \in \{0,1\}^*$. We will show that H has the desired average-case performance. That is,

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{x\sim\{0,1\}^n}[H(x) = f_L(x)] &= \mathop{\mathbf{Pr}}_{x\sim\{0,1\}^n}[H^*(0x) = f_{L^*}(0x)] \\
&= \mathop{\mathbf{Pr}}_{y\sim\{0,1\}^{n+1}}[H^*(y) = f_{L^*}(y) \mid y_1 = 0] \\
&\geq \mathop{\mathbf{Pr}}_{y\sim\{0,1\}^{n+1}}[H^*(y) = f_{L^*}(y)] - \mathop{\mathbf{Pr}}_{y\sim\{0,1\}^{n+1}}[y_1 = 1] \\
&\geq 1 - \alpha^*(n+1) - \frac{1}{2} \\
&= 1 - \left(\alpha((n+1)-1) - \frac{1}{2}\right) - \frac{1}{2} \\
&= 1 - \alpha(n)\,.
\end{aligned}$$

□