Electronic Colloquium on Computational Complexity, Report No. 173 (2018)

The Non-Hardness of Approximating Circuit Size

Eric Allender[1]
Rutgers University, Piscataway, NJ, USA
allender@cs.rutgers.edu

Rahul Ilango[2]
Rutgers University, Piscataway, NJ, USA
rahul.ilango@rutgers.edu

Neekon Vafa[3]
Harvard University, Cambridge, MA, USA
nvafa@college.harvard.edu

Abstract

The Minimum Circuit Size Problem (MCSP) has been the focus of intense study recently; MCSP is hard for SZK under rather powerful reductions [4], and is provably not hard under "local" reductions computable in $\mathsf{TIME}(n^{0.49})$ [22]. The question of whether MCSP is NP-hard (or indeed, hard even for small subclasses of P) under some of the more familiar notions of reducibility (such as many-one or Turing reductions computable in polynomial time or in AC^0) is closely related to many of the longstanding open questions in complexity theory [7, 8, 17, 18, 19, 20, 22].

All known hardness results for MCSP hold also for computing somewhat weak approximations to the circuit complexity of a function [3, 4, 9, 17, 21, 25]. Some of these results were proved by exploiting a connection to a notion of time-bounded Kolmogorov complexity (KT) and the corresponding decision problem (MKTP). More recently, a new approach for proving improved hardness results for MKTP was developed [5, 7], but this approach establishes only hardness of extremely good approximations of the form $1 + o(1)$, and these improved hardness results are not yet known to hold for MCSP. In particular, it is known that MKTP is hard for the complexity class DET under nonuniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, implying MKTP is not in $\mathsf{AC}^0[p]$ for any prime p [7]. It is still open if similar circuit lower bounds hold for MCSP. One possible avenue for proving a similar hardness result for MCSP would be to improve the hardness of approximation for MKTP beyond $1 + o(1)$ to $\omega(1)$. In this paper, we show that this is impossible.

More specifically, we prove that PARITY does not reduce to the problem of computing superlinear approximations to KT-complexity or circuit size via AC^0-Turing reductions that make $O(1)$ queries. This is significant, since it is known that just *one* query to a much worse approximation of circuit size or KT-complexity suffices, for an AC^0 reduction to compute an approximation to any set in P/poly [23]. For weaker approximations, we also prove non-hardness under more powerful reductions. Our non-hardness results are unconditional, in contrast to conditional results presented in [7] (for more powerful reductions, but for much worse approximations). This highlights obstacles that would have to be overcome by any proof that MKTP or MCSP is hard for NP under AC^0 reductions. It may also be a step toward confirming a conjecture of Murray and Williams, that MCSP is not NP-complete under logtime-uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions [22].

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Circuit complexity

[1] Supported by NSF grant CCF-1514164. This work was done [in part] while the author was visiting the Simons Institute for the Theory of Computing.
[2] Supported by NSF grant CCF-1559855.
[3] Supported by NSF grant CCF-1559855.

ISSN 1433-8092

Keywords and phrases Minimum Circuit Size Problem, reductions, NP-completeness, time-bounded Kolmogorov complexity

Acknowledgements Much of this work was done during the 2018 DIMACS REU program, which was organized by Lazaros Gallos, Parker Hund, and many others. We would also like to thank Michael Saks, Shuichi Hirahara, Avishay Tal, and John Hitchcock for helpful discussions.

1 Introduction

The Minimum Circuit Size Problem (MCSP) is the problem of determining whether a (given) Boolean function f (represented as a bitstring of length 2^k for some k) has a circuit of size at most a (given) threshold θ. Although the complexity of MCSP has been studied for more than half a century (see [26, 21] for more on the history of the problem), recent interest in MCSP traces back to the work of Kabanets and Cai [21], who connected the problem to questions involving the natural proofs framework of Razborov and Rudich [24].

Since then, there has been a flurry of research on MCSP [3, 6, 4, 8, 19, 22, 18, 23, 17, 7, 20, 5, 16], but still the exact complexity of MCSP remains unknown. MCSP is in NP, but it remains an important open question whether MCSP is NP-complete.

MCSP is likely not in P. There is good evidence for believing MCSP $\notin$ P. If MCSP is in P, then there are no cryptographically-secure one-way functions [21]. Furthermore, [4] shows MCSP is hard for SZK under BPP-Turing reductions, so if MCSP $\in$ P then SZK $\subseteq$ BPP, which seems unlikely.

Showing MCSP is NP-hard would be difficult. Murray and Williams [22] have shown that if MCSP is NP-hard under polynomial-time many-one reductions, then EXP $\neq$ ZPP, which is a likely separation but one that escapes current techniques. Results from [4, 19, 22] also give various likely (but difficult to show) consequences for MCSP being hard under more restrictive forms of reduction. We note that it has been suggested that MCSP might well be complete for NP [20]. In this regard, it may also be relevant to note that $\mathsf{MCSP}^{\mathsf{QBF}}$ is complete for PSPACE under ZPP-Turing reductions [3].

MCSP is not hard for NP in limited settings. Murray and Williams [22] show MCSP is not NP-hard under a certain type of "local" reductions computable in TIME($n^{0.49}$). This is significant, since many well-known NP-complete problems are complete under local reductions computable in even logarithmic time. (A list of such problems is given in [22].)

Many hardness results for MCSP also hold for approximate versions of MCSP. In various settings, the power of MCSP to distinguish between circuits of size θ and $\theta + 1$ is not fully used. Rather, in [3, 9, 4, 25, 23, 20], the reduction succeeds assuming only that reliable answers are given to queries on instances of the form (T, θ), where either the truth table T requires circuits of size $\geq \theta = |T|/2$ or T can be computed by circuits of size $\leq |T|^\delta$, for some $\delta > 0$.

This is an appropriate time to call attention to one such reduction to approximations to MCSP. Corollary 6 of [23] shows that, for every $\delta > 0$, for every solution S to MCSP[$n^\delta, n/2$], for every set $A \in$ P/poly, there is a $c > 1$ and a set A' that differs from A on at most

$(1/2 - 1/n^c)2^n$ of the strings of each length n, such that $A' \leq_{\mathrm{tt}}^{\mathsf{AC}^0} S$ via a reduction[4] that makes only *one query*. (That is, $A' \leq_{1-\mathrm{tt}}^{\mathsf{AC}^0} S$.) Stated another way, any set in P/poly can be "approximated" with just one query to a weak approximation of MCSP. (Changing the solution S will yield a different set A'.)

There is no known many-one hardness result for MCSP, but one is known for a related problem. MKTP, the minimum time-bounded Kolmogorov complexity problem, is loosely the "program version" of MCSP. It is known [7] that MKTP is hard for DET under (non-uniform) NC^0 many-one reductions; it is conjectured that the same is true for MCSP. Time-bounded Kolmogorov complexity is polynomially related to circuit complexity [3], so one natural way to extend the hardness result of [7] from MKTP to MCSP would be to stretch the very small gap given in the reduction of DET to MKTP.

1.1 Our Contributions, and Related Prior Work

We address the following questions raised by prior work:

- Can the non-hardness result of [22] be extended to more powerful reductions? Note that it has been conjectured that MCSP is not NP-complete under uniform AC^0 reductions [22, 8].
- Can the conditional theorem of [7], establishing non-NP-hardness-of-approximation for MCSP under cryptographic assumptions (for very weak approximations), be improved, to show non-NP-hardness of MCSP with a smaller gap?
- Finally, can the result of [7], showing that MKTP is hard for DET under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, be extended, to hold for MCSP as well, by increasing the gap?

We make progress on all of these questions by proving an impossibility result in the setting of $\epsilon(\theta)$-GapMCSP, which is the promise version of MCSP with a multiplicative $\epsilon(\theta)$ gap where θ is the threshold.

▶ **Theorem 1.** PARITY $\not\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ $\epsilon(\theta)$-GapMCSP *where* $\epsilon(\theta) = o(\theta)$.

This is not the first work to describe non-hardness of approximation under AC^0 reductions. Arora [11] is credited by [1], with showing that no AC^0 reduction f can have the property that $x \in$ PARITY implies $f(x)$ has a very large clique, and $x \notin$ PARITY implies $f(x)$ has only very small cliques. (In Section 3, we present a similar result for Max-3-SAT, so that the reader can compare the techniques.) Our work differs from that of [11] in several respects. Arora shows that AC^0 reductions cannot prove very *strong* hardness of approximations for a problem where strong inapproximability results are already known. We show that AC^0 reductions cannot establish even very *weak* inapproximability results for MCSP. Also, our techniques allow us to move beyond $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, to consider AC^0-Turing reducibility.

To our knowledge, this is the first known non-hardness result for any variant of MCSP under non-uniform AC^0 reductions. While AC^0 reductions are provably less powerful than polynomial time reductions, most natural examples of NP-complete problem are easily seen to be complete under AC^0 (and even NC^0!) reductions [10].

It is shown in [7] that, if cryptographically-secure one-way functions exist, then $\epsilon(n)$-GapMCSP is not hard for NP under P/poly-Turing reductions[5] for some $\epsilon(n) = n^{o(1)}$. Our result gives

---

[4] Although Corollary 6 of [23] does not mention the number of queries, inspection of the proof shows that only one query is performed.

[5] The problem ϵ-GapMCSP is defined somewhat differently in [7] than here. See Section 2. Thus the form of $\epsilon(n)$ looks different here than in [7].

a trade-off, where we reduce the gap dramatically but also weaken the type of reduction. In particular, our results imply that if one-way functions exist, then $\epsilon(n)$-GapMCSP is NP-intermediate under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ and $\leq_{k-\mathrm{tt}}^{\mathsf{AC}^0}$ reductions, where $\epsilon(n) = o(n)$.

Finally, our work rules out one natural way to extend the MKTP hardness results to MCSP. One might have hoped that the reduction given by [7] could be extended to a larger gap and hence apply to MCSP (since MKTP and MCSP are polynomially related [3]). However, we show that this is impossible.

All of the theorems that we state in terms of MCSP hold also for MKTP, with identical proofs. For the sake of readability, we present the theorems and proofs only in terms of MCSP.

2 Preliminaries

We use \ to denote set difference. For a natural number n, we let $[n]$ denote the set $\{1, \ldots, n\}$.

2.1 Defining MCSP

For any binary string T of length 2^k, we define $\mathrm{CC}(T)$ to be the size of the smallest circuit (using only NOT gates and AND and OR gates of fan-in 2) that computes the function given by truth table T written in lexicographic order, where, for concreteness, circuit size is defined to be the number of AND and OR gates, although our arguments work for other reasonable notions of circuit size.

Throughout the paper, we use various approximate notions of the minimum circuit size problem, given as follows:

▶ **Definition 2** (Gap MCSP). For any function $\epsilon : \mathbb{N} \to \mathbb{N}$, We define $\epsilon(n)$-GapMCSP to be the promise problem (Y, N) where

$$Y := \{(T, \theta) \mid \mathrm{CC}(T) < \epsilon(\theta)\}, \text{ and}$$
$$N := \{(T, \theta) \mid \mathrm{CC}(T) > \theta\},$$

where θ is written in binary.

Note that this definition differs in minor ways from the way that ϵ-GapMCSP was defined in [7]. The definition presented here allows for finer distinctions than the definition that was used in [7].

Our results for non-hardness under $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reductions are best stated in terms of a restricted version of ϵ-GapMCSP, where the thresholds are fixed, for inputs of a given size: This variant of MCSP has been studied previously in [22, 17]; the analogous problem defined in terms of KT-complexity is denoted R_{KT} in [3].

▶ **Definition 3** (Parameterized Gap MCSP). For any functions $\ell, g : \mathbb{N} \to \mathbb{N}$ such that $\ell(n) \leq g(n)$, We define the language $\mathsf{MCSP}[\ell, g]$ to be the promise problem (Y, N) where

$$Y := \{T \mid \mathrm{CC}(T) < \ell(|T|)\}, \text{ and}$$
$$N := \{T \mid \mathrm{CC}(T) > g(|T|)\}.$$

2.2 Complexity classes and Reductions

We assume the reader is familiar with basic complexity classes such as P and NP. As we work extensively with non-uniform NC^0 and AC^0, we refer to the text by Vollmer [27] for background on these circuit classes. Throughout this paper, unless otherwise explicitly mentioned, we refer to the non-uniform versions of these circuit classes.

Let $\mathcal{C}$ be a class of circuits. For any languages A and B, we write $A \leq_{\mathrm{m}}^{\mathcal{C}} B$ if there is a function f computed by a circuit family $\{C_n\} \in \mathcal{C}$ such that $f(x) \in B \iff x \in A$. We write $A \leq_{\mathrm{T}}^{\mathcal{C}} B$ if there is a circuit family in $\mathcal{C}$ computing A with B-oracle gates. In particular, since we are primarily concerned with $\mathcal{C} = \mathsf{AC}^0$, we denote this as $A \leq_{\mathrm{T}}^{\mathsf{AC}^0} B$. We write $A \leq_{\mathrm{tt}}^{\mathsf{AC}^0} B$ if there is an AC^0 circuit family computing A with B-oracle gates, where there is no directed path from any oracle gate to another, i.e. if the reduction is non-adaptive. If, furthermore, the non-adaptive reduction has the property that each of the oracle circuits contains at most k oracle gates, then we write $A \leq_{k-\mathrm{tt}}^{\mathsf{AC}^0} B$.

Let $Y \subseteq \{0,1\}^\star$ and $N \subseteq \{0,1\}^\star$ be disjoint. Then $\Pi = (Y, N)$ is a *promise problem*. A language L is a *solution* to a promise problem $\Pi = (Y, N)$ if $Y \subseteq L$ and $N \cap L = \emptyset$. For two promise problems Π_1 and Π_2, some type of reducibility r (many-one, truth table, or Turing), and a circuit class $\mathcal{C}$, we say $\Pi_1 \leq_r^{C} \Pi_2$ if there is a *single* family of oracle circuits $\{C_n\}$ in $\mathcal{C}$ such that for every solution S_2 of Π_2, there is a solution S_1 of Π_1 such that C_n computes an r-reduction from S_1 to S_2.

2.3 Boolean Strings and Functions

For an $x \in \{0,1\}^n$ and a set of indices $B \subseteq [n]$, we let x^B denote the Boolean string obtained by flipping the ith bit of x for each $i \in B$.

A *partial string* (or *restriction*) is an element of $\{0,1,?\}^\star$. Define the *size* of a partial string p to be the number of bits in which it is $\{0,1\}$-valued. We say a partial string $p \in \{0,1,?\}^n$ *agrees* with a binary string $x \in \{0,1\}^n$ if they agree on all $\{0,1\}$-valued bits. If $x \in \{0,1\}^n$ is a binary string and $B \subseteq [n]$, then $x|_B$ denotes the partial string given by replacing the jth bit of x with ? for each $j \in [n] \setminus B$. We say a partial string p_1 *extends* a partial string p_2 if p_1 is equal to p_2 on all bits where p_2 is $\{0,1\}$-valued.

A *partial Boolean function* on n variables is a function $f : I \to \{0,1\}$ where $I \subseteq \{0,1\}^n$. For a promise problem $\Pi = (Y, N)$ and $n \in \mathbb{N}$, we let $\Pi|_n$ be the partial Boolean function that decides membership in Y on instances of length n which satisfy the promise. (In particular, $\Pi|_n : I \coloneqq (Y \cup N) \cap \{0,1\}^n \to \{0,1\}$.)

We will make use of two well-studied complexity measures on Boolean functions: block sensitivity and certificate complexity. We refer the reader to a detailed survey by Hatami, Kulkarni, and Pankratov [15] for background on these notions. For completeness, we provide the definitions of the two measures that we need. In our context, we will use these measures on partial Boolean functions. Let $I \subseteq \{0,1\}^n$ and let $f : I \to \{0,1\}$ be a partial Boolean function. For an input $x \in I$, define the *block sensitivity of f at x*, denoted $bs(f, x)$, to be the maximum number of non-empty, disjoint sets $B_1, \ldots, B_k$ such that $x^{B_i} \in I$ and $f(x) \neq f(x^{B_i})$ for all i. (Here, by "$f(y) \neq f(z)$" we require that f is defined at both y and z.) Define the 0-*block sensitivity of f* be $bs_0(f) \coloneqq \max_{x: f(x)=0} bs(f, x)$. For an input $x \in I$, define the *certificate complexity of f at x*, denoted $c(f, x)$, to be the size of the smallest set $B \subseteq [n]$ such that $f(y) = f(x)$ for all $y \in I$ that agree with $x|_B$. Define the 0-*certificate complexity of f* to be $c_0(f) \coloneqq \max_{x: f(x)=0} c(f, x)$.

3 Prior Work

In this section, we present a result that is similar in spirit to a result reported by Arora in an unpublished manuscript [11]. There, it was shown that there is no AC^0-computable function f with the property that $x \in \mathsf{PARITY}$ implies $f(x)$ has a very large clique, and $x \notin \mathsf{PARITY}$

implies $f(x)$ has only very small cliques. Here, in order to illustrate the techniques that were employed in [11], we observe that no AC^0 reduction can establish the known inapproximability of Max-3-SAT [14].

▶ **Proposition 4.** *Let $0 < \epsilon < 1$. No AC^0 reduction f can have the property that $x \in$ PARITY implies $f(x) \in$ 3-SAT, and $x \notin$ PARITY implies $f(x)$ has at most an ϵ fraction of the clauses satisfied.*

Proof. By appealing to Lemma 9, we may assume that the function f is an NC^0 reduction, as in the proof of Theorem 10. Let d be the constant, such that each output bit of $f(x)$ depends on at most d bits of x, and let $x \in$ PARITY have length n. Let $f(x)$ consist of m clauses, each encoded using $c \log m$ bits for some constant c (which we can assume since the number of clauses is polynomially-related to the number of variables). Then since $|f(x)| = cm \log m$, and each output bit depends on at most d input bits, there is some $i \leq n$ such that the i-th bit of x affects at most $(dc \log m)/n$ output bits. Flipping the i-th bit of x, to obtain a new string $x' \notin$ PARITY can affect at most $(dcm \log m)/n$ clauses. Since $f(x) \in$ 3-SAT, there is an assignment that satisfies at least $m - (dcm \log m)/n$ clauses of $f(x')$. The theorem is proved, by observing that $m - (dcm \log m)/n > \epsilon m$ for all large m. ◀

4 Non-Hardness Under NC^0 Reductions

In this section, we prove our main lemmas, showing that problems that are NC^0-reducible to ϵ-GapMCSP have bounded 0-block sensitivity and also have sublinear 0-certificate complexity. Whenever we will have occasion to use these lemmas, it will be in situations when we are able to assume that the NC^0 reduction is computing a function f satisfying the condition that there is a bound $\gamma(n) > 0$ such that, for all n, there is a $\theta \geq \gamma(n)$ such that, for all x of length n, $f(x)$ is of the form $(T(x), \theta)$. (In particular, the threshold θ is the same for all inputs of length n.) We will call such an NC^0 reduction a *γ-honest reduction.*

▶ **Lemma 5.** *Let $\epsilon(\theta) = o(\theta)$, and let $\Pi = (Y, N)$ be a promise problem, where $\Pi \leq_{\mathrm{m}}^{\mathsf{NC}^0}$ ϵ-GapMCSP via a γ-honest reduction f computed by an NC^0 circuit family C_n of depth $\leq d$, where $\gamma(n) \geq \log \log n$. Then there is an n_0 (that depends only on ϵ and d) such that for all $n \geq n_0$, if $N|_n \neq \emptyset$, then $bs_0(\Pi|_n) < s$, where s is a constant that depends only on d.*

Proof. Let $s = 2^{d+1} + 1$. Since $\epsilon(n) = o(n)$, we can pick a constant $r_0 > 4s$ such that $\epsilon(r) < r/(2s)$ for all $r \geq r_0$.

Pick $n_0 \geq 2^{2^{r_0}}$, and let $n \geq n_0$.

For the sake of contradiction, suppose $bs_0(\Pi|_n) \geq s$, and let $x \in N \cap \{0,1\}^n$ be a 0-valued instance with $bs(\Pi|_n, x) \geq s$. Then we can find disjoint sets $B_1, \ldots B_s \subseteq [n]$ such that $\Pi|_n(x^{B_j}) = 1$ for all $j \in [s]$. (That is, each x^{B_j} is in Y.)

Let $f(x) = (T, \theta)$, and note that $\mathrm{CC}(T) > \theta \geq \gamma(n)$ (since f is γ-honest). Since $x \in N$ and C_n is a reduction to ϵ-GapMCSP, we know that any circuit that computes the function with truth table T has size at least θ. For each $j \in [s]$, let T_j be the truth table produced by C_n on input x^{B_j}. Since $x^{B_j} \in Y$, we know that each T_j has a circuit D_j computing T_j of size at most $\epsilon(\theta)$. (Here, it is important that the same threshold θ is used for all inputs of length n, by γ-honesty.)

We aim to build a "small" circuit computing T, which would contradict T having high complexity. Our circuit C for computing T works as follows: on input i, output the majority of $D_1(i), \ldots, D_s(i)$. The size of C is at most $s \cdot \epsilon(\theta) + 2s$ (each D_j has size at most $\epsilon(\theta)$, and computing the majority of s bits can be done with a circuit of size $2s$).

Now, we argue that this circuit correctly computes the ith bit of T for all i. Let i be arbitrary. Recall the ith bit of T is defined to be the ith output of $C_n(x)$. Since C_n is a depth d circuit of fan-in 2, the ith output of C_n depends on at most 2^d input wires $W \subseteq [m]$. Hence, on any input y such that $y|_W = x|_W$, we have that the ith output of $C_n(y)$ equals the ith output of $C_n(x)$. In particular, if B is disjoint from W, then the ith output of $C_n(x^B)$ equals the ith output of $C_n(x)$. Since $B_1, \ldots B_s$ are disjoint and $|W| \leq 2^d$, it follows that at most 2^d of the sets $B_1, \ldots, B_s$ have a non-empty intersection with W. Hence, since $s = 2^{d+1} + 1$, the majority of the sets $B_1, \ldots, B_s$ are disjoint with W, so the majority of the circuits $D_1, \ldots, D_s$ when run on input i output the ith output of $C_n(x)$.

We thus have that $\mathrm{CC}(T) \leq s \cdot \epsilon(\theta) + 2s$. But $\theta > \gamma(n) \geq \log\log n$ (since the reduction f is γ-honest). By the choice of n_0 we have $\epsilon(\theta) < \theta/2s$ (since $\theta > \log\log n \geq r_0$). Thus $\mathrm{CC}(T) \leq s \cdot \theta/2s + 2s = \theta/2 + 2s < \theta$ (since $\theta > \log\log n > 4s$). This contradicts $\mathrm{CC}(T) > \theta$. ◀

▶ **Lemma 6.** *Let $\epsilon(\theta) = o(\theta)$, and let $\Pi = (Y, N)$ be a promise problem, where $\Pi \leq_m^{\mathsf{NC}^0}$ ϵ-GapMCSP via a γ-honest reduction f computed by an NC^0 circuit family C_n of depth $\leq d$, where $\gamma(n) \geq \log\log n$. Let $k \geq 1$. Then there is an n_0 (that depends only on ϵ, k and d) such that for all $n \geq n_0$, if $N|_n \neq \emptyset$, then $c_0(\Pi|_n) \leq n/k$.*

Proof. Let $p = 2^d$, let $p' = \binom{2pk+1}{p}$, and let K be a constant that is specified later (and which depends only on k and d). Since $\epsilon(\theta) = o(\theta)$, we can pick a constant s_0 such that $\binom{p'}{2}\epsilon(s) + K < s$ for all $s \geq s_0$.

Pick $n_0 \geq 2^{2^{s_0}}$, and let $n \geq n_0$.

For contradiction, suppose $c_0(\Pi|_n) > n/k$. Let $x \in N \cap \{0,1\}^n$ be a 0-valued instance with $c_0(\Pi|_n, x) > n/k$. Then, for all $S \subseteq [n]$ with $|S| \leq n/k$, there is an x_S such that x_S agrees with $x|_S$ and such that $\Pi|_n(x_S) = 1$. (That is, $x_S \in Y$.)

Let (T, θ) be the truth table produced by C_n on input x. Since $x \in N$ and C_n is a reduction, we know that any circuit computing T has size at least θ.

For each $S \subseteq [n]$ with size at most n/k, let T_S be the truth table produced by C_n on input x_S. Since $x_S \in Y$, we know that T_S has a circuit D_S of size at most $\epsilon(\theta)$.

We aim to build a "small" circuit computing T, which would contradict that T has high complexity. Recall that $p = 2^d$, and that $p' = \binom{2pk+1}{p}$.

▶ **Claim 6.1.** *There exists sets $S_1, \ldots S_{p'} \subseteq [n]$ such that*
- *$|S_i| \leq \frac{n}{2k}$ for all i, and*
- *for any set $P \subseteq [n]$ with $|P| \leq p$, we have that $P \subseteq S_i$ for some i.*

Proof. (Proof of Claim) Pick sets $V_1, \ldots, V_{2pk+1} \subseteq [n]$ of size at most $\frac{n}{2pk}$ whose union is $[n]$. Let $\mathcal{V} = \{V_1, \ldots, V_{2pk+1}\}$. Now let each of $S_1, \ldots, S_{\binom{2pk+1}{p}}$ be the union of some p sets chosen from $\mathcal{V}$. Each S_i has size at most $p\frac{n}{2pk} = \frac{n}{2k}$. Let $P \subseteq [n]$ be an arbitrary set of size p. Since $\bigcup_{V \in \mathcal{V}} V = [n]$, every element e of P lies within some $V \in \mathcal{V}$. Then P is contained in the union of some p sets from $\mathcal{V}$, so $P \subseteq S_i$ for some i. ◀

For each $i \neq j \in [p']$, let $S_{i,j} = S_{j,i} = S_i \cup S_j$. Note that $|S_{i,j}| \leq n/k$.

Our circuit C for computing T works as follows. On input r, for each $i \in [p']$, see if $D_{S_{i,1}}(r) = \cdots = D_{S_{i,p'}}(r)$. If so, then output $D_{S_{i,1}}(r)$. The size of this circuit is at most $\binom{p'}{2}\epsilon(\theta) + K$ (for some fixed constant K) since each of the $\binom{p'}{2}$ $D_{S_{i,j}}$ circuits has size at most $\epsilon(\theta)$ and the other "unanimity" condition is a Boolean function on $\binom{p'}{2}$ variables (of in fact linear size) and so can be computed with circuit of some size $K = O(p')^2$ (that depends only on k and d).

Now, we argue that C on input r correctly computes the rth bit of T. Let $r \in [m]$ be arbitrary. For convenience, on an input $y \in \{0,1\}^n$ let $C_n^r(y)$ denote the rth output of $C_n(x)$. Recall the rth bit of T is defined to be $C_n^r(x)$. We must show two things. First, that there exists an i such that $D_{S_{i,1}}(r) = \cdots = D_{S_{i,p'}}(r)$ and second, that if for some i we have that $D_{S_{i,1}}(r) = \cdots = D_{S_{i,p'}}(r)$, then $D_{S_{i,1}}(r) = C_n^r(x)$.

Since C_n has depth d, the rth output of C_n can depend on at most 2^d input wires $W \subseteq [m]$. Hence, on any input y such that $y|_W = x|_W$, we have that $C_n^r(y) = C_n^r(x)$. Since $p = 2^d$, by the claim, there exists some $S_{i^\star}$ such that $W \subseteq S_{i^\star}$. Therefore, for all j we have that $x_{S_{i^\star,j}}|_W = x|_W$, so $D_{S_{i^\star,j}}(r) \overset{\text{def}}{=} C_n^r(x_{S_{i^\star,j}}) = C_n^r(x)$.

This implies both things we must show. First, we know that $D_{S_{i^\star,1}}(r) = \cdots = D_{S_{i^\star,p'}}(r)$ since they each equal $C_n^r(x)$. Second, if for some i, we have that $D_{S_{i,1}}(r) = \cdots = D_{S_{i,p'}}(r)$, then we also have that $D_{S_{i,1}}(r) = D_{S_{i,i^\star}}(r) = C_n^r(x)$.

Thus we have that T can be computed by a circuit of size at most $\binom{p'}{2}\epsilon(\theta) + K$, which is less than θ, since $\theta \geq \log\log n \geq s_0$. This contradicts that $\mathrm{CC}(T) > \theta$. ◀

Next, we present a variant of Lemma 6 stated in terms of a larger gap.

▶ **Lemma 7.** *Let $\epsilon(\theta) < \theta^\alpha$, and let $\Pi = (Y, N)$ be a promise problem, where $\Pi \leq_{\mathrm{m}}^{\mathsf{NC}^0}$ ϵ-*GapMCSP *via a γ-honest reduction f computed by an NC^0 circuit family C_n of depth $\leq d$, where $\gamma(n) \geq n^\beta$. Then for all δ such that $\delta_0 = \beta(1-\alpha)/2^{d+1} > \delta > 0$ there is an n_0 such that for all $n \geq n_0$, if $N|_n \neq \emptyset$, then $c_0(\Pi|_n) \leq n^{1-\delta}$.*

Proof. Let $p = 2^d$. Suppose for contradiction that for some $\delta > 0$ with $\delta < \delta_0 = \beta(1-\alpha)/2p$ we have $c_0(\Pi|_n) > n^{1-\delta}$ infinitely often. We can follow the same argument (and notation) as above, except we have to be more careful since $n/c_0(\Pi|_n)$ is no longer a constant, and hence $p' = \binom{2pn/c_0(\Pi|_n)+1}{p} \leq \binom{2pn^\delta+1}{p} = O(n^{p\delta})$ is no longer constant. Since the unanimity condition can be implemented by a circuit of size linear in $\binom{p'}{2}$, we can construct a circuit computing truth table T of size

$$\epsilon(\theta) \cdot c_1 p'^2 = \epsilon(\theta) \cdot c_1 \binom{2pn^\delta + 1}{p}^2 \leq c_2 \epsilon(\theta) n^{2p\delta}$$

infinitely often for some positive constants c_1, c_2. By γ-honesty, we have $\theta \geq \gamma(n) \geq n^\beta$. This implies that we can construct a circuit computing T of size

$$c_2 \epsilon(\theta) n^{2p\delta} \leq c_2 \epsilon(\theta) (\theta^{1/\beta})^{2p\delta} < c_2 \theta^\alpha \theta^{2p\delta/\beta} < \theta$$

infinitely often. This is a contradiction since T is a truth table with circuit complexity $\geq \theta$. ◀

Next, we present a variant of Lemma 7, but restricted to the parameterized version of MCSP. This variant is useful in extending our non-hardness results to $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reductions that make $n^{o(1)}$ queries.

▶ **Lemma 8.** *Let $\Pi = (Y, N)$ be a promise problem. If $\Pi \leq_{\mathrm{m}}^{\mathsf{NC}^0}$* MCSP*$[\ell, g]$ with $\ell(m) = o(g(m)/m^\delta)$ for some $\delta > 0$, then $c_0(\Pi|_n) \leq n^\epsilon$ for some $\epsilon < 1$ for all but finitely many n where $N|_n \neq \emptyset$, where ϵ depends only on the depth of the NC^0 circuit family and δ.*

Proof. Suppose for contradiction that for all $\epsilon < 1$ we have $c_0(\Pi|_n) > n^\epsilon$ infinitely often. Once again, we follow the same argument (and notation) as above. We can construct a circuit computing truth table T of size

$$\ell(m) \cdot c_1 p'^2 \leq \ell(m) \cdot c_1 \binom{2pn/c_0(\Pi|_n) + 1}{p}^2 \leq \ell(m) c_1 \binom{2pn^{1-\epsilon} + 1}{p}^2 \leq c_2 \ell(m) n^{2p(1-\epsilon)},$$

infinitely often for some positive constants c_1, c_2. (Here, m denotes the length of the truth table T.) Note that since $c_0(\Pi|_n) > n^\epsilon$, we know $\Pi|_n$ depends on $\geq n^\epsilon$ input bits. Since the circuit has depth at most d and gates of fan-in 2, we must have $m \geq n^\epsilon/2^d$. This implies that we can construct a circuit computing T of size

$$c_2\ell(m)(n^\epsilon)^{\frac{2p(1-\epsilon)}{\epsilon}} \leq c_3\ell(m)m^{\frac{2p(1-\epsilon)}{\epsilon}},$$

infinitely often for some positive constant c_3. Setting $\epsilon = \frac{2p}{2p+\delta}$, we have that T can be computed by a circuit of size $\leq c_3\ell(m) \cdot m^\delta$ infinitely often, which is a contradiction since T is a truth table with circuit complexity $\geq g(m) = \omega(\ell(m) \cdot m^\delta)$. ◀

5 Non-Hardness Under Many-One AC^0 Reductions

To extend our non-hardness results to AC^0 we make use of a version of a theorem given in [1] that was first proved by [2, 12] that says randomly restricting a family of AC^0 circuits yields a family of NC^0 circuits with high probability.

▶ **Lemma 9** (Lemma 7 in [1]). *Let C_n be a family of n-input (multi-output) AC^0 circuits. Then there exists an $a > 0$ such that for all $n \in \mathbb{N}$ there exists a restriction of C_n to $\Omega(n^{1/a})$ input variables that transforms C_n into a (multi-output) NC^0 circuit.*

▶ **Theorem 10.** $\mathsf{PARITY} \not\leq_m^{\mathsf{AC}^0}$ ϵ-$\mathsf{GapMCSP}$ *where $\epsilon(n) = o(n)$.*

Proof. Suppose not. Then there is a family of AC^0 circuits C_n that many-one reduces PARITY to ϵ-$\mathsf{GapMCSP}$. By Lemma 9, there is an a such that we can transform each C_n into an NC^0 circuit D_m on $m = \Omega(n^{1/a})$ variables, computing a reduction f from either PARITY or $\neg\mathsf{PARITY}$ (depending on the parity of the restriction) to ϵ-$\mathsf{GapMCSP}$. For each input x of length n, $f(x)$ is of the form $(T(x), \theta(x))$. Since there are only $O(\log n)$ output gates in the $\theta(x)$ field, and each output gate depends on only $O(1)$ input variables, all of the output gates for $\theta(x)$ can be fixed by setting only $O(\log n)$ input variables. Furthermore, we claim that there is some setting of these $O(\log n)$ input variables, such that the resulting value of θ is greater than $\log n$. If this were not the case, then the $\leq_m^{\mathsf{AC}^0}$ reduction of PARITY (or $\neg\mathsf{PARITY}$) on $m = \Omega(n^{1/a})$ variables to ϵ-$\mathsf{GapMCSP}$ has the property that $\theta(x)$ is always less than $\log n$. But, as in the proof of Theorem 1.3 of [22], instances of MCSP where θ is $O(\log n)$ can be solved with a CNF circuit of polynomial size. Thus this would give rise to AC^0 circuits for PARITY, contradicting the well-known circuit lower bounds of [2, 12].

Thus we can set $O(\log n)$ additional variables, and obtain circuits that reduce PARITY (or $\neg\mathsf{PARITY}$) on $m' = m - O(\log n) = \Omega(n^{1/(a+1)})$ variables to ϵ-$\mathsf{GapMCSP}$, where furthermore this reduction satisfies the hypotheses of Lemmas 5 and 6. But this contradicts the fact that both PARITY and $\neg\mathsf{PARITY}$ on m' variables have 0-certificate complexity and 0-block-sensitivity m'. ◀

6 Non-Hardness Under Limited Turing AC^0 Reductions

With some work, we can extend our non-hardness results beyond many-one reductions to some limited Turing reductions.

In our proofs that deal with AC^0-Turing reductions, we will need to replace some oracle gates with "equivalent" hardware – where this hardware will provide answers that are consistent with *some* solution to the promise problem ϵ-$\mathsf{GapMCSP}$, but might not be consistent with the particular solution that is provided as an oracle. In order to ensure that this doesn't cause any problems, we introduce the notion of a "sturdy" AC^0-Turing reduction:

▶ **Definition 11.** Let $\Pi_1 = (Y_1, N_1)$ and $\Pi_2 = (Y_2, N_2)$ be promise problems. A family $\{C_n\}$ of AC^0-oracle circuits is a *sturdy* $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reduction from Π_1 to Π_2 if, for every pair of solutions S, S' to Π_2, every oracle gate G in C_n, and every $x \in Y_1 \cup N_1$, there is a solution S'' such that $C_n^S(x) = C_n^{S''}(x) = C_n^S[G \to S'](x)$, where the notation $C_n^S[g \to S']$ refers to the circuit C_n with oracle S, but where the oracle gate G answers queries according to the solution S' instead of S.

▶ **Lemma 12.** *Let Π be any promise problem. If $\Pi \leq_{\mathrm{T}}^{\mathsf{AC}^0} \epsilon(n)$-$\mathsf{GapMCSP}$ via a reduction of depth d, then $\Pi \leq_{\mathrm{T}}^{\mathsf{AC}^0} \epsilon(n)$-$\mathsf{GapMCSP}$ via a sturdy reduction of depth $5d$.*

Proof. Briefly: We modify C_n, so that each oracle query is checked against queries that were asked "earlier" in the computation, and the computation uses only the oracle answer from the first time a query was asked. Since each query is given an answer that is consistent with *some* solution, the new circuit gives the same answers as a new solution (which we denote as S''). Since C_n is a reduction, we get the same answer when using S or S''.

Label the oracle gates $G_1, \ldots, G_k$ of C_n in topological order so that there is no directed path from G_i to G_j for all $i < j$. Let q_i denote the query asked by G_i. Let C_n' be the circuit where we replace any wire that leaves G_i by a wire connected to the following subfunction:

$$G_i(x) \wedge \forall j < i(q_i \neq q_j)$$
$$\text{or}$$
$$\exists j < i(q_i = q_j \wedge \forall k < j(q_k \neq q_j) \wedge G_j(q_j))$$

The reader can verify that this additional circuitry can be implemented in depth five, and thus C_n' has depth at most $5d$.

Now let S and S' be any two solutions to $\epsilon(n)$-$\mathsf{GapMCSP}$. Consider any input x of length n that satisfies the promise of $\Pi = (Y, N)$. (That is, $x \in Y \cup N$.) Thus $C_n^S(x) = C_n^{S'}(x)$. Now consider the the operation of $C_n'(x)$ where some oracle gate G_i answers queries according to S', rather than S. By construction, the behavior of this computation ${C'}_n^S[G_i \to S']$ is the same as that of $C_n^{S''}(x)$, where

$$S''(q(x)) \coloneqq \begin{cases} S(q(x)) & \text{if } q(x) \neq q_i(x), \text{ or if } q_i(x) = q_j(x) \text{ for some } j < i, \\ S'(q(x)) & \text{otherwise.} \end{cases}$$

S'' is also a solution to ϵ-$\mathsf{GapMCSP}$, since it agrees with either S or S' on each query, and both S and S' agree on all queries that satisfy the promise. Thus ${C'}_n^S[G_i \to S'](x) = C_n^{S''}(x) = C_n^{S'}(x) = C_n^S(x)$, since C_n is a reduction. Also, ${C'}_n^{S''}(x) = C_n^{S''}(x)$ and ${C'}_n^S(x) = C_n^S(x)$, since each oracle gate of C_n' answers each query the same way that C_n does, if the same oracle is provided to each gate. Thus ${C'}_n^S(x) = {C'}_n^{S''}(x) = {C'}_n^S[g \to S'](x)$. This establishes that ${C'}_n$ is computing a sturdy reduction.

◀

▶ **Theorem 13.** *Let $k \geq 1$, and let $\epsilon(n) = o(n)$. Then* $\mathsf{PARITY} \not\leq_{k-\mathrm{tt}}^{\mathsf{AC}^0} \epsilon$-$\mathsf{GapMCSP}$.

Proof. We show that, for all $k \geq 1$, if $\mathsf{PARITY} \leq_{k-\mathrm{tt}}^{\mathsf{AC}^0} \epsilon$-$\mathsf{GapMCSP}$, then $\mathsf{PARITY} \leq_{(k-1)-\mathrm{tt}}^{\mathsf{AC}^0}$ ϵ-$\mathsf{GapMCSP}$. This suffices, since a 0-truth-table reduction is simply an AC^0 circuit computing PARITY, which cannot exist.

Given the oracle circuit family C_n, (where by Lemma 12 we may assume that the $\leq_{k-\mathrm{tt}}^{\mathsf{AC}^0}$ reduction is sturdy), let D_n be the subcircuit consisting of those gates that are on a path from an input variable to any oracle gate. D_n is simply an AC^0 circuit on n variables, and thus by Lemma 9, there is an a such that we can transform each D_n into an NC^0 circuit

E_m on $m = \Omega(n^{1/a})$ variables. Replacing D_n by E_m in C_n yields a k-tt reduction F_m from PARITY or ¬PARITY on m variables to ϵ-GapMCSP. For any input length r, computing PARITY on r bits can be accomplished by computing either PARITY or ¬PARITY on m bits, where m is only polynomially-larger than r. Thus, without any loss of generality, we may assume that our circuit family C_n has the property that the subcircuit D_n consisting of the gates on a path from an input gate to an oracle gate consists of NC^0 circuitry.

For each n, select the first oracle gate G_1 (in some order). Consider the circuit family B_n consisting of all of the gates that are on a path from any input to G_1. Note that B_n is an NC^0 circuit family computing some function f, where $f(x)$ is of the form $(T(x), \theta(x))$. If it is possible to set some of the input variables of B_n so that the output gates for $\theta(x)$ take on a value $\theta \geq \log n$, do so. Note that this leaves $m = n - O(\log n)$ variables unset. (If it is not possible to do so, then (as in the proof of Theorem 10), G_1 can be replaced in C_n by a polynomial-sized CNF circuit, thereby yielding a (sturdy) $(k-1)$-tt reduction, as desired.) Call C'_m and B'_m the circuits that result by restricting the $O(\log n)$ input variables of C_n and B_n, respectively.

We now aim to find a restriction of the inputs and a solution to ϵ-GapMCSP such that the output of G_1 is constant. Define $\Pi = (Y, N)$ to be the promise problem where for all x we put $x \in Y$ if and only if $\mathrm{CC}(T(x)) \leq \epsilon(\theta)$ and $x \in N$ if and only if $\mathrm{CC}(T(x)) > \theta$ where $B'_m(x) = (T(x), \theta)$. Observe that B'_m is a $\log n$-honest NC^0 reduction of Π to ϵ-GapMCSP.

There are two cases, depending on whether $N = \emptyset$ or not. If $N = \emptyset$, then $S' = \{(T, \theta) : \mathrm{CC}(T) < \epsilon(\theta)\}$ is a solution to ϵ-GapMCSP such that every query to G_1 is answered affirmatively. By the sturdiness of the reduction, G_1 can be replaced by a constant 1, transforming C'_m into a $(k-1)$-tt reduction.

If $N \neq \emptyset$, then by Lemma 6, for all large m $c_0(\Pi|_m) \leq m/(k+1)$. That is, there is a way to set some $r \leq m/(k+1)$ input variables, obtaining restriction ρ, and thereby obtain a circuit $B''_{m-r} = B'_m|_\rho$ on $m - r$ variables, such that for any string z of length $m - r$, $\mathrm{CC}(T_{m-r}(z)) > \epsilon(\theta)$ where $B''_{m-r}(z) = (T_{m-r}(z), \theta)$. That is, every query to G_1 is answered negatively in $C'_m|_\rho$, and hence G_1 can be replaced by a constant 0, transforming $C'_m|_\rho$ into a $(k-1)$-tt reduction from PARITY to ϵ-GapMCSP on $m - r = \Omega(n)$ variables in this case.

In both cases, we obtain a $(k-1)$-tt reduction from PARITY to ϵ-GapMCSP, as desired. ◀

With a larger gap, we can rule out nonadaptive reductions that use $n^{o(1)}$ queries.

▶ **Theorem 14.** *Let $\epsilon(n) < n^\alpha$ for some $1 > \alpha > 0$. Then for any circuit family $\{C_n\}$ computing an $\leq_{\mathrm{tt}}^{\mathsf{AC}^0}$ reduction of* PARITY *to ϵ-*GapMCSP*, there is a $\delta > 0$ such that, for all large n, $\{C_n\}$ makes at least n^δ queries.*

Proof. Let $\{C_n\}$ be a circuit family computing an $\leq_{\mathrm{tt}}^{\mathsf{AC}^0}$ reduction of PARITY to ϵ-GapMCSP. By Lemma 12 we may assume that each C_n is sturdy. As in the proof of the preceding theorem, we assume without loss of generality that C_n has the property that the subcircuit D_n consisting of those gates that lie on paths from input gates to oracle gates consists of NC^0 circuitry of depth d. (We will assume without loss of generality that, if the gates in D_n are removed from C_n, the depth of the circuit that remains is also at most d. Otherwise, let d be the maximum of these two constants.)

We will show that, for all large n, C_n contains at least n^δ oracle gates $G_1, G_2, \ldots, G_t$, where δ is chosen to be less than $(1-\alpha)/9d2^{d+1}$. For the sake of a contradiction, assume that $t < n^\delta$.

As in the proof of the preceding theorem, we construct a sequence of restrictions (one for each oracle gate), so that when the input bits of C_n are set according to the restrictions, each oracle gate either has a very small threshold θ, or else it can be replaced by a constant.

In this way, we transform C_n into a circuit on $m \geq n/2$ input bits where each oracle gate G_i has a threshold $\theta_i < n^{1/3d}$. Replacing each such oracle gate by a CNF of size $2^{O(n^{1/3d})}$ (as in the proof of the preceding theorem) results in an AC^0 circuit of depth at most $d+1$ computing PARITY, in contradiction to the lower bound of [13]. Details follow.

Our argument proceeds in t stages, where oracle gate G_i is considered in stage i. At the start of stage i we have a partial restriction ρ_{i-1} that has at most $(i-1)n^{1-2\delta}$ bits set. Here is a detailed description of stage i:

Consider the circuit family B_n consisting of all of the gates that are on a path from any input to G_i. Note that B_n is an NC^0 circuit family computing some function f_i, where $f_i(x)$ is of the form $(T_i(x), \theta_i(x))$. If for all x that agree with ρ_{i-1}, $\theta_i(x) < n^{1/(3d)}$, then stage i is done; set $\rho_i = \rho_{i-1}$ and go on to the next stage. Otherwise, there is a way to set an additional $O(\log n)$ additional variables, thereby extending ρ_{i-1} to obtain a new restriction ρ'_i, so that for all x which agree with ρ'_i, $\theta_i(x)$ takes on a constant value $\theta_i \geq n^{1/(3d)}$.

We now aim to find a restriction of the inputs and a solution to ϵ-GapMCSP such that the output of G_i is constant. Define $\Pi_i = (Y_i, N_i)$ to be the promise problem where for all x that agree with ρ'_i we put $x \in Y_i$ if and only if $\mathrm{CC}(T_i(x)) \leq \epsilon(\theta_i)$ and $x \in N_i$ if and only if $\mathrm{CC}(T_i(x)) > \theta_i$ where $B_n(x) = (T_i(x), \theta_i)$. Observe that B_n is a $n^{1/(3d)}$-honest NC^0 reduction of Π_i to ϵ-GapMCSP.

There are two cases, depending on whether $N_i = \emptyset$ or not. If $N_i = \emptyset$, then $S = \{(T, \theta) : \mathrm{CC}(T) \leq \theta\}$ is a solution to ϵ-GapMCSP such that every query to G_i is answered affirmatively. By the sturdiness of the reduction, the output of G_i can be replaced by the constant 1, and let $\rho_i = \rho'_i$.

If $N_i \neq \emptyset$, then by Lemma 7, for all large n, $c_0(\Pi_i|_{\rho'_i}) \leq n^{1-3\delta}$. (The conditions of Lemma 7 are satisfied, since $(1/3d)(1-\alpha)/2^{d+1} > 3\delta$.) That is, there is a way to set at most $n^{1-3\delta}$ additional variables, thereby extending ρ'_i to obtain a new restriction ρ_i, such that for any string x of length n that agrees with ρ_i, $\mathrm{CC}(T_i(x)) > \epsilon(\theta_i)$. Therefore, $S = \{(T, \theta) : \mathrm{CC}(T) \leq \epsilon(\theta)\}$ is a solution to ϵ-GapMCSP such that every query to G_i is answered negative. Hence, by the sturdiness of the reduction, gate G_i can be replaced by a constant 0.

This completes stage i. Note that, in obtaining ρ_i from ρ_{i-1} we set an additional $O(\log n) + n^{1-3\delta} < n^{1-2\delta}$ variables.

Since $t < n^\delta$, we have that ρ_t has $m \geq n - tn^{1-2\delta} > n - n^\delta n^{1-2\delta} = n - n^{1-\delta} > n/2$ unset variables. Let C''_m be the circuit $C_n|_{\rho_t}$. Each oracle gate in C''_m has the property that the threshold that is computed is always no more than $n^{1/3d}$. Since the reduction is sturdy, the circuit still behaves correctly if each oracle gate is replaced by a circuit that computes MCSP exactly, and (as in the proof of Theorem 1.3 of [22]), instances of MCSP where θ is bounded by $n^{1/3d}$ can be computed by a CNF of size $2^{O(n^{1/3d})}$. Replacing each oracle gate by such a CNF yields a circuit of depth at most $d+1$, of size $2^{O(n^{1/3d})}$, computing PARITY, thereby violating the lower bound established in [13]. ◀

If we consider the parameterized version of MCSP, rather than ϵ-GapMCSP, we obtain non-hardness even under $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reductions.

▶ **Theorem 15.** *Let $\ell(m) = o(g(m)/m^\delta)$ for some $1 > \delta > 0$. Then for any circuit family $\{C_n\}$ computing an $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reduction of* PARITY *to* MCSP$[\ell, g]$*, there is an $\epsilon > 0$ such that, for all large n, $\{C_n\}$ makes at least n^ϵ queries.*

Proof. Define the *oracle depth* of a gate G to be the largest number of oracle gates on any directed path ending with G.

Let $\{C_n\}$ be a circuit family computing an $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reduction of PARITY to MCSP$[\ell, g]$. As above, we may assume that each C_n is sturdy, and that the subcircuit D_n consisting of those gates at oracle depth 1 consists of NC^0 circuitry of depth at most d. Let k be the maximum oracle depth of any gate in $\{C_n\}$.

Similar to the proof of the preceding theorem, we construct a sequence of t restrictions $\rho_1, \ldots, \rho_t$, so that in $C_n|_{\rho_i}$ the first i gates $G_1, \ldots, G_i$ can be replaced a constant. In this way, we transform C_n into a circuit on $n' \geq n/2$ input bits of oracle depth $k-1$.

We will first show that there is a value $\epsilon > 0$ (specified later) such that if C_n does not have at least n^ϵ gates at oracle depth 1, then C_n can be replaced by an $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reduction of oracle depth $k-1$, by eliminating all of the oracle gates $G_1, \ldots, G_t$ at oracle depth 1.

Our argument proceeds in t stages, where oracle gate G_i is considered in stage i. At the start of stage i we have a partial restriction ρ_{i-1} that has at most $(i-1)n^{1-2\epsilon}$ bits set. Here is a detailed description of stage i:

Consider the circuit family B_n consisting of all of the gates that are on a path from any input to G_i. Note that B_n is an NC^0 circuit family computing some function $f_i(x) = T_i(x)$. Let $m = |T_i(x)|$.

We now aim to find a restriction of the inputs and a solution to MCSP$[\ell, g]$ for which the output of G_i is constant. Define $\Pi_i = (Y_i, N_i)$ to be the promise problem where for all x that agree with ρ_{i-1} we put $x \in Y_i$ if and only if $\mathrm{CC}(T_i(x)) \leq \ell(m)$ and $x \in N_i$ if and only if $\mathrm{CC}(T_i(x)) > g(m)$. Observe that B_n is an NC^0 reduction of Π_i to ϵ-GapMCSP.

There are two cases, depending on whether $N = \emptyset$ or not. If $N = \emptyset$, then $S = \{T : CC(T) \leq g(|T|)\}$ is a solution to MCSP$[\ell, g]$ such that every query to G_i is answered affirmatively. By the sturdiness of the reduction, the output of G_i can be replaced by the constant 1, and we let $\rho_i = \rho_{i-1}$.

If $N \neq \emptyset$, then, by Lemma 8, for all large n, $c_0(\Pi_i|_{\rho_{i-1}}) \leq n^{\epsilon'}$ for some $\epsilon' < 1$ that depends only on d and δ. That is, there is a way to set at most $n^{\epsilon'}$ additional variables, thereby extending ρ_{i-1} to obtain a new restriction ρ_i, such that for any string x of length n that agrees with ρ_i, $\mathrm{CC}(T_i(x)) > \ell(m)$. Thus, $S = \{T : CC(T) \leq \ell(m)\}$ is a solution to MCSP$[\ell, g]$ such that every query to G_i is answered negatively. Therefore, by the sturdiness of the reduction, gate G_i can be replaced by a constant 1.

This completes stage i. Note that, in obtaining ρ_i from ρ_{i-1} we set an additional $n^{\epsilon'}$ variables.

It is now time to set the constant ϵ to be $1 - (\epsilon'/2)$.

Since $t < n^\epsilon$, we have that ρ_t has $r \geq n - tn^{\epsilon'} = n - n^{1-(\epsilon'/2)}n^{\epsilon'} = n - n^{1-(\epsilon'/2)} > n/2$ unset variables.

A minor complication arises, when we want to repeat this argument, to reduce the oracle depth to $k-2$, etc. Namely, the constant ϵ' depends on the depth d of the NC^0 circuitry that feeds into the oracle gates at the bottom level of C_n. $C_n|_{\rho_t}$ has oracle depth $k-1$, as desired, but it now has AC^0 circuitry feeding into the lowest level of oracle gates, and when we appeal to Lemma 9 to apply a random restriction to convert that AC^0 circuitry to NC^0 circuitry, the depth of the NC^0 circuitry increases to a depth that we can denote d_2. This problem is resolved by observing that the choice of ϵ' in Lemma 8 is monotone in the depth d. Thus, if we carry out the argument above, but pick ϵ' using the parameter d_2 instead of d when we appeal to Lemma 8, and then repeat the argument to reduce the oracle depth to $k-2$, the parameters still work out. If we let d_3 be the depth of the NC^0 circuitry that results by starting with C_n with depth-d NC^0 circuitry at the bottom, eliminating lowest level of oracle gates and applying a random restriction to obtain a circuit family of oracle depth $k-1$ with NC^0 circuitry of depth d_2 at the bottom, and then repeating the process to obtain a circuit family of oracle depth $k-2$ with NC^0 circuitry of depth d_3 at the bottom, then the argument above is sufficient to obtain a circuit family of depth $k-3$, etc. Thus, there is a choice of ϵ' that suffices to convert an arbitrary $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reduction of oracle depth k (with fewer than n^ϵ oracle gates) to an AC^0 circuit computing parity on $n^{\Omega(1)}$ input bits, thereby obtaining the desired contradiction. ◀

7 Open Questions

There remain several open questions. The true complexity of MCSP remains a mystery. We have made progress in understanding the hardness of an approximation to MCSP, but how far can Theorem 10 be extended? Can we prove the result for general truth-table and Turing reductions? Can we reduce the gap in the theorem to some constant factor approximations? Does the impossibility result hold when AC^0 is replaced with, say, $\mathsf{AC}^0[2]$ many-one reductions? Does the DET-hardness of MKTP [7] also hold for MCSP, given that we have ruled out any large gap reduction?

References

1 Manindra Agrawal, Eric Allender, and Steven Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *J. Comput. Syst. Sci.*, 57(2):127–143, October 1998. doi:10.1006/jcss.1998.1583.

2 M. Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

3 Eric Allender, Harry Buhrman, Michal Kouckỳ, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006.

4 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Information and Compution*, 256:2–8, 2017. doi:10.1016/j.ic.2017.04.004.

5 Eric Allender, Joshua A Grochow, Dieter van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum circuit size, graph isomorphism, and related problems. *SIAM Journal on Computing*, 47(4):1339–1372, 2018.

6 Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM Journal on Computing*, 38(1):63–84, 2008.

7 Eric Allender and Shuichi Hirahara. New insights on the (non)-hardness of circuit minimization and related problems. In *Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS '17)*, 2017. doi:10.4230/LIPIcs.MFCS.2017.54.

8 Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. *computational complexity*, 26(2):469–496, Jun 2017. doi:10.1007/s00037-016-0124-0.

9 Eric Allender, Michal Kouckỳ, Detlef Ronneburger, and Sambuddha Roy. The pervasive reach of resource-bounded Kolmogorov complexity in computational complexity theory. *Journal of Computer and System Sciences*, 77(1):14–40, 2011.

10 Eric Allender, Michael C Loui, and Kenneth W Regan. Reducibility and completeness. In *Algorithms and theory of computation handbook*, pages 23–23. Chapman & Hall/CRC, 2010.

11 Sanjeev Arora. AC^0-reductions cannot prove the PCP theorem. Unpublished Manuscript., 1995.

12 Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. doi:10.1007/BF01744431.

13 Johan Håstad. *Computational Limitations for Small Depth Circuits.* MIT Press, Cambridge, MA, 1987.

14 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

15 Pooya Hatami, Raghav Kulkarni, and Denis Pankratov. Variations on the sensitivity conjecture. *Theory of Computing, Graduate Surveys*, 4:1–27, 2011. doi:10.4086/toc.gs.2011.004.

16 Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. *Electronic Colloquium on Computational Complexity*, 138, 2018. To appear in FOCS 2018.

17 Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 79. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

18 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

19 John Hitchcock and Aduri Pavan. On the NP-completeness of the minimum circuit size problem. In *FSTTCS*, 2015.

20 Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich. The power of natural properties as oracles. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 102. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

21 Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 73–79, New York, NY, USA, 2000. ACM. doi:10.1145/335305.335314.

22 Cody Murray and Ryan Williams. On the (non) NP-hardness of computing circuit complexity. In *Proceedings of the 30th Conference on Computational Complexity*, CCC '15, pages 365–380, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CCC.2015.365.

23 Igor Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds and pseudorandomness. *arXiv preprint arXiv:1611.01190*, 2016.

24 Alexander Razborov and Steven Rudich. Natural proofs. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 204–213, New York, NY, USA, 1994. ACM. doi:10.1145/195058.195134.

25 Michael Rudow. Discrete logarithm and minimum circuit size. *Information Processing Letters*, 128:1–4, 2017.

26 Boris Trakhtenbrot. A survey of Russian approaches to perebor (brute-force searches) algorithms. *IEEE Ann. Hist. Comput.*, 6(4):384–400, October 1984. doi:10.1109/MAHC.1984.10036.

27 Heribert Vollmer. *Introduction to circuit complexity: a uniform approach.* Springer Science & Business Media, 2013.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il