Simple reductions to circuit minimization: DIMACS REU Report

Vishal Ramesh, Sasha Sami, Noah Singer

July 2021

Abstract

The complexity of the minimum circuit size problem (MCSP) — and its many variants — is linked intricately to countless other questions in theoretical computer science. For instance, NP-hardness of MCSP or the closely-related *minimum Kolgomorov time-complexity problem* (MKTP) is known to imply ZPP $\neq$ EXP, and even if such reductions exist, they cannot be nearly as simple as the standard NP-completeness reductions for other problems [MW17]. In this project, we study whether various variants of circuit minimization can be complete for NP, or smaller classes, under simple types of reductions.

We investigate these questions in three specific ways. Firstly, we study whether recent results [AGM20; AGHR21] showing MKTP is hard for DET or even coNISZK$_\mathsf{L}$ under non-uniform projections may be replicated for MCSP. We build on techniques of [GII+19] and construct a non-uniform projection from Majority to MCSP, modulo an unproven conjecture on the monotonicity of expected circuit complexity of p-biased functions. Secondly, we consider AC^0_d-MFSP, the problem of minimizing depth-d formula sizes. We suggest (but do not fully prove) that the reduction of [Ila20b] can be modified to yield the NP-hardness of AC^0_d-MFSP under quasipolynomial-size uniform AC^0 reductions which are *randomized* and *adaptive*. We adapt the techniques of [Fu20] to show that (non-depth-bounded-)MFSP cannot be complete for NP under quasipolynomial-size uniform AC^0, *deterministic* and *non-adaptive* reductions, unless EXP $\neq$ ZPP. These (partial) results shed light on the power of randomness and adaptivity in reductions, at least in the setting of quasipolynomial-size AC^0 computation. Finally, we investigate the robustness of the definition of the class NISZK$_\mathsf{L}$ (as defined in [AGHR21]) and observe that NISZK$_\mathsf{L}$ = NISZK$_{\mathsf{AC}^0[\oplus]}$.

1 Introduction

In this REU project, we study the computational complexity of various *circuit minimization* problems. The prototypical such problem, the *minimum circuit size problem* (MCSP), is defined as follows. The input is a truth table of a Boolean function $f : \{0,1\}^n \to \{0,1\}$, as well as a size parameter θ, and MCSP$(f, \theta) = 1$ iff size$(f) \leq \theta$, where size(f) is simply the size of the smallest Boolean circuit computing f (over some fixed basis of gates). MCSP has numerous variants, often resulting from changing the notion of size used in the definition. In this report, we will mostly focus on MCSP, as well as the analogous *formula* minimization problem MFSP, and its constant-depth variant AC^0_d-MFSP.

How complex is circuit minimization? For simplicity, let us begin by focusing on MCSP. Note that if f is an n-bit function, then MCSP's input has size $O(2^n)$. MCSP is thus in the class NP; indeed, an NP verifier need only compare the output of a witness circuit C to f on every input in $\{0,1\}^n$. But how does MCSP fit into the class NP?

Kabanets and Cai [KC00] showed that if MCSP $\in$ P, then one-way functions do not exist, and hence "cryptography breaks".[1] But conversely, a line of work beginning with the seminal paper of Murray and Williams [MW17] establishes significant barriers towards proving that MCSP is NP-complete. Firstly, [MW17] showed (unconditionally) that MCSP cannot be complete for NP under "super-simple" reductions, e.g., deterministic many-one reductions where each output bit is computable in $o(\sqrt{N})$ time (where N is the input length). This is in contrast to essentially every other natural NP-complete problem, such as SAT and 3Coloring. The intuition behind this result of [MW17] is that a hardness reduction to MCSP must output the truth table of a complex function, and so the hardness reduction itself cannot be too simple. Murray and Williams [MW17] furthermore showed that the NP-completeness of MCSP would imply new complexity

[1] One-way functions are widely believed to exist.

lower bounds, specifically, $\mathsf{EXP} \neq \mathsf{ZPP}$.[2] Fu [Fu20] recently improved this result by weakening the hypothesis: $\mathsf{EXP} \neq \mathsf{ZPP}$ is implied even by the hardness of MCSP for sparse tally languages within ZPP under non-adaptive reductions.

In general, one may ask questions of the form, "Is MCSP variant X complete for class Y under type Z of reductions", and hope to either resolve them or connect them to other questions in complexity theory. In terms of types of reductions, we recall the distinction between *many-one* (a.k.a. *m-*) reductions from A to B, which simply convert instances of A to instances of B and return the decision on the B-instance, and *many-many* (a.k.a. *Turing*) reductions, which use an *oracle* of B — which may be invoked many times — to solve an instance of A. One important special type of many-many reduction is the *non-adaptive* (a.k.a. *truth-table*) reduction, where each B-oracle query does not depend on the output of previous oracle queries. See [All20] for more general background on circuit minimization and reductions.

(Almost-)NP-hardness for MCSP's "distant cousins". Although resolving the NP-completeness of MCSP itself is certainly out of reach of current methods, recent works have given strong evidence for the hardness of many variants of circuit minimization, in the form of NP-hardness or ETH-hardness (under reductions which are possibly quasipolynomial-time, many-many, and randomized[3]) [Ila20a; ILO20; Ila20b; ACM+21; LP21]. Of particular interest to us is the recent result of Ilango [Ila20b], which gives an *adaptive*, *quasipolynomial-time*, *randomized* NP-hardness reduction to AC^0_d-MFSP (i.e., constant-depth formula minimization).

Hardness under weak reductions for weak classes. Other works have analyzed the hardness of MCSP for classes that are much weaker than NP. Many of the same works have studied the closely-related *minimum Kolgomorov time-complexity problem* (MKTP), which is roughly a variant of MCSP using Turing machines instead of circuits. (See [All20] for more details on MKTP.) All currently known results about MCSP, such as [MW17; Fu20], also hold for MKTP. But interestingly, the past several years have seen new results which hold for MKTP but have not been extended to MCSP.

Allender and Hirahara [AH19] showed that MKTP is hard for DET under non-uniform NC^0 (many-one) reductions. Subsequently, Allender, Garvia Bosshard, and Musipatla [AGM20] strengthened this result to give non-uniform *projections* (i.e., reductions computable by circuits with only Not and constant gates). Most recently, Allender, Gouwar, Hirahara, and Robelle [AGHR21] showed that $\overline{\mathsf{MKTP}}$ is hard for $\mathsf{NISZK}_\mathsf{L}$ — the class of problems with noninteractive statistical zero-knowledge protocols with logspace-computable verifiers and simulators (see [AGHR21, Definition 2] for details), which contains DET — under non-uniform projections.[4]

Hardness results when the type of reduction is weak, such as the above results of [AH19; AGM20; AGHR21], are interesting for a number of reasons. Firstly, they "brush up" against the limits on hardness of $\mathsf{MCSP}/\mathsf{MKTP}$ under weak reductions established by [MW17; Fu20]. Secondly, they may actually imply lower bounds for $\mathsf{MCSP}/\mathsf{MKTP}$ in weak computational models. Indeed, if variant X is hard for class Y under reductions of type Z, and Y is closed under Z, then one may conclude X is in fact hard for Y. [AH19] used these insights to show that MKTP does not have small $\mathsf{AC}^0[p]$ circuits, and [AGHR21] used their improved reduction to show that MKTP does not have small $\mathsf{Majority} \circ \mathsf{Tr}$ (majority of threshold) circuits.

2 Questions addressed in this project

In this project we studied a number of questions around simple reductions to variants of circuit minimization. Some of these questions led to "dead ends", while others still may be productive research directions. In this section, we summarize all the directions we studied, including those which proved unsuccessful. When possible, we cross-reference to later sections of the report.

[2] The consensus on this statement is that it is true, but impossible to prove with current techniques.

[3] Randomization is typically helpful because it allows the reduction to explicitly construct hard functions. Randomized reductions may typically be made non-uniform by hardwiring an appropriate choice of random bits.

[4] [AGHR21] also showed that $\overline{MKTP}$ is hard for NISZK under randomized many-one reductions. It was previously known that $\overline{\mathsf{MKTP}}$ — as well as MCSP — is hard for SZK under randomized many-many reductions [AD17].

Can MKTP hardness results be replicated for MCSP? As discussed above, [AH19] showed that MKTP is hard for DET under non-uniform NC^0 reductions and used this to prove $\mathsf{AC}^0[p]$ lower bounds against MKTP. But a similar DET-hardness result for MCSP is not known. (Similar $\mathsf{AC}^0[p]$ lower bounds against MCSP were later proven by Golovnev *et al.* [GII+19], using an unrelated set of techniques. More on this below.) In general, MKTP seems to be easier to analyze in hardness reductions than MCSP, because KT complexity is easier to handle than circuit size. In particular, we consider the $\mathsf{NISZK_L}$-hardness-of-MKTP results of [AGHR21]. The reduction of [AGHR21] is from a gap-promise problem about calculating the entropy of NC^0 circuits (which Allender *et al.* also show is complete for $\mathsf{NISZK_L}$). Its analysis uses some earlier tools from work of Allender, Grochow, van Melkebeek, Moore, and Morgan [AGvM+18] for calculating the KT complexity of polynomially-many independent samples from a distribution as a function of its entropy, and this argument uses in an essential way the fact that the KT complexity of the hardest n-bit function can be characterized very tightly (it is $\approx n$). Unfortunately, such tight bounds are lacking for the circuit size of the hardest function on n bits, and so we lack a corresponding reduction to MCSP. (See [AGvM+18, §7.1] for more on this difficulty.)

Thus, in this project, we examine other ways of proving hardness results for MCSP. Specifically, we look to extend the approach of Golovnev *et al.* [GII+19] for proving $\mathsf{AC}^0[p]$ bounds against MCSP to explicitly construct a simple, many-one hardness reduction for MCSP from some interesting problem. It will be useful for us to first sketch [GII+19]'s approach. Firstly, consider the distributional distinguishing problem called the *coin problem*; on inputs of length N, the (p, q)-coin problem is to distinguish a p-biased N-bit string from a q-biased N-bit string with high probability. [GII+19] shows that there is a reduction from the $(1/2, 1/2 - \epsilon)$-coin problem to MCSP for any $\epsilon < 1/N^{0.49}$; indeed, the reduction is a trivial reduction, which follows (roughly) from showing that circuit complexities of $1/2$-biased strings are higher than $(1/2-\epsilon)$-biased strings.[5] Then, [GII+19] invokes a result of Shaltiel and Viola [SV10], which yields an AC^0, many-many reduction Majority to the $(1/2, 1/2 - \epsilon)$-coin problem. Finally, $\mathsf{AC}^0[p]$ lower bounds for MCSP follow from standard $\mathsf{AC}^0[p]$ lower bounds for Majority.

It will also be useful to briefly sketch [SV10]'s reduction. The crucial observation made by [SV10] is that given an N-bit string x, sampling an M-bit string in which each bit is chosen randomly from x is the same as sampling a $(\mathrm{wt}(x)/N)$-biased string, where $\mathrm{wt}(x)$ denotes the Hamming weight of x. Picking $M = N^2$, we hence have a randomized reduction from the promise problem of distinguishing N-bit strings of weight $N/2$ vs. N-bit strings of weight $N/2 - 1$ to the $(1/2, 1/2 - 1/\sqrt{M})$-coin problem (on M bits). This reduction can indeed be made deterministic using amplification and the fact that approximate majority is in AC^0 [Ajt83]. Finally, Majority reduces to the above promise problem in AC^0 (the reduction is many-many and uses an appropriate padding argument).

Our first hope was that we could somehow directly implement the [SV10] reduction (which, recall, is a many-many reduction) to give a deterministic, many-one reduction from Majority to MCSP. This would work if we had "AC^0 gadgets for MCSP"; e.g., a $\vee$ gadget would given two instances $(f, \theta), (f', \theta')$ of MCSP, would produce an instance $(f_\vee, \theta_\vee)$, such that $\mathsf{MCSP}(f_\vee, \theta_\vee) = \mathsf{MCSP}(f, \theta) \vee \mathsf{MCSP}(f', \theta')$. Unfortunately, such objects seem not to exist in AC^0 (although we did not prove that they are impossible). At this point, we also considered instead attempting a reduction from Majority to MFSP, the minimum *formula* size problem, since we do have direct-sum theorems for formula size. That is, given two functions $f, f' : \{0,1\}^n \to \{0,1\}$, the function $f \vee f' : \{0,1\}^{2n} \to \{0,1\}$ which maps (x, y) to $f(x) \vee f'(y)$ has formula size equal to the sum of the formula sizes of f and f', ± 1 (see e.g. [Weg87, §10.2]); this would at least give $\vee$ and $\wedge$ gadgets in the case $\theta \approx \theta'$. But unfortunately, the size of $(f \vee f')$'s truth table is N^2 (for $N = 2^n$), so this operation can only be used a constant number of times in a polynomial-time reduction, ruling out its applicability in our setting.

Finally, we recognized that since we were only seeking randomized/non-uniform reductions, we could sidestep the issue of using approximate majority, and hope to use the [SV10] sampling trick directly, along with bounds on complexity of biased functions à la [GII+19], to reduce Majority to MCSP. Indeed, we manage to do this in §3, modulo an unproven conjecture on the circuit complexity of biased functions (see Theorem 3.7).

[5] Actually, [GII+19] cannot quite conclude something quite this strong from existing bounds on the circuit complexity of $(1/2 - o(1/\operatorname{polylog}(N)))$-biased functions; instead, [GII+19] uses bounds on the circuit complexity of $O(1)$-biased functions and a hybrid argument. See [GII+19, p. 3] for a discussion. We use the same technique in §3.

How important is adaptivity in circuit minimization reductions? Ilango [Ila20b] shows that AC^0_d-MFSP (depth-d formula minimization) is NP-hard under quasipolynomial-time randomized reductions. [Ila20b]'s reduction is adaptive. Fu [Fu20] shows, on the other hand, that MCSP cannot be NP-hard (or even ZPP-hard) under polynomial-time non-adaptive reductions, unless $\mathsf{ZPP} = \mathsf{EXP}$. We hoped to (1) show that the reduction of [Ila20b] could be implemented as an (*adaptive*, quasipolynomial-sized, randomized) AC^0 reduction, and (2) extend [Fu20] to show that unless $\mathsf{EXP} \neq \mathsf{ZPP}$, AC^0_d-MFSP cannot be NP-hard under *non-adaptive*, quasipolynomial-sized, uniform AC^0 reductions. If (1) and (2) were both true, we would have the following disjunction of interesting statements: Either $\mathsf{EXP} \neq \mathsf{ZPP}$, or there is a problem (i.e., AC^0_d-MFSP) which is NP-complete under adaptive, randomized quasipolynomial-sized AC^0 reductions but not under non-adaptive, uniform quasipolynomial-sized AC^0 reductions.

We have good evidence that (1) may be true, but a full proof is out of scope for this REU report. We briefly sketch our reasoning here. The reduction of [Ila20b] occurs in three stages, and it is ultimately a series of constant-approximation-preserving reductions.

Firstly, AC^0_d-MFSP is reduced to the problem of constant-approximating a minimization over depth-d formulae with a $\vee$ gate at the top; we will denote this latter problem by $\mathsf{AC}^0_{d,\vee}$-MFSP. This reduction roughly consists of $\vee$-ing the input function with a function whose $\vee$-top depth-d complexity is much smaller than its $\wedge$-top depth-d complexity. The particular function used is due to Håstad, Rossman, Servedio, and Tan [HRST17] and is AC^0-computable; so this step seems implementable in AC^0.

The second step of the reduction is depth-reducing: Constant-approximating $\mathsf{AC}^0_{d,\vee}$-MFSP is reduced to (larger-)constant-approximating $\mathsf{AC}^0_{d-1,\vee}$-MFSP. In this step, a collection of functions, parametrized by a complexity parameter t, are sampled. When a function g is sampled with parameter t, certain nondeterministic complexity measures of g satisfy certain inequalities in terms of t with high probability; if this high probability event does occur, then [Ila20b] deduces information about the complexity of the input function f. The functions g appear to be implementable in AC^0. [Ila20b] uses amplification (sampling many g's for the same complexity parameter t) and is phrased using a $\mathsf{Majority}$ operation, but we should be able to implement this (given sufficient amplification) in AC^0 using the fact that approximate majority is in AC^0. We remark that the sampling of the functions g with certain complexity parameters is the crucial use of randomization in the reduction of [Ila20b].

Finally, [Ila20b] observes that $\mathsf{AC}^0_{2,\vee}$-MFSP is equivalent to DNF minimization, and invokes the almost-optimal NP-hardness results for DNF minimization due to Khot and Saket [KS08], who showed that DNF minimization on n variables cannot even be $n^{1-\epsilon}$-approximated, for any $\epsilon > 0$.[6] The [KS08] result uses sophisticated hardness-of-approximation tools (i.e., probabilistically checkable proofs) and appears very unlikely to be implementable in AC^0. However, it is much stronger than what is required to make the reduction of [Ila20b]; indeed, [Ila20b] only needs a (large) constant hardness-of-approximation factor. Hence here we consider invoking earlier, less-optimal hardness results, in particular the one due to Allender, Hellerstein, McCabe, Pitassi, and Saks [AHM+08], which proved only n^ϵ hardness for some $\epsilon > 0$, and is technically simpler. However, it is not completely clear whether the reduction of [AHM+08] is implementable in AC^0. In particular, it relies on some tricks, attributed to Gimpel by Czort [Czo99], for reducing partial DNF minimization to total DNF minimization, which employ parity functions as gadgets to force particular terms to be in minimal DNFs for some functions. We believe these gadgets may be replaceable by something computable in AC^0, but we have not sufficiently explored this direction. [AHM+08] ultimately invokes further hardness-of-approximation results for set-covering problems, the AC^0-implementability of which must also be investigated further.

On the side of (2), i.e., [Fu20]-style results demonstrating the difficulty of proving hardness of AC^0_d-MFSP under non-adaptive, uniform, quasipolynomial-sized AC^0 reductions, we make partial progress in this report. Specifically, we are able to show the same conclusion for MFSP (i.e., without the depth-d restriction), and we give an exposition in §5 which may shed additional light on the proof of [Fu20]. However, the techniques appear to be incapable of replicating this result in the setting of depth-bounded formula complexity.

How robust is the definition of $\mathsf{NISZK_L}$? The class $\mathsf{NISZK_L}$, of problems with non-interactive, statistical zer oknowledge proofs with logspace-computable verifiers and simulators, was introduced by [AGHR21], who

[6] There is one additional step here, which is required to switch measures of DNF complexity, but it appears implementable in AC^0 as well.

showed that $\overline{\mathsf{MKTP}}$ is hard for it under non-uniform projections. We are interested in better understanding how robust the logspace-computability requirements, i.e., if we make stronger or weaker restrictions on the simulator and verifier, does it change the set of problems? In §4, we include a proof of the fact that $\mathsf{NISZK_L} = \mathsf{NISZK}_{\mathsf{AC}^0[\oplus]}$, i.e., it does not change the set of problems to restrict the simulator and verifier to be constant-depth circuits augmented with parity gates. We remark that this result follows immediately from the work of [AGHR21].

3 Projection from Majority to MCSP

Let μ_p^M denote the distribution over p-biased M-bit strings, i.e., each of the M bits is independently sampled from a Bernoulli(p) distribution. Let $\exp(\lambda) = e^{-\lambda}$.

Definition 3.1. *Given $N \in \mathbb{N}$ and $t \leq \frac{N}{2}$, define the* symmetric threshold *set* STr *as*

$$\mathsf{STr}_{N,t} = \{x \in \{0,1\}^N : t \leq \mathrm{wt}(x) \leq N - t\}.$$

Lemma 3.2. *For sufficiently large $n \in \mathbb{N}$, let $N = 2^n$. Assuming Conjecture 3.6 (to be stated below), there exists $t \in [0.02N, 0.48N]$, such that there exists randomized and nonuniform projections from $\mathsf{STr}_{N,t}$ to* MCSP.

To prove Lemma 3.2, we use tools and the overall structure from Golovnev et al. [GII+19]:

Theorem 3.3 (Lupanov)**.** *Let $f : \{0,1\}^m \to \{0,1\}$ be any Boolean function that is supported on $k \leq 2^{m-1}$ inputs, where $k \geq \Omega(2^m)$. Then, for all sufficiently large $m \in \mathbb{N}$,*

$$\mathrm{size}(f) \leq \frac{\log \binom{2^m}{k}}{\log \log \binom{2^m}{k}} + O\left(\frac{2^m \log m}{m^2}\right).$$

Moreover, all but $o(1)$ fraction of random functions f require

$$\mathrm{size}(f) \geq \frac{\log \binom{2^m}{k}}{\log \log \binom{2^m}{k}} + \Omega\left(\frac{2^m \log m}{m^2}\right).$$

Recall that for $k = pM$ for $p \in (0,1)$, as $M \to \infty$ we have the asymptotic approximation $\log \binom{M}{k} \approx MH(p)$ where $H(\cdot)$ is the binary entropy function.

We also require the following form of McDiarmid's inequality:

Theorem 3.4 (McDiarmid)**.** *Let $X_1, \ldots, X_M \in \{0,1\}$ be independent random variables for $M = 2^m$. Let $f : \{0,1\}^m \to \mathbb{R}$ be any function and $c \in \mathbb{R}$ such that for all $1 \leq i \leq N$ and $b_1, \ldots, b_M, \widetilde{b_i} \in \{0,1\}$,*

$$\left| f(b_1, \ldots, b_M) - f(b_1, \ldots, b_{i-1}, \widetilde{b_i}, b_{i+1}, \ldots, b_M) \right| \leq c.$$

Then, for any $\lambda > 0$,

$$\Pr\left[|f(X_1, \ldots, X_M) - \mathbb{E}[f(X_1, \ldots, X_M)]| \geq \lambda\right] \leq 2\exp\left(\frac{-2\lambda^2}{Mc^2}\right).$$

We use Theorem 3.4 to prove the following corollary. It is analogous to [GII+19, Theorem 3.1], but we use a slightly different setting of parameters; in particular, we get a tighter probability bound by relaxing the distance. For $p \in [0,1]$, let $s_p^M = \mathbb{E}_{f \sim \mu_p^M}[\mathrm{size}(f)]$.

Corollary 3.5. *Let $M = 2^m$ and let $p \in [0,1]$. Then*

$$\Pr_{f \sim \mu_p^M}[|\mathrm{size}(f) - s_p^M| \geq M^{2/3} m] \leq \exp(-O(M^{-1/3})).$$

Proof. Follows by setting $\lambda = M^{2/3}m$, since changing one bit of an M-bit truth table changes the circuit complexity by at most $O(m)$. Indeed, the resultant probability bound is

$$2\exp\left(-\frac{2M^{4/3}m^2}{Mm^2}\right) = \exp(-O(M^{1/3})).$$

□

We make the following conjecture about s_p^M:

Conjecture 3.6. *Let M be sufficiently large. Then for any p, q such that $0.02 \leq p < q \leq \frac{1}{2}$, $s_p^M \leq s_q^M$.*

We feel that Conjecture 3.6 is very natural and very likely to be true. We communicated with Rahul Ilango about the conjecture, and he shares our view as to its likelihood, but believes it may be difficult to prove.

Now, we have:

Proof of Lemma 3.2. Let $m = 10n$ and $M = 2^M = N^{10}$. Define $\delta = 0.46$, $p_0 = 0.02$, and for $0 < i \leq \delta N$, define $p_i = p_0 + i \cdot \frac{1}{N}$. In particular, $p_{\delta N} = p_0 + \delta = 0.48$.

The Chernoff bound implies that a random sample from $\mu_{p_0}^M$ contains has Hamming weight at most $0.04M$ with probability $1 - o(1)$, and hence

$$s_{p_0}^M \leq \frac{MH(0.04)}{m + \log(H(0.04))} + o\left(\frac{M}{m}\right) \leq (0.25 + o(1))\frac{M}{m}$$

since $H(0.04) \approx 0.242$. Similarly,

$$s_{p_{\delta N}}^M \geq (0.99 + o(1))\frac{M}{m}$$

since $H(0.46) \approx 0.995$. Hence there exists some $i \in \{0, \ldots, \delta N - 1\}$ such that

$$s_{p_{i+1}}^M - s_{p_i}^M \geq \Omega\left(\frac{M}{mN}\right) = \Omega\left(\frac{N^9}{n}\right). \tag{$*$}$$

Let $t = p_{i+1}N$, and consider the following randomized projection $\mathcal{P}$ from $\mathsf{STr}_{N,t}$ to MCSP: On input $x \in \{0,1\}^N$, sample a string $y \in \{0,1\}^M$ by independently sampling each of the M bits as a random bit from x. Then output (y, θ) where $\theta = (s_{p_{i+1}}^M + s_{p_i}^M)/2$.

It remains to analyze the randomized correctness of this projection, and to show that it may be made nonuniform by hard-wiring randomness. The crucial observation, as in [SV10], is that y is sampled from $\mu_{p_x}^M$ where $p_x = \mathrm{wt}(x)/N$. Note that if x is a **YES**-instance of $\mathsf{STr}_{N,t}$, then $p_x \in [p_{i+1}, 1 - p_{i+1}]$. WLOG $p_x \leq \frac{1}{2}$,[7] and by Conjecture 3.6, we have $s_{p_x}^M \geq s_{p_{i+1}}^M$, and Corollary 3.5 implies that with probability $\exp(-O(N^{10/3}))$, if $y \sim \mu_{p_x}^M$, $\mathrm{size}(y) \geq s_{p_{i+1}}^M - M^{2/3}m = s_{p_{i+1}}^M - O(N^{20/3}n) \geq \theta$ (the last inequality follows from $(*)$). Similarly, if x is a **NO**-instance of $\mathsf{STr}_{N,t}$, with probability $\exp(-O(N^{10/3}))$, if $y \sim \mu_{p_x}^M$, $\mathrm{size}(y) \leq \theta$. This is sufficient to establish randomized correctness. Finally, we note that union-bounding over incorrectness for any input $x \in \{0,1\}^N$, the error probability is still $2^N \cdot \exp(-O(N^{10/3})) < 1$, and so we may hardwire an appropriate choice of randomness and make the projection non-uniform. □

Theorem 3.7. *For sufficiently large $N \in \mathbb{N}$, assuming Conjecture 3.6 (to be stated below), there exists randomized and nonuniform projections from $\mathsf{Majority}_n$ to MCSP.*

Proof. Follows from Lemma 3.2 and padding $\mathsf{STr}_{N,t}$. □

Interestingly, [MW17] seems to rule out randomized projections from $\mathsf{Majority}$ to MCSP using much less randomness (in particular, $o(N^{0.2})$).

[7] Note that $s_{1-p}^M \in [s_p^M - 1, s_p^M + 1]$, since we can biject every string with its complement and add a $\neg$ gate at the output of every circuit.

4 Robustness of definition of class $\mathsf{NISZK_L}$

[GSV99] introduced the class NISZK of promise problems Π, having a non-interactive statistical zero knowledge proof system. Recently, [AGHR21] defined $\mathsf{NISZK_L}$; the definition is identical, except that the verifier and simulator are restricted to log-space, instead of polynomial time (in the size of the input). See [AGHR21] for details on the definition.

[GSV99] showed that a promise version of *entropy approximation* for circuits is complete for NISZK, and [AGHR21] showed that the analogous problem restricted to a weak subset of $\mathsf{NC^0}$ circuits is complete for $\mathsf{NISZK_L}$. The problems are defined formally as follows:

Definition 4.1. *Let a circuit $C : \{0,1\}^m \to \{0,1\}^n$ represent a probability distribution X on $\{0,1\}^n$ induced by the uniform distribution on $\{0,1\}^m$. Then $\mathsf{Promise \cdot EA}$ is defined to be the promise problem*

$$\mathsf{EA}^{\mathbf{YES}} = \{(C,k) \mid H(X) > k+1\}$$
$$\mathsf{EA}^{\mathbf{NO}} = \{(C,k) \mid H(X) < k-1\}$$

where $H(X)$ denotes the entropy of X. $\mathsf{Promise \cdot EA_{NC^0}}$ is defined similarly, except that the circuits C considered in the input have gates of fan-in at most 2 and each output bit depends on at most 4 input bits.

Then we note the following theorem:

Theorem 4.2 ([AGHR21])**.** $\mathsf{Promise \cdot EA_{NC^0}}$ *is complete for* $\mathsf{NISZK_L}$*, under* $\leq_m^{proj}$*.*[8]

We prove the following observation, which follows quite readily from the techniques of [AGHR21]:

Observation 4.3. $\mathsf{NISZK_L} = \mathsf{NISZK_{AC^0[\oplus]}}$*, where* $\mathsf{NISZK_{AC^0[\oplus]}}$ *is defined similar to* $\mathsf{NISZK_L}$*, except that the verifier and simulator are implemented by log-space uniform* $\mathsf{AC^0[\oplus]}$ *circuits.*

Proof. As logspace-uniform $\mathsf{AC^0[\oplus]}$ computations may clearly be implemented in logarithmic space, $\mathsf{NISZK_{AC^0[\oplus]}} \subseteq \mathsf{NISZK_L}$. Hence, in light of Theorem 4.2, to prove Observation 4.3 it suffices to show that $\mathsf{Promise \cdot EA_{NC^0}} \in \mathsf{NISZK_{AC^0[\oplus]}}$. In order to do this, we follow [AGHR21] in re-using [GSV99]'s NISZK protocol for $\mathsf{Promise \cdot EA} \in \mathsf{NISZK}$. We show that for inputs as in $\mathsf{Promise \cdot EA_{NC^0}}$, the protocol may be implemented in $\mathsf{NISZK_{AC^0[\oplus]}}$ (whereas [AGHR21] proved it only for $\mathsf{NISZK_L}$).

The protocol proceeds by transforming the input $y = (C,k)$ to a distribution Z, encoded by a circuit D_y of size $\mathrm{poly}(s)$, where $s = |y|$. The protocol in essence involves the verifier and the simulator evaluating D_y over some input. Thus showing that there exists a uniform $\mathsf{AC^0[\oplus]}$ family of circuits $\{B_n\}$, such that $B_{\mathrm{poly}(s)}(y,\cdot) = D_y$ (for inputs satisfying the promise $\mathsf{Promise \cdot EA_{NC^0}}$), proves Observation 4.3. We verify this by going through the steps in transformation by [GSV99] (using the notation of [AGHR21]):

1. Let X' consist of $\mathrm{poly}(s)$ copies of X.
2. Let $Y = (h, h(X'))$, where h is chosen uniformly at random from a 2-universal hash family $\mathcal{H}$, such that the elements of $\mathcal{H}$ can be represented using $\mathrm{poly}(s)$ bits.
3. Let Y' consist of $\mathrm{poly}(s)$ copies of Y.
4. Let $Z = (Y'(r), h', h'(r))$, where r denotes the input to Y'. Again h' is chosen uniformly at random from a 2-universal hash family $\mathcal{H}$ whose elements can be represented using $\mathrm{poly}(s)$ bits.

Let $z = (C,k)$ denote an instance of $\mathsf{Promise \cdot EA_{NC^0}}$. Let $|z| = s$. As C is a $\mathsf{NC^0}$ circuit, it can be simulated by a $\mathsf{AC^0[\oplus]}$ circuit. So the $\mathrm{poly}(s)$ copies of X can be generated by taking $\mathrm{poly}(s)$ random bits as input. The hash families used in steps 2 and 4 can be computed in $\mathsf{AC^0[\oplus]}$ and require only $\mathrm{poly}(s)$ bits to represent. [9] Thus Y can be computed by an $\mathsf{AC^0[\oplus]}$ circuit. And so $\mathrm{poly}(s)$ copies of Y can be generated similarly. D_y is obtained by restricting the bits corresponding to z, to y, in the final $\mathsf{AC^0[\oplus]}$ circuit. □

We hypothesize that the requirements for the verifier and simulator in $\mathsf{NISZK_L}$ may also be *relaxed* without changing the class of problems. Specifically, we conjecture that $\mathsf{NISZK_L} = \mathsf{NISZK_{DET}}$.

[8] $\leq_m^{\mathrm{proj}}$ stands for many-one projection reduction.

[9] Indeed, the hash family given by multiplication of the input by polynomially-large Boolean matrices is 2-universal.

5 Attempt to extend result by [Fu20] to AC^0_d-MCSP

We note the following result by [Fu20].

Theorem 5.1 ([Fu20, Corollary 21])**.** *If* MCSP *is* ZPP*-hard under polynomial-time non-adaptive reductions (denoted by* $\leq^{\mathsf{P}}_{tt}$*)*[10]*, then* EXP $\neq$ ZPP*.*

We sketch a proof (slightly simplifying the exposition of [Fu20], who proved a stronger statement). Firstly, we use the following lemma:

Lemma 5.2 (Corollary of [Fu20, Lemma 17])**.** *There exists a unary language* $L \in \mathsf{ZTIME}(2^{O(n)})$*,*[11] *such that* $L \notin$ ZPP*.*

Proof sketch of Theorem 5.1. Let L be as in the statement of Lemma 5.2. Define $L' = \mathsf{Pad}(L)$, where $\mathsf{Pad}(L) = \{1^{2^n+n} \mid 1^n \in L\}$. Thus $L' \in$ ZPP.

By assumption, $L' \leq^{\mathsf{P}}_{tt}$ MCSP. Thus, there exists a machine $M(\cdot)$ that carries out the reduction in polynomial time $p(n)$.

Define the language $R = \{(1^n, i, j) \mid$ bit i of query j outputted by $M(1^{2^n+n})$ is 1, and $i, j \leq p(2^n+n)\}$. We have that $R \in$ EXP: Given $(1^n, i, j)$, compute $M(y)$ for $y = 1^{2^n+n}$, and check bit i of query j of the output.

Assume, towards contradiction, that EXP = ZPP. Thus $R \in$ ZPP $\subseteq$ P/poly. So R is computed by some family of circuits $\{C_n\}$ of polynomial size.

Finally, we claim that we can conclude that $L \in$ EXP, and thus $L \in$ ZPP, a contradiction. $L \in$ EXP as follows: Given $x = 1^n$ (an instance of L), generate $y = 1^{2^n+n}$ (the corresponding instance of L'), and compute $M(y)$. Get all the queries to MCSP from the output of $M(y)$. For a query q, if the threshold s is larger than the size of C_n, the query has answer **YES**, otherwise brute-force over all possible circuits of size $\leq s$ to get the answer to q. □

Consider the following uniformity condition for a quasi-polynomial sized circuits $\{D_n\}$: Given (n, h, g) output 1 iff there is an edge from gate g to gate h in D_n. In linear time; we can extend Theorem 5.1 to the following:

Theorem 5.3. *If* MCSP *is* ZPP*-hard under deterministic uniform quasi-polynomial time non-adaptive Turing reductions, it would imply* EXP $\neq$ ZPP*. The same holds for* MFSP*.*

Theorem 5.3 can be proved by the following minor changes to the proof of Theorem 5.1 (keeping rest of the steps unchanged):

1. Assume L' is reducible to MCSP (resp. MFSP) under uniform quasi-polynomial time non-adaptive reductions. Then $L' \leq^{\text{quasi}}_{tt}$ MCSP (resp. MFSP), and there exists a machine $M(\cdot)$ that carries out the reduction in quasi-polynomial time $u(n) = 2^{\log^c n}$ (for some constant c).

2. $R = \{(1^n, i, j) \mid$ bit i of query j outputted by $M(1^{2^n+n})$ is 1, and $i, j \leq u(2^n+n)\}$. $R \in$ EXP, as $M(y)$, for $y = 1^{2^n+n}$ takes time $u(2^n+n) = 2^{\log^c(2^n+n)} \in O(2^{n^{c+1}})$.

$M(y)$ for $y = 1^{2^n+n}$ takes $O(2^{n^{c+1}})$ time. And as the number of queries is $2^{O(n^{c+1})}$ (obtained by output of $M(y)$) and brute forcing over polynomial sized circuits takes exponential time, $L \in$ EXP and the original argument goes through.

5.3 cannot be extended to AC^0_d-MFSP, as it seems to break down when trying to show $L \in$ EXP. It is so because the brute force for MFSP queries with "large" threshold (larger than some polynomial) was avoided by the fact that $R \in$ P/poly. But for AC^0_d-MFSP, $R \in$ P/poly does not suffice, as the circuits C_n can potentially have non-constant depth (and are not guaranteed to be formulae). Indeed, for the argument to go through even in the AC^0_d-MCSP case, one would need to assume EXP $\in$ ZPP $\cap$ AC^0_d, which is false.

[10] $A \leq^T_{tt} B$ iff there exists some function f computable in time T, such that $f(x) = (q_1, q_2, \cdots, q_t, C')$ for $x \in A$, where C' is a circuit with t inputs and $x \in A \Leftrightarrow C'(B(q_1), B(q_2), \cdots, B(q_t)) = 1$. Here q_i are referred to as *queries* to B. (This is equivalent to the characterization of non-adaptive reductions as many-many reductions whose oracle queries do not depend on results of previous queries.)

[11] ZTIME$(\cdot)$ stands for zero error probabilistic time.

Acknowledgements

We would like to thank the DIMACS REU program for hosting us this summer, and especially our mentor Eric Allender for all of his support. N.S. was supported by NSF grant CCF-1852215. V.R. and S.S. were supported by CoSP, a project funded by European Union's Horizon 2020 research and innovation programme, grant agreement No. 823748. We would like to thank Rahul Ilango for helpful discussions over the course of the summer.

References

[ACM+21] Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. "One-Way Functions and a Conditional Variant of MKTP". Electronic Colloquium on Computational Complexity. Electronic Colloquium on Computational Complexity. Apr. 2021. URL: https://eccc.weizmann.ac.il/report/2021/009.

[AD17] Eric Allender and Bireswar Das. "Zero Knowledge and Circuit Minimization". In: *Information and Computation* 256 (2017). Conference version in MFCS 2017, pp. 2–8. DOI: 10.1016/j.ic.2017.04.004.

[AGHR21] Eric Allender, John Gouwar, Shuichi Hirahara, and Caleb Robelle. "Cryptographic Hardness under Projections for Time-Bounded Kolmogorov Complexity". Electronic Colloquium on Computational Complexity. Electronic Colloquium on Computational Complexity. Feb. 2021. URL: https://eccc.weizmann.ac.il/report/2021/010.

[AGM20] Eric Allender, Azucena Garvia Bosshard, and Amulya Musipatla. "A Note on Hardness under Projections for Graph Isomorphism and Time-Bounded Kolmogorov Complexity". Electronic Colloquium on Computational Complexity. Electronic Colloquium on Computational Complexity. Oct. 2020. URL: https://eccc.weizmann.ac.il/report/2020/158/.

[AGvM+18] Eric Allender, Joshua A. Grochow, Dieter van Melkebeek, Cristopher Moore, and Andrew Morgan. "Minimum Circuit Size, Graph Isomorphism, and Related Problems". In: *SIAM Journal on Computing* 47.4 (2018). Conference version in ITCS 2018, pp. 1339–1372. DOI: 10.1137/17M1157970.

[AH19] Eric Allender and Shuichi Hirahara. "New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems". In: *ACM Transactions on Computation Theory* 11.4 (Sept. 2019). Conference version in MFCS 2017. DOI: 10.1145/3349616.

[AHM+08] Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael Saks. "Minimizing Disjunctive Normal Form Formulas and AC^0 Circuits given a Truth Table". In: *SIAM Journal on Computing* 38.1 (Jan. 2008), pp. 63–84. DOI: 10.1137/060664537.

[Ajt83] Miklós Ajtai. "Σ^1_1-Formulae on Finite Structures". In: *Annals of Pure and Applied Logic* 24.1 (July 1983), pp. 1–48. DOI: 10.1016/0168-0072(83)90038-6.

[All20] Eric Allender. "The New Complexity Landscape Around Circuit Minimization". In: *Language and Automata Theory and Applications: 14th International Conference,* LATA 2020 (Milan, Italy, Mar. 4–6, 2020). Vol. 12038. Jan. 7, 2020, pp. 3–16. DOI: 10.1007/978-3-030-40608-0_1. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7206620/.

[Czo99] Sebastian Czort. "The Complexity of Minimizing Disjunctive Normal Form Formulas". Masters Thesis. University of Aarhus, 1999.

[Fu20] Bin Fu. "Hardness of Sparse Sets and Minimal Circuit Size Problem". In: *Computing and Combinatorics*. COCOON 2020 (Aug. 29–31, 2020). Ed. by Donghyun Kim, R. N. Uma, and Zhipeng Cai. Vol. 12273. LNCS. Springer, 2020, pp. 484–495. DOI: 10.1007/978-3-030-58150-339.

[GII+19] Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. "$\mathsf{AC}^0[p]$ Lower Bounds against MCSP via the Coin Problem". In: *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*. ICALP 2019 (Patras, Greece, July 9–Dec. 9, 2019). Ed. by Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi. Vol. 132. LIPIcs. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019, 66:1–66:15. DOI: 10.4230/LIPIcs.ICALP.2019.66.

[GSV99] Oded Goldreich, Amit Sahai, and Salil Vadhan. "Can Statistical Zero Knowledge Be Made Non-Interactive? Or On the Relationship of SZK and NISZK". In: *Advances in Cryptology*. CRYPTO 1999 (Santa Barbara, CA, USA, Aug. 15–19, 1999). Ed. by Michael Wiener. Springer Berlin Heidelberg, 1999, pp. 467–484. DOI: 10.1007/3-540-48405-1_30.

[HRST17] Johan Håstad, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. "An Average-Case Depth Hierarchy Theorem for Boolean Circuits". In: *Journal of the ACM* 64.5 (Oct. 2017), 35:1–35:27. DOI: 10.1145/3095799.

[Ila20a] Rahul Ilango. "Approaching MCSP from above and below: Hardness for a Conditional Variant and $\mathsf{AC}^0[p]$". In: *11th Innovations in Theoretical Computer Science Conference*. ITCS 2020. Ed. by Thomas Vidick. Vol. 151. LIPIcs. Keywords: Minimum Circuit Size Problem, reductions, NP-completeness, circuit lower bounds. Dagstuhl, Germany: Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2020, 34:1–34:26. DOI: 10.4230/LIPIcs.ITCS.2020.34.

[Ila20b] Rahul Ilango. "Constant Depth Formula and Partial Function Versions of MCSP Are Hard". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*. FOCS 2020 (Nov. 16–19, 2020). IEEE Computer Society, 2020, pp. 424–433. DOI: 10.1109/FOCS46700.2020.00047.

[ILO20] Rahul Ilango, Bruno Loff, and Igor C Oliveira. "NP-Hardness of Circuit Minimization for Multi-Output Functions". In: (2020), p. 36.

[KC00] Valentine Kabanets and Jin-Yi Cai. "Circuit Minimization Problem". In: *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*. STOC 2000 (Portland, Oregon, USA, May 21–23, 2000). Association for Computing Machinery, 2000, pp. 73–79. DOI: 10.1145/335305.335314.

[KS08] Subhash Khot and Rishi Saket. "Hardness of Minimizing and Learning DNF Expressions". In: *2008 49th Annual IEEE Symposium on Foundations of Computer Science*. FOCS 2008 (Philadelphia, PA, USA, Oct. 25–28, 2008). 2008, pp. 231–240. DOI: 10.1109/FOCS.2008.37.

[LP21] Yanyi Liu and Rafael Pass. "On One-Way Functions from NP-Complete Problems". Cryptology ePrint Archive. Cryptology ePrint Archive. Apr. 2021. URL: https://eprint.iacr.org/2021/513.pdf.

[MW17] Cody D. Murray and R. Ryan Williams. "On the (Non) NP-Hardness of Computing Circuit Complexity". In: *Theory of Computing* 13.4 (2017). Conference version in CCC 2015, pp. 1–22. DOI: 10.4086/toc.2017.v013a004.

[SV10] Ronen Shaltiel and Emanuele Viola. "Hardness Amplification Proofs Require Majority". In: *SIAM Journal on Computing* 39.7 (2010). Conference version in STOC 2008, pp. 3122–3154. DOI: 10.1145/1374376.1374461.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions*. Wiley-Teubner Series in Computer Science. John Wiley & Sons, Ltd., and B. G. Teubner, Stuttgart, 1987.