New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems*

Eric Allender[1] and Shuichi Hirahara[2]

1 **Department of Computer Science, Rutgers University, Piscataway, NJ, USA**
allender@cs.rutgers.edu
2 **Department of Computer Science, The University of Tokyo, Tokyo, Japan**
hirahara@is.s.u-tokyo.ac.jp

Abstract

The Minimum Circuit Size Problem (MCSP) and a related problem (MKTP) that deals with time-bounded Kolmogorov complexity are prominent candidates for NP-intermediate status. We show that, under very modest cryptographic assumptions (such as the existence of one-way functions), the problem of approximating the minimum circuit size (or time-bounded Kolmogorov complexity) within a factor of $n^{1-o(1)}$ is *indeed* NP-intermediate. To the best of our knowledge, these problems are the first natural NP-intermediate problems under the existence of an arbitrary one-way function.

We also prove that MKTP is hard for the complexity class DET under non-uniform NC^0 reductions. This is surprising, since prior work on MCSP and MKTP had highlighted weaknesses of "local" reductions such as $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$. We exploit this local reduction to obtain several new consequences:

- MKTP is not in $\mathsf{AC}^0[p]$.
- Circuit size lower bounds are equivalent to hardness of a relativized version MKTP^A of MKTP under a class of uniform AC^0 reductions, for a large class of sets A.
- Hardness of MCSP^A implies hardness of MKTP^A for a wide class of sets A. This is the first result directly relating the complexity of MCSP^A and MKTP^A, for any A.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases computational complexity, Kolmogorov complexity, circuit size

Digital Object Identifier 10.4230/LIPIcs.MFCS.2017.54

1 Introduction

The Minimum Circuit Size Problem (MCSP) has attracted intense study over the years, because of its close connection with the natural proofs framework of Razborov and Rudich [23], and because it is a prominent candidate for NP-intermediate status. It has been known since [18] that NP-intermediate problems exist, if P $\neq$ NP, but "natural" candidates for this status are rare. Problems such as factoring and Graph Isomorphism are sometimes put forward as candidates, but there are not strong complexity-theoretic arguments for why these problems should not lie in P. We prove that a very weak cryptographic assumption implies that a $n^{1-o(1)}$ approximation for MCSP is NP-intermediate.

* Supported by NSF grant CCF-1555409 (Allender) and JSPS KAKENHI Grant Numbers JP16J06743 (Hirahara). Proofs of some results have been omitted due to space limits; more details can be found at [6].

42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017).
Editors: Kim G. Larsen, Hans L. Bodlaender, and Jean-Francois Raskin; Article No. 54; pp. 54:1–54:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

MCSP is hard for SZK [4] under BPP reductions, but the situation is quite different, when more restricted notions of reducibility are considered. Recent results [14, 19, 7] have suggested that MCSP might not even be hard for P under logspace reductions (although the evidence is still inconclusive).

The input to MCSP consists of a pair (T, s), where T is a bit string of length 2^m representing the truth-table of an m-variate Boolean function, and $s \in \mathbb{N}$; $(T, s) \in \mathsf{MCSP}$ if there is a circuit computing T having size at most s. Note that, for different models of circuit (type of gates, allowable fan-in, etc.) and different measures of size (number of gates, number of wires, size of the description of the circuit, etc.) the resulting MCSP problems might have different complexity. No efficient reduction is known between different variants of the problem. However, all prior work on MCSP (such as [16, 3, 9, 19, 4, 25, 7, 14]) applies equally well to any of these variants. MCSP is also closely related to a type of time-bounded Kolmogorov complexity known as KT, which was defined in [3]. The problem of determining KT complexity, formalized as the language $\mathsf{MKTP} = \{(x, s) : \mathrm{KT}(x) \leq s\}$ has often been viewed as just another equivalent "encoding" of MCSP in this prior work. (In particular, our results mentioned in the paragraphs above apply also to MKTP.) Recently, however, some reductions were presented that are not currently known to apply to MCSP [5].

In this section, we outline the ways in which this paper advances our understanding of MCSP and related problems, while reviewing some of the relevant prior work.

Hardness is equivalent to circuit size lower bounds. Significant effort (e.g. [16, 19, 7, 14])) has been made in order to explain why it is so difficult to show NP-hardness of MCSP or MKTP. Most of the results along this line showed implications from hardness of MCSP to circuit size lower bounds: If MCSP or MKTP is NP-hard under some restricted types of reductions, then a circuit size lower bound (which is quite difficult to obtain via current techniques of complexity theory) follows. For example, if MCSP or MKTP is hard for TC^0 under Dlogtime-uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, then $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ and $\mathsf{DSPACE}(n) \not\subseteq \mathsf{io\text{-}SIZE}(2^{\epsilon n})$ [19, 7].

Murray and Williams [19] asked if, in general, circuit lower bounds imply hardness of the circuit minimization problems. We answer their questions affirmatively in certain settings: A stronger lower bound $\mathsf{DSPACE}(n) \not\subseteq \mathsf{io\text{-}SIZE}^{\mathsf{MKTP}}(2^{\epsilon n})$ implies that MKTP is hard for DET under logspace-uniform $\leq_{\mathrm{tt}}^{\mathsf{AC}^0}$ reductions (Theorem 11).

At this point, it is natural to ask if the circuit lower bounds are in fact *equivalent* to hardness of MKTP. We indeed show that this is the case, when we consider the minimum *oracle* circuit size problem. For an oracle A, MCSP^A is the set of pairs (T, s) such that T is computed by a size-s circuit that has "oracle gates" for A in addition to standard AND and OR gates. The related MKTP^A problem asks about the time-bounded Kolmogorov complexity of a string, when the universal Turing machine has access to the oracle A. For many oracles A that are hard for PH, we show that $\mathsf{DSPACE}(n) \not\subseteq \mathsf{io\text{-}SIZE}^A(2^{\epsilon n})$ for some $\epsilon > 0$ *if and only if* MKTP^A is hard for DET under a certain class of reducibilities (Theorem 12).

That is, it is impossible to prove hardness of MKTP^A (under some reducibilities) without proving circuit lower bounds, and vice versa. Our results clearly connect the fact that it is difficult to obtain hardness of MKTP^A with the fact that circuit size lower bounds are difficult.

Hardness under local reductions, and unconditional lower bounds. Murray and Williams [19] showed that MCSP and MKTP are not hard for TC^0 under so-called *local* reductions computable in time less than $\sqrt{n}$ – and thus in particular they are not hard under NC^0

reductions that are very uniform (i.e., there is no routine computable in time $t(n) < n^{.5-\epsilon}$ that, on input (n, i) outputs the $O(1)$ queries upon which the i-th output bit of such an NC^0 circuit depends). Murray and Williams speculated that this might be a promising first step toward showing that MCSP is not hard for NP under Dlogtime-uniform AC^0 reductions, since it follows from [1] that any set that is hard for TC^0 under P-uniform AC^0 reductions is also hard for TC^0 under P-uniform NC^0 reductions. Indeed, the results of Murray and Williams led us to expect that MCSP and MKTP are not even hard for PARITY under non-uniform NC^0 reductions.

Contrary to these expectations, we show that MKTP is hard not only for TC^0 but even for the complexity class DET under non-uniform NC^0 reductions (Theorem 9). Consequently, MKTP is not in $\mathsf{AC}^0[p]$ for any prime p.[1] Note that it is still not known whether MCSP or $R_{\mathrm{KT}} = \{x : \mathrm{KT}(x) \geq |x|\}$ is in $\mathsf{AC}^0[p]$. It is known[2] [3] that neither of these problems is in AC^0. Under a plausible derandomization hypothesis, this non-uniform reduction can be converted into a logspace-uniform $\leq_{\mathrm{tt}}^{\mathsf{AC}^0}$ reduction that is an AND of NC^0-computable queries. Thus "local" reductions are more effective for reductions to MKTP than may have been suspected.

Implications among hardness conditions for MKTP and MCSP. No $\leq_{\mathrm{T}}^{\mathsf{P}}$ reductions are known between MKTP^A or MCSP^A for any A. Although most previous complexity results for one of the problems have applied immediately to the other, via essentially the same proof, there has not been any proven relationship among the problems. For the first time, we show that, for many oracles A, hardness for MCSP^A implies hardness for MKTP^A (Theorem 12).

A reduction that is not "oracle independent". Hirahara and Watanabe [14] observed that all of the then-known reductions to MCSP and MKTP were "oracle-independent", in the sense that, for any class $\mathcal{C}$ and reducibility $\leq_r$, all proofs that MCSP (or MKTP) is hard for $\mathcal{C}$ under $\leq_r$ also show that MCSP^A (MKTP^A) is also hard for $\mathcal{C}$. They showed that oracle-independent $\leq_{\mathrm{T}}^{\mathsf{P}}$-reductions cannot show hardness for any class larger than P.

This motivates the search for reductions that are *not* oracle-independent. We give a concrete example of a logspace-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$ reduction that (under a plausible complexity assumption) reduces DET to MKTP. This is *not* an oracle independent reduction, since $\mathsf{MKTP}^{\mathsf{QBF}}$ is not hard for DET under this same class of reductions (Corollary 13).

A clearer picture of how hardness "evolves". It is instructive to contrast the evolution of the class of problems reducible to MKTP^A under different types of reductions, as A varies from very easy ($A = \emptyset$) to complex ($A = \mathsf{QBF}$). For this thought experiment, we assume the very plausible hypothesis that $\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}(2^{\epsilon n})$. Restrictions of QBF give a useful parameterization for the complexity of A. Consider A varying from being complete for each level of PH (that is, quantified Boolean formulas with $O(1)$ alternations between $\forall$ and $\exists$ quantifiers), to instances of QBF with $\log^* n$ alternations, then to $O(\log n)$ alternations etc.,

---

[1] Subsequent to our work, a stronger average-case lower bound against $\mathsf{AC}^0[p]$ was proved [13]. The techniques of [13] do not show how to reduce DET, or even smaller classes such as TC^0, to MKTP. Thus our work is incomparable to [13].

[2] Somewhat remarkably, Oliveira and Santhanam [20] have independently shown that MCSP and MKTP are hard for DET under non-uniform $\leq_{\mathrm{tt}}^{\mathsf{TC}^0}$ reductions. Their proof relies on self-reducibility properties of the determinant, whereas our proof relies on the fact that Graph Isomorphism is hard for DET [27]. Their results have the advantage that they apply to MCSP rather than merely to MKTP, but because it is not known whether $\mathsf{TC}^0 = \mathsf{P}$ they do not obtain unconditional lower bounds, as in Corollary 10.

through to $2^{\sqrt{\log n}}$ alternations, and until finally $A = \mathsf{QBF}$. Since $\mathsf{DSPACE}(n) \subseteq \mathsf{P}^A/\mathsf{poly}$, at some point in this evolution we have $\mathsf{DSPACE}(n) \subseteq \text{io-}\mathsf{SIZE}^A(2^{\epsilon n})$; it is plausible to assume that this doesn't happen until A has at least $\log n$ quantifier alternations, or more.

At all stages in this evolution $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathsf{MKTP}^\mathsf{A}}$ [4], until at some point $\mathsf{BPP}^{\mathsf{MKTP}^\mathsf{A}}$ expands to coincide with PSPACE [3]. Also, at all stages in this evolution DET $\leq_\mathrm{m}^{\mathsf{NC}^0}$-reduces to MKTP^A (and even when $A = \mathsf{QBF}$ we do not know, for instance, if $\mathsf{NC}^3 \leq_\mathrm{m}^{\mathsf{NC}^0}$-reduces to MKTP^A). Thus these reductions behave "monotonically", in the sense that as the complexity of A increases, the class of problems reducible to MKTP^A does not shrink noticeably, and sometimes appears to grow markedly.

The situation is much more intriguing when we consider the *uniform* class of $\leq_\mathrm{T}^{\mathsf{AC}^0}$ reductions that arise from derandomizing the nonuniform $\leq_\mathrm{m}^{\mathsf{NC}^0}$ reductions from DET. At the start, when $A = \emptyset$, we have DET reducing to MKTP^A, and this is maintained until A becomes complex enough so that $\mathsf{DSPACE}(n) \subseteq \text{io-}\mathsf{SIZE}^A(2^{\epsilon n})$. At this point, not only does DET not reduce to MKTP^A, but neither does PARITY! (See Theorem 12.)

This helps place the results of [7] in the proper context. In [7] strong evidence was presented against $\mathsf{MCSP}^{\mathsf{QBF}}$ being hard for, say, P under $\leq_\mathrm{m}^{\mathsf{L}}$ reductions, and this was taken as indirect evidence that MCSP itself should not be hard for P, since $\mathsf{MCSP} \in \mathsf{NP}$ and thus is much "easier" than the PSPACE-complete problem $\mathsf{MCSP}^{\mathsf{QBF}}$. However, we expect that MCSP^A and MKTP^A should behave somewhat similarly to each other, and it *can* happen that a class can reduce to MKTP (Theorem 11) and *not* reduce to MKTP^A for a more powerful oracle A (Corollary 13).

Hardness of the Gap problem. Our new hardness results for MKTP^A share with earlier reductions the property that they hold even for "Gap" versions of the problem. That is, for some $\epsilon > 0$, the reduction works correctly for any solution to the promise problem with "yes" instances $\{(x, s) : \mathrm{KT}^A(x) \leq s\}$ and "no" instances $\{(x, s) : \mathrm{KT}^A(x) > s + |x|^\epsilon\}$. However, we do not know if they carry over to instances with a wider "gap" between the Yes and No instances; earlier hardness results such as those of [3, 9, 4, 25] hold for a much wider gap (such as with the Yes instances having $\mathrm{KT}(x) < |x|^\epsilon$, and the no instances with $\mathrm{KT}(x) \geq |x|$), and this is one reason why they applied both to MKTP and to MCSP. Thus there is interest in whether it is possible to reduce MCSP with small "gap" to MCSP with large "gap". If this were possible, then MCSP and MKTP would be interreducible in some sense.

Earlier work [7] had presented unconditional results, showing that "gap" versions of MCSP could not be hard for TC^0 under $\leq_\mathrm{m}^{\mathsf{AC}^0}$ reductions, unless those reductions had large "stretch" (mapping short inputs to long outputs). In [6], we show that BPP-Turing reductions among gap MCSP problems require large stretch, unless $\mathsf{MCSP} \in \mathsf{BPP}$.

Natural NP-intermediate Problems. In Section 3 we also consider gap MCSP problems where the "gap" is quite large (i.e., problems of approximating the minimum circuit size for a truth table of size n within a factor of $n^{1-o(1)}$). Problems of this sort are of interest, because of the role they play in the natural proofs framework of [23], if one is trying to prove circuit lower bounds of size $2^{o(n)}$. Our Theorem 6 shows that these problems are NP-intermediate in the sense that these do not lie in P/poly and are not NP-hard under P/poly reductions, under modest cryptographic assumptions (weaker than assuming that factoring or discrete log requires superpolynomial-size circuits, or assuming the existence of a one-way function). To the best of our knowledge, these problems are the first natural NP-intermediate problems under the existence of an arbitrary one-way function.

Our new insight on MCSP here is that, if the gap problems are NP-hard, then MCSP is "strongly downward self-reducible": that is, any instance of MCSP of size n can be reduced to instances of size n^ϵ. In the past, many natural problems have been shown to be strongly downward self-reducible (see [8]); Our contribution is to show that MCSP also has such a property (under the assumption that the gap MCSP problems are NP-hard).

2 Preliminaries

We assume the reader is familiar with standard DTIME and DSPACE classes. We also occasionally refer to classes defined by time-bounded *alternating* Turing machines: $\mathsf{ATIME}(t(n))$, or by simultaneously bounding time and the number of alternations between existential and universal configurations: $\mathsf{ATIME\text{-}ALT}(t(n), a(n))$.

We refer the reader to the text by Vollmer [29] for background and more complete definitions of the standard circuit complexity complexity classes

$$\mathsf{NC}^0 \subsetneq \mathsf{AC}^0 \subsetneq \mathsf{AC}^0[p] \subsetneq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{P/poly},$$

as well as the standard complexity classes $\mathsf{L} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PH} \subseteq \mathsf{PSPACE}$. Between L and P in this list, there is one more class that plays an important role for us: DET is the class of problems that are reducible to the problem of computing the determinant of integer matrices, by NC^1-Turing reductions.

This brings us to the topic of reducibility. Let $\mathcal{C}$ be either a class of functions or a class of circuits. We say that $A \leq_{\mathrm{m}}^{\mathcal{C}} B$ if there is a function $f \in \mathcal{C}$ (or f computed by a circuit family in $\mathcal{C}$, respectively) such that $x \in A$ iff $f(x) \in B$. We will make use of $\leq_{\mathrm{m}}^{\mathsf{L}}, \leq_{\mathrm{m}}^{\mathsf{TC}^0}, \leq_{\mathrm{m}}^{\mathsf{AC}^0}$ and $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reducibility. The more powerful notion of Turing reducibility also plays an important role in this work. Here, $\mathcal{C}$ is a complexity class that admits a characterization in terms of Turing machines or circuits, which can be augmented with an "oracle" mechanism, either by providing a "query tape" or "oracle gates". We say that $A \leq_{\mathrm{T}}^{\mathcal{C}} B$ if there is a oracle machine in $\mathcal{C}$ (or a family of oracle circuits in $\mathcal{C}$) accepting A, when given oracle B. We make use of $\leq_{\mathrm{T}}^{\mathsf{P/poly}}, \leq_{\mathrm{T}}^{\mathsf{BPP}}, \leq_{\mathrm{T}}^{\mathsf{P}}, \leq_{\mathrm{T}}^{\mathsf{L}}$ and $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reducibility; instead of writing $A \leq_{\mathrm{T}}^{\mathsf{P/poly}} B$ or $A \leq_{\mathrm{T}}^{\mathsf{BPP}} B$, we will more frequently write $A \in \mathsf{P}^B/\mathsf{poly}$ or $A \in \mathsf{BPP}^B$. Turing reductions that are "nonadaptive" – in the sense that the list of queries that are posed on input x does not depend on the answers provided by the oracle – are called *truth-table reductions*. We make use of $\leq_{\mathrm{tt}}^{\mathsf{AC}^0}$ and $\leq_{\mathrm{tt}}^{\mathsf{TC}^0}$ reducibility.

Kabanets and Cai [16] sparked renewed interest in MCSP and highlighted connections between MCSP and more recent progress in derandomization. They introduced a class of reductions to MCSP, which they called *natural reductions*. Recall that instances of MCSP are of the form (T, s) where s is a "size parameter". A $\leq_{\mathrm{m}}^{\mathsf{P}}$ reduction f is called *natural* if $f(x)$ is of the form $f(x) = (f_1(x), f_2(|x|))$. That is, the "size parameter" is the same, for all inputs x of the same length.

Whenever circuit families are discussed (either when defining complexity classes, or reducibilities), one needs to deal with the issue of *uniformity*. For example, the class AC^0 (corresponding to families $\{C_n : n \in \mathbb{N}\}$ of unbounded fan-in AND, OR, and NOT gates having size $n^{O(1)}$ and depth $O(1)$) comes in various flavors, depending on the complexity of computing the mapping $1^n \mapsto C_n$. When this is computable in polynomial time (or logarithmic space), then one obtains P-uniform AC^0 (logspace-uniform AC^0, respectively). If no restriction at all is imposed, then one obtains non-uniform AC^0. As discussed in [29], the more restrictive notion of Dlogtime-uniform AC^0 is frequently considered to be the "right" notion of uniformity to use when discussing small complexity classes such as $\mathsf{AC}^0, \mathsf{AC}^0[p]$ and TC^0. If these classes are mentioned with no explicit mention of uniformity, then Dlogtime-uniformity is intended. For uniform NC^1 the situation is somewhat more complicated, as discussed in [29]; there is wide agreement that the "correct" definition coincides with $\mathsf{ATIME}(O(\log n))$.

There are many ways to define time-bounded Kolmogorov complexity. The definition $\mathrm{KT}(x)$ was proposed in [3], and has the advantage that it is polynomially-related to circuit size (when a string x is viewed as the truth-table of a function). $\mathrm{KT}(x)$ is the minimum, over all d and t, of $|d| + t$, such that the universal Turing machine U, on input (d, i, b) can determine in time t if the i-th bit of x is b. (More formal definitions can be found in [3].)

A *promise problem* consists of a pair of disjoint subsets (Y, N). A language A is a *solution* to the promise problem (Y, N) if $Y \subseteq A \subseteq \overline{N}$. A language B reduces to a promise problem via a type of reducibility $\leq_r$ if $B \leq_r A$ for *every* set A that is a solution to the promise problem.

3 GapMCSP

In this section, we consider the "gap" versions of MCSP and MKTP. We focus primarily on MCSP, and for simplicity of exposition we consider the "size" of a circuit to be the number of AND and OR gates of fan-in two. (NOT gates are "free"). The arguments can be adjusted to consider other circuit models and other reasonable measures of "size" as well. Given a truth-table T, let $\mathrm{CC}(T)$ be the size of the smallest circuit computing T, using this notion of "size".

▶ **Definition 1.** For any function $\epsilon\colon \mathbb{N} \to (0, 1)$, let $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ be the approximation problem that, given a truth-table T, asks for outputting a value $f(T) \in \mathbb{N}$ such that

$$\mathrm{CC}(T) \leq f(T) \leq |T|^{1-\epsilon(|T|)} \cdot \mathrm{CC}(T).$$

Note that this approximation problem can be formulated as the following promise problem. (See also [11] for similar comments.)

▶ **Fact 2.** $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ *is polynomial-time Turing equivalent to the following promise problem* (Y, N)*:*

$$Y := \{\, (T, s) \mid \mathrm{CC}(T) < s/|T|^{1-\epsilon(|T|)} \,\},$$
$$N := \{\, (T, s) \mid \mathrm{CC}(T) > s + 1 \,\},$$

where T is a truth-table and $s \in \mathbb{N}$.

Note that $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ becomes easier when ϵ becomes smaller. If $\epsilon(n) = o(1)$, then (using the promise problem formulation) it is easy to see that $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ has a solution in $\mathsf{DTIME}(2^{n^{o(1)}})$, since the Yes instances have witnesses of length $|T|^{o(1)}$. However, it is worth emphasizing that, even when $\epsilon(n) = o(1)$, $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ is a canonical example of a combinatorial property that is useful in proving circuit size lower bounds of size $2^{o(n)}$, in the sense of [23]. Thus it is of interest that MCSP cannot reduce to $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ in this regime under very general notions of reducibility, unless MCSP itself is easy.

▶ **Theorem 3.** *For any polynomial-time-computable nonincreasing $\epsilon(n) = o(1)$, if* $\mathsf{MCSP} \in \mathsf{BPP}^{\mathrm{Gap}_\epsilon\mathrm{MCSP}}$ *then* $\mathsf{MCSP} \in \mathsf{BPP}$.

A new idea is that $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ is "strongly downward self-reducible." We will show that any $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ instance of length n is reducible to $n^{1-\epsilon}$ MCSP instances of length n^ϵ. To this end, we will exploit the following simple fact.

▶ **Lemma 4.** *For a function $f\colon \{0,1\}^n \to \{0,1\}$, a string $x \in \{0,1\}^k$ and $k \in \mathbb{N}$, let $f_x\colon \{0,1\}^{n-k} \to \{0,1\}$ be a function defined as $f_x(y) := f(x,y)$. Then, the following holds:*

$$\max_{x\in\{0,1\}^k} \mathrm{CC}(f_x) \le \mathrm{CC}(f) \le 2^k \cdot \left(\max_{x\in\{0,1\}^k} \mathrm{CC}(f_x) + 3\right),$$

(In other words, $\max_{x\in\{0,1\}^k} \mathrm{CC}(f_x)$ gives an approximation of $\mathrm{CC}(f)$ within a factor of 2^k.)

Proof of Theorem 3. Let M be an oracle BPP Turing machine which reduces MCSP to $\mathrm{Gap}_\epsilon\mathrm{MCSP}$. Let $|T|^c$ be an upper bound for the running time of M, given a truth-table T, and let $|T| = 2^n$.

We recursively compute the circuit complexity of T by the following procedure: Run M on input T. If M makes a query S to the $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ oracle, then divide S into consecutive substrings $S_1, \cdots, S_{2^k}$ of length $|S| \cdot 2^{-k}$ such that $S_1 \cdot S_2 \cdots S_{2^k} = S$ (where k is a parameter, chosen later, that depends on $|S|$), and compute the circuit complexity of each S_i recursively for each $i \in [2^k]$. Then continue the simulation of M, using the value $2^k \cdot \left(\max_{i\in[2^k]} \mathrm{CC}(S_i) + 3\right)$ as an approximation to $\mathrm{CC}(S)$.

We claim that the procedure above gives the correct answer. For simplicity, let us first assume that the machine M has zero error probability. It suffices to claim that the simulation of M is correct in the sense that every query of M is answered with a value that satisfies the approximation criteria of $\mathrm{Gap}_\epsilon\mathrm{MCSP}$. Suppose that M makes a query S. By the assumption on the running time of M, we have $|S| \le |T|^c = 2^{nc}$. By Lemma 4, we have

$$\mathrm{CC}(S) \le 2^k \cdot \left(\max_{i\in[2^k]} \mathrm{CC}(S_i) + 3\right) \le 2^k \cdot (\mathrm{CC}(S) + 3).$$

In particular, the estimated value satisfies the promise of $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ if $2^k \cdot (\mathrm{CC}(S) + 3) \le |S|^{1-\epsilon(|S|)} \cdot \mathrm{CC}(S)$. Since we may assume without loss of generality that $\mathrm{CC}(S) \ge 3$, it suffices to make sure that $2^{k+1} \cdot \mathrm{CC}(S) \le |S|^{1-\epsilon(|S|)} \cdot \mathrm{CC}(S)$. Let $|S| = 2^m$. Then, in order to satisfy $k + 1 \le (1 - \epsilon(|S|)) \cdot m$, let us define $k := (1 - \epsilon(|S|)) \cdot m - 1$. For this particular choice of k, the estimated value $2^k \cdot \left(\max_{i\in[2^k]} \mathrm{CC}(S_i) + 3\right)$ of the circuit complexity of S satisfies the promise of $\mathrm{Gap}_\epsilon\mathrm{MCSP}$, which implies that the reduction M computes the correct answer for MCSP.

Now we analyze the time complexity of the algorithm. Each recursive step makes at most 2^{2cn} many recursive calls, because there are potentially 2^{cn} many queries S of M, each of which may produce at most $2^k \le 2^{cn}$ recursive calls. The length of each truth-table S_i that arises in one of the recursive calls is $|S_i| = |S| \cdot 2^{-k} = 2^{m-k} = 2^{\epsilon(|S|)\cdot m+1}$. We claim that $|S_i| \le 2^{1+(n/2)}$ holds for sufficiently large n. Let us take n to be large enough so that $\epsilon(2^{n/2}) \le 1/2c$. If $m \ge n/2$, then $|S_i| \le 2^{\epsilon(2^m)\cdot m+1} \le 2^{\epsilon(2^{n/2})\cdot cn+1} \le 2^{1+(n/2)}$. Otherwise, since $m \le n/2$ and $\epsilon(|S|) < 1$, we obtain $|S_i| \le 2^{\epsilon(|S|)\cdot m+1} \le 2^{1+(n/2)}$. Therefore, on inputs of length 2^n, each recursive call produces instances of length at most $2^{1+(n/2)}$. The overall time complexity can be estimated as $2^{c'n} \cdot 2^{c'n/2} \cdot 2^{c'n/4} \cdots = 2^{2c'n}$ for some constant c' (say, $c' = 3c$), which is a polynomial in the input length 2^n.

We note that the analysis above works even for *randomized* reductions that may err with exponentially small probability. Since we have proved that the algorithm runs in polynomial time, the probability that the algorithm makes an error is at most a polynomial times an exponentially small probability, which is still exponentially small probability (by the union bound). ◀

▶ Remark. If we drop the assumption that $\epsilon(n)$ be computable, then the proof of Theorem 3 still shows that if $\mathsf{MCSP} \in \mathsf{P}^{\mathrm{Gap}_\epsilon\mathrm{MCSP}}/\mathsf{poly}$ then $\mathsf{MCSP} \in \mathsf{P}/\mathsf{poly}$.

▶ **Corollary 5.** *Let $\epsilon(n) = o(1)$. If* $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ *has no solution in* $\mathsf{P/poly}$ *then* $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ *is not hard for* NP *(or even for* MCSP*) under* $\leq_{\mathrm{T}}^{\mathsf{P/poly}}$ *reductions, and is thus* NP*-intermediate.*

Proof. This is immediate from the preceding remark. If $\mathsf{MCSP} \in \mathsf{P}^{\mathrm{Gap}_\epsilon\mathrm{MCSP}}/\mathsf{poly}$ then $\mathsf{MCSP} \in \mathsf{P/poly}$, which in turn implies that $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ has a solution in $\mathsf{P/poly}$. ◀

In what follows, we show that the assumption of Corollary 5 is true under very modest cryptographic assumptions. It is known that, for any constant $\epsilon > 0$, $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ is SZK-hard under $\leq_{\mathrm{T}}^{\mathsf{P/poly}}$ reductions [4]. Here, we show that if SZK is not in $\mathsf{P/poly}$, then for some $\epsilon(n) = o(1)$, $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ has no solution in $\mathsf{P/poly}$. In fact, we can prove something *stronger*: If auxiliary-input one-way functions exist, then $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ is not in $\mathsf{P/poly}$. We now describe auxiliary-input one-way functions.

Usually, the existence of cryptographically-secure one-way functions is considered to be essential for meaningful cryptography. That is, one requires a function f computed in polynomial time such that, for any algorithm A computed by polynomial-sized circuits, $\Pr_x[f(A(f(x))) = f(x)] = 1/n^{\omega(1)}$ where x is chosen uniformly at random from $\{0,1\}^n$. A weaker notion that has been studied in connection with SZK goes by the name *auxiliary-input one-way functions*. This is an indexed family of functions $f_y(x) = F(y, x)$, where $|x| = p(|y|)$ for some polynomial p, and F is computable in time polynomial in $|y|$, such that for some infinite set I, for any algorithm[3] A computed by polynomial-sized circuits, for all $y \in I$, $\Pr_x[f_y(A(f_y(x))) = f_y(x)] = 1/n^{\omega(1)}$ where $n = |y|$ and x is chosen uniformly at random from $\{0,1\}^{p(n)}$. It is known that there are promise problems in SZK that have no solution in $\mathsf{P/poly}$ only if auxiliary-input one-way functions exist. (This is due to [22]; a good exposition can be found in [28, Theorems 7.1 & 7.5], based on earlier work of [21].)

▶ **Theorem 6.** *If auxiliary-input one-way functions exist, then there is a function $\epsilon(n) = o(1)$ such that* $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ *is* NP*-intermediate. (Namely,* $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ *has no solution in* $\mathsf{P/poly}$ *and* $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ *is not* NP*-hard under* $\leq_{\mathrm{T}}^{\mathsf{P/poly}}$ *reductions.)*

▶ Remark. In particular, either one of the following implies that some $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ is NP-intermediate, since each implies the existence of auxiliary-input one-way functions:
1. the existence of cryptographically-secure one-way functions.
2. SZK is not in $\mathsf{P/poly}$.

4 Hardness for DET

In this section, we give some of our main contributions. We show that MKTP is hard for DET under $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reductions (Theorem 9); prior to this, no variant of MCSP was known to be hard for any complexity class under any type of many-one reducibility. The $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reduction that we present is nonuniform; we show that hardness under *uniform* reductions is related to lower bounds in circuit complexity, and in some cases we show that circuit lower bounds are *equivalent* to hardness results under uniform notions of reducibility (Theorem 12). These techniques yield the first results relating the complexity of MCSP^A and MKTP^A problems.

Here is the outline of this section. We will build on a randomized reduction of Allender, Grochow and Moore [5]: They showed that there is a ZPP reduction from the rigid[4] graph

[3] We have chosen to define one-way functions in terms of security against non-uniform adversaries. It is also common to use the weaker notion of security against probabilistic polynomial-time adversaries, as in [28].

[4] A graph is *rigid* if it has no nontrivial automorphisms.

isomorphism problem to MKTP. Here we show that the reduction is in fact an AC^0 reduction (Corollary 8). Combining Torán's AC^0 reduction [27] from DET to the rigid graph isomorphism as well as the Gap theorem [2], we will show $\mathsf{DET} \leq_{\mathrm{m}}^{\mathsf{NC}^0} \mathsf{MKTP}$ (Theorem 9).

To show that circuit size lower bounds are equivalent to hardness under uniform AC^0 reductions, we will derandomize the reduction of [5] (Theorem 11). To this end, we give an AC^0 reduction f from the rigid graph isomorphism problem to MKTP and an "encoder" e that encodes random binary strings into a random permutation in Lemma 7 below.

▶ **Lemma 7.** *Let A be any oracle. There is a function f computable in Dlogtime-uniform AC^0 and a function e computable in Dlogtime-uniform TC^0 such that, for any two rigid graphs G, H with n vertices:*

- $\Pr_r[f(G,H,e(r)) \notin \mathsf{MKTP}^A] > 1 - \frac{1}{2^{4n^2}}$ *if $G \not\equiv H$, and*
- $\Pr_r[f(G,H,e(r)) \in \mathsf{MKTP}^A] = 1$ *if $G \equiv H$.*

▶ **Corollary 8.** *Let A be any oracle. The rigid graph isomorphism problem is reducible to MKTP^A via a non-uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reduction.*

Proof. A standard counting argument shows that there is a value of $e(r)$ that can be hardwired into the reduction of Lemma 7 that works correctly for all pairs (G, H) of n-vertex graphs. (Note that the input length is $2n^2$, and the error probability is at most $1/2^{4n^2}$.) ◀

▶ **Theorem 9.** *Let A be any oracle. DET is reducible to MKTP^A via a non-uniform $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reduction. Furthermore, this reduction is "natural" in the sense of [16].*

Proof. Since DET is closed under $\leq_{\mathrm{m}}^{\mathsf{TC}^0}$ reductions, it suffices to show that MKTP^A is hard under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, and then appeal to the "Gap" theorem of [2], to obtain hardness under $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reducibility. Torán [27] shows that DET is AC^0-reducible to GI, and the proofs of Theorem 5.3 and Corollary 5.4 of [27] show that DET is AC^0-reducible to GI via a reduction that outputs only pairs of rigid graphs. Composing this reduction with the non-uniform AC^0 reduction given by Corollary 8 completes the argument. (Since DET is closed under complement, there is also a non-uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reduction to the complement of MKTP^A.)

Since the same θ is used for all inputs of the same length, the reduction is "natural". ◀

The lower bounds of Razborov and Smolensky [24, 26] yield the following corollary:

▶ **Corollary 10.** *MKTP^A is not in $\mathsf{AC}^0[p]$ for any oracle A and any prime p.*

(An alternate proof of this circuit lower bound can be obtained by applying the pseudorandom generator of [10] that has sublinear stretch and is secure against $\mathsf{AC}^0[p]$. Neither argument seems easy to extend, to provide a lower bound for MCSP.)

One may wonder if the non-uniform reduction can be made uniform under a derandomization hypothesis. We do not know how to obtain a uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reduction, but we can come close, if A is not too complex. Recall the definition of ctt-reductions: $B \leq_{\mathrm{ctt}}^{\mathcal{C}} C$ if there is a function $f \in \mathcal{C}$ with the property that $f(x)$ is a list $f(x) = (y_1, \ldots, y_m)$, and $x \in B$ if and only if $y_j \in C$ for all j. Furthermore, we say that f is a *natural* logspace-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$-reduction to MKTP if each query y_j has the same length (and this length depends only on $|x|$), and also each y_j is of the form (z_j, θ) where the threshold θ depends only on $|x|$.

The following theorem can be viewed as a "partial converse" to results of [19, 7], which say that problems in $\mathsf{LTH} \subseteq \mathsf{E}$ require exponential size circuits if MCSP or MKTP is hard for TC^0 under Dlogtime-uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions. That is, the earlier results show that very uniform hardness results imply circuit lower bounds, whereas the next theorem shows that somewhat stronger circuit lower bounds imply uniform hardness results (for a less-restrictive notion of uniformity, but hardness for a larger class). Later on, in Theorem 12, we present a related condition on reductions to MKTP^A that is *equivalent* to circuit lower bounds.

▶ **Theorem 11.** *Let A be any oracle. If there is some $\epsilon > 0$ such that $\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$, then every language in DET reduces to MKTP^A via a natural logspace-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$-reduction.*

Proof. Let $B \in \mathsf{DET}$. Thus there is an AC^0 reduction g reducing B to the Rigid Graph Isomorphism Problem [27]. Consider the following family of statistical tests $T_x(r)$, indexed by strings x:

On input r:
Compute $z = f(g(x), e(r))$, where $f(G, H, e(r))$ is the function from Lemma 7. Accept iff ($x \in B$ iff $z \in \mathsf{MKTP}^A$).

Since $B \in \mathsf{DET} \subseteq \mathsf{P}$, the test $T_x(r)$ has a polynomial-size circuit with one MKTP^A oracle gate. (In fact, the statistical test is an NC^2 circuit with one oracle gate.) If $x \in B$, then T_x accepts *every* string r, whereas if $x \notin B$, T_x accepts most strings r.

Klivans and van Melkebeek [17] (building on the work of Impagliazzo and Wigderson [15]) show that, if $\mathsf{DSPACE}(n)$ requires exponential-size circuits from a given class $\mathcal{C}$, then there is a hitting set generator computable in logspace that hits all large sets computable by circuits from $\mathcal{C}$ having size n^k. In particular, under the given assumption, there is a function h computable in logspace such that $h(0^n) = (r_1, r_2, \ldots, r_{n^c})$ with the property that, for all strings x of length n, there is an element of $h(0^n)$ that is accepted by T_x.

Now consider the logspace-uniform AC^0 oracle circuit family, where the circuit for inputs of length n has the strings $e(h(0^n)) = (e(r_1), e(r_2), \ldots, e(r_{n^c}))$ hardwired into it. The circuit computes the queries $f(g(x), e(r_i))$ for $1 \leq i \leq n^c$, and accepts if, for all i, $f(g(x), e(r_i)) \in \mathsf{MKTP}^A$. Note that if $x \notin B$, then one of the r_i is accepted by T_x, which means that $f(g(x), e(r_i)) \notin \mathsf{MKTP}^A$; if $x \in B$, then $f(g(x), e(r_i)) \in \mathsf{MKTP}^A$ for all i. This establishes that the reduction is correct. ◀

Theorem 11 deals with the oracle problem MKTP^A, but the most interesting case is the case where $A = \emptyset$. The hypothesis is false when $A = \mathsf{QBF}$, since the KT^A measure is essentially the same as the KS measure studied in [3], where it is shown that $\mathsf{PSPACE} = \mathsf{ZPP}^{R_{\mathrm{KS}}}$, and thus has polynomial-size $\mathsf{MKTP}^{\mathsf{QBF}}$-circuits. Strikingly, not only is the hypothesis false in this case – but the conclusion is false as well. (See Corollary 13.)

For certain oracles (and we discuss below how broad this class of oracles is), the existence of uniform reductions is *equivalent to* certain circuit lower bounds.

▶ **Theorem 12.** *Let $\mathsf{MKTP}^A \in \mathsf{P}^A/\mathsf{poly}$. Then the following are equivalent:*

- PARITY *reduces to* MKTP^A *via a natural logspace-uniform* $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$*-reduction.*
- *For some* $\epsilon > 0$, $\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}^A(2^{\epsilon n})$.
- *For some* $\epsilon > 0$, $\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$.
- DET *reduces to* MKTP^A *via a natural logspace-uniform* $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$*-reduction.*

Furthermore, if PARITY *reduces to* MCSP^A *via a natural logspace-uniform* $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$*-reduction, then all of the above hold.*

Proof. First, we show that the first condition implies the second.

Let $\{C_n : n \in \mathbb{N}\}$ be a logspace-uniform family of oracle circuits computing PARITY, consisting of AC^0 circuitry feeding into oracle gates, which in turn are connected to an AND gate as the output gate. Let the oracle gates in C_n be $g_1, g_2, \ldots, g_{n^c}$. On any input string x, let the value fed into gate g_i on input x be $(q_i(x), \theta)$, and recall that, since the reduction is natural, the threshold θ depends only on n, and thus it is a constant in C_n.

Now, we appeal to [7, Claim 3.11], and conclude that each $\mathsf{MKTP}^{\mathsf{QBF}}$ oracle gate can be replaced by a DNF formula of size at most $n^{O(1)}2^{O(\theta^2 \log \theta)}$. Inserting these DNF formulae into C_n (in place of each oracle gate) results in a circuit of size $n^{O(1)}2^{O(\theta^2 \log \theta)}$ computing PARITY. Let the depth of this circuit be some constant d. It follows from [12] that $n^{O(1)}2^{O(\theta^2 \log \theta)} \geq 2^{\Omega(n^{1/(d-1)})}$, and hence that $\theta \geq n^{1/4d}$.

Note that all of the oracle gates g_i must output 1 on input $0^{n-1}1$, and one of the oracle gates g_{i_0} must output 0 on input 0^n. Thus we have $\mathrm{KT}^A(q_{i_0}(0^n)) \geq \theta \geq n^{1/4d}$. It follows from [3, Theorem 11] that the function with truth-table $q_{i_0}(0^n)$ has no circuit (with oracle gates for A) of size less than $(\mathrm{KT}^A(q_{i_0}(0^n)))^{1/3} \geq \theta^{1/3} \geq n^{1/12d}$.

Note that, in order to compute the j-th bit of some query $q_i(0^n)$, it suffices to evaluate a logspace-uniform AC^0 circuit where all of the input bits are 0. Since this computation can be done in logspace on input $(0^n1^i0^j)$, note that the language $H = \{(n, i, j) : \text{the } j\text{-th bit of query } q_i(0^n) \text{ is } 1\}$ is in linear space. Let $m = |(n, i, j)|$, and let $s(m)$ be the size of the smallest circuit D_m computing H for inputs of length m. Hardwire the bits for n and also set the bits for i to i_0. The resulting circuit on $|j| < m$ bits computes the function given by $q_{i_0}(0^n)$, and it was observed above that this circuit has size at least $n^{1/12d} \geq 2^{m/12d}$.

This establishes the first implication. (Note also that a similar argument yields the same conclusion from the assumption that PARITY reduces to MCSP^A via a natural logspace-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$-reduction.)

The assumption that $\mathsf{MKTP}^A \in \mathsf{P}^A/\mathsf{poly}$ suffices to show that the second condition implies the third. More formally, we'll consider the contrapositive. Assume that $\mathsf{DSPACE}(n) \subseteq \text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$ for every $\epsilon > 0$. An oracle gate for MKTP^A on inputs of size m can be replaced by a circuit (with oracle gates for A) of size m^c for some constant c. Carrying out this substitution in a circuit (with oracle gates for MKTP^A) of size $2^{\epsilon n}$ yields a circuit of size at most $2^{\epsilon n} + 2^{\epsilon n}(2^{\epsilon n})^c$.

Let $\delta > 0$. Then we can pick ϵ so that $2^{\epsilon n} + 2^{\epsilon n}(2^{\epsilon n})^c < 2^{\delta n}$, thereby establishing that $\mathsf{DSPACE}(n) \subseteq \text{io-}\mathsf{SIZE}^A(2^{\delta n})$ for every $\delta > 0$. This establishes the second implication.

The 3rd condition implies the 4th by Theorem 11. The 4th obviously implies the 1st. ◀

To the best of our knowledge, this is the first theorem that has given conditions where the existence of a reduction to MCSP^A implies the existence of a reduction to MKTP^A. We know of no instance where the implication goes in the opposite direction.

At this point, we should consider the class of oracles for which Theorem 12 applies. That is, what is the set of oracles A for which $\mathsf{MKTP}^A \in \mathsf{P}^A/\mathsf{poly}$? First, we observe that this condition holds for any PSPACE-complete set, which yields the following corollary:

▶ **Corollary 13.** PARITY *does not reduce to either* $\mathsf{MKTP}^{\mathsf{QBF}}$ *or* $\mathsf{MCSP}^{\mathsf{QBF}}$ *via a natural logspace-uniform* $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$*-reduction.*

Another example is $A = \{(M, x, 1^m) : M \text{ is an alternating Turing machine that accepts } x\text{, and runs in time at most } m \text{ and makes at most } \log m \text{ alternations}\}$. A is complete for the class $\mathsf{ATIME\text{-}ALT}(n^{O(1)}, O(\log n))$ under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions. Note that $\mathsf{MKTP}^A \in \mathsf{ATIME\text{-}ALT}(n^{O(1)}, O(\log n))$, and thus $\mathsf{MKTP}^A \in \mathsf{P}^A$. (Other examples can easily be created in this way, using an even smaller number of alternations. Note that, for this oracle A, it seems plausible that all four conditions in Theorem 12 hold.

Nonetheless, we grant that this seems to be a strong condition to place upon the oracle A – and it has even stronger consequences than are listed in Theorem 12. For instance, note that the proof that the first condition in Theorem 12 implies the second relies only on the fact that PARITY requires large AC^0 circuits. Thus, an identical proof shows that these four conditions are also equivalent to the condition that PARITY is reducible to MKTP^A via a natural ctt-reduction where the queries are computed by logspace-uniform $\mathsf{AC}^0[7]$ circuits. (One can substitute any other problem and class of mod circuits, where an exponential lower bound follows from [24, 26].) In fact, as in [7, Lemma 3.10] we can apply random restrictions in a logspace-uniform way (as described in [1]) and obtain a reduction from PARITY to MKTP^A where the queries are computed by logspace-uniform NC^0 circuits! That is, for example, $\mathsf{MAJORITY}$ is reducible to MKTP^A via reductions of this sort computed by logspace-uniform $\mathsf{AC}^0[3]$ circuits iff PARITY is reducible to the same set via reductions where the queries are computed by logspace-uniform NC^0 circuits. We find these implications to be surprising. The "gap" phenomenon that was described in [2] (showing that completeness under one class of reductions is equivalent to completeness under a more restrictive class of reductions) had not previously been observed to apply to $\mathsf{AC}^0[p]$ reducibility.

We want to highlight some contrasts between Theorem 11 and Corollary 13. $\mathsf{MKTP}^{\mathsf{QBF}}$ is hard for PSPACE under ZPP-Turing reductions [3], whereas MKTP is in NP. Thus $\mathsf{MKTP}^{\mathsf{QBF}}$ appears to be much harder than MKTP. Yet, under a plausible hypothesis, MKTP is hard for a well-studied subclass of P under a type of reducibility, where the "harder" problem $\mathsf{MKTP}^{\mathsf{QBF}}$ cannot even be used as an oracle for PARITY under this same reducibility.

In other words, the (conditional) natural logspace-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$ reductions from problems in DET to MKTP given in Theorem 11 are not "oracle independent" in the sense of [14]. Prior to this work, there had been no reduction to MCSP or MKTP that did not work for every MCSP^A or MKTP^A, respectively.

Prior to this work, it appears that there was no evidence for any variant of MCSP or MKTP being hard for a reasonable complexity class under $\leq_{\mathrm{T}}^{\mathsf{L}}$ reductions. All prior reductions (such as those in [4, 3, 5]) had been probabilistic and/or non-uniform, or (even under derandomization hypotheses) seemed difficult to implement in NC. We had viewed the results of [7] as providing evidence that none of these variants would be hard for P under, say, logspace reducibility. Now, we are no longer sure what to expect.

5 Conclusions and Open Questions

Conclusions. At a high level, we have advanced our understanding about MCSP and MKTP in the following two respects:

1. On one hand, under a very weak cryptographic assumption, the problem of approximating MCSP or MKTP is indeed NP-intermediate under *general* types of reductions when the approximation factor is quite *huge*. This complements the work of [19] for very *restricted* reductions.
2. On the other hand, if the gap is *small*, MKTP is DET-hard under nonuniform NC^0 reductions (contrary to previous expectations). This suggests that nonuniform reductions are crucial to understanding hardness of MCSP. While there are many results showing that NP-hardness of MCSP under *uniform* reductions is as difficult as proving circuit lower bounds, can one show that MCSP is NP-hard under $\mathsf{P/poly}$ reductions (without proving circuit lower bounds)?

Open Questions. It should be possible to prove unconditionally that MCSP is not in $\mathsf{AC}^0[2]$; we conjecture that the hardness results we give for MKTP hold also for MCSP.

We suspect that it should be possible to prove more general results of the form "If MCSP^A is hard for class $\mathcal{C}$, then so is MKTP^A". We view Theorem 12 to be just a first step in this direction. One way to prove such a result would be to show that MCSP^A reduces to MKTP^A, but (with a few exceptions such as $A = \mathsf{QBF}$) no such reduction is known. Of course, the case $A = \emptyset$ is the most interesting case.

Is MKTP hard for P? Or for some class between DET and P? Is it more than a coincidence that DET arises both in this investigation of MKTP and in the work of [20] on MCSP?

Is there evidence that Gap_ϵMCSP has intermediate complexity when ϵ is a fixed constant, similar to the evidence that we present for the case when $\epsilon(n) = o(1)$?

Acknowledgments. We thank Ryan Williams, Rahul Santhanam, Salil Vadhan, Marina Knittel, and Prashant Nalini Vasudevan for helpful discussions.

References

1 Manindra Agrawal. The isomorphism conjecture for constant depth reductions. *Journal of Computer and System Sciences*, 77(1):3–13, 2011. doi:10.1145/28395.28404.

2 Manindra Agrawal, Eric Allender, and Steven Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *Journal of Computer and System Sciences*, 57(2):127–143, 1998. doi:10.1006/jcss.1998.1583.

3 Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006. doi:10.1137/050628994.

4 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Information and Computation*, 2017. to appear. doi:10.1016/j.ic.2017.04.004.

5 Eric Allender, Joshua Grochow, and Cristopher Moore. Graph isomorphism and circuit size. Technical Report TR15-162, Electronic Colloquium on Computational Complexity, 2015. URL: https://eccc.weizmann.ac.il/report/2015/162/.

6 Eric Allender and Shuichi Hirahara. New insights on the (non)-hardness of circuit minimization and related problems. Technical Report TR17-073, Electronic Colloquium on Computational Complexity, 2017. URL: https://eccc.weizmann.ac.il/report/2017/073/.

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

8 Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57:14:1–14:36, 2010. doi:10.1145/1706591.1706594.

9 Eric Allender, Michal Koucký, Detlef Ronneburger, and Sambuddha Roy. The pervasive reach of resource-bounded Kolmogorov complexity in computational complexity theory. *Journal of Computer and System Sciences*, 77:14–40, 2010. doi:10.1016/j.jcss.2010.06.004.

10 Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On beating the hybrid argument. *Theory of Computing*, 9:809–843, 2013. doi:10.4086/toc.2013.v009a026.

11 Oded Goldreich. On promise problems: A survey. In Oded Goldreich, Arnold L. Rosenberg, and Alan L. Selman, editors, *Theoretical Computer Science, Essays in Memory of Shimon Even*, volume 3895 of *Lecture Notes in Computer Science*, pages 254–290. Springer, 2006. doi:10.1007/11685654_12.

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

13 Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of mcsp and its variants. In *32nd Conference on Computational Complexity, CCC*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. to appear.

14 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *31st Conference on Computational Complexity, CCC*, LIPIcs, pages 18:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.18.

15 Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC'97*, pages 220–229, 1997. doi:10.1145/258533.258590.

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

17 Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002. doi:10.1137/S0097539700389652.

18 Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975. doi:10.1145/321864.321877.

19 Cody Murray and Ryan Williams. On the (non) NP-hardness of computing circuit complexity. In *30th Conference on Computational Complexity, CCC*, volume 33 of *LIPIcs*, pages 365–380. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.CCC.2015.365.

20 Igor Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds and pseudorandomness. In *32nd Conference on Computational Complexity, CCC*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. to appear.

21 Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *IEEE Conference on Structure in Complexity Theory*, pages 133–138. IEEE Computer Society, 1991. doi:10.1109/SCT.1991.160253.

22 Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *Second Israel Symposium on Theory of Computing Systems (ISTCS)*, pages 3–17. IEEE Computer Society, 1993. doi:10.1109/ISTCS.1993.253489.

23 Alexander Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:24–35, 1997. doi:10.1006/jcss.1997.1494.

24 Alexander A. Razborov. Lower bounds on the size of bounded depth networks over a complete basis with logical addition. *Matematicheskie Zametki*, 41:598–607, 1987. In Russian. English translation in *Mathematical Notes of the Academy of Sciences of the USSR* 41:333–338, 1987.

25 Michael Rudow. Discrete logarithm and minimum circuit size. Technical Report TR16-23, Electronic Colloquium on Computational Complexity, 2016. URL: https://eccc.weizmann.ac.il/report/2016/108/.

26 Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings 19th Symposium on Theory of Computing*, pages 77–8. ACM Press, 1987. doi:10.1145/28395.28404.

27 Jacobo Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004. doi:10.1137/S009753970241096X.

28 Salil P. Vadhan. An unconditional study of computational zero knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006. doi:10.1137/S0097539705447207.

29 Heribert Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer-Verlag New York Inc., 1999. doi:10.1007/978-3-662-03927-4.