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

ERIC ALLENDER, Rutgers University, USA
SHUICHI HIRAHARA, National Institute of Informatics, JAPAN

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. Our technique is quite general; we use it also to show that approximating the size of the largest clique in a graph within a factor of $n^{1-o(1)}$ is also NP-intermediate unless $\mathsf{NP} \subseteq \mathsf{P/poly}$.

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 $\leq_m^{\mathsf{NC}^0}$. 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 significant class of sets A.
- Hardness of MCSP^A implies hardness of MKTP^A for a significant class of sets A. This is the first result directly relating the complexity of MCSP^A and MKTP^A, for any A.

CCS Concepts: • **Theory of computation → Complexity classes**; **Circuit complexity**.

Additional Key Words and Phrases: computational complexity, Kolmogorov complexity, Circuit size, MCSP

ACM Reference Format:
Eric Allender and Shuichi Hirahara. 2019. New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems. *ACM Trans. Comput. Theory* 1, 1 (July 2019), 26 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

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 [RR97], and because it is a prominent candidate for NP-intermediate status. It has been known since the work of Ladner [Lad75] that NP-intermediate problems exist if $\mathsf{P} \neq \mathsf{NP}$, 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

Authors' addresses: Eric Allender, Rutgers University, Computer Science, 110 Frelinghuysen Rd., Hill Center, Piscataway, NJ, 08854, USA, allender@cs.rutgers.edu; Shuichi Hirahara, National Institute of Informatics, Principle of Informatics Research Division, Tokyo, JAPAN, s_hirahara@nii.ac.jp.

1942-3454/2019/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

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.

MCSP is hard for SZK [AD17] under BPP reductions, but the situation is quite different when more restricted notions of reducibility are considered. Recent results [AHK17, HW16, MW17] have suggested that MCSP might not even be hard for P under logspace or AC^0 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 $s \in \mathbb{N}$; $(T, s) \in$ 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 [ABK$^+$06, AD17, AHK17, AKRR10, HW16, HP15, IKV18, KC00, MW17, Rud17]) applies equally well to any of these variants. MCSP is also closely related to a type of time-bounded Kolmogorov complexity known as KT, which was defined in [ABK$^+$06]. 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 [AGvM$^+$18, HS17]. For instance, it was shown in [AGvM$^+$18] that the Graph Isomorphism problem and several other problems with an algebraic flavor are contained in $\mathsf{ZPP}^{\mathsf{MKTP}}$, and it is shown in [HS17] that the "random 3SAT problem" reduces to MKTP. It is not known if these statements are true for MCSP.

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. [KC00, MW17, AHK17, HW16])) 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 $\mathsf{DSPACE}(n) \not\subseteq \mathsf{io\text{-}SIZE}(2^{\epsilon n})$ [MW17, AHK17].

Murray and Williams [MW17] asked if, in general, circuit lower bounds imply hardness of the circuit minimization problems. We answer their questions affirmatively in certain settings: An oracle circuit 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 4.5).

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, OR, and NOT 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 a significant class of 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. (See Theorem 4.7, and the Remark after Corollary 4.10.)

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 [MW17] 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 [Agr11] 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 4.3). 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 [ABK+06] that neither of these problems is in AC^0. Under a plausible derandomization hypothesis, this non-uniform reduction can be converted into a Dlogtime-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.

Our DET-hardness result is proved by building on a randomized reduction of [AGvM+18] reducing Graph Isomorphism to MKTP. We modify that construction, to obtain a nonuniform AC^0 reduction (Corollary 4.2). The restricted version of Graph Isomorphism that we use is known to be hard for DET [Tor04]. Our proof of Theorem 4.3 then appeals to the "Gap Theorem" of [AAR98], in order to conclude that $\mathsf{DET} \leq_{\mathrm{m}}^{\mathsf{NC}^0} \mathsf{MKTP}$; the Gap Theorem states that, for any class C closed under TC^0 reductions, C-hardness under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions implies C-hardness under $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reductions.

Somewhat remarkably, Oliveira and Santhanam [OS17] 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 [Tor04]. Their results have the advantage that they apply to MCSP rather than merely to MKTP, but because their reduction is more complex (TC^0, as contrasted with AC^0), they do not obtain unconditional lower bounds, as in Corollary 4.4.

Our hardness results (both unconditional hardness results under nonuniform reductions, and conditional uniform hardness results) are summarized in Table 1.

Table 1. Hardness for MKTP^A: MKTP^A is hard for DET under the type of reducibility listed in the first column, if oracle A satisfies the condition listed in the second column. The last column shows the theorem where the result is stated in the paper.

reductions $\mathcal{R}$	condition on A	Theorem
nonuniform $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$	every A	Theorem 4.3
P-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$	$\mathsf{E} \not\subseteq \text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$	Corollary 4.9
L-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$	$\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$	Theorem 4.5
Dlogtime-uniform $\leq_{\mathrm{ctt}}^{\mathsf{AC}^0}$	$\Sigma_d\mathsf{TIME}(n)$ hard on average for $\text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$	Theorem 4.6

Implications among hardness conditions for MKTP and MCSP. No $\leq_{\mathrm{T}}^{\mathrm{P}}$ 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

[1] Subsequent to our work, a stronger average-case lower bound against $\mathsf{AC}^0[p]$ was proved [HS17]. The techniques of [HS17] do not show how to reduce DET, or even smaller classes such as TC^0, to MKTP. Thus our work is incomparable to [HS17].

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 4.7).

A reduction that is not "oracle independent". Hirahara and Watanabe [HW16] observed that all of the then-known reductions to MCSP and MKTP were "oracle-independent", in the sense that, for any class C and reducibility $\leq_r$, all proofs that MCSP (or MKTP) is hard for C under $\leq_r$ also show that MCSP^A (MKTP^A) is also hard for C, for *every* A. In addition, they showed an inherent limitation of oracle-independent proofs: 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 Dlogtime-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 4.10).

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., through to $2^{\sqrt{\log n}}$ alternations, and so on, until finally $A = \mathsf{QBF}$. Since $\mathsf{DSPACE}(n) \subseteq \mathsf{P}^{\mathsf{QBF}}/\mathrm{poly}$, 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 $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathsf{MKTP}^A}$ [AD17], until at some point $\mathsf{BPP}^{\mathsf{MKTP}^A}$ expands to coincide with PSPACE [ABK+06]. Also, at all stages in this evolution $\mathsf{DET} \leq_{\mathrm{m}}^{\mathsf{NC}^0}$-reduces to MKTP^A. No larger class is known to $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$-reduce to MKTP^A; 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 4.7.)

This helps place the results of [AHK17] in the proper context. In [AHK17] strong evidence was presented against $\mathsf{MCSP}^{\mathsf{QBF}}$ being hard for 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 4.5) and *not* reduce to MKTP^A for a more powerful oracle A (Corollary 4.10).

Some of these results are summarized in the following tables. Table 2 indicates what is known to follow for the (unrelativized) MCSP and MKTP problems.

In the "intermediate" case, where the oracle A is a complete set for PP, Table 3 shows both hardness and non-hardness consequences of some plausible hypotheses.

When the oracle A is the PSPACE-complete set QBF, the situation is summarized in Table 4.

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

Table 2. Consequences of hardness for MCSP and MKTP: If MCSP or MKTP is $\mathcal{C}$-hard under $\mathcal{R}$, then condition $\mathcal{S}$ holds. The last column shows where the result is found.

class $\mathcal{C}$	reductions $\mathcal{R}$	statement $\mathcal{S}$	Reference
TC^0	Dlogtime-uniform $\leq_m^{\mathsf{AC}^0}$	NP $\not\subseteq$ P/poly and LTH $\not\subseteq$ io-SIZE$[2^{\epsilon n}]$	[AHK17]
PARITY	L-uniform $\leq_{ctt}^{\mathsf{AC}^0}$	MKTP $\notin$ P/poly or DSPACE$(n) \not\subseteq$ io-SIZE$[2^{\epsilon n}]$	4.7
TC^0	Dlogtime-uniform $\leq_T^{\mathsf{AC}^0}$	NP $\not\subseteq$ P/poly or CH = PH (hence NP $\neq \mathsf{TC}^0$)	4.15
NC^1	Dlogtime-uniform $\leq_T^{\mathsf{AC}^0}$	NP $\not\subseteq$ P/poly or PSPACE = PH (hence NP $\neq$ NC)	4.15
NP	Dlogtime-uniform $\leq_T^{\mathsf{AC}^0}$	NP $\not\subseteq$ P/poly or NEXP = MA (hence NP $\neq$ MA $\cap$ P/poly)	4.16
NP	L-uniform $\leq_T^{\mathsf{AC}^0}$	NP $\not\subseteq$ P/poly or NEXP = PSPACE	4.19
NP	P-uniform $\leq_T^{\mathsf{AC}^0}$	NP $\not\subseteq$ P/poly or NEXP = EXP	4.19

Table 3. Hardness and non-hardness for $\mathsf{MKTP}^{\mathsf{PP}}$ and $\mathsf{MCSP}^{\mathsf{PP}}$: If condition $\mathcal{S}$ holds, then the problems in the first column are hard (or not-hard, as indicated) for the class $\mathcal{C}$ under $\mathcal{R}$. Note that, in going from the first line to the second, $\mathcal{C}$ becomes larger, and (except for the uniformity notion) the class of reductions becomes more restrictive. This highlights the importance of the uniformity condition, in determining if these problems are hard for a given class. The last column shows the theorem where the result is stated in the paper.

Problem	Hard?	class $\mathcal{C}$	reductions $\mathcal{R}$	statement $\mathcal{S}$	Theorem
$\mathsf{MKTP}^{\mathsf{PP}}$ & $\mathsf{MCSP}^{\mathsf{PP}}$	Not hard	NC^1	Dlogtime-uniform $\leq_{tt}^{\mathsf{AC}^0}$	PSPACE $\neq \mathsf{PH}^{\mathsf{PP}}$	4.12
$\mathsf{MKTP}^{\mathsf{PP}}$	Is hard	DET	L-uniform $\leq_{ctt}^{\mathsf{AC}^0}$	DSPACE$(n) \not\subseteq$ io-SIZE$^{\mathsf{MKTP}^{\mathsf{PP}}}(2^{\epsilon n})$	4.5

Table 4. Non-hardness for $\mathsf{MKTP}^{\mathsf{QBF}}$ and $\mathsf{MCSP}^{\mathsf{QBF}}$: Neither $\mathsf{MKTP}^{\mathsf{QBF}}$ nor $\mathsf{MCSP}^{\mathsf{QBF}}$ is $\mathcal{C}$-hard under $\mathcal{R}$, assuming that the condition $\mathcal{S}$ holds. The last column shows the theorem where the result is stated in the paper.

class $\mathcal{C}$	reductions $\mathcal{R}$	condition $\mathcal{S}$	Theorem/Corollary
PARITY	L-uniform natural $\leq_{ctt}^{\mathsf{AC}^0}$	True	4.10
PSPACE	$\leq_T^{\mathsf{L}}$	True	4.11
NP	$\leq_T^{\mathsf{L}}$	PSPACE $\neq$ NEXP	4.11
NP	L-uniform $\leq_T^{\mathsf{AC}^0}$	PSPACE $\neq$ NEXP	4.18
NP	P-uniform $\leq_T^{\mathsf{AC}^0}$	EXP $\neq$ NEXP	4.18
P	$\leq_T^{\mathsf{L}}$	PSPACE $\neq$ EXP	4.11
DET	nonuniform $\leq_m^{\mathsf{NC}^0}$	False	4.3

$\epsilon > 0$, 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 [ABK+06, AKRR10, AD17, Rud17] 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 [AHK17] had presented unconditional results, showing that "gap" versions of MCSP could not be hard for TC^0 under $\leq_m^{\mathsf{AC}^0}$ reductions, unless those reductions had large "stretch" (mapping short inputs to long outputs). In Section 3.2, we show that BPP-Turing reductions among gap MCSP problems require large stretch, unless MCSP $\in$ BPP.

Natural NP-intermediate Problems. Using very similar techniques, 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 [RR97], if one is trying to prove circuit lower bounds of size $2^{o(n)}$. Our Theorem 3.6 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 [AK10]); Our contribution is to show that MCSP also has such a property (under the assumption that the gap MCSP problems are NP-hard). In fact, we also present a similar argument showing that a $n^{1-o(1)}$ approximation for CLIQUE is NP-intermediate if NP $\not\subseteq$ P/poly.

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: ATIME($t(n)$), or by simultaneously bounding time and the number of alternations between existential and universal configurations: ATIME-ALT($t(n), a(n)$).

We refer the reader to the text by Vollmer [Vol99] 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 $\mathsf{L} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PH} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXP} \subseteq \mathsf{NEXP}$. E denotes $\mathsf{DTIME}(2^{O(n)})$; E contains many of the standard complete sets for EXP. We shall also have occasion to refer to the *counting hierarchy*, CH [Tor91], which consists of the classes PP, $\mathsf{PP}^{\mathsf{PP}}$, $\mathsf{PP}^{\mathsf{PP}^{\mathsf{PP}}}$, etc.

This brings us to the topic of reducibility. Let C be either a class of functions or a class of circuits. We say that $A \leq_{\mathrm{m}}^{C} B$ if there is a function $f \in C$ (or f computed by a circuit family in 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, 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 $A \leq_{\mathrm{T}}^{C} B$ if there is a oracle machine in C (or a family of oracle circuits in C) accepting A, when given oracle B. We make use of $\leq_{\mathrm{T}}^{\mathsf{P/poly}}$, $\leq_{\mathrm{T}}^{\mathsf{BPP}}$, $\leq_{\mathrm{T}}^{\mathsf{P}}$, $\leq_{\mathrm{T}}^{\mathsf{L}}$, $\leq_{\mathrm{T}}^{\mathsf{NC}^1}$ 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 $A \in \mathsf{P}^B/\mathrm{poly}$ 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.

Now that circuit-based notions of reducibility have been introduced, we can present the definition of one more class that plays a large role in this work: DET is the class of problems that are reducible to the problem Det of computing the determinant of integer matrices, by NC^1-Turing reductions. DET lies between L and P. Although $\mathsf{AC}^0 \subsetneq \mathsf{NC}^1 \subseteq \mathsf{L}$, the reader should be aware that $\mathsf{L}^{\mathsf{Det}}$ is contained in the class of problems $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reducible to Det, which in turn is contained in DET.

(See [AO96, All04] for more details.) In fact, since $\leq_T^L$ reductions are nonadaptive [LL76], in many situations $\leq_T^L$ reductions can be simulated by $\leq_T^{AC^0}$ reductions.

Kabanets and Cai [KC00] 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 $\leq_m^P$ 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.

The notation io-SIZE($s(n)$) denotes the class of all languages A such that, for infinitely many lengths n, there is a circuit of size at most $s(n)$ accepting exactly the strings of length n in A. (Although this definition can depend upon the precise notion of "circuit size" being considered, every statement that we make using this notation holds using any reasonable notion of "size".) If B is a language, then io-SIZE$^B(s(n))$ denotes the class of all languages A such that, for infinitely many lengths n, there is an "oracle circuit" (that is, a circuit that has "oracle gates" in addition to the standard Boolean gates) of size at most $s(n)$ accepting exactly the strings of length n in A, when given B as an oracle.

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 [Vol99], 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 AC^0, $AC^0[p]$ 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 [Vol99]; there is wide agreement that the "correct" definition coincides with ATIME($O(\log n)$).

There are many ways to define time-bounded Kolmogorov complexity. The definition KT(x) was proposed in [ABK+06], 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).

Definition 2.1. Let U be a Turing machine and let A be an oracle. The measure $KT_U^A(x)$ is defined to be

$$KT_U^A(x) = \min\{\, |d| + t \quad : \quad \forall b \in \{0,1,*\}\ \forall i \leq |x|+1, \text{ the machine } U^{A,d}(i,b) \text{ accepts in } t \text{ steps iff } x_i = b \,\}.$$

We omit the superscript A if $A = \emptyset$

It is observed in [ABK+06] that, for any two universal machines U and U', there is a constant c such that $KT_U^A(x) \leq c \cdot KT_{U'}^A(x) \log KT_{U'}^A(x)$. We pick one such universal machine U and define KT(x) to be $KT_U(x)$. The definition of KT was designed in such a way as to make it useful in proving theorems about MCSP. The fact that the machine U is allowed to have random access to the bits of the "description" d has the by-product that certain algorithms are easier to implement using the machines of the KT formalism than using the hardware formalisms of circuit complexity. This helps explain why some of the theorems that we prove for MKTP are not currently known to hold for MCSP.

For completeness, we include a definition of MCSP^A (although this already appears informally in the introduction).

Definition 2.2. Let A be any oracle. MCSP^A is the set of pairs (T, s), such that T is a bit string of length 2^m representing the truth table of an m-variate Boolean function, and $s \in \mathbb{N}$, where there is an oracle circuit computing the function represented by T, which has at most s gates. (The gates that are allowed are bounded fan-in AND, OR, and NOT gates, along with unbounded-fan-in oracle gates, although our theorems also hold using other reasonable modifications to the circuit model.) We use the notation MCSP for $\mathsf{MCSP}^{\emptyset}$.

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 3.1. For any function $\epsilon : \mathbb{N} \to (0, 1)$, let $\mathrm{Gap}_\epsilon\mathsf{MCSP}$ 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 [Gol06] for similar comments.)

Proposition 3.2. $\mathrm{Gap}_\epsilon\mathsf{MCSP}$ *is polynomial-time Turing equivalent to the following promise problem* (Y, N)*:*

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

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

Proof. Given a solution A of the promise problem (Y, N), one can compute an approximation $f(T)$ of $\mathrm{CC}(T)$ as follows:

$$f(T) := \max\{\, s \in \mathbb{N} \mid (T, s) \notin A \,\}$$

We claim that $f(T)$ satisfies the approximation guarantee as given in Definition 3.1. By the definition of $f(T)$, we have $(T, f(T)) \notin A$, which implies that $(T, f(T)) \notin Y$, and thus $\mathrm{CC}(T) \geq f(T)/|T|^{1-\epsilon(|T|)}$. Similarly, by the definition of $f(T)$, we have $(T, f(T) + 1) \in A$, which implies that $(T, f(T) + 1) \notin N$, and thus $\mathrm{CC}(T) \leq f(T)$. To summarize, we have $\mathrm{CC}(T) \leq f(T) \leq |T|^{1-\epsilon(|T|)} \cdot \mathrm{CC}(T)$ and thus $f(T)$ satisfies Definition 3.1.

On the other hand, suppose that an approximation $f(T)$ of $\mathrm{CC}(T)$ is given. We can define a solution A of the promise problem (Y, N) as $A := \{\, (T, s) \mid f(T) < s \,\}$. We claim that A indeed is a solution of (Y, N). If $(T, s) \in Y$, then $f(T) \leq |T|^{1-\epsilon(|T|)} \cdot \mathrm{CC}(T) < s$ and therefore $(T, s) \in A$. On the other hand, if $(T, s) \in N$, then $f(T) \geq \mathrm{CC}(T) \geq s$, which implies $(T, s) \notin A$. □

Note that $\mathrm{Gap}_\epsilon\mathsf{MCSP}$ becomes easier when ϵ becomes smaller. If $\epsilon(n) = o(1)$, then (using the promise problem formulation) it is easy to see that $\mathrm{Gap}_\epsilon\mathsf{MCSP}$ 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 $\epsilon(n) = o(1)$, $\mathrm{Gap}_\epsilon\mathsf{MCSP}$ 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 [RR97]. Thus it is of interest that MCSP cannot reduce to Gap_ϵMCSP in this regime under very general notions of reducibility, unless MCSP itself is easy.

Theorem 3.3. *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 the Gap_ϵMCSP is "strongly downward self-reducible." We will show that any Gap_ϵMCSP 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 3.4. *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. We first claim that $\max_{x\in\{0,1\}^k} \mathrm{CC}(f_x) \le \mathrm{CC}(f)$. Indeed, let C be a minimum circuit that computes f and let x be an arbitrary string of length k. For each $x \in \{0,1\}^k$, define a circuit C_x as $C_x(y) := C(x,y)$ on input $y \in \{0,1\}^{n-k}$. Then, since C_x computes f_x and the size of C_x is at most that of C, we have $\mathrm{CC}(f_x) \le \mathrm{CC}(f)$.

Next, we claim that $\mathrm{CC}(f) \le 2^k \cdot \left(\max_{x\in\{0,1\}^k} \mathrm{CC}(f_x) + O(1)\right)$. For any $x \in \{0,1\}^k$, let C_x be a minimum circuit that computes f_x. We build a circuit that computes $f =: f_\epsilon$ recursively as follows: $f_z(x,y) = (\neg x_1 \wedge f_{z0}(x_2,\dots,x_k,y)) \vee (x_1 \wedge f_{z1}(x_2,\dots,x_k,y))$ for any string z of length less than k, and $f_x(y) = C_x(y)$ for any $x \in \{0,1\}^k$. Since $\mathrm{CC}(f_z) \le \mathrm{CC}(f_{z0}) + \mathrm{CC}(f_{z1}) + 3$ we obtain

$$\begin{aligned} \mathrm{CC}(f) &\le \sum_{x\in\{0,1\}^k} C_x(y) + 3\cdot(2^k - 1) \\ &< 2^k \cdot \left(\max_{x\in\{0,1\}^k} \mathrm{CC}(f_x) + 3 \right). \end{aligned}$$

□

Proof of Theorem 3.3. Let M be an oracle BPP Turing machine which reduces MCSP to Gap_ϵMCSP. 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 Gap_ϵMCSP 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 Gap_ϵMCSP. 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 3.4, 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 Gap_ϵMCSP if $2^k \cdot (\mathrm{CC}(S)+3) \leq |S|^{1-\epsilon(|S|)} \cdot \mathrm{CC}(S)$. Since we may assume without loss of generality that $\mathrm{CC}(S) \geq 3$, it suffices to make sure that $2^{k+1} \cdot \mathrm{CC}(S) \leq |S|^{1-\epsilon(|S|)} \cdot \mathrm{CC}(S)$. Let $|S| = 2^m$. Then, in order to satisfy $k+1 \leq (1-\epsilon(|S|)) \cdot m$, 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 Gap_ϵMCSP, 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 $2^k \leq 2^{cn}$ 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 $|S_i| \leq 2^{1+(n/2)}$ holds for sufficiently large n. Let us take n to be large enough so that $\epsilon(2^{n/2}) \leq 1/2c$. If $m \geq n/2$, then $|S_i| \leq 2^{\epsilon(2^m)\cdot m+1} \leq 2^{\epsilon(2^{n/2})\cdot cn+1} \leq 2^{1+(n/2)}$. Otherwise, since $m \leq n/2$ and $\epsilon(|S|) < 1$, we obtain $|S_i| \leq 2^{\epsilon(|S|)\cdot m+1} \leq 2^{1+(n/2)}$. 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 3.3 still shows that if MCSP $\in \mathsf{P}^{\mathrm{Gap}_\epsilon\mathrm{MCSP}}/\mathrm{poly}$ then MCSP $\in \mathsf{P}/\mathrm{poly}$.

Corollary 3.5. *Let $\epsilon(n) = o(1)$. If* Gap_ϵMCSP *has no solution in* P/poly *then* Gap_ϵMCSP *is not hard for* NP *(or even for* MCSP*) under* $\leq_{\mathrm{T}}^{\mathsf{P}/\mathrm{poly}}$ *reductions, and is thus* NP*-intermediate.*

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

In what follows, we show that the assumption of Corollary 3.5 is true under very modest cryptographic assumptions. It is known that, for any constant $\epsilon > 0$, Gap_ϵMCSP is SZK-hard under $\leq_{\mathrm{T}}^{\mathsf{P}/\mathrm{poly}}$ reductions [AD17]. Here, we show that if SZK is not in P/poly, then for some $\epsilon(n) = o(1)$, Gap_ϵMCSP has no solution in P/poly. In fact, we can prove something *stronger*: If auxiliary-input one-way functions exist, then Gap_ϵMCSP is not in P/poly. We now describe auxiliary-input one-way functions.

Most researchers consider the existence of cryptographically-secure one-way functions to be essential for meaningful cryptography [IL89]. 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[2] 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 P/poly only if auxiliary-input one-way functions exist.

[2]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 [Vad06].

(This is due to [OW93]; a good exposition can be found in [Vad06, Theorems 7.1 & 7.5], based on earlier work of [Ost91].)

Theorem 3.6. *If auxiliary-input one-way functions exist, then there is a function $\epsilon(n) = o(1)$ such that Gap_ϵMCSP is NP-intermediate. (Namely, Gap_ϵMCSP has no solution in P/poly and Gap_ϵMCSP is not NP-hard under $\leq_T^{\mathrm{P/poly}}$ reductions.)*

Remark: In particular, either one of the following implies that some Gap_ϵMCSP 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 P/poly.

Proof. Let $F(y, x)$ define an auxiliary-input one-way family of functions $f_y(x)$ where $|x| = p(|y|)$ for some polynomial p. Let $S(n)$ be the size of the smallest circuit A such that for some y of length n, $\Pr_x[f_y(A(f_y(x))) = f_y(x)] \geq 1/S(n)$ where $n = |y|$ and x is chosen uniformly at random from $\{0,1\}^{p(n)}$. By assumption $S(n)$ is not bounded by any polynomial. Let $e(n)$ be a nondecreasing unbounded function such that $n^{c_0 e(n^{c_0})} < S(n)$ for infinitely many n, where c_0 is a constant that we will pick later.

At this point, we make use of some standard derandomization tools, including the HILL pseudorandom generator [HILL99], and pseudorandom function generators [GGM86, RR97]. First, we recall the HILL construction, phrased in terms of non-uniform adversaries:

Theorem 3.7 (see [HILL99]). *Let $F(y, x)$ be computable uniformly in time polynomial in $|y|$, and let $\mu : \mathbb{N} \to [0, 1]$. For any oracle L and any oracle circuit M of size $s(n)$, there is a size $(s(n)^{O(1)}/\mu(n^{O(1)}))$ circuit N such that the following holds for any n and y: If*

$$\left| \Pr_{|r|=2n}[M^L(y, r) = 1] - \Pr_{|x|=n}[M^L(y, G_{f_y}^{\mathrm{HILL}}(x)) = 1] \right| \geq \mu(n),$$

then

$$\Pr_{|x|=n}[F(y, N^L(y, F(y, x))) = F(y, x)] \geq \mu(n^{O(1)})/n^{O(1)},$$

where r and x are chosen uniformly at random. Here $G_{f_y}^{\mathrm{HILL}}$ is a pseudorandom generator, where $G_{f_y}^{\mathrm{HILL}}(x)$ is computable in time polynomial in $|y|$, as described in [HILL99].

Theorem 3.7 states that if there exists a distinguisher with access to an oracle L that distinguishes the output of $G_{f_y}^{\mathrm{HILL}}$ from the uniform distribution, then oracle access to L suffices to invert f_y on a significant fraction of the inputs. We now argue that such a distinguisher can be computed by a circuit of size $n^{O(e(n))}$ with oracle gates for $\mathrm{Gap}_{1/e(n)}$MCSP, where $e(n)$ is the slow-growing function that we defined earlier.

Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a pseudorandom generator mapping strings of length n to strings of length $2n$, constructed from the generator $G_{f_y}^{\mathrm{HILL}}$. Furthermore, let $G_0(x)$ be the first n bits of $G(x)$, and let $G_1(x)$ be the second n bits of $G(x)$, so that $G(x) = G_0(x)G_1(x)$. We now make use of the pseudorandom function generator of Razborov and Rudich [RR97], with the following parameters.

For any string w of length k, let $G_w(x) : \{0,1\}^n \to \{0,1\}^n$ be defined as $G_w(x) = G_{w_1}(G_{w_2}(\ldots(G_{w_n}(x))\ldots))$. Define $G'_y(x, w)$ to be the first bit of $G_w(x)$, where here the subscript y refers back to the fact that G is defined from $G_{f_y}^{\mathrm{HILL}}$.

Now let $z(x)$ be the truth table of $G'_y(x, w)$ (viewed as a function of w). Since $|w| = k$, $|z(x)| = 2^k$. Since G'_y is computed in time polynomial in the length of x and w, $\mathrm{CC}(z(x)) < (n + k)^c$ for some constant c.

Now, let us choose k to be $(c+1)e(n)\log n$. It follows that

$$\mathrm{CC}(z(x)) < (n + (c+1)e(n)\log n)^c < n^{c+1} = (2^{(c+1)e(n)\log n})^{1/e(n)} = |z(x)|^{1/e(n)}.$$

That is, from the pseudorandom distribution, we always have $\mathrm{CC}(z(x)) < |z(x)|^{1/e(n)}$, whereas a random string has CC complexity at least roughly equal to the length of the string with high probability.

Thus, an oracle gate for any oracle L that satisfies the $\mathrm{Gap}_{1/e(n)}$MCSP promise problem can distinguish random functions from the output of the pseudorandom function generator. And [RR97] shows how to obtain a circuit M of size $n^{O(e(n))}$ such that

$$\left| \Pr_{|r|=2n}[M^L(y,r)=1] - \Pr_{|x|=n}[M^L(y, G^{\mathrm{HILL}}_{f_y}(x))=1] \right| \geq 1/n^{e(n)}.$$

By Theorem 3.7, there are constants c_1, c_2, c_3, c_4 and there is a circuit N of size $n^{c_1 e(n^{c_2})}$ such that the following holds for any n and y:

$$\Pr_{|x|=n}[F(y, N^L(y, F(y,x))) = F(y,x)] \geq n^{c_3 e(n^{c_4})}$$

where x is chosen uniformly at random.

Now if we pick c_0 to be greater than $\max\{c_1, c_2, c_3, c_4\}$, it follows that N is a circuit of size less than $S(n)$ that inverts $f_y(x)$ with probability greater than $1/S(n)$, contrary to the definition of S.

This establishes that no solution to the $\mathrm{Gap}_{1/e(n)}$MCSP promise problem lies in P/poly. By Corollary 3.5, we conclude that $\mathrm{Gap}_{1/e(n)}$MCSP is NP-intermediate. □

Remark: Observe that Theorem 3.6 can also be rephrased in terms of *uniform* probabilistic adversaries, if we assume that the one-way functions require time $n^{e(n)}$ to invert, for some easy-to-compute function e.

3.1 Other NP-intermediate problems.

Although our focus is primarily on MCSP, we observe here that the strongly downward self-reducibility property that we exploited above is fairly common. For instance, it has been noticed previously that CLIQUE also has this property [Sri03, AK10]. It appears to be a new observation, however, that this property yields a natural NP-intermediate optimization problem.

Definition 3.8. For any function $\epsilon : \mathbb{N} \to (0,1)$, let Gap_ϵCLIQUE be the approximation problem that, given an n-vertex graph G, asks for outputting a value $f(G) \in \mathbb{N}$ such that

$$\omega(G) \leq f(G) \leq n^{1-\epsilon(n)} \cdot \omega(G).$$

Here, as usual $\omega(G)$ denotes the clique number of G: the size of the largest clique in G.

Theorem 3.9. NP $\not\subseteq$ P/poly *if and only if there is an* $\epsilon(n) = o(1)$ *such that* Gap_ϵCLIQUE *has no solution in* P/poly *and is not hard for* NP *under* $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$ *reductions.*

Proof. Assume NP $\not\subseteq$ P/poly. Define $e(n)$ to be the least c such that, for all $m \leq n$, there is a circuit of size m^c that computes a function $f(G)$ (for m-vertex graphs G) such that $\omega(G) \leq f(G) \leq m^{1-1/c} \cdot \omega(G)$. If $e(n) = O(1)$, it follows from [Hås99] that CLIQUE $\in$ P/poly, contrary to assumption. Thus $e(n) = \omega(1)$.

Let $\epsilon = \epsilon(n) = 1/e(n)$; thus $\epsilon(n) = o(1)$. It follows immediately from the definition of $e(n)$ that Gap_ϵCLIQUE has no solution in P/poly.

If we partition the vertices of an n-node graph G into $n^{1-\epsilon}$ parts $V_1, \ldots, V_{n^{1-\epsilon}}$ of size at most $\lceil n^\epsilon \rceil$, then $\omega(G) \leq (n^{1-\epsilon}) \cdot \max_i \omega(G_i)$, where G_i is the induced subgraph of G with vertices in V_i. (See [Sri03, AK10] for other applications of this observation.)

Now, precisely as in the proof of Theorem 3.3, it follows that if CLIQUE were P/poly-Turing reducible to Gap_ϵCLIQUE, then CLIQUE $\in$ P/poly, contrary to our assumption. This shows that Gap_ϵCLIQUE is not NP-hard under P/poly reductions, and thus completes the "only if" direction of the Theorem. (The converse is trivial.) □

3.2 Reductions among GapMCSPs Require Large Stretch

In the previous section, we studied Gap_ϵMCSP where $\epsilon(n) = o(1)$. In this section, we focus on the case where ϵ is a fixed positive constant. (When Gap_ϵMCSP is considered as a "natural property", this is the range of ϵ that would be considered when trying to prove a circuit size lower bound of $2^{\Omega(n)}$. It is also the range of ϵ for which the hardness results of [AD17] hold.)

In what follows, we say that a reduction from Gap_δMCSP to Gap_ϵMCSP *has stretch* n^c if, on input T, the reduction makes queries of length at most $|T|^c$.

Theorem 3.10. *Let* $0 < \epsilon < \delta < 1$. *If* Gap_δMCSP *is reducible to* Gap_ϵMCSP *via a randomized Turing reduction of stretch at most* n^c *for some* $c < \delta/\epsilon$, *then* Gap_δMCSP $\in$ BPP.

Proof. The argument is almost identical to the argument in the preceding section. Given an input to Gap_δMCSP, simulate the reduction from Gap_δMCSP to Gap_ϵMCSP. As before, if the reduction makes a query S, then divide S into consecutive substrings $S_1, \ldots, S_{2^k}$ of length 2^{m-k}, where m is defined as $|S| = 2^m$ and k is a parameter chosen later depending on m. For each $i \in [2^k]$, recursively solve Gap_δMCSP on the instance S_i, and let $f(S_i)$ be the answer of the recursive call. Now, we estimate the circuit complexity of S as $2^k \cdot (\max_{i\in[2^k]} f(S_i) + 3)$ and continue the simulation.

We claim the correctness of the simulation for a certain choice of parameter $k = k(m)$. Let e denote the estimated circuit complexity of S, that is, $e := 2^k \cdot (\max_{i\in[2^k]} f(S_i) + 3)$. The goal is to show that e satisfies the promise of Gap_ϵMCSP, or equivalently,

$$\mathrm{CC}(S) \le e \le |S|^{1-\epsilon} \cdot \mathrm{CC}(S). \tag{1}$$

We may assume that answers of recursive calls satisfy the promise of Gap_δMCSP by induction: that is, $\mathrm{CC}(S_i) \le f(S_i) \le |S_i|^{1-\delta} \cdot \mathrm{CC}(S_i)$. Thus, by Lemma 3.4, we have

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

as required in the first inequality of (1). Now we turn to the second inequality of (1). We may assume, without loss of generality, that $e \le 2^{k+1} \cdot \max_{i\in[2^k]} f(S_i)$. Therefore, we obtain

$$\begin{aligned}
e &\le 2^{k+1} \cdot \max_{i\in[2^k]} f(S_i) \\
&\le 2^{k+1} \cdot \max_{i\in[2^k]} |S_i|^{1-\delta} \cdot \mathrm{CC}(S_i) && \text{(by the promise of } \mathrm{Gap}_\delta\text{MCSP)} \\
&= 2^{k+1+(m-k)(1-\delta)} \cdot \max_{i\in[2^k]} \mathrm{CC}(S_i) && \text{(since } |S_i| = 2^{m-k}\text{)} \\
&\le 2^{k+1+(m-k)(1-\delta)} \cdot \mathrm{CC}(S) && \text{(by Lemma 3.4)} \\
&\le |S|^{1-\epsilon} \cdot \mathrm{CC}(S),
\end{aligned}$$

where the last inequality holds if $k + 1 + (m-k)(1-\delta) \le m \cdot (1-\epsilon)$, that is, $k \le m - m\epsilon/\delta - 1/\delta$. Thus we define k as $k := m - m\epsilon/\delta - 1/\delta$, which ensures the second inequality of (1).

Now we turn to analysis of the running time of the algorithm. Let 2^n be the length of the input to the algorithm. By the assumption on the stretch of the reduction, we have $|S| \le 2^{nc}$, that is, $m \le nc$. Therefore, $|S_i| = 2^{m-k} = 2^{m\epsilon/\delta+1/\delta} \le 2^{nc\epsilon/\delta+1/\delta}$. Since $c\epsilon/\delta < 1$, the algorithm above runs in polynomial time. (Indeed, let $t(N)$ be an upper bound of the running time of the algorithm

on inputs of length N and $\rho := c\epsilon/\delta < 1$. We have $t(N) \leq N^{O(1)}t(N^{\rho})$. Solving this recursive inequality, we obtain $t(N) = N^{O(1)}$.) □

4 HARDNESS FOR DET

In this section, we present some of our main contributions. We show that MKTP is hard for DET under $\leq_m^{NC^0}$ reductions (Theorem 4.3); prior to this, no variant of MCSP has been shown to be hard for any complexity class under any type of many-one reducibility. The $\leq_m^{NC^0}$ reduction that we present is nonuniform; we show that hardness under *uniform* reductions is closely 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 4.7). These techniques allow us to prove 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 [AGvM+18, Section 3.1]: It is proved there that there is a ZPP reduction from the rigid[3] graph isomorphism problem to MKTP. Here we modify that construction, to obtain a nonuniform AC^0 reduction (Corollary 4.2). Combining Torán's AC^0 reduction [Tor04] from DET to the rigid graph isomorphism problem as well as the Gap theorem [AAR98], we will show $\mathsf{DET} \leq_m^{NC^0} \mathsf{MKTP}$ (Theorem 4.3).

Next, we will establish that certain circuit size lower bounds are *equivalent to* the existence of certain *uniform* AC^0 reductions to MKTP. This will be accomplished, by derandomizing the reduction of Theorem 4.3. Using this equivalence as a tool, we then close the section with a series of results presenting consequences of MKTP or MCSP being hard for various complexity classes, under different types of reducibility.

4.1 Hardness of MKTP under nonuniform many-one reductions

We now modify the ZPP reduction of [AGvM+18, Section 3.1], which reduces the rigid graph isomorphism problem to MKTP, showing that it can be replaced by a nonuniform AC^0 reduction.

Lemma 4.1. *Let A be any oracle. There is a function f computable in Dlogtime-uniform AC^0 such that, for any two rigid graphs G_0, G_1 with n vertices:*

- $\Pr_r[f(G_0, G_1, r) \notin \mathsf{MKTP}^A] > 1 - \frac{1}{2^{4n^2}}$ *if* $G_0 \not\equiv G_1$, *and*
- $\Pr_r[f(G_0, G_1, r) \in \mathsf{MKTP}^A] = 1$ *if* $G_0 \equiv G_1$.

Proof. We present the proof for $A = \emptyset$; however, it is immediate that the proof carries over for any oracle A. The function f is given by the reduction presented in [AGvM+18, Section 3.1], showing that the Rigid Graph Isomorphism Problem is in $\mathsf{Promise\text{-}ZPP}^{\mathsf{MKTP}}$. This reduction takes graphs G_0 and G_1 as input, and interprets the random coin flip sequence r as a tuple (w, Π) where Π is a sequence of t random permutations $\pi_1, \ldots, \pi_t$, and $|w| = t$.

We make use of a result of Hagerup [Hag91], showing that there is a function e computed by Dlogtime-uniform AC^0 circuits,[4] generating a nearly-uniform distribution on permutations of n elements. More precisely, let S_n denote the symmetric group on $[n]$, where permutation σ is represented as a binary string of the form $\sigma(1)\ldots\sigma(n)$. Then for every ℓ there is a $k > \ell$ and a Dlogtime-uniform AC^0-computable function $e : \{0,1\}^{n^k} \to S_n \cup \{0^{n \log n}\}$ such that, for every $\sigma \in S_n$

$$\Pr_{s \in \{0,1\}^{n^k}}[e(s) = \sigma] \geq 1/n! - 2^{-n^\ell}$$

and $\Pr_{s \in \{0,1\}^{n^k}}[e(s) = 0^{n \log n}] \leq 2^{-n^\ell}$.

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

[4] Hagerup states this result in terms of CRCW PRAMs [Hag91] with a polynomial number of processors, running for $O(1)$ steps. It is known [BIS90] that this class coincides with Dlogtime-uniform AC^0.

Following the presentation in [AGvM+18], our AC^0 reduction takes two graphs G_0 and G_1, along with a random string $r = ws_1s_2 \dots s_t$ where $|w| = t = n^{O(1)}$ and each s_i has length n^k, where k and ℓ (from the previous paragraph) are chosen so that $2^{-t/2} + t/2^{n^\ell} < 2^{-4n^2}$.) Thus, for a randomly-chosen r, with probability at least $1 - (t/2^{n^\ell})$, in AC^0 we can compute the pair (w, Π) where $\Pi = \pi_1, \pi_2, \dots \pi_t = e(s_1), e(s_2), \dots, e(s_t)$. Next we compute the string $x_r = \pi_1(G_{w_1}), \dots, \pi_t(G_{w_t})$. (With probability at most $t/2^{n^\ell}$, some $e(s_i)$ consists of a block of zeros, in which case our AC^0 function will set x_r equal to a string of zeros, indicating failure.) We observe this is computable in AC^0: Graphs are encoded as adjacency matrices. Thus, given a graph G and a permutation π, the bit (r, s) of $\pi(G)$ is the same as the bit (i, j) in G, where $\pi(i) = r$ and $\pi(j) = s$. That is, position (r, s) in the output is the $\mathrm{OR}_{i,j}$ (taken over all relevant positions (i, j) in the encoding of π) of [$G_{i,j}$ AND [the encoding of π contains the strings (i, r) and (j, s)]]. This latter condition can easily be checked in AC^0.

The proof in [AGvM+18, Section 3.1] shows that, if $G_0 \equiv G_1$, then $\mathrm{KT}(x_r) \leq t(\log n!) + t/2$.

On the other hand, [AGvM+18] observes that if $G_0 \not\equiv G_1$ then the entropy of the distribution on strings x_r (assuming t uniformly random permutations and a uniformly-randomly chosen string w) is at least $t + t\log(n!)$, and hence the probability that $\mathrm{KT}(x_r) < (t + t\log(n!)) - t/2$ is at most $2^{-t/2}$. In our setting, the permutations are *very nearly* uniformly random (and it approaches the uniform distribution as ℓ increases), and there is also the possibility that x_r does not consist of t permuted graphs, but instead is all zeros. This latter condition arises with probability at most $t/2^{n^\ell}$. Recalling that $2^{-t/2} + t/2^{n^\ell} < 2^{-4n^2}$, we now have the following:

- If $G_0 \equiv G_1$, then $\mathrm{KT}(x_r) \leq t(\log n!) + t/2$.
- If $G_0 \not\equiv G_1$, then with probability $> 1 - 2^{-4n^2}$, we have $\mathrm{KT}(x_r) \geq t(\log n!) + t/2$.

We are now ready to define the AC^0-computable function f: $f(G_0, G_1, r) = (x_r, \theta)$, where $\theta = t(\log n!) + t/2$. We have established that f has the desired properties. □

Corollary 4.2. *Let A be any oracle. The rigid graph isomorphism problem is reducible to MKTP^A via a non-uniform $\leq_{\mathrm{m}}^{\mathrm{AC}^0}$ reduction.*

Proof. A standard counting argument shows that there is a value of r that can be hardwired into the probabilistic reduction of Lemma 4.1 that works correctly for all pairs (G_0, G_1) of n-vertex graphs. (Note that the length of the input is $2n^2$, and the error probability is at most $1/2^{4n^2}$.) □

Theorem 4.3. *Let A be any oracle.* DET *is reducible to MKTP^A via a non-uniform $\leq_{\mathrm{m}}^{\mathrm{NC}^0}$ reduction. Furthermore, this reduction is "natural" in the sense of [KC00].*

Proof. Since DET is closed under $\leq_{\mathrm{m}}^{\mathrm{TC}^0}$ reductions, it suffices to show that MKTP^A is hard under $\leq_{\mathrm{m}}^{\mathrm{AC}^0}$ reductions, and then appeal to the "Gap" theorem of [AAR98], to obtain hardness under $\leq_{\mathrm{m}}^{\mathrm{NC}^0}$ reducibility. Torán [Tor04] shows that DET is AC^0-reducible to GI. In fact it is shown in the proofs of Theorem 5.3 and Corollary 5.4 of [Tor04] that DET is AC^0-reducible to GI via a reduction that produces only pairs of rigid graphs as output. Composing this reduction with the non-uniform AC^0 reduction given by Corollary 4.2 completes the argument.

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

An appeal to the circuit lower bounds of Razborov and Smolensky [Raz87, Smo87] now yields the following corollary:

Corollary 4.4. *MKTP^A is not in $\mathrm{AC}^0[p]$ 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 [FSUV13] that has sublinear stretch and is secure against $\mathrm{AC}^0[p]$; see [HS17], where

a stronger separation from $\mathsf{AC}^0[p]$ is obtained in this way. Neither our argument nor those of [HS17, AGvM+18] seems easy to extend, to provide a lower bound for MCSP.)

4.2 Equivalence between hardness of MKTP and circuit lower bounds

The reader may wonder whether the non-uniform reduction can be made uniform under a suitable derandomization hypothesis. We do not know how to obtain a uniform AC^0-many-one reduction, but we can come close, if the oracle A is not too complex. Recall the definition of ctt-reductions: $B \leq_{\mathrm{ctt}}^{C} C$ if there is a function $f \in C$ 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 furthermore 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 [MW17, AHK17], 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.[5] 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 4.7, we present a related condition on reductions to MKTP^A that is *equivalent* to circuit lower bounds.

Theorem 4.5. *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 [Tor04]. Consider the following family of statistical tests $T_x(r)$, indexed by strings x:

> On input r:
> Compute $z = f(g(x), r)$, where $f(G_0, G_1, r)$ is the function from Lemma 4.1.
> 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 [KvM02] (building on the work of Impagliazzo and Wigderson [IW97]) show that, if $\mathsf{DSPACE}(n)$ requires exponential-size circuits from a given class C, then there is a hitting set generator computable in logspace that hits all large sets computable by circuits from C that have 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 $h(0^n) = (r_1, r_2, \ldots, r_{n^c})$ hardwired into it. On input x, the circuit computes the queries $f(g(x), r_i)$ for $1 \leq i \leq n^c$, and accepts if, for all i, $f(g(x), r_i) \in \mathsf{MKTP}^A$. Note that if $x \notin B$, then one of the r_i is accepted by T_x, which means that $f(g(x), r_i) \notin \mathsf{MKTP}^A$; if $x \in B$, then $f(g(x), r_i) \in \mathsf{MKTP}^A$ for all i. This establishes that the reduction is correct. □

It is also possible to prove a result analogous to Theorem 4.5, in terms of Dlogtime-uniform AC^0 reductions, in place of logspace-uniform AC^0 reductions. However, this requires a much stronger, *average case* circuit lower bound, for sets in LTH (as opposed to $\mathsf{DSPACE}(n)$):

[5] Recall that $\mathsf{LTH} = \bigcup_k \Sigma_k \mathsf{TIME}(O(n))$ is the linear-time analog of the polynomial time hierarchy.

Theorem 4.6. *Let A be any oracle. There is a constant c such that, if there is some $\epsilon > 0, b \geq 1$ and a set B in the d-th level of* LTH *such that, for all large n and every oracle circuit C of size $2^{\epsilon n}$,*

$$\Pr_{x \in \{0,1\}^n}[B(x) = C^{\mathsf{MKTP}^A}(x)] < 1 - 1/n^b,$$

then every language in DET *reduces to* MKTP^A *via a natural Dlogtime-uniform* $\leq^{\mathsf{AC}^0}_{\mathrm{ctt}}$*-reduction of depth $d + c$.*

Proof. The idea is similar to the proof of Theorem 4.5. Let $B \in$ DET. We consider the same family of statistical tests $T_x(r)$.

Viola [Vio05, Theorem 4.3] shows that, under the hypothesis of Theorem 4.6, there is a pseudo-random generator $G : \{0,1\}^{O(\log n)} \to \{0,1\}^{n^c}$ that is secure against all statistical tests computable by circuits of size n^c. In particular, as in the proof of Theorem 4.5, we obtain a hitting set generator h. The depth required by the construction in [Vio05] is $d + O(1)$.

The rest of the proof proceeds precisely as in the proof of Theorem 4.5. □

We remark that the hardness assumption of Theorem 4.5 ($\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$) can probably be weakened (saying that $\mathsf{DSPACE}(n)$ requires large circuits of some restricted sort), since the class of statistical tests that need to be fooled consists only of NC^2 circuits with one oracle gate. On the other hand, Theorem 4.7 indicates that the hardness assumption that we use is *equivalent* to the existence of uniform reductions, for certain oracles A – so it is not clear that there is much to be gained by searching for a weaker hardness assumption.

Theorem 4.5 deals with the oracle problem MKTP^A, but the most interesting case is the case where $A = \emptyset$, both because the hypothesis seems most plausible in that case, and because MKTP has been studied in connection with MCSP, which has been studied more than the associated oracle circuit problem MCSP^A. The hypothesis is false when $A =$ QBF, since the KT^A measure is essentially the same as the KS measure studied in [ABK+06], where it is shown that $\mathsf{PSPACE} = \mathsf{ZPP}^{R_{\mathrm{KS}}}$, and thus PSPACE has polynomial-size $\mathsf{MKTP}^{\mathsf{QBF}}$-circuits. Strikingly, it is of interest that not only the hypothesis is false in this case – but the conclusion is false as well. (See Corollary 4.10.)

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

- PARITY *reduces to* MKTP^A *via a natural logspace-uniform* $\leq^{\mathsf{AC}^0}_{\mathrm{ctt}}$*-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* $\leq^{\mathsf{AC}^0}_{\mathrm{ctt}}$*-reduction.*

Furthermore, if PARITY *reduces to* MCSP^A *via a natural logspace-uniform* $\leq^{\mathsf{AC}^0}_{\mathrm{ctt}}$*-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.

At this point, it is useful to recall a lemma from [AHK17] (distilled from [MW17]) that describes how the complexity depends on θ:

Lemma 4.8. *[AHK17, Claim 3.11] For any language A and any $0 \leq v \leq m$, MCSP^A on inputs $f \in \{0,1\}^m$, with the size parameter fixed to θ, is solved by a DNF formula of size $m \cdot 2^{O(\theta^2 \log \theta)}$.*

Thus, by Lemma 4.8, 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 [Hås87] 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 [ABK+06, 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 $(0^n 1^i 0^j)$, 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 $n^{1/20d} \geq 2^{m/20d}$.

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 $\leq^{\mathsf{AC}^0}_{\mathsf{ctt}}$-reduction.)

The assumption that $\mathsf{MKTP}^A \in \mathsf{P}^A/\mathrm{poly}$ 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 small enough 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.

Theorem 4.5 establishes that the third condition implies the fourth. The fourth condition obviously implies the first. □

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.

The logspace uniformity condition in Theorem 4.7 can be replaced by other less-restrictive uniformity conditions. We mention the following example:

Corollary 4.9. *Let $\mathsf{MKTP}^A \in \mathsf{P}^A/\mathrm{poly}$. Then the following are equivalent:*

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

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

At this point, we should consider the class of oracles for which Theorem 4.7 applies. That is, what is the set of oracles A for which $\mathsf{MKTP}^A \in \mathsf{P}^A/\mathrm{poly}$? First, we observe that this condition holds for any PSPACE-complete set, which yields the following corollary:

Corollary 4.10. PARITY *does not reduce to either* $\mathsf{MKTP}^{\mathsf{QBF}}$ *or* $\mathsf{MCSP}^{\mathsf{QBF}}$ *via a natural logspace-uniform* $\leq_{\mathrm{ctt}}^{\mathrm{AC}^0}$*-reduction.*

Remark: As an instructive example of a set for which $\mathsf{MKTP}^A \in \mathsf{P}^A/\mathrm{poly}$, consider the set $A = \{(M, x, 1^m) : M$ is an alternating Turing machine that accepts x, and runs in time at most m and makes at most $\log m$ alternations$\}$. A is complete for the class $\mathsf{ATIME\text{-}ALT}(n^{O(1)}, O(\log n))$ under $\leq_{\mathrm{m}}^{\mathrm{AC}^0}$ reductions. It is easy to see 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 4.7 hold.

Nonetheless, we do grant that this does seem to be a strong condition to place upon the oracle A – and it has even stronger consequences than are listed in Theorem 4.7. For instance, note that the proof that the first condition in Theorem 4.7 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. (Or you can substitute any other problem and class of mod circuits, where an exponential lower bound is known because of [Raz87, Smo87].) In fact, as in the proof of [AHK17, Lemma 3.10] we can apply random restrictions in a logspace-uniform way (as described in [Agr11]) and obtain a reduction from PARITY to MKTP^A where the queries are computed by logspace-uniform NC^0 circuits! For example, here is an argument showing that MAJORITY is reducible to MKTP^A (for oracle A satisfying the hypotheses of Theorem 4.7) via natural ctt-reductions computed by logspace-uniform $\mathsf{AC}^0[3]$ circuits iff PARITY is reducible to MKTP^A via reductions where the queries are computed by logspace-uniform NC^0 circuits:

Assume first that MAJORITY is reducible to MKTP^A via natural ctt-reductions computed by logspace-uniform $\mathsf{AC}^0[3]$ circuits. The proof that the first condition in Theorem 4.7 implies the second also shows that the second condition holds if MAJORITY is reducible to MKTP^A via natural ctt-reductions computed by logspace-uniform $\mathsf{AC}^0[3]$ circuits. (The only things that need to be changed, are (1) every occurrence of "PARITY" should be changed to MAJORITY" (2) the phrase "consisting of AC^0 circuitry feeding into oracle gates" should be changed to "consisting of $\mathsf{AC}^0[3]$ circuitry feeding into oracle gates", and (3) "Note that all of the oracle gates g_i must output 1 on input $0^{n-1}1$" should be replaced by "Note that all of the oracle gates g_i must output 1 on input 1^n". Thus, under our assumption, all four of the conditions in Theorem 4.7 hold. In particular, PARITY reduces to MKTP^A via a natural logspace-uniform $\leq_{\mathrm{ctt}}^{\mathrm{AC}^0}$-reduction. The AC^0-computable function f that computes the list of oracle queries has the property that there is a logspace-computable restriction ρ that leaves n^ϵ input variables unset (for some $\epsilon > 0$) with the property that the function $f|_\rho$ on n^ϵ variables is NC^0-computable. (See, e.g., [AAR98, Lemma 7] and see [Agr11] to see how ρ can be computed in logspace.) This yields the claimed reduction from PARITY to MKTP^A where the queries are NC^0-computable.

Conversely, assume now that PARITY is reducible to MKTP^A via a natural $\leq_{\mathrm{ctt}}^{\mathrm{AC}^0}$-reduction where the queries are computed by logspace-uniform NC^0 circuits. This is a stronger condition than the first condition on Theorem 4.7, and hence all four of these conditions hold. In particular, DET reduces to MKTP^A via $\leq_{\mathrm{ctt}}^{\mathrm{AC}^0}$ reductions. The claim now follows, since MAJORITY $\in$ DET.

We find these implications to be surprising. The "gap" phenomenon that was described in [AAR98] (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 4.5 and Corollary 4.10. $\mathrm{MKTP}^{\mathsf{QBF}}$ is hard for PSPACE under ZPP-Turing reductions [ABK+06], whereas MKTP is in NP. Thus $\mathrm{MKTP}^{\mathsf{QBF}}$ appears to be *much harder* than MKTP. Yet, Theorem 4.5 shows that, under a plausible hypothesis, the "easier" set MKTP is hard for DET, whereas (by Corollary 4.10) the "harder" problem $\mathrm{MKTP}^{\mathsf{QBF}}$ cannot even be used as an oracle for PARITY under this same reducibility.

In other words, the (conditional) natural logspace-uniform $\leq_{\mathrm{ctt}}^{\mathrm{AC}^0}$ reductions from problems in DET to MKTP given in Theorem 4.5 are not "oracle independent" in the sense of [HW16]. 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 $\leq_{\mathrm{T}}^{\mathrm{L}}$ reductions. All prior reductions (such as those in [AD17, ABK+06, AGvM+18]) had been probabilistic and/or non-uniform, or (even under derandomization hypotheses) seemed difficult to implement in NC. But Theorem 4.7 shows that it is quite likely that MKTP is hard for DET under $\leq_{\mathrm{T}}^{\mathrm{L}}$ reductions (and even under much more restrictive reductions). Previously, we had viewed the results of [AHK17] 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.

4.3 On the importance of uniformity

Surprisingly (to us), the notion of uniformity appears to be central. In particular, the reader is probably wondering whether the logspace-uniformity condition in Theorem 4.5 (relating hardness of MKTP^A to worst-case circuit lower bounds) can be improved to Dlogtime-uniformity. As a partial answer to this question, we note that Viola [Vio05] shows that there is no black-box construction of a pseudorandom generator computable in AC^0 that is based on worst-case circuit lower bounds. In this section, in Theorem 4.12, we show that, when considering hardness of MKTP and MCSP, small details about the complexity of the reduction (including the precise depth, and the notion of uniformity) cannot be ignored.

First, we recall Corollary 3.7 of [AHK17], which states that $\mathrm{MKTP}^{\mathsf{QBF}}$ is not hard for P under $\leq_{\mathrm{m}}^{\mathrm{L}}$ reductions unless PSPACE = EXP. It turns out that this holds even for logspace-Turing reductions.

Theorem 4.11. *$\mathrm{MKTP}^{\mathsf{QBF}}$ is not hard for* P *(or* NP*) under $\leq_{\mathrm{T}}^{\mathrm{L}}$ reductions unless* PSPACE = EXP *(*PSPACE = NEXP*, respectively).* *$\mathrm{MKTP}^{\mathsf{QBF}}$ is not hard for* PSPACE *under $\leq_{\mathrm{T}}^{\mathrm{L}}$ reductions. The same holds for $\mathrm{MCSP}^{\mathsf{QBF}}$.*

We include this proof here, both because it improves a Corollary in [AHK17], and because the proof can be viewed as a warm-up for the proof of Theorem 4.12.

Proof. First, note that $\leq_{\mathrm{T}}^{\mathrm{L}}$ and $\leq_{\mathrm{tt}}^{\mathrm{L}}$ reducibilities coincide [LL76]. Thus assume that $\mathrm{MKTP}^{\mathsf{QBF}}$ is hard for P under $\leq_{\mathrm{tt}}^{\mathrm{L}}$ reductions; we will show that PSPACE = EXP. (The proof for $\mathrm{MCSP}^{\mathsf{QBF}}$ is identical, and the variant concerning hardness for NP is analogous.)

The proof idea is based on [HW16]: Assume that $\mathsf{P} \subseteq \mathsf{L}_{\mathrm{tt}}^{\mathrm{MKTP}^{\mathsf{QBF}}}$. (Here, L_{tt} means a $\leq_{\mathrm{tt}}^{\mathrm{L}}$ reduction.) By standard padding, we obtain $\mathsf{EXP} \subseteq \mathsf{PSPACE}_{\mathrm{tt}}^{\mathrm{MKTP}^{\mathsf{QBF}}}$. Any query of a $\mathsf{PSPACE}_{\mathrm{tt}}$ machine has low $\mathrm{KT}^{\mathsf{QBF}}$ complexity. Moreover, one can check whether a string has low $\mathrm{KT}^{\mathsf{QBF}}$ complexity in PSPACE. Combining these two facts, we obtain $\mathsf{EXP} \subseteq \mathsf{PSPACE}_{\mathrm{tt}}^{\mathrm{MKTP}^{\mathsf{QBF}}} = \mathsf{PSPACE}$. A formal proof follows.

Let $B \in \mathsf{EXP}$. Let $B' = \{x10^{2^{|x|}} : x \in B\}$ and note that $B' \in \mathsf{P}$. Consider the $\leq_{\mathrm{tt}}^{\mathrm{L}}$ reduction that reduces B' to MKTP^A. On any input string y, let the i-th oracle query be $q_i(y)$. The language $\{(i, j, x)$

: the j-th bit of $q_i(x10^{2^{|x|}})$ is 1} is in PSPACE and thus is in $\mathsf{P}^{\mathsf{QBF}}$. It follows that $q_i(x10^{2^{|x|}})$ is of the form (y_i, θ_i), where $\mathrm{KT}^{\mathsf{QBF}}(y_i) = |x, i, j|^{O(1)}$. Thus, to evaluate the oracle query q_i on input $x10^{2^{|x|}}$, this PSPACE computation (on input x) suffices: Compute the bits of θ_i; this can be done in PSPACE, since the number of bits in θ_i is at most $|x|^{O(1)}$, and each bit is computable in PSPACE. If $\theta_i > |x, i, j|^c$ (for the appropriate value of c), then return "1" since the query y_i certainly has KT^A complexity less than this. Otherwise, try all descriptions d of length at most θ_i, to determine whether there is some such d for which $U^{\mathsf{QBF}}(d, j)$ is equal to the j-th bit of q_i (allowing at most $|x, i, j|^c$ steps for the computation of U).

The rest of the $\leq^{\mathsf{L}}_{\mathrm{tt}}$ reduction on input $x10^{2^{|x|}}$ can be computed in space $|x|^{O(1)}$, by re-computing the values of the oracle queries, as required.

The unconditional result that $\mathsf{MKTP}^{\mathsf{QBF}}$ is not hard for PSPACE under $\leq^{\mathsf{L}}_{\mathrm{T}}$ reductions follows along the same lines, choosing $B \in \mathsf{EXPSPACE}$, and leading to the contradiction EXPSPACE = PSPACE. □

A similar approach yields the following result:

Theorem 4.12. *For each $d \geq 0$, if $\Sigma^p_{d+2} \subseteq \mathsf{P}^A/\mathrm{poly}$ and $\mathsf{PSPACE} \not\subseteq \mathsf{PH}^A$, then neither MKTP^A nor MCSP^A is hard for NC^1 under Dlogtime-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{tt}}$ reductions of depth d.*

Proof. We present the proof for MKTP^A; the proof for MCSP^A is identical.

Assume that MKTP^A is hard for NC^1 under Dlogtime-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{tt}}$ reductions of depth d; we will show that $\mathsf{PSPACE} \subseteq \mathsf{PH}^A$.

By the closure properties of PH, it will suffice to show that $\mathsf{ATIME}(n) \subseteq \mathsf{PH}^A$.

Let $B \in \mathsf{ATIME}(n)$. Let $B' = \{x10^{2^{|x|}} : x \in B\}$ and note that $B' \in \mathsf{NC}^1$. Consider the oracle family (C_m) that reduces B' to MKTP^A. Let the oracle gates in C_{2^n+n+1} be $g_1, g_2, \ldots, g_\ell$. On any input string y, let the query that is fed into gate g_i be $q_i(y)$. The language $\{(2^{|x|} + |x| + 1, i, j, x) :$ the j-th bit of $q_i(x10^{2^{|x|}})$ is 1} is in Σ^p_{d+2} and thus is in $\mathsf{P}^A/\mathrm{poly}$. It follows that $q_i(x10^{2^{|x|}})$ is of the form (y_i, θ_i), where $\mathrm{KT}^A(y_i) = |x, i, j|^{O(1)}$. Thus, to evaluate oracle gate g_i on input $x10^{2^{|x|}}$, this PH^A computation (on input x) suffices: Compute the bits of θ_i; this can be done in PH, since the number of bits in θ_i is at most $|x|^{O(1)}$, and each bit is computable in PH. If $\theta_i > |x, i, j|^c$ (for the appropriate value of c), then return "1" since the query y_i certainly has KT^A complexity less than this. Otherwise, guess a description d of length at most θ_i, and universally check (for each j) that $U^A(d, j)$ is equal to the j-th bit of q_i (allowing at most $|x, i, j|^c$ steps for the computation of U).

To evaluate the rest of the circuit, note that the unbounded fan-in AND and OR gates that sit just above the oracle gates can also be evaluated in PH^A (at one level higher in the hierarchy than is required to evaluate the oracle gates). Repeating this process through the remaining $O(1)$ levels of the circuit yields the desired PH^A algorithm for B. □

Remark: The significance of Theorem 4.12 is best viewed by combining it with Theorem 4.5. If we choose A to be any PP-complete set, or if we choose A to be one of the sets discussed in the Remark after Corollary 4.10, then for all d we have $\Sigma^p_{d+2} \subseteq \mathsf{P}^A$ and both of the hypotheses

- $\mathsf{PSPACE} \not\subseteq \mathsf{PH}^A$, and
- $\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$

are plausible. Thus, for such oracles A, under a plausible hypothesis, we have both MKTP^A is *not* hard for NC^1 under Dlogtime-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{tt}}$ reductions, and MKTP^A *is* hard for DET under logspace-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{ctt}}$ reductions. Thus different notions of uniformity are a key part of the puzzle, when trying to understand the hardness of problems such as MKTP and MCSP.

As another example, choose A to be any set that is complete for Σ^p_{d+2}, and assume PSPACE $\neq$ PH. Then under the strong-but-plausible hypothesis that there is a set $B \in \Sigma_{d'}\mathsf{TIME}(n)$ that has large symmetric difference with any set in $\mathsf{SIZE}^{\mathsf{MKTP}^A}(2^{\epsilon n})$, we have $\Sigma^p_{d+2} \subseteq \mathsf{P}^A$ and PSPACE $\not\subseteq \mathsf{PH}^A = \mathsf{PH}$, thereby satisfying the hypotheses of both Theorem 4.12 and Theorem 4.6. Thus, for this choice of A, under a plausible hypothesis, we have both MKTP^A is *not* hard for NC^1 under Dlogtime-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{tt}}$ reductions of depth d, and MKTP^A *is* hard for DET under Dlogtime-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{ctt}}$ reductions of depth $d' + c$ (where c is the constant from Theorem 4.6).

In both of these examples, the key ingredient seems to be that, in order for AC^0 to be able to reduce problems to MCSP^A or MKTP^A, it is essential to be able to formulate useful queries, by either having sufficient depth, or by having sufficient power in the uniformity condition.

We are even able to extend our approach in some cases, to apply to AC^0-Turing reducibility.

Theorem 4.13. *Let* $\mathsf{NP}^A \subseteq \mathsf{P}^A/\mathrm{poly}$. *If* PSPACE $\not\subseteq \mathsf{PH}^A$, *then neither* MKTP^A *nor* MCSP^A *is hard for* NC^1 *under Dlogtime-uniform* $\leq^{\mathsf{AC}^0}_{\mathrm{T}}$ *reductions.*

Proof. The proof is similar to that of Theorem 4.12. Assume that MKTP^A is hard for NC^1 under Dlogtime-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{T}}$ reductions; we will show that $\mathsf{ATIME}(n) \subseteq \mathsf{PH}^A$ by presenting a PH^A algorithm to evaluate the gates in the $\leq^{\mathsf{AC}^0}_{\mathrm{T}}$ reduction of the NC^1 language B' from the proof of Theorem 4.12.

Note that in a circuit computing an $\leq^{\mathsf{AC}^0}_{\mathrm{T}}$ reduction, there is an "initial" layer of oracle gates, whose queries are computed nonadaptively, while all oracle gates at deeper levels have queries whose values depend upon oracle gates at earlier levels in the circuit. Note also that, under the given assumption $\mathsf{NP}^A \subseteq \mathsf{P}^A/\mathrm{poly}$, we can conclude that $\mathsf{PH}^A \subseteq \mathsf{P}^A/\mathrm{poly}$.

The proof now proceeds along precisely the same lines as the proof of Theorem 4.12, which shows that a PH^A computation can compute the value of each wire that feeds into the "initial" layer of oracle gates. Similarly, as in the proof of Theorem 4.12, all of the AND, OR, and NOT gates at higher levels can be computed in PH^A, given that the gates at lower levels can be evaluated in PH^A. Thus, we need only show how to deal with oracle gates at deeper levels.

Consider any such oracle gate g. On any input string y, let the query that is fed into gate g when evaluating the circuit on input y be $q_g(y)$. The language $\{(2^{|x|} + |x| + 1, g, j, x) :$ the j-th bit of $q_g(x10^{2^{|x|}})$ is 1$\}$ is in PH^A and thus (by our new assumption) is in $\mathsf{P}^A/\mathrm{poly}$. It follows that $q_g(x10^{2^{|x|}})$ is of the form (y, θ), where $\mathrm{KT}^A(y) = |x, g, j|^{O(1)}$. Thus, to evaluate oracle gate g on input $x10^{2^{|x|}}$, this PH^A computation (on input x) suffices: Compute the bits of θ; this can be done in PH^A, since the number of bits in θ_i is at most $|x|^{O(1)}$, and each bit is computable in PH. If $\theta_i > |x, g, j|^c$ (for the appropriate value of c), then return "1" since the query y certainly has KT^A complexity less than this. Otherwise, guess a description d of length at most θ, and universally check (for each j) that $U^A(d, j)$ is equal to the j-th bit of q_g (allowing at most $|x, i, j|^c$ steps for the computation of U). □

In order to compare our results with those of [AHK17, MW17], we also state a related theorem, whose proof is similar:

Theorem 4.14. *Let* $\mathsf{NP}^A \subseteq \mathsf{P}^A/\mathrm{poly}$. *If* $\mathsf{CH} \not\subseteq \mathsf{PH}^A$, *then neither* MKTP^A *nor* MCSP^A *is hard for* TC^0 *under Dlogtime-uniform* $\leq^{\mathsf{AC}^0}_{\mathrm{T}}$ *reductions.*

Proof. The proof is nearly identical to that of Theorem 4.13.

Under the assumption that MKTP^A is hard for TC^0 under Dlogtime-uniform $\leq^{\mathsf{AC}^0}_{\mathrm{T}}$ reductions; it suffices to show that the linear-time counting hierarchy (see [AKR+01] for a definition) is contained in PH^A by presenting a PH^A algorithm for a set B in the linear-time counting hierarchy. The language B' is now in TC^0, instead of merely being in NC^1. The rest of the proof proceeds virtually

unchanged. (One can modify the statement of Theorem 4.12 in a similar way, but we do not include this modification here.) □

A consequence of Theorems 4.13 and 4.14 is the following corollary, which has the same flavor of results of the form "MCSP is hard for class C implies a likely but hard-to-prove consequence" as presented by Murray and Williams [MW17], but moving beyond the $\leq_m^{AC^0}$ reductions considered by them, to the more general $\leq_T^{AC^0}$ reductions.

Corollary 4.15. *If either of* MKTP *or* MCSP *is hard for* NC^1 *(or* TC^0*) under Dlogtime-uniform* $\leq_T^{AC^0}$ *reductions, then* NP ≠ NC *(*NP ≠ TC^0*, respectively).*

Proof. This follows from Theorems 4.13 and 4.14 when $A = \emptyset$. If NP = NC, then NP ⊆ P/poly, and PH = NC ≠ PSPACE. Thus neither MKTP nor MCSP is hard for NC^1 under Dlogtime-uniform $\leq_T^{AC^0}$ reductions. Also, if NP = TC^0, then NP ⊆ P/poly, and PH = TC^0 ≠ CH [All99]. Thus neither MKTP nor MCSP is hard for TC^0 under Dlogtime-uniform $\leq_T^{AC^0}$ reductions. □

Corollary 4.15 should be compared to the earlier work of [MW17, AHK17]. Murray and Williams presented nonuniform lower bounds that would follow from MCSP or MKTP being hard for NP under Dlogtime-uniform $\leq_m^{AC^0}$ reductions. In [AHK17] even stronger nonuniform consequences were shown to follow from the weaker assumption of hardness for TC^0. (See Table 2.) In Corollary 4.13, we present a weaker *uniform* lower bound that follows from the weaker assumption that MCSP or MKTP is hard for TC^0 under a *more powerful* notion of reducibility.

We also present another result in this vein, about NP-completeness. Prior work [MW17, AHK17] had obtained stronger consequences from the stronger assumption that MCSP is NP-complete under Dlogtime-uniform $\leq_m^{AC^0}$ reductions.

Corollary 4.16. *If either of* MKTP *or* MCSP *is hard for* NP *under Dlogtime-uniform* $\leq_T^{AC^0}$ *reductions, then*

$$NP \neq MA \cap P/poly.$$

Proof. If you modify the proof of Theorem 4.13, replacing NC^1 by NP and replacing PSPACE by NEXP, you obtain that, if NP ⊆ P/poly, then NEXP ≠ PH implies that neither MKTP nor MCSP is hard for NP under Dlogtime-uniform $\leq_T^{AC^0}$ reductions. (That is, if we assume that MKTP is hard for NP under Dlogtime-uniform $\leq_T^{AC^0}$ reductions, then the argument from Theorem 4.13 shows that NEXP ⊆ PH, by presenting a PH algorithm to evaluate the gates in an AC^0 oracle circuit reducing an NP language B' to MKTP.)

Or, restating this using the same hypothesis as in the statement of the corollary, if MKTP or MCSP is hard for NP under Dlogtime-uniform $\leq_T^{AC^0}$, then either NP $\not\subseteq$ P/poly or NEXP = PH. Since (NP ⊆ P/poly and NEXP = PH) is equivalent to NEXP ⊆ P/poly, and since NEXP ⊆ P/poly is equivalent to NEXP = MA [IKW02], we obtain that NP-hardness of MCSP or MKTP implies NP $\not\subseteq$ P/poly or NEXP = MA. (Murray and Williams obtain essentially this same consequence under the stronger assumption that MCSP is complete under $\leq_m^{AC^0}$ reductions, but are also able to show that NEXP $\not\subseteq$ P/poly in this case.)

In either case, we obtain the consequence NP ≠ MA ∩ P/poly. □

We close this section with another variant of Theorem 4.13, proved via the same technique:

Theorem 4.17. *Let* $NP^A \subseteq P^A/poly$. *If* NEXP $\not\subseteq$ $PSPACE^A$ *(or* NEXP $\not\subseteq$ EXP^A*), then neither* $MKTP^A$ *nor* $MCSP^A$ *is hard for* NP *under logspace-uniform* $\leq_T^{AC^0}$ *reductions (*P*-uniform* $\leq_T^{AC^0}$ *reductions, respectively).*

Corollary 4.18. $\mathrm{MKTP}^{\mathsf{QBF}}$ *is not hard for* NP *under logspace-uniform* $\leq_{\mathrm{T}}^{\mathrm{AC}^0}$ *reductions (*P*-uniform* $\leq_{\mathrm{T}}^{\mathrm{AC}^0}$ *reductions) unless* PSPACE = NEXP *(*EXP = NEXP*, respectively). The same holds for* $\mathrm{MCSP}^{\mathsf{QBF}}$.

Although the following corollary discusses $\leq_{\mathrm{T}}^{\mathrm{AC}^0}$ reductions, it also says something about $\leq_{\mathrm{T}}^{\mathsf{L}}$ reducibility. This is because, assuming $\mathsf{DSPACE}(n) \not\subseteq \text{io-}\mathsf{SIZE}^{\mathrm{MKTP}^A}(2^{\epsilon n})$, any $\leq_{\mathrm{T}}^{\mathsf{L}}$ reduction to MKTP can be simulated by a logspace-uniform $\leq_{\mathrm{T}}^{\mathrm{AC}^0}$ reduction to MKTP. (To see this, note that, by Theorem 4.5, MKTP is hard for DET under this class of reductions, and hence each of the logspace-computable (nonadaptive) queries can be computed using oracle gates for MKTP, and similarly the logspace computation that uses the queries can also be simulated using MKTP. Similar observations arise in [AO96].)

Corollary 4.19. *If either of* MKTP *or* MCSP *is hard for* NP *under logspace-uniform* $\leq_{\mathrm{T}}^{\mathrm{AC}^0}$ *reductions (*P*-uniform* $\leq_{\mathrm{T}}^{\mathrm{AC}^0}$ *reductions), then* $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ *or* NEXP = PSPACE *(*NEXP = EXP*, respectively).*

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 [MW17] 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 P/poly reductions (without proving circuit lower bounds)?

Open Questions. It should be possible to prove unconditionally that MCSP is not in $\mathrm{AC}^0[2]$; we conjecture that the hardness results that we are able to prove for MKTP hold also for MCSP. (We refer the reader to [AGvM$^+$18, Section 7] for a more detailed discussion about the obstacles that need to be overcome, in order to extend these theorems to MCSP.)

We suspect that it should be possible to prove more general results of the form "If MCSP^A is hard for class C, then so is MKTP^A". We view Theorem 4.7 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 Oliveira and Santhanam on MCSP [OS17]?

Is there evidence that $\mathrm{Gap}_\epsilon\mathrm{MCSP}$ 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, Rahul Ilango, and Prashant Nalini Vasudevan for helpful discussions. We thank Emanuele Viola for calling our attention to the work of Hagerup [Hag91]. We thank the anonymous referees for their careful reading and helpful suggestions. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing. E. A. was supported by National Science Foundation grants CCF-1514164

and CCF-1555409; some of the work was done while this author was visiting the Institute for Mathematical Sciences, National University of Singapore in 2017, on a visit that was supported by the Institute. S.H. was supported by JSPS KAKENHI Grant Number JP16J06743.

REFERENCES

[AAR98] 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.

[ABK+06] Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.

[AD17] Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Information and Computation*, 256:2–8, 2017. Special issue for MFCS '14.

[Agr11] Manindra Agrawal. The isomorphism conjecture for constant depth reductions. *Journal of Computer and System Sciences*, 77(1):3–13, 2011.

[AGvM+18] Eric Allender, Joshua Grochow, Dieter van Melkebeek, Andrew Morgan, and Cristopher Moore. Minimum circuit size, graph isomorphism and related problems. *SIAM Journal on Computing*, 47:1339–1372, 2018.

[AHK17] Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. *Computational Complexity*, 26(2):469–496, 2017.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57:14:1 – 14:36, 2010.

[AKR+01] Eric Allender, Michal Koucký, Detlef Ronneburger, Sambuddha Roy, and V. Vinay. Time-space tradeoffs in the counting hierarchy. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*, pages 295–302, 2001.

[AKRR10] 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.

[All99] Eric Allender. The permanent requires large uniform threshold circuits. *Chicago J. Theor. Comput. Sci.*, 1999.

[All04] Eric Allender. Arithmetic circuits and counting complexity classes. In J. Krajíček, editor, *Complexity of Computations and Proofs*, volume 13 of *Quaderni di Matematica*, pages 33–72. Seconda Università di Napoli, 2004.

[AO96] Eric Allender and Mitsunori Ogihara. Relationships among PL, #L, and the determinant. *RAIRO - Theoretical Information and Application*, 30:1–21, 1996.

[BIS90] D. A. M. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.

[FSUV13] Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On beating the hybrid argument. *Theory of Computing*, 9:809–843, 2013.

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

[Gol06] 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.

[Hag91] Torben Hagerup. Fast parallel generation of random permutations. In *Proc., 18th International Colloquium on Automata, Languages and Programming, (ICALP)*, volume 510 of *Lecture Notes in Computer Science*, pages 405–416. Springer, 1991.

[Hås87] Johan Håstad. *Computational Limitations for Small Depth Circuits*. MIT Press, Cambridge, MA, 1987.

[Hås99] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:1364–1396, 1999.

[HP15] John M. Hitchcock and Aduri Pavan. On the NP-completeness of the minimum circuit size problem. In *Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS)*, volume 45 of *LIPIcs*, pages 236–245. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[HS17] Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In *32nd Conference on Computational Complexity, CCC*, volume 79 of *LIPIcs*, pages 7:1–7:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[HW16] Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *31st Conference on Computational Complexity, CCC*, volume 50 of *LIPIcs*, pages 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[IKV18] R. Impagliazzo, V. Kabanets, and I. Volkovich. The power of natural properties as oracles. In *33rd Conference on Computational Complexity, CCC*, volume 102 of *LIPIcs*, pages 7:1–7:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[IKW02] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *J. Comput. Syst. Sci.*, 65(4):672–694, 2002.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 230–235, 1989.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *ACM Symposium on Theory of Computing (STOC)*, pages 220–229, 1997.

[KC00] Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *ACM Symposium on Theory of Computing (STOC)*, pages 73–79, 2000.

[KvM02] 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.

[Lad75] Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975.

[LL76] Richard E. Ladner and Nancy A. Lynch. Relativization of questions about log space computability. *Mathematical Systems Theory*, 10:19–32, 1976.

[MW17] Cody Murray and Ryan Williams. On the (non) NP-hardness of computing circuit complexity. *Theory of Computing*, 13(4):1–22, 2017.

[OS17] Igor Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds and pseudorandomness. In *32nd Conference on Computational Complexity, CCC*, volume 79 of *LIPIcs*, pages 18:1–18:49. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[Ost91] 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.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Second Israel Symposium on Theory of Computing Systems (ISTCS)*, pages 3–17. IEEE Computer Society, 1993.

[Raz87] 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.

[RR97] Alexander Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:24–35, 1997.

[Rud17] Michael Rudow. Discrete logarithm and minimum circuit size. *Information Processing Letters*, 128:1–4, 2017.

[Smo87] 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.

[Sri03] Aravind Srinivasan. On the approximability of clique and related maximization problems. *Journal of Computer and System Sciences*, 67(3):633–651, 2003.

[Tor91] Jacobo Torán. Complexity classes defined by counting quantifiers. *J. ACM*, 38(3):753–774, 1991.

[Tor04] Jacobo Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.

[Vad06] Salil P. Vadhan. An unconditional study of computational zero knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006.

[Vio05] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2005.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer-Verlag New York Inc., 1999.

Received January 2018