On the NP-Completeness of the Minimum Circuit Size Problem

John M. Hitchcock*
Department of Computer Science
University of Wyoming

A. Pavan†
Department of Computer Science
Iowa State University

Abstract

We study the Minimum Circuit Size Problem (MCSP): given the truth-table of a Boolean function f and a number k, does there exist a Boolean circuit of size at most k computing f? This is a fundamental NP problem that is not known to be NP-complete. Previous work has studied consequences of the NP-completeness of MCSP. We extend this work and consider whether MCSP may be complete for NP under more powerful reductions. We also show that NP-completeness of MCSP allows for amplification of circuit complexity. We show the following results.

- If MCSP is NP-complete via many-one reductions, the following circuit complexity amplification result holds: If NP $\cap$ co-NP requires $2^{n^{\Omega(1)}}$-size circuits, then E^{NP} requires $2^{\Omega(n)}$-size circuits.
- If MCSP is NP-complete under truth-table reductions, then EXP $\neq$ NP $\cap$ SIZE(2^{n^ϵ}) for some $\epsilon > 0$ and EXP $\neq$ ZPP. This result extends to polylog Turing reductions.

1 Introduction

Many natural NP problems are known to be NP-complete. Ladner's theorem [14] tells us that if P is different from NP, then there are NP-intermediate problems: problems that are in NP, not in P, but also not NP-complete. The examples arising out of Ladner's theorem come from diagonalization and are not natural. A canonical candidate example of a natural NP-intermediate problem is the Graph Isomorphism (GI) problem. If GI is NP-complete, then the polynomial-time hierarchy collapses [17, 8]. This gives very strong evidence that GI is unlikely to be NP-complete.

In this paper, we study another candidate example of NP-intermediate problem—the Minimum Circuit Size Problem (MCSP): given the truth-table of a Boolean function f and a number k, does there exist a Boolean circuit of size at most k computing f? We do not have a good understanding of the complexity of this fundamental problem. Clearly MCSP is in NP. It is believed that MCSP is not in P, however we do not know whether it is NP-complete. Unlike the GI problem, we do not currently have complexity-theoretic evidence that MCSP is not NP-complete. If MCSP is not NP-complete, then it implies that P does not equal NP. Previous work has considered whether MCSP (and its variants) is complete via various notions of reductions [13, 16, 4]. These works establish that if MCSP is complete, then certain consequences happen for complexity classes –

*This research was supported in part by NSF grant 0917417.
†This research was supported in part by NSF grants 0916797 and 1421163.

some plausible, some not. These results indicate that settling whether MCSP is NP-complete is outside the scope of current techniques.

Kabanets and Cai [13] showed that if MCSP is NP-complete under *natural* reductions, then (i) $\mathrm{E} \not\subseteq \mathrm{P/poly}$ and (ii) E requires $2^{\Omega(n)}$-size circuits or NP can be solved in subexponential time. On the contrary, they obtained a host of interesting consequences under the assumption that MCSP is in P. For example, they showed that if MCSP is in P, then Blum integers can be factored in time 2^{n^ϵ}. They also related this assumption to *circuit complexity amplification*. They showed that the assumption "MCSP is in P" yields the following: if there exists a language in E with circuit complexity $2^{\delta n}$ (for some $\delta > 0$), then there is a language in E with essentially maximal circuit complexity (close to $2^n/n$). Such a circuit complexity amplification result, even though believable, is surprising.

Recently Murray and Williams [16] showed that MCSP is not complete under *local reductions* where each output bit of the reduction can be computed in time $n^{1/2-\epsilon}$. They also showed that if MCSP is complete via AC_0 reductions then E has languages with circuit complexity $2^{\delta n}$. For the case of polynomial-time reductions, they showed that if MCSP is NP-complete via polynomial-time reductions,then $\mathrm{EXP} \not\subseteq \mathrm{P/poly}$ or $\mathrm{EXP} = \mathrm{NEXP}$. In particular, it follows that $\mathrm{EXP} \neq \mathrm{NP} \cap \mathrm{P/poly}$ and $\mathrm{EXP} \neq \mathrm{ZPP}$. Even though we strongly believe that statements such as EXP differs from ZPP and E has high circuit complexity hold, we are far away from proving them. These results explain the difficult of proving a NP-completeness result for MCSP (if it is indeed NP-complete). Allender, Holden, and Kabanets studied the oracle version of MCSP problem. Given the truth-table of a Boolean function f and a parameter k, does f admit circuits of size k that have access to an oracle A? They showed that $\mathrm{MCSP}^{\mathrm{QBF}}$ is unlikely to be hard for various complexity classes under reductions that are more restrictive than polynomial-time reductions. For example, they showed that if $\mathrm{MCSP}^{\mathrm{QBF}}$ is hard for NP under logspace reductions, then nondeterministic exponential time (NEXP) collapses to P$SPACE$.

The known results that concern NP-completeness of MCSP and oracle versions of MCSP can be summarized as follows: For the case of "restricted" reductions, these problems are unlikely to complete NP-hard; Establishing NP-hardness under polynomial-time reductions would resolve a few major open problems in complexity theory.

Our Results. In this paper we obtain additional results regarding NP-completeness of MCSP under polynomial-time reductions. Our first result relates completeness of MCSP with *circuit complexity amplification*. If a complexity class $\mathcal{C}$ requires superpolynomial-size circuits, then can we amplify this hardness to show that a complexity class $\mathcal{D}$ requires circuits of much higher size? Ideally, we want $\mathcal{D}$ to be the same as $\mathcal{C}$. However, we do not know how to prove such results even when the class $\mathcal{D}$ is a superclass of $\mathcal{C}$. Buresh-Oppenheim and Santhanam [10] showed that if the *nondeterministic* circuit complexity of E is $2^{\delta n}$, then $\mathrm{E}/O(n)$ has languages with maximal circuit complexity. They established a negative result that shows that known proof techniques can not amplify deterministic circuit complexity. As our first result, we show that NP-completeness of MCSP implies certain circuit complexity amplification. Assume that MCSP is NP-complete and suppose further that we have a moderately exponential-size ($2^{n^{\Omega(1)}}$) circuit-size lower bound for $\mathrm{NP} \cap \text{co-NP}$. We show that this hardness can be amplified into a strongly exponential ($2^{\Omega(n)}$) circuit-size lower bound for E^{NP}. Admittedly, the gap between these classes is large, but we know of no unconditional method of doing this. This result should be contrasted with the previously mentioned result of Kabanets and Cai. Interestingly, both the statements "MCSP is in P" and

"MCSP is NP-complete" imply that circuit complexity amplification is possible.

The statement "If NP $\cap$ co-NP requires circuits of size $2^{n^{\Omega(1)}}$, then E^{NP} requires circuits of size $2^{\Omega(n)}$" can also be viewed as an *upward separation result*—if a complexity class is hard, then a higher complexity class is much harder. In general, such upward separation results are rare. For example, we do not know if NP differs from P, then NEXP differs from EXP. Thus NP-completeness of MCSP implies an upward separation result.

Next we consider the completeness of MCSP under reductions that are more general than polynomial-time, many-one reductions. We do not know whether GI polynomial-time, many-one reduces to MCSP, however Allender and Das [3] showed that GI reduces to MCSP if we allow *probabilistic, Turing reductions*. These reductions use randomness and are allowed to ask multiple (adaptive) queries. This result suggests that allowing more general reductions to MCSP yields more power. This raises the following question: Is it possible to establish completeness of MCSP under more general reductions? In our second result, we show that it would be difficult to establish completeness MCSP under *truth-table/nonadaptive reductions*. We show that if MCSP is NP-complete under truth-table reductions, then EXP $\neq$ NP $\cap$ SIZE(2^{n^ϵ}) for some $\epsilon > 0$. This is an extension of Murray and Williams' result. We first provide an alternate proof of Murray and Williams result for the case of many-one reductions using different techniques, and extend this proof to the case of truth-table reductions. Our alternate proof could be of independent interest. We also note that the proof of Murray and Williams can also be extended to the case of truth-table reductions. Additionally, our proof extends to polylog-Turing reductions. It is worth noting that our results for truth-table completeness and polylog-Turing completeness are not directly comparable.

Techniques. Our approaches are based on ideas from honest reductions. A many-one reduction f is honest if $|f(x)| \geq |x|^\epsilon$ for some $\epsilon > 0$. From early work of Berman and Hartmanis [7], we know that all natural NP-complete problems are complete under honest reductions. Let f be a many-one reduction from L to MCSP. We say that this reduction is *parametric honest* if there is an $\epsilon > 0$ such that for every x, the output $f(x) = \langle y, k\rangle$ satisfies $k > n^\epsilon$. Note that L reduces to MCSP via honest reductions does not imply that L reduces to MCSP via parametric honest reductions. Suppose that MCSP is NP-complete via parametric honest reductions. Consider such a reduction f from 0^* to MCSP. Note that $f(1^n) = \langle y, k\rangle$ is a negative instance of MCSP, and since $k > n^\epsilon$, we have that the circuit complexity of y (when viewed a truth-table of a boolean function) is at least n^ϵ. Thus we have a polynomial-time algorithm that outputs strings with high circuit complexity which in turn implies that E has high circuit complexity.

We show that under certain plausible hypotheses, NP-completeness of MCSP implies that there exist parametric honest reductions to MCSP. We combine this with the above observation to obtain our results.

- In our first result, the hypothesis is that NP $\cap$ co-NP requires moderately exponential-size circuits. We show this implies MCSP is complete under parametric honest, SNP (strong nondeterministic) reductions. Informally, a reduction is an SNP reduction if it is computable by a NP $\cap$ co-NP machine. This yields the strong exponential-size circuit lower bound for E^{NP}.

- In our second result, the hypothesis is that there is a hard tally language T in NP. Using this hypothesis, we show that if MCSP is truth-table complete, then there is a truth-table reduction from T to MCSP where at least one query is parametric, honest. This yields a

circuit lower bound for E. We build on this to show that truth-table completeness of MCSP implies a separation of EXP from ZPP.

Even though we know that all known NP-complete sets are complete via honest reductions, we do not whether this is true for all NP-complete sets. In recent years there have been a few results that show that, under some believable hypotheses, every NP-complete set is complete via honest reductions whose resource bounds are slightly larger than polynomial [2, 12, 11, 9]. We use ideas from these works to show that if MCSP is complete, then it is complete via parametric honest reductions (under certain hypotheses).

This paper is organized as follows. Section 2 covers preliminaries and previous work. Our results on truth-table completeness of MCSP are in Section 3. The consequences for amplification of circuit complexity are in section 4.

2 Preliminaries

For the standard notation and notions in complexity theory we refer the reader to [6]. Our alphabet is $\Sigma = \{0,1\}$ and we use Σ^n to denote all binary strings of length n. Given an n bit string (where n is a power of 2) x, we view x as the truth-table of a function, denoted f_x, from $\{0,1\}^{\log n}$ to $\{0,1\}$. Given a function $f : \Sigma^n \to \{0,1\}$, we use $CC(f)$ to denote the size of the smallest Boolean circuit that computes f. For a string x (whose length is a power of two), we use $CC(x)$ to denote $CC(f_x)$. For a language L, $L(x) = 1$ if $x \in L$; otherwise $L(x) = 0$. Given a language L, we use $L_n : \Sigma^n \to \{0,1\}$ to denote the characteristic function of L restricted to strings of length n. We say that $CC(L) > s(n)$ if there exist infinitely many n for which $CC(L_n) > s(n)$. We say that $CC(L) > s(n)$ a.e. if $CC(L_n) > s(n)$ for all but finitely many n. A complexity class $\mathcal{C}$ does not have circuits of size $s(n)$ if there exists $L \in \mathcal{C}$ such that $CC(L) > s(n)$.

Definition 2.1. MCSP is the set of tuples $\langle x, k\rangle$ such that $CC(f_x)$ is at most k.

An instance $\langle x, k\rangle$ of MCSP is called *ℓ-large* is if $k \geq \ell$. In our proofs we use *strong nondeterministic reductions* [1, 15] and approximable sets. We define these notions.

Definition 2.2. A language A reduces to a language B via strong, nondeterministic, polynomial-time reductions (SNP reductions), if there exists a polynomial-time bounded, nondeterministic Turing machine N such that for every $x \in \Sigma^*$, the following conditions hold:

- Every path of $N(x)$ outputs a string y or outputs a special symbol $\perp$.
- If $x \in A$, then every output y of $N(x)$ belongs to B; if $x \notin A$, every output y of $N(x)$ does not belong to B.

Definition 2.3. A language L is $t(n)$-time 2-approximable [5], if there is exists a function f computable in time $O(t(n))$ such that for every pair of strings x and y, $f(x,y) \neq L(x)L(y)$. A language L is io-lengthwise, $t(n)$-time, 2-approximable if there exists a $O(t(n))$-time computable function f such that for infinitely many n for every pair of strings x and y of length n, $f(x,y) \neq L(x)L(y)$.

It is known that every polynomial-time, 2-approximable set has polynomial-size circuits [5]. This proof can be extended.

Theorem 2.4 ([5]). *If a language L is* io*-length wise, $t(n)$-time 2-approximable, then for infinitely many n, $CC(L_n) \leq O(t^2(n))$.*

Definition 2.5. A language A is polynomial-time, truth-table reduces to a language B if there exist a pair of polynomial-time computable functions f and g such that for every x, $A(x) = f(x, B(q_1), \cdots, B(q_m))$, where $g(x) = \langle q_1, \cdots q_m \rangle$.

Definition 2.6. Let L be a language that polynomial-time, many-one reduces to MCSP. We say that L reduces to MCSP via *parametric, honest reduction* if there exists an $\epsilon > 0$, and a polynomial-time, many-one reduction f from L to MCSP if $f(x)$ is $|x|^\epsilon$ large for every $x \in \Sigma^*$.

The above definition can be extended to the case of SNP reductions.

Definition 2.7. We say that a language L reduces to MCSP via *parametric, honest,* SNP *reduction*, if there exists an $\epsilon > 0$ and a polynomial-time nondeterministic machine N such that L SNP reduces to MCSP via N and every output of $N(x)$, that does not equal $\perp$, is $|x|^\epsilon$-large.

The following observations are proved using the standard techniques.

Observation 2.8. *Suppose that there is a* $\mathrm{P}/O(\log n)$ *algorithm A and an $\epsilon > 0$ such that for all but infinitely many n the output of $A(1^n)$ has circuit complexity greater than n^ϵ. Then there is a language L is* E *such that $CC(L) \geq 2^{\delta n}$ for some $\delta > 0$.*

Observation 2.9. *Suppose that there is a non-deterministic, polynomial-time algorithm A and an $\epsilon > 0$ such that for infinitely many n the following holds: Every output of $A(1^n)$ that does not equal $\perp$, has circuit complexity greater than n^ϵ. Then there is a language L is* E^{NP} *such that $CC(L) \geq 2^{\delta n}$ for some $\delta > 0$.*

3 Amplification of Circuit Complexity

In this section we show that completeness of MCSP implies that circuit complexity can be amplified.

Theorem 3.1. *Assume that* MCSP *is* NP*-complete via polynomial-time, many-one reductions. If there exists a language L in* $\mathrm{NP} \cap$ co-NP *such that for some $\epsilon > 0$, $CC(L) \geq 2^{n^\epsilon}$ a.e., then there exists $\delta > 0$ such that then* E^{NP} *does not have circuits of size $2^{\delta n}$.*

Before we proceed with proof, we give a brief overview of the proof. Gu, Hitchcock, and Pavan [11] showed that if NP can not be solved in sub-exponential time (at all lengths), then every NP-complete set is complete via P/poly, length-increasing reductions. We borrow ideas from this work. Let L be a hard language in $\mathrm{NP} \cap$ co-NP whose circuit complexity is high.

Our first step in the proof is that under this hypothesis, completeness of MCSP implies completeness via parametric, honest reductions. For this we define an intermediate language I that embeds both SAT and L. This language consists of tuples $\langle x, y, z \rangle$ so that $Maj(x \in L, y \in \mathrm{SAT}, z \in L)$ is 1. This language is clearly in NP. Consider a reduction f from I to MCSP. Suppose that $f(\langle x, y, z \rangle) = \langle u, k \rangle$. If k is small (less than n^δ), then we can solve the membership of $\langle u, k \rangle$ in time roughly 2^{n^δ}. If $\langle u, k \rangle$ is in MCSP then $\langle x, y, z \rangle \in I$. Thus it must be the case that at least one of x or z are in L. Thus $L(x)L(z)$ cannot be equal to 00. Thus in time 2^{n^δ} time we learned some information about the collective membership of x and z in L (even though this information

does help us solve individual memberships of x and z in L). Now suppose that for every pair x and z, we have that $f(\langle x, y, z\rangle)$ is small, then for every pair of strings x and z we can exclude one possibility for $L(x)L(z)$ in time 2^{n^δ}. This implies that L must be io-2-approximable and thus L has low enough circuit complexity (by Theorem 2.4). From this we conclude that for at least one pair x and z, $f(\langle x, y, z\rangle)$ is large. Using this we build a parametric, honest reduction from SAT to I. We now proceed with details.

Proof. Let L be a language in NP$\cap$co-NP that does not have 2^{n^ϵ}-size circuits at almost all lengths. We will first prove that if MCSP is NP-complete, then MCSP is complete via parametric, honest, SNP reductions.

Lemma 3.2. *Suppose that there exists a language in* NP$\cap$co-NP *that requires* 2^{n^ϵ}*-size circuits a.e. for some* $\epsilon > 0$*. If* MCSP *is* NP*-complete, then* MCSP *is complete via parametric, honest,* SNP *reductions.*

Proof. Let L be the hard language in NP $\cap$ co-NP that requires 2^{n^ϵ}-size circuits. We define the following intermediate language I. Let $\delta = \epsilon/2$.

$$I = \{\langle x, y, z\rangle \mid Maj\{x \in L, y \in \text{SAT}, z \in L\} = 1, |x| = |z| = |y|^{1/\delta}\}.$$

Clearly I is in NP. Let f be a many-one reduction from I to MCSP. Our goal is to exhibit a large query, SNP reduction from SAT to MCSP. For this we will first show that for every string y of length n^δ, there exist $x \in L$, $z \notin L$ (of length n) such that $f(\langle x, y, z\rangle)$ is n^δ-large.

Let

$$T_n = \{\langle x, z\rangle \mid |x| = |z| = n, L(x) \neq L(z), \forall y \in \Sigma^{n^\delta}, f(\langle x, y, z\rangle) \text{ is not } n^\delta\text{-large}\}.$$

We will next claim that T_n must be the empty set for all but finitely many n.

Claim 3.2.1. *For all but finitely many* n*,* $T_n = \emptyset$*.*

Proof. We prove by contradiction. Suppose that there exist infinitely many n at which T_n is not empty. We show that for infinitely many lengths n, $CC(L_n) \leq 2^{n^\epsilon}$, which contradicts the hardness of L. This contradiction is achieved by showing that L is io-lengthwise, 2-approximable in time 2^{n^ϵ}. Consider the following approximator function h:

1. Input: x, z of length n.
2. For every y from Σ^{n^δ} compute $f(\langle x, y, z\rangle)$.
3. If every $f(\langle x, y, z\rangle)$ is n^δ-large, then output 01 and stop.
4. If for some $y \in \Sigma^{n^\delta}$, $f(x, y, z)$ is not n^δ large, compute the membership of $f(x, y, z)$ in MCSP.
5. If $f(x, y, z) \in$ MCSP, then output 00; otherwise output 11.

Let n be a length at which $T_n \neq \emptyset$. We show that for every x, z of length n the output of the above algorithm does not equal $L(x)L(z)$. Since T_n is not empty, there exists a $y \in \{0,1\}^{n^\delta}$ such that $f(\langle x, y, z\rangle$ is not n^δ large. Thus the above algorithm reaches Step 4. If $f(x, y, z) \in$ MCSP, then the algorithm outputs 00. In this case, since f is a many-one reduction from I to MCSP, $\langle x, y, z\rangle \in I$. Thus at least one of x or z must belong to L. Thus $L(x)L(z) \neq 00$. Similarly, if

$f(x,y,z) \notin$ MCSP, then $\langle x,y,z\rangle \notin I$, and this implies that at least one of x or z does not belong to L. Thus the output of the algorithm 11 does not equal $L(x)L(z)$.

We now bound the running time of the above algorithm. Step 2 takes $O(2^{n^\delta} \cdot poly(n))$ time. Consider Step 4. This step is performed only when $f(x,y,z) = \langle u,k\rangle$ is not n^δ-large. Thus $k \leq n^\delta$. Thus to decide the membership of $\langle u,k\rangle$, we have to cycle through all circuits of size $\leq n^\delta$ and check if any of them computes the function f_u. This step takes $2^{O(\log nn^\delta)}$ time. Thus the total time taken by the above algorithm is bounded by $2^{O(\log nn^\delta)}$.

If T_n is not empty for infinitely many n, the language L is io-lengthwise, 2-approximable in time $2^{O(\log nn^\delta)}$. Thus by Theorem 2.4, $CC(L_n) \leq 2^{n^\epsilon}$ for infinitely many n as $\delta \leq \epsilon/2$.This is a contradiction. □

We will now return to the proof of Lemma 3.2. Thus $T_n \neq \emptyset$ for all but finitely many lengths n. This suggests the following SNP reduction from SAT to MCSP: On an input y of length n, guess a string $x \in L$ and a string $z \notin L$ of lengths $n^{1/\delta}$ and compute $f(\langle x,y,z\rangle) = \langle u,k\rangle$. If $k < n$ output $\perp$, otherwise output $\langle u,k\rangle$. By claim 3.2.1, for all but finitely many n, $T_{n^{1/\delta}}$ is not empty. Thus for all but finitely many n, there exist strings x and z of length $n^{1/\delta}$ such that $x \in L$, $z \notin L$ and $f(\langle x,y,z\rangle)$ is n-large for every y of length n. Since L is in NP $\cap$ co-NP, at least one path of the reduction guesses such x and z and the output along this path is n-large. Thus MCSP is complete via parametric, honest, SNP-reductions.

□

We now complete the proof of Theorem 3.1. Let $T = 0^*$, by Lemma 3.2, there is a SNP reduction f from T to MCSP that is parametric honest. Let $x_n = \langle y_n,k\rangle$ be the lexicographically smallest output produced by f on input 1^n. Since $1^n \notin T$, we have that $\langle y_n,k\rangle \notin$ MCSP and $k \geq n^\delta$. Thus $CC(y_n) \geq n^\delta$. By Observation 2.9, it follows that E^{NP} has high circuit complexity.

□

4 Truth-Table Completeness

Our results in this section are based on the following hypothesis.

> *Hypothesis H:* There exists an $\epsilon > 0$ and a tally language in NP that cannot be solved deterministically in time 2^{n^ϵ}.

Before moving on to more powerful reductions, we begin by examining the case of many-one reducibility.

Theorem 4.1. *Assume that Hypothesis H holds. If MCSP is NP-complete via polynomial-time, many-one reductions, then there exists a $\delta > 0$ such that* $\mathrm{E} \not\subseteq \mathrm{SIZE}(2^{\delta n})$.

Proof. Assume that MCSP is NP-complete and let T be the hard tally language that is not in DTIME(2^{n^ϵ}). Let f be a many-one reduction from T to MCSP. Fix $\delta < \epsilon$.

Claim 4.1.1. *There exist infinitely many n such that $0^n \notin T$ and $f(0^n)$ is n^δ-large.*

Proof. Suppose not. For all but finitely many n at which $0^n \notin T$ we have that $f(0^n)$ is not δ-large. This means that if $f(0^n)$ is n^δ-large for some n, then $0^n \in T$. This suggests the following algorithm for T: On input 0^n, compute $f(0^n) = \langle x,k\rangle$. If $f(0^n)$ is n^δ-large, then accept 0^n. Otherwise, we

have that $k < n^\delta$. Now cycle through all circuits of size at most k to determine the membership of $\langle x, k\rangle$ in MCSP. This lets us decide the membership of 0^n in T.

The time taken for this procedure is dominated by the time taken to cycle through all circuits of size at most k. Since there are at most $2^{O(\log n n^\delta)}$ such circuits, the language T can be decided in time less than 2^{n^ϵ}. This is a contradiction. □

Now consider the following polynomial-time algorithm that on input 0^n computes $f(0^n) = \langle x, k\rangle$ and outputs x. Note that for infinitely many n, this algorithm outputs the truth-table of a function whose circuit complexity is at least n^δ. This implies that there is a language in E whose circuit complexity is $2^{\delta n}$. □

The above theorem yields the following corollary, similar to Murray and Williams [16]. The consequence here EXP $\neq$ NP$\cap$SIZE(2^{n^ϵ}) is stronger than EXP $\neq$ NP$\cap$P/poly obtained by Murray and Williams, though we note that their proof may be adapted to obtain this as well.

Corollary 4.2. *If* MCSP *is* NP*-complete, then* EXP $\neq$ ZPP *and* EXP $\neq$ NP $\cap$ SIZE(2^{n^ϵ}) *for some* $\epsilon > 0$.

Proof. Assume that MCSP is NP-complete. We consider two cases.

- If Hypothesis H does not hold, then NP $\neq$ EXP as EXP has tally languages that can not be solved in time 2^n. Since ZPP is a subset of NP, EXP $\neq$ ZPP.
- If Hypothesis H holds, then by the above theorem, E does not have circuits of size $2^{\delta n}$ (at infinitely many lengths). This implies that ZPP can be derandomized to P at infinitely many length and which in turn implies that EXP $\neq$ ZPP. Finally note that, if E does not have circuits of size $2^{\delta n}$, then EXP does not have circuits of size 2^{n^ϵ} for some $\epsilon > 0$.

In both cases, the conclusion of the corollary is true. □

Next we extend the above theorem (and its proof) to the case of truth-table reductions. We note that the proof of Murray and Williams can also be extended to the case of truth-table reductions.

Theorem 4.3. *Assume that the hypothesis H holds. If* MCSP *is truth-table complete for* NP*,* E *does not have circuits of size* $2^{\delta n}$ *for some* $\delta > 0$.

Proof. Let T be the hard tally language in NP and let f a truth-table reduction from T to MCSP. On input 0^n, let $q_1^n, \cdots q_m^n$ be the queries produced by f. Fix $\delta < \epsilon$. We first claim that at least one of the queries produced is large and is a negative instance of MCSP.

Claim 4.3.1. *There exist infinitely many* n *for which there exists* i, $1 \leq i \leq m$, *such that* q_i^n *is* n^δ*-large and does not belong to* MCSP.

Proof. Suppose not. For all but finitely many n, the following holds. For every i, $1 \leq i \leq m$, either q_i^n is not n^δ-large or $q_i^n \in$ MCSP. This suggests the following algorithm to decide T:

On input 0^n, run the reduction f and produce queries $q_1^n, \cdots q_m^n$.

- If q_i^n is not n^δ-large then use a brute-force search algorithm to decide the membership of q_i^n in MCSP.
- If q_i^n is n^δ-large, then $q_i^n \in$ MCSP.

Use all answers to the queries decide the membership of 0^n in T.

Clearly, the algorithm correctly decides T. The most expensive step of the algorithm is to decide the membership of q_i^n in MCSP using the brute-force algorithm. Note that we run the brute-force algorithm only when q_i^n is not n^δ-large. Thus the time taken for this step is $2^{O(\log nn^\delta)}$. Thus the total time taken by the algorithm is $O(m2^{O(\log nn^\delta)})$. Since m is polynomial in n and $\delta < \epsilon$, this is bounded by 2^{n^ϵ}. This contradicts our hypothesis. □

Using the above claim, we show that there is an efficient algorithm (with a logarithmic amount of advice) that outputs strings with high circuit complexity.

Claim 4.3.2. *There is a* $\mathrm{P}/O(\log n)$ *algorithm* $\mathcal{A}$ *that on input* 0^n *outputs a string* x_n *and for infinitely many* n, $CC(x_n) \geq n^\delta$.

Proof. Let n^ℓ bound the run time of the truth-table reduction from T to MCSP. The algorithm on input 0^n gets a tuple $\langle b, r\rangle$ as advice where b is a bit and $r < n^\ell$. The bit b is set to 1 if at least one of q_i^n is n^δ-large and does not belong to MCSP; otherwise b is set to 0. When b is 1, then the number r indicates the first index i, $1 \leq i \leq m$, for which q_i^n is n^δ-large and does not belong to MCSP. When b equals 0, r is set to 0. Note that the length of the advice is $O(\log n)$.

The algorithm on input 0^n first looks at the advice bit b. If b is 0, then it outputs 0^n. Otherwise it runs the reduction from T to MCSP to produce queries $q_1^n, \cdots q_m^n$. Let $q_r^n = \langle x_n, k\rangle$. The algorithm outputs x_n.

By Claim 4.3.1, there exist infinitely many n at which at least one of q_i^n is n^δ-large and does not belong to MCSP. At every such length the above algorithm (on correct advice bits) outputs a string x_n for which $CC(x_n) > n^\delta$. □

By Observation 2.8, there is a language in E that requires circuits of size $2^{\rho n}$ for some $\rho > 0$. This completes the proof of the theorem. □

As before we have the following corollary.

Corollary 4.4. *If* MCSP *is truth-table complete for* NP*, then* EXP $\neq$ ZPP *and* EXP $\neq$ NP $\cap$ SIZE(2^{n^ϵ}) *for some* $\epsilon > 0$.

Using similar ideas we can prove the following.

Theorem 4.5. *If* MCSP *is polylog-Turing complete for* NP*, then* EXP $\neq$ ZPP *and* EXP $\neq$ NP $\cap$ SIZE(2^{n^ϵ}) *for some* $\epsilon > 0$.

References

[1] L. Adleman and K. Manders. Reducibility, randomness, and intractability. In *Proc. 9th ACM Symp. Theory of Computing*, pages 151–163, 1977.

[2] M. Agrawal. Pseudo-random generators and structure of complete degrees. In *17th Annual IEEE Conference on Computational Complexity*, pages 139–145, 2002.

[3] E. Allender and B. Das. Zero knowledge and circuit minimization. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, pages 25–32, 2014.

[4] E. Allender, D. Holden, and V. Kabanets. The minimum oracle circuit size problem. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 21–33, 2015.

[5] A. Amir, R. Beigel, and W. Gasarch. Some connections between bounded query classes and non-uniform complexity. *Inf. Comput.*, 186(1):104–139, 2003.

[6] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[7] L. Berman and H. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.*, 6:305–322, 1977.

[8] R. Boppanna, J. Hastad, and S. Zachos. Does Co-NP have short interactive proofs? *Information Processing Letters*, 25(2):125–132, 1987.

[9] H. Buhrman, B. Hescott, S. Homer, and L. Torenvliet. Non-uniform reductions. *Theory Comput. Syst.*, 47(2):317–341, 2010.

[10] J. Bursesh-Oppenheim and R. Santhanam. Making hard problems harder. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 73–87, 2006.

[11] X. Gu, J. M. Hitchcock, and A. Pavan. Collapsing and separating completeness notions under average-case and worst-case hypotheses. *Theory of Computing Systems*, 51(2):248–265, 2011.

[12] J. M. Hitchcock and A. Pavan. Comparing reductions to NP-complete sets. *Information and Computation*, 205(5):694–706, 2007.

[13] V. Kabanets and J. Y. Cai. Circuit minimization problem. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79, 2000.

[14] R. Ladner. On the structure of polynomial time reducibility. *J. Assoc. Comput. Mach.*, 22:155–171, 1975.

[15] T. Long. Strong nondeterministic polynomial-time reducibilities. *Theor. Comput. Sci.*, 21:1–25, 1982.

[16] C. Murray and R. Williams. On the (non) np-hardness of computing circuit complexity. In *Computational Complexity Conference*, 2015.

[17] U. Schöning. The power of counting. In A. Selman, editor, *Complexity Theory Retrospective*, pages 204–223. Springer-Verlag, 1990.