The Complexity of Complexity
New Insights on the (Non)hardness of Circuit Minimization and Related Problems

Eric Allender
Rutgers University

Joint work with
Shuichi Hirara (U. Tokyo)

RodFest, January 5, 2017
NP-Intermediate Problems
Ladner (1975) showed that intermediate problems exist unless $P=NP$. But these problems are “unnatural”.
NP-Intermediate Problems

Some prominent candidates were studied early on…
NP-Intermediate Problems

Some prominent candidates were studied early on…

…but subsequently “moved”.

Eric Allender: The Complexity of Complexity
NP-Intermediate Problems

MCSP is still here:
The Context

- The **Minimum Circuit Size Problem (MCSP)** = \{ (f,i) : f is the truth-table of a function that has a circuit of size ≤ i \}.

- That is, MCSP is the overgraph of the “circuit size” complexity measure SIZE(f).
The Context

- The **Minimum Circuit Size Problem (MCSP)** = \{(f, i) : f is the truth-table of a function that has a circuit of size \( \leq i \)\}.

- In NP, but not known (or widely believed) to be NP-complete. [Kabanets, Cai], [Murray, Williams], [A, Holden, Kabanets], [Hirahara, Watanabe]

- Lots of reasons to believe it’s not in P.
MCSP is more like a family of problems, than a single problem.

For instance “size” could mean “# of wires” or “# of gates”, or “# of bits to describe the circuit”, etc.

None of these is known to be reducible to any other – but all can stand in for “MCSP”.

One more such variant:

\[ O_{KT} = \{(x,i) : KT(x) \leq i\} \]
Complexity Classes

EXPSPACE

NEX

PSPACE

NP

P

BPP

P/poly
More Complexity Classes

- BPP
- RP
- ZPP
- P
- DET
- L
- Uniform AC⁰
- AC⁰
- P/poly
Yet More Context

- Factoring and Discrete Log are in $\text{ZPP}^{\text{MCSP}}$.
- Graph Isomorphism is in $\text{RP}^{\text{MCSP}}$.
- Every promise problem in SZK is in $(\text{Promise}) \text{BPP}^{\text{MCSP}}$.
- But is MCSP NP-hard?

Statistical Zero Knowledge
SZK is contained in $\text{NP/poly} \cap \text{coNP/poly}$
Yet More Context

- Factoring and Discrete Log are in $\text{ZPP}^{\text{MCSP}}$.
- Graph Isomorphism is in $\text{RP}^{\text{MCSP}}$.
- Every promise problem in $\text{SZK}$ is in $\text{(Promise) BPP}^{\text{MCSP}}$.

- But is $\text{MCSP}$ NP-hard?

All these hardness results use probabilistic reductions.
Consequences of Hardness

- From [MW],[AHK], we know:
  - MCSP NP-hard under $\leq_p^m$ implies $\text{EXP} \neq \text{ZPP}$.
  - MCSP NP-hard under $\leq^{\log}_m$ implies $\text{PSPACE} \neq \text{ZPP}$.
  - MCSP NP-hard under $\leq^{\text{AC}^0}_m$ implies $\text{BPP} = \text{P}$, $\text{NP}$ not in $\text{P/poly}$, $\text{DSPACEN}(n)$ not in $\text{SIZE}(2^{\varepsilon n})$.
  - MCSP NP-hard under $\text{TIME}(\sqrt[3]{n})$ local reductions implies FALSE.
Local Reductions

Given $i$, can compute $i^{th}$ bit of output in time $t(n)$
Local Reductions

- SAT is NP-complete under $\text{TIME}(\log n)$ local reductions.
- PARITY does not reduce to MCSP under $\text{TIME}(\sqrt[3]{n})$ local reductions.
- How about $\text{NC}^0$ reductions?
  - For MCSP we still don’t know.
  - For $O_{KT}$, we have a theorem:
    - $O_{KT}$ is hard for DET under non-uniform $\text{NC}^0$ reductions.
First Theorem

> $O_{KT}$ is hard for DET under non-uniform $NC^0$ reductions.

> Corollary: $O_{KT}$ is not in $AC^0[p]$ for any prime $p$.

> Circuit lower bounds imply $O_{KT}$ is hard for DET under *uniform* $AC^0$–Turing reductions.

> $O_{KT}$ hard for PARITY under these same uniform reductions implies (similar) circuit lower bounds.
First Theorem

- $O_{KT}$ is hard for DET under non-uniform $NC^0$ reductions.

- Corollary: $O_{KT}$ is not in $AC^0[p]$ for any prime $p$.

- Independently, [Oliveira & Santhanam] have shown that DET reduces to MCSP via non-uniform $TC^0$ reductions. (This still leaves open the question of whether MCSP is in $AC^0[p]$.)
First Theorem

- \( O_{KT} \) is hard for DET under non-uniform \( NC^0 \) reductions.

- Corollary: \( O_K \) and \( O_C \) are hard for DET under non-uniform \( NC^0 \) reductions.

- **Open question:** Is \( O_K \) or \( O_C \) hard for \( P \) or hard for \( \Sigma^0_1 \) under non-uniform \( AC^0 \) or \( NC^0 \leq_m \) reductions?

- Non-uniformity is key. No uniform \( \leq_m \) reduction can reduce anything nontrivial to \( O_K \) or \( O_C \).
First Theorem

- $O_{KT}$ is hard for DET under non-uniform NC$^0$ reductions.

- **Corollary:** $O_K$ and $O_C$ are hard for DET under non-uniform NC$^0$ reductions.

- **Open question:** Is $O_K$ or $O_C$ hard for P or hard for $\Sigma^0_1$ under non-uniform AC$^0$ or NC$^0 \leq_m$ reductions?

- $O_K$ and $O_C$ are hard for $\Sigma^0_1$ under (non-uniform) P/poly-Turing reductions [ABKMR].
Three Bizarre Inclusions

- NEXP is contained in \( \text{NP}^{O_K} \) (for every \( U \)).

- PSPACE is contained in \( \text{P}^{O_K} \) (for every \( U \)).
- BPP is contained in \( \text{P}^{tt \text{O}_K} \) (for every \( U \)).
  - The decidable sets that are in \( \text{P}^{tt \text{O}_K} \) for every \( U \) are in PSPACE. [AFG11]
  - [CDELM] The sets that are in \( \text{P}^{tt \text{O}_K} \) for every \( U \) are decidable.
Three Bizarre Inclusions

- NEXP is contained in $\text{NP}^O_K$ (for every $U$).
  - The sets that are in $\text{NP}^O_K$ for every $U$ are in EXPSPACE.

- PSPACE is contained in $\text{P}^O_K$ (for every $U$).

- BPP is contained in $\text{P}^O_{tt}$ (for every $U$).
  - The sets that are in $\text{P}^O_{tt}$ for every $U$ are in PSPACE.

- Thus there are reasons to care about efficient reductions to $O_K$ and $O_C$. 
Recall: Every promise problem in SZK is in (Promise) $\text{BPP}^{\text{MCSP}}$.

SZK is usually defined in terms of “promise problems”.
Promise Problems

Ordinary decision problems.
Promise Problems

Ordinary decision problems.

Yes No

Yes Don’t Care No

A “solution”
Promise Problems

- A promise problem $(Y,N)$ is NP-hard if SAT reduces to every solution.

- $\text{Gap}_{\epsilon}\text{MCSP}$:
  - $Y = \{(f,s) : \text{SIZE}(f) < s/|f|^{1-\epsilon(|f|)}\}$
  - $N = \{(f,s) : \text{SIZE}(f) > s\}$

- Note: $\text{Gap}_{\epsilon}\text{MCSP}$ becomes easier when $\epsilon$ is smaller.

- If $\epsilon(n) = o(1)$, then $\text{Gap}_{\epsilon}\text{MCSP} \in \text{DTIME}(2^{n^{o(1)}})$.

- Related to Natural Proofs for $\text{SIZE}(2^{o(n)})$. 
The 2\textsuperscript{nd} Theorem

- If cryptographically-secure one-way functions exist, then there is a function $\varepsilon(n) = o(1)$ such that
  - $\text{Gap}_\varepsilon\text{MCSP}$ is not in $\text{P/poly}$, and
  - $\text{Gap}_\varepsilon\text{MCSP}$ is not $\text{NP}$-hard under $\text{P/poly}$-Turing reductions.
The 2\textsuperscript{nd} Theorem

- If cryptographically-secure one-way functions exist, then there is a function $\varepsilon(n) = o(1)$ such that
  - $\text{Gap}_\varepsilon O_{KT}$ is not in $P/poly$, and
  - $\text{Gap}_\varepsilon O_{KT}$ is not $NP$-hard under $P/poly$-Turing reductions.
Three Bizarre Inclusions, Reprise

- NEXP is contained in $NP^{\text{GapO}_K}$, which is contained in EXPSPACE.
- PSPACE is contained in $P^{\text{GapO}_K}$.
- BPP is contained in $P_{\text{tt}}^{\text{GapO}_K}$, which is contained in PSPACE.
- I’m on record as conjecturing $\text{BPP} = P_{\text{tt}}^{\text{OK}}$.
- Might it be easier to show $\text{BPP} = P_{\text{tt}}^{\text{GapO}_K}$?
- In particular, is there an easier way than [CDELM] to show that $P_{\text{tt}}^{\text{GapO}_K}$ is decidable?
1st theorem ($O_{KT}$ hard for DET)

- [Toran] showed that DET $\text{AC}^0$-reduces to the isomorphism problem for rigid graphs.
- [A, Grochow, Moore] showed that rigid graph isomorphism is in $\text{RP}^{O_{KT}}$. We’ll modify that proof.
1st Theorem: Proof

On input \((G_0, G_1)\)

- Randomly pick a bit string \(w=w_1w_2\ldots w_t\).
- Pick random permutations \(\pi_1\ldots \pi_t\).
- Let \(z= w\pi_1(G_{w_1})\pi_2(G_{w_2})\ldots \pi_t(G_{w_t})\)

If \(G_0\) and \(G_1\) are not isomorphic, then \(z\) allows us to reconstruct \(w\) and \(\pi_1\ldots \pi_t\), so that \(z\) has (non-time-bounded) \(K\)-complexity around \(t+ts\) (where \(s = \log n!\)), \text{whp}. Hence \(KT(z) > t+ts\).

Otherwise, \(KT(z)\) is around \(n^2+ts\).
1st Theorem: Proof

- On input \((G_0, G_1)\)
  - Randomly pick a bit string \(w=w_1w_2\ldots w_t\).
  - Pick random permutations \(\pi_1\ldots \pi_t\).
  - Let \(z= w\pi_1(G_{w_1})\pi_2(G_{w_2})\ldots \pi_t(G_{w_t})\).

- The mapping \((G_0, G_1, \pi_1, \pi_2, \ldots \pi_t) \mapsto z\) is computable in \(AC^0\) if the permutations are presented as a list of the form \((i, \pi(i))\).

- But a random bit string is not going to encode a sequence of permutations.
1st Theorem: Proof

- On input \((G_0, G_1)\)
  - Randomly pick a bit string \(w = w_1 w_2 \ldots w_t\).
  - Pick random permutations \(\pi_1 \ldots \pi_t\).
  - Let \(z = w \pi_1(G_{w_1}) \pi_2(G_{w_2}) \ldots \pi_t(G_{w_t})\).

- Solution: There is a logspace-computable function \(f\) that takes \((nt)^{O(1)}\) random bits and with high probability outputs a (nearly)-uniformly random sequence \(\pi_1, \pi_2, \ldots, \pi_t\).
1st Theorem: Proof

- On input \((G_0, G_1)\)
  - Randomly pick a bit string \(w = w_1 w_2 \ldots w_t\).
  - Pick random permutations \(\pi_1 \ldots \pi_t\).
  - Let \(z = w \pi_1(G_{w_1}) \pi_2(G_{w_2}) \ldots \pi_t(G_{w_t})\)

- Thus there is an \(\text{AC}^0\) function \(g\) such that \(g(G_0, G_1, f(r))\) is a probabilistic reduction from Rigid Graph Isomorphism to \(O_{KT}\). Hardwiring in a good choice of \(f(r)\) yields a non-uniform \(\text{AC}^0\) reduction.
1st Theorem: Proof

- On input \((G_0, G_1)\)
  - Randomly pick a bit string \(w=w_1w_2\ldots w_t\).
  - Pick random permutations \(\pi_1 \ldots \pi_t\).
  - Let \(z= w\pi_1(G_{w_1})\pi_2(G_{w_2})\ldots \pi_t(G_{w_t})\)

- Combining this with [Toran], \(O_{K_T}\) is hard for DET under \(AC^0\) reductions.

- Now, by [Agrawal,A,Rudich], \(O_{K_T}\) is also hard for DET under \(NC^0\) reductions.
Congratulations, Rod!

Thanks!