Metacomplexity or the Complexity of Complexity

Eric Allender
Rutgers University

3 Decades of DIMACS, November 22, 2019
Most of the attention in this area focuses on MCSP (the Minimum Circuit Size Problem):

\[ \text{MCSP} = \{(x,i) : \text{Size}(x) \leq i\} \]

- $x$ is a bit string of length $2^n$ (the truth table of a Boolean function).
- \text{Size}(x) is the number of gates in the smallest circuit computing the function $x$. 
Meta-Complexity

- Is complexity theory difficult?
- Is it hard to tell if a function is computationally complex?
- Studying the complexity of MCSP captures this question.
- This self-referential aspect has driven much recent progress on MCSP and related problems.
A Complex Landscape
MCSP is also closely connected to learning theory.

Canonical learning problem: given a list of “yes” instances and “no” instances, find an efficient circuit that “explains” this data, and will attempt to classify new instances.

This is MCSP for partially-specified functions (Partial-MCSP).
Some Basic MCSP Facts

- MCSP is in NP.
- [RR,KC]: If MCSP is in P/poly, then there are no cryptographically-secure one-way functions.
- [AD]: MCSP is hard for SZK under BPP-Turing reductions.
- [MW]: If MCSP is NP-hard under \( \leq^p_m \) reductions, then EXP ≠ ZPP.
- [MW]: MCSP is not hard for PARITY under \( n^{1/3} \)-time local reductions.
Local Reductions

Given $i$, can compute $i^{th}$ bit of output in time $t(n)$
Some Basic MCSP Facts

- MCSP is in NP.
- [RR,KC]: If MCSP is in P/poly, then there are no cryptographically-secure one-way functions.
- [AD]: MCSP is hard for SZK under BPP-Turing reductions.
- [MW]: If MCSP is NP-hard under $\leq^p_m$ reductions, then EXP ≠ ZPP.
- [MW]: MCSP is not hard for PARITY under $n^{1/3}$-time local reductions.
Complexity Classes

- EXP
- PSPACE
- NP
- coNP
- coRP
- RP
- ZPP
- BPP
- P
- P/poly
Some Basic MCSP Facts

- MCSP is in NP.

- [RR,KC]: If MCSP is in P/poly, then there are no cryptographically-secure one-way functions.

- [AD]: MCSP is hard for SZK under BPP-Turing reductions.

- [MW]: If MCSP is NP-hard under $\leq^p_m$ reductions, then $\text{EXP} \neq \text{ZPP}$.

- [MW]: MCSP is not hard for PARITY under $n^{1/3}$-time local reductions.
Statistical Zero Knowledge

- SZK is “nearly” inside NP \( \cap \) coNP (and is probably completely inside NP \( \cap \) coNP).
Some Basic MCSP Facts

- MCSP is in NP.
- [RR,KC]: If MCSP is in P/poly, then there are no cryptographically-secure one-way functions.
- [AD]: MCSP is hard for SZK under BPP-Turing reductions.
- [MW]: If MCSP is NP-hard under $\leq^p_m$ reductions, then EXP $\neq$ ZPP.
- [MW]: MCSP is not hard for PARITY under $n^{1/3}$-time local reductions.
Kolmogorov Complexity

K(x) = min \{|d| : U(d)=x\}.

“Invariance:” If U and U’ are two Universal TMs, then K_U(x) ≈ K_{U'}(x).
Kolmogorov Complexity

- $K(x) = \min \{|d| : U(d) = x\}$.

Great for many purposes... but not computable.
Time-bounded Kolmogorov Complexity

A first attempt:

\[ K^{t(n)}(x) = \min \{|d| : U(d) = x \text{ in time } t(|x|) \}. \]

Problem: No invariance!

There are workarounds, but they’re messy.

Still, we’ll need to refer to this notion, especially when \( t(n) = n^k \) for some \( k \), or when \( t(n) = 2^{n^\varepsilon} \) for some \( \varepsilon > 0 \).

Let’s call these \( K^{\text{poly}} \) and \( K^{\text{exp}} \), respectively.
Time-bounded Kolmogorov Complexity

\[ K_t(x) = \min \{ |d| + \log t : U(d) = x \text{ in time } t \}. \]

Great for many purposes… but captures an odd type of circuit size.
Circuit Complexity

- Let $D$ be a circuit of AND and OR gates (with negations at the inputs). $\text{Size}(D) = \# \text{ of wires in } D$.
- $\text{Size}(f) = \min\{\text{Size}(D) : D \text{ computes } f\}$
- We may allow oracle gates for a set $A$, along with AND and OR gates.
- $\text{Size}^A(f) = \min\{\text{Size}(D) : D^A \text{ computes } f\}$
What is an Oracle Gate?

An oracle gate for oracle $B$ is a piece of hardware with $k$ wires coming in (for some $k$). If those wires take on the value $x$, then the gate outputs 1 if $x$ is in $B$, and 0 otherwise.
Time-Bounded Kolmogorov Complexity

» Levin’s definition:

» \( K_t(x) = \min \{|d| + \log t : U(d) = x \text{ in time } t \} \).

» …but captures an odd type of circuit size.

» Let \( A \) be complete for \( E = \text{Dtime}(2^{O(n)}) \).

» Then \( K_t(x) \approx \text{Size}^A(x) \).

» This turns out to be useful in showing that computing \( K_t \) is complete for \( \text{EXP} \).
Time-Bounded Kolmogorov Complexity

- Levin’s definition:
  \[ K_t(x) = \min\{|d| + \log t : U(d) = x \text{ in time } t \}. \]
- Why \( \log t \)?
  - This gives an optimal search order for NP search problems.
  - Adding \( t \) instead of \( \log t \) would give every string complexity \( \geq |x| \).
- …So let’s look at how to make the run-time be much smaller.
Revised Kolmogorov Complexity

- \( K(x) = \min\{|d| : \text{for all } i \leq |x| + 1, U(d,i,b) = 1 \text{ iff } b \text{ is the } i\text{-th bit of } x\} \) (where bit \# i+1 of x is *).
  - This is identical to the original definition.

- \( K_t(x) = \min\{|d|+\log t : \text{for all } i \leq |x| + 1, U(d,i,b) = 1 \text{ iff } b \text{ is the } i\text{-th bit of } x, \text{ in time } t\} \).
  - The new and old definitions are within \( O(\log |x|) \) of each other.

- Define \( KT(x) = \min\{|d|+t : \text{for all } i \leq |x| + 1, U(d,i,b) = 1 \text{ iff } b \text{ is the } i\text{-th bit of } x, \text{ in time } t\} \).
Kolmogorov Complexity is Circuit Complexity

- $KT(x) \approx \text{Size}(x)$.
- $K(x) \approx KT^H \approx \text{Size}^H(x)$.
- $Kt(x) \approx KT^E \approx \text{Size}^E(x)$. 
Kolmogorov Complexity is Circuit Complexity

- $KT(x) \approx \text{Size}(x)$.
- $K(x) \approx KT^H \approx \text{Size}^H(x)$.
- $Kt(x) \approx KT^E \approx \text{Size}^E(x)$.
- $KS(x) = \min\{|d|+s : U(d) = x, \text{ in space } s\} \approx KT^{QBF}(x) \approx \text{Size}^{QBF}(x)$.
- Let's review what is known about the complexity of computing $KT$, $KS$, and $Kt$ (and more generally $KT^A$ for various $A$).
The Minimum Circuit Size Problem (and friends)

- **MCSP** = \{ (x,i) : Size(x) \leq i \}
- **MKTP** = \{ (x,i) : KT(x) \leq i \}
- **MKtP** = \{ (x,i) : Kt(x) \leq i \}
- **MK^{poly}P** = \{ (x,i) : K^{poly}(x) \leq i \}
- **MK^{exp}P** = \{ (x,i) : K^{exp}(x) \leq i \}, etc.
Known Hardness Results

- MKtP and $\text{MK}^{\text{expP}}$ are complete for EXP under P/poly-Turing reductions (and NP-Turing reductions).

- MKSP is complete for PSPACE under ZPP-Turing reductions.

- MKTP and $\text{MK}^{\text{polyP}}$ (and MCSP) are hard for SZK under BPP-Turing reductions.

- The hardness results above are all proved using the connection to pseudorandom function generators.
Pseudorandom Generators

\[ G_f \]

\[
\text{seed} \rightarrow \text{PseudoRandom bits } b_1, b_2, \ldots
\]

[HILL]: For any poly-time \( f \), we can build a generator \( G_f \).

If \( T \) is a test that accepts a smaller fraction of \( G_f \)-pseudorandom strings than uniformly random strings, then we can use \( T \) to invert \( f \) whp.
Pseudorandom Generators

$G_f$

seed $\rightarrow$ PseudoRandom bits $b_1, b_2, \ldots$

The output of $G_f$ has small time-bounded K-complexity.
Pseudorandom Generators

The output of $G_f$ has small time-bounded K-complexity.

$$KT(x) \approx \text{Circuit.size}(x).$$
Pseudorandom Generators

The output of $G_f$ has small time-bounded K-complexity.

$$KT(x) \approx \text{Circuit.size}(x).$$

Most $x$ require very large circuits.
Pseudorandom Generators

The output of $G_f$ has small time-bounded $K$-complexity. 

$$KT(x) \approx \text{Circuit.size}(x).$$

Most $x$ require very large circuits. 

MKTP gives us a great test $T$ to distinguish random and pseudorandom strings.

seed $\rightarrow$ PseudoRandom bits $b_1, b_2, \ldots$
More Recent Developments

- There are too many new developments to mention them all, even briefly. Thus I’ll mention only a few highlights.

- In order to tell a coherent story, I’ll be suppressing some important but distracting details. (That is, I’ll be lying just a bit.)
NP-completeness

- [ILO]: Partial-MCSP is NP-complete under $\leq_p^m$ reductions.
- This can also be phrased in terms of Partial-MKTP.
- Ker-I Ko showed in 1990 that there are oracles relative to which Partial-MK$^{\text{poly}}_{\text{P}}$ is not NP-complete.
- This highlights a significant difference between $K^{\text{poly}}$ and KT.
NP-completeness

Many important functions produce more than one bit of output. How about the question of determining the size of the smallest circuit computing a function \( \{0,1\}^n \rightarrow \{0,1\}^m \)?

[ILO] Multi-MCSP is NP-complete under randomized reductions.
Masek proved in the 1970s that it is NP-complete to find the size of the smallest depth-2 formula for a given truth table.

Extending this to larger depth has been open for decades. (Some weaker hardness results were known [AHMPS’06].)

Ilango has now shown that finding the size of the smallest depth-d formula is NP-hard under randomized reductions.

This is still wide open for constant-depth circuits.
EXP completeness

- Recall: MKtP and MK^{exp}P are complete for EXP under P/poly-Turing reductions (and NP-Turing reductions).

- [Hirahara ’19]: MK^{exp}P is complete for EXP under ZPP-Turing reductions, and MK^{exp}P is not in P.

- These are both still open for MKtP!

- [Hirahara ’19] also shows that and MK^{t(n)}P is not in P for any superpolynomial t(n).
MCSP and One-Way Functions

- MCSP($2^{\varepsilon n}$) is the set \(\{x : \text{Size}(x) \leq 2^{\varepsilon n}\}\).

- Note that this problem is “easy on average” because only a small fraction of \(x\) instances are in the set. (Just say “No”.)

- A more appropriate notion of “easy on average” is to have an algorithm provide answers in \(\{0,1,?\}\), where the \(\{0,1\}\) answers are always correct, and the algorithm seldom gives a “?” answer.
MCSP and One-Way Functions

- MCSP($2^\varepsilon n$) is the set \{x : Size(x) \leq 2^\varepsilon n\}.

- Santhanam (2019) introduces a conjecture that implies the following are equivalent:
  - Cryptographically-Secure one-way functions exist.
  - MCSP($2^\varepsilon n$) is hard on average.
  - MKTP($n^\varepsilon$) is hard on average.

- Would imply that we don’t live in “Pessiland” (a world in which problems are hard on average, but there are no one-way functions).
Worst-case-to-Average-case

- GapMCSP is the problem of approximating circuit size.
- Hirahara (2018) shows that GapMCSP has no BPP solution iff MCSP(2^{\varepsilon n}) is hard on average.
- This holds also for KT complexity.
- A related result holds for K^{poly} complexity.
- If MCSP is NP-hard, then this would suggest that we don’t live in Heuristica (a world in which P < NP but all NP problems are easy on average).
More recent developments

- Most of the hardness results mentioned hold also for GapMCSP, GapMKTP, GapMKtP, etc.

- That is, the arguments using pseudorandom generators work if we can distinguish between strings of “very high” complexity and “very low” complexity.

- But some results are known only for a “small Gap” for MKTP, and are open for MCSP:
  - \( \text{DET} \leq_{m}^{\text{AC}^0} \text{MKTP} \)
  - Graph Isomorphism is in \( \text{ZPP}^{\text{MKTP}} \).
More recent developments

- Most of the hardness results mentioned hold also for GapMCSP, GapMKTP, GapMKtP, etc.

- That is, the arguments using pseudorandom generators work if we can distinguish between strings of “very high” complexity and “very low” complexity.

- But some results are known only for a “small Gap” for MKTP, and are open for MCSP:
  - $\text{DET} \leq_{\text{AC}^0}^{m} \text{MKTP}$
  - Is there a way to “boost the Gap”?
No Boosting

- [AIV]: GapMCSP and GapMKTP are not hard for PARITY under AC$^0$ reductions.
Lower Bounds and Frontiers

- Frustrating history of Complexity Theory:
- Some nearly-exponential bounds for limited constant-depth families of circuits, but …
- The best AND-OR formula size lower bounds are of size about $n^3$.
- For other models (branching programs, general formulas) the best is only about $n^2$.
- Other lower bounds for “streaming” models (limited memory, one-way access).
For various MCSP variants, lower bounds in these limited models are known that are (nearly) as strong as for any problems in NP [CKLM].

And new “magnification” techniques show that very slight improvements to these MCSP bounds (or proving bounds for MCSP on “streaming” models) would resolve major open questions (including P < NP) [OS,OPS,MMW,CJW].

Some of these bounds seem achievable!
Summary

➢ The slow pace of progress in complexity has been the source of some pessimism about resolving the big open questions.

➢ But the study of MCSP and related problems has led to new approaches that hold some promise of doing an end-run around the barriers that had seemed to block progress.

➢ New developments are coming at a fast pace! It promises to be an exciting ride.

➢ Thanks, DIMACS, for all you’ve done thus far!