Minimum Circuit Size, Graph Isomorphism, and Related Problems

Andrew Morgan
University of Wisconsin–Madison
November 1st, 2018

Based on work with E. Allender, J. Grochow, D. van Melkebeek, and C. Moore
Minimum Circuit Size

$$\text{MCSP} = \{(x, \theta) : x \text{ has circuit complexity at most } \theta\}$$

How hard is MCSP?
How hard is MCSP?

Some known reductions:

- **Factoring** $\in \text{ZPP}^{\text{MCSP}}$
  
  [Allender–Buhrman–Koucký–van Melkebeek–Ronneburger]

- **DiscreteLog** $\in \text{ZPP}^{\text{MCSP}}$
  
  [Allender–Buhrman–Koucký–van Melkebeek–Ronneburger, Rudow]

- **GI** $\in \text{RP}^{\text{MCSP}}$
  
  [Allender–Das]

- **SZK** $\subseteq \text{BPP}^{\text{MCSP}}$
  
  [Allender–Das]

where

GI = graph isomorphism

SZK = problems with statistical zero knowledge protocols

Can replace MCSP by $M_\mu P$ for any complexity measure $\mu$

 polynomially related to circuit size
Describe a string $x$ by a program $p$ so that $p(i) = i$-th bit of $x$

$\text{KT}(x) = \text{smallest } |p| + T$, where

- $p$ describes $x$
- $p$ runs in at most $T$ steps for all $i$

$\text{MKTP} = \{(x, \theta) : \text{KT}(x) \leq \theta\}$

Time-bounded Turing machines with advice $\cong$ Circuits

$\implies$ KT polynomially-related to circuit complexity
How hard is $M_{\mu}P$?

Some known reductions:

- Factoring $\in \text{ZPP}^{M_{\mu}P}$
- DiscreteLog $\in \text{ZPP}^{M_{\mu}P}$
- GI $\in \text{RP}^{M_{\mu}P}$
- SZK $\subseteq \text{BPP}^{M_{\mu}P}$

where
GI = graph isomorphism
SZK = problems with statistical zero knowledge protocols
$M_{\mu}P = \text{MCSP}, \text{MKTP}, \ldots$

Eliminate error in GI and SZK reductions?
A zero-error reduction

Theorem. GI ∈ ZPP^{MKTP}

Fundamentally different reduction from before

Extends to any ‘explicit’ isomorphism problem, including several where the best known algorithms are still exponential

 Doesn’t (yet) work for MCSP
How do the old reductions work?

Hinge on $M\mu P$ breaking PRGs

PRG from any one-way function [Håstad–Impagliazzo–Levin–Luby]

**Inversion Lemma.** There is a poly-time randomized Turing machine $M$ using oracle access to $M\mu P$ so that the following holds. For any circuit $C$, if $\sigma \sim \{0,1\}^n$,

$$\Pr[C(\tau) = C(\sigma)] \geq \frac{1}{\text{poly}(|C|)}$$

where $\tau = M(C, C(\sigma))$

[Allender–Buhrman–Koucký–van Melkebeek–Ronneburger]

Example: Fix a graph $G$. Let $C$ map a permutation $\sigma$ to $\sigma(G)$.

$M$ inverts $C$: if $\sigma(G)$ is a random permutation of $G$, then $M(C, \sigma(G))$ finds $\tau$ s.t. $\tau(G) = \sigma(G)$ with good probability
Example: GI in $\text{RP}^\text{MCSP}$

**Theorem.** $\text{GI} \in \text{RP}^{\text{M} \mu \text{P}}$ [Allender–Das]

Given $G_0 \cong G_1$, use $M$ to find an isomorphism

Let $C(\sigma) = \sigma(G_0)$ where $\sigma \sim S_n$

$M$ inverts $C$: given random $\sigma(G_0)$, $M$ finds $\tau$ with $\tau(G_0) = \sigma(G_0)$

$G_0 \cong G_1$ implies that $\sigma(G_1)$ is distributed the same as $\sigma(G_0)$

So $M(C, \sigma(G_1))$ finds $\tau$ with $\tau(G_0) = \sigma(G_1)$

$\implies \text{GI} \in \text{RP}^{\text{M} \mu \text{P}}$
Eliminating error?

Similar results:

- Factoring $\in \text{ZPP}^{M\mu P}$
- DiscreteLog $\in \text{ZPP}^{M\mu P}$
- GI $\in \text{RP}^{M\mu P}$
- SZK $\subseteq \text{BPP}^{M\mu P}$

How to eliminate error?

$M\mu P$ is only used to *generate* witnesses, which are then checked in deterministic polynomial time.

Thus, showing GI $\in \text{coRP}^{M\mu P}$ using a similar approach implicitly requires GI $\in \text{coNP}$, *i.e.*, NP-witnesses for *non* isomorphism.

*Approach uses MKTP to help with verification*
A zero-error reduction

Theorem. \( GI \in ZPP^{MKTP} \)

Nonisomorphism has \( NP^{MKTP} \) witnesses.

Key idea: \( KT \) complexity is a good estimator for the entropy of samplable distributions.
Graph Isomorphism in $\mathsf{ZPP}^\mathsf{MKTP}$
Graph Isomorphism

$GI = \text{decide whether two given graphs } (G_0, G_1) \text{ are isomorphic}$

$\text{Aut}(G) = \text{group of automorphisms of } G$

Number of distinct permutations of $G = n!/|\text{Aut}(G)|$

To show $GI \in \text{ZPP}^{\text{MKTP}}$, suffices to show $GI \in \text{coRP}^{\text{MKTP}}$, i.e., to witness non-isomorphism
Recall: $KT(x) = \text{smallest } |p| + T \text{ where } p \text{ describes } x \text{ in time } T$

Intuition for bounding $KT(x)$: describe a string $x$ by a program $p$ taking advice $\alpha$ so that $p^\alpha(i) = i$-th bit of $x$

$KT(x)$ is smallest $|p| + |\alpha| + T$ where

- $p$ with advice $\alpha$ describes $x$
- $p$ runs in at most $T$ steps for all $i$
Examples:

1. $\text{KT}(0^n) = \text{polylog}(n)$
   
   Store $n$ in advice, define $p(i)$ to output 0 if $i \leq n$, and end-of-string otherwise

2. $G =$ adjacency matrix of a graph
   
   $\text{KT}(G) \leq \binom{n}{2} + \text{polylog}(n)$

3. Let $y = t$ copies of $G$
   
   $\text{KT}(y) \leq \text{KT}(G) + \text{polylog}(nt)$

4. Let $y =$ sequence of $t$ numbers from $\{5, 10, 10^{300}, -46\}$
   
   $O(1)$ bits to describe the set, plus $2t$ bits to describe the sequence given the set
   
   $\text{KT}(y) \leq 2t + \text{polylog}(t)$
Let $G_0, G_1$ be rigid graphs, i.e., no non-trivial automorphisms.

Key fact: if $G_0 \cong G_1$, there are $n!$ distinct graphs among permutations of $G_0$ and $G_1$; if $G_0 \not\cong G_1$, there are $2(n!)$. 

Consider sampling $r \sim \{0, 1\}$ and $\pi \sim S_n$ uniformly, and outputting the adjacency matrix of $\pi(G_r)$.

- If $G_0 \cong G_1$, this has entropy $s = \log(n!)$
- If $G_0 \not\cong G_1$, this has entropy $s + 1$

Main idea: use KT-complexity of a random sample to estimate the entropy.
Witnessing nonisomorphism: rigid graphs

Let $y = \pi(G_r), \pi \sim S_n, r \sim \{0, 1\}$.

Hope: $\text{KT}(y)$ is *typically near* the entropy, *never* much larger

\[
\begin{align*}
G_0 &\cong G_1 \\
G_0 &\not\cong G_1
\end{align*}
\]

where $s = \log(n!)$

Then $\text{KT}(y) > \theta$ is a witness of nonisomorphism.
Witnessing nonisomorphism: rigid graphs

Let \( y = \pi_1(G_{r_1})\pi_2(G_{r_2}) \cdots \pi_t(G_{r_t}), \pi_i \sim S_n, r_i \sim \{0, 1\} \).

Truth: \( KT(y)/t \) is typically near the entropy, never much larger.

\[ s \quad \theta \quad s + 1 \]

\[ G_0 \cong G_1 \]

\[ G_0 \not\cong G_1 \]

where \( s = \log(n!) \)

Then \( KT(y)/t > \theta \) is a witness of nonisomorphism.
Bounding KT in isomorphic case

Let \( y = \pi_1(G_{r_1})\pi_2(G_{r_2}) \ldots \pi_t(G_{r_t}) \). Goal: \( \text{KT}(y) \ll ts + t \).

Since \( G_0 \cong G_1 \), rewrite \( y = \tau_1(G_0)\tau_2(G_0) \ldots \tau_t(G_0) \).

Describe \( y \) as

- **Fixed data:** \( n, t \), adjacency matrix of \( G_0 \)
- **Per-sample data:** \( \tau_1, \ldots, \tau_t \)
- **Decoding algo:** to output \( j \)-th bit of \( y \), look up appropriate \( \tau_i \) and compute \( \tau_i(G_0) \)

Suppose each \( \tau_i \) can be encoded into \( s \) bits:

\[
\text{KT}(y) < O(1) + \underbrace{\text{poly}(n, \log t)}_{|p|} + ts + \underbrace{\text{poly}(n, \log t)}_{|\alpha|} + \underbrace{\text{poly}(n, \log t)}_T \\
= ts + \text{poly}(n, \log t) \ll ts + t \ (t \text{ large})
\]
**Rigid graphs: Isomorphic case**

**Lehmer Code.** There is an indexing of $S_n$ by the numbers $1, \ldots, n!$ so that the $i$-th permutation can be decoded from the binary representation of $i$ in time $\text{poly}(n)$.

Naïve conversion to binary: $\text{KT}(y) < t \lceil s \rceil + \text{poly}(n, \log t)$

$\ll ts + t$?

Blocking trick: amortize encoding overhead across samples

Yields for some $\delta > 0$, $\text{KT}(y) \leq ts + t^{1-\delta} \text{poly}(n)$,

i.e., $\text{KT}(y)/t \leq s + \text{poly}(n)/t^\delta$
Let \( y = \pi_1(G_{r_1})\pi_2(G_{r_2}) \cdots \pi_t(G_{r_t}) \).

If \( G_0 \cong G_1 \), then \( \frac{KT(y)}{t} \leq s + o(1) \) always holds.

If \( G_0 \not\cong G_1 \), then as \( y \) is \( t \) independent samples from a distribution of entropy \( s + 1 \), \( \frac{KT(y)}{t} \geq s + 1 - o(1) \) holds w.h.p.

\[ \implies \text{coRP}^{\text{MKTP}} \text{ algorithm for GI on rigid graphs} \]
Assume for simplicity that there are as many distinct permutations of $G_0$ as of $G_1$.

Let $s$ be entropy in random permutation of $G_i$: $\log(n!/|\text{Aut}(G_i)|)$

Sample $y = \pi_1(G_{r_1}) \cdots \pi_t(G_{r_t})$, hope $\text{KT}(y)/t$ looks the same:

If $G_0 \not\approx G_1$, $\text{KT}(y)/t > s + 1 - o(1)$ w.h.p.

If $G_0 \approx G_1$, $y$ has entropy $ts$, hope a similar encoding shows $\text{KT}(y)/t \leq s + o(1)$. 
Assume for simplicity that there are as many distinct permutations of $G_0$ as of $G_1$.

Let $s$ be entropy in random permutation of $G_i$: $\log(n!/|\text{Aut}(G_i)|)$

Sample $y = \pi_1(G_{r_1}) \cdots \pi_t(G_{r_t})$, hope $KT(y)/t$ looks the same:

Two complications:

- Encoding distinct permutations of $G_0$ as numbers $1, \ldots, n!$ is too expensive
- Knowing $\theta$ requires knowing $|\text{Aut}(G_i)|$
Indexing the various permutations of a non-rigid graph $G$ as numbers $1, \ldots, n!$ is too expensive.

Need to use numbers $1, \ldots, N$ where $N = n!/|\text{Aut}(G)|$.

Such a specific encoding exists, but will see a more general-purpose substitute soon.
General graphs: computing $\theta$

It suffices to give a probably-approximately-correct overestimator (PAC overestimator) for $\theta$:

$$\tilde{\theta} = \frac{KT(y)}{t}, \ G_0 \cong G_1$$

$$KT(y)/t, \ G_0 \not\cong G_1$$

Equivalently, it suffices to give a PAC underestimator for $\log |\text{Aut}(G_i)|$, since $\theta = (\log n! - \log |\text{Aut}(G_i)|) + 1/2$
Claim. There is an efficient randomized algorithm using MKTP to PAC underestimate $\log |\text{Aut}(G)|$ when given $G$.

Proof. Recall that there is a deterministic algorithm using an oracle for GI that computes generators for $\text{Aut}(G)$.

Plug in an existing $\text{RP}^{\text{MKTP}}$ algorithm for the oracle: this gives us generators for a group $A$ with $A = \text{Aut}(G)$ w.h.p.

Prune generators of $A$ not in $\text{Aut}(G)$ $\implies A \leq \text{Aut}(G)$

$|A|$ can be computed efficiently from its generators. Output $\log |A|$.
General graphs: Recap

\[ y = \pi_1(G_{r_1})\pi_2(G_{r_2}) \cdots \pi_t(G_{r_t}) \]

\[ s = \log n!/|\text{Aut}(G_i)| \]

\[ \widetilde{\theta} \]

\[ KT(y)/t, G_0 \cong G_1 \]

\[ KT(y)/t, G_0 \not\cong G_1 \]

Witness of nonisomorphism: \( KT(y)/t > \widetilde{\theta} \)

**Theorem.** \( GI \in ZPP^{MKTP} \)
Generic Encoding Lemma
We saw that for any rigid graph $G$, the $n!$ distinct permutations of $G$ can be encoded as integers $1, \ldots, n!$.

This can be extended to general graphs, but still involves heavy use of the structure of the symmetric group.

What about other groups?

Is algebraic structure necessary?
Encoding outputs of samplable distributions

Turns out: can encode the outcomes of any samplable distribution.

Flatter distributions $\implies$ better encodings.

Rare events are hard to encode. So assume that all outcomes are somewhat likely.

Define the max-entropy of a distribution to be the smallest $s$ such that all outcomes occur with probability at least $2^{-s}$ (or zero).

**Encoding Lemma.** Let $C$ be a circuit sampling a distribution of max-entropy $s$. There is a circuit $D$ of size $\text{poly}(|C|)$ and, for each outcome $y$, a string $i_y$ of length $s + \log s + O(1)$, s.t. $D(i_y) = y$.

Proof based on hashing
Encoding outputs of samplable distributions

Example: $C$ samples a random permutation of a graph $G$. Then each permutation of $G$ can be decoded from a string of length $s + \log s + O(1)$, where $s = \log(n!/|\text{Aut}(G)|)$

Overhead of $\log s + O(1)$ is worse than $\lceil s \rceil - s$, but can still be amortized out.

End result: for any graph $G$, any $t$ permutations of $G$ has KT-complexity at most $ts + t^{1-\delta}\text{poly}(n)$.

In general, for any circuit $C$ of max-entropy $s$, any $t$ samples from $C$ has KT-complexity at most $ts + t^{1-\delta}\text{poly}(|C|)$.
Entropy Estimator Theorem.
Let $C$ be any circuit sampling a distribution of max-entropy $s_{\text{max}}$ and min-entropy $s_{\text{min}}$. Let $y$ be the concatenation of $t$ independent samples from $C$. Then $KT(y)/t$ is typically between $s_{\text{min}} - o(1)$ and $s_{\text{max}} + o(1)$, and never much larger.

\[ KT(y)/t \]

Nice case: $s_{\text{max}} - s_{\text{min}} = o(1)$. $C$ is “almost flat”. 
Extensions for General Isomorphism Problems
General isomorphism problem

Group $H$ acts on a universe $\Omega$. Given $\omega_0, \omega_1 \in \Omega$, decide whether some $h \in H$ sends $\omega_0$ to $\omega_1$. Assume products, inverses, etc. are efficiently computable.

Example 1: With GI, $H = S_n$, $\Omega =$ labeled $n$-vertex graphs, where $H$ acts by permuting labels. $\omega_0 = G_0$ and $\omega_1 = G_1$. Find a permutation sending $G_0$ to $G_1$.

Example 2: “Matrix Subspace Conjugacy”. $\Omega =$ subspaces of $\mathbb{F}^{n \times n}$, given by a basis (a set of matrices). $H = \text{GL}_n(\mathbb{F})$, acting by conjugation. Given $\{M_1, M_2, \ldots, M_k\}$ and $\{N_1, N_2, \ldots, N_k\}$, is there $T$ so that

$$\text{span}\{ T^{-1}M_1 T, T^{-1}M_2 T, \ldots, T^{-1}M_k T \} = \text{span}\{ N_1, N_2, \ldots, N_k \}?$$
With the entropy estimator theorem in hand, the techniques for GI mostly generalize

Only obstacle: PAC overestimating $\theta$ / underestimating $\log |\text{Aut}(G)|$

Recall, we did this by

1. using a search-to-decision reduction to find generators for $\text{Aut}(G)$, and
2. computing $|\text{Aut}(G)|$ efficiently from its generators

What to do when a search-to-decision reduction isn’t known?
What if the ambient group isn’t $S_n$?
Idea: PAC underestimate log $|\text{Aut}(G)|$ using the entropy estimator theorem again

$\text{Aut}(G)$ efficiently samplable implies amortized KT-complexity

PAC underestimates log $|\text{Aut}(G)|$

Let $y' = \pi_1 \pi_2 \cdots \pi_t$ be $t$ random elements of $\text{Aut}(G)$

Need $\text{Aut}(G)$ efficiently samplable twice:
- Construction of $y'$ in the algorithm
- Analysis of $\text{KT}(y')$

Note: these sampling procedures need not be the same.

Show
- how to use MKTP to sample $\text{Aut}(G)$ with only $G$ on hand
- for every $G$, there is a circuit $C_G$ which samples $\text{Aut}(G)$ uniformly
Recall that MKTP can be used to invert circuits.

**Inversion Lemma.** There is a poly-time randomized Turing machine $M$ using oracle access to $M\mu P$ so that the following holds. For any circuit $C$, if $\sigma \sim \{0,1\}^n$,

$$\Pr[C(\tau) = C(\sigma)] \geq 1/\text{poly}(|C|) \text{ where } \tau = M(C, C(\sigma))$$

[Allender–Buhrman–Koucký–van Melkebeek–Ronneburger]

Let $C$ sample a random permutation $\pi$ and output $\pi(G)$.

Pick $\pi \sim S_n$ at random, let $\tau = M(C, \pi(G))$. With probability $1/\text{poly}(n)$, $\tau(G) = \pi(G)$, so $\tau^{-1} \circ \pi \in \text{Aut}(G)$.

Conditioned on $\pi(G)$, $\tau$ and $\pi$ are independent, so $\tau^{-1} \circ \pi$ is uniform on $\text{Aut}(G)$.
A sequence $\pi_1, \pi_2, \ldots, \pi_k$ of elements of a finite group $\Gamma$ is said to be Erdős–Rényi if the ‘random subproduct’

$$\pi_1^{r_1} \pi_2^{r_2} \cdots \pi_k^{r_k}, \quad r_i \sim \{0, 1\}$$

is distributed approximately uniformly on $\Gamma$. ($s_{\text{max}} - s_{\text{min}} = o(1)$)

Erdős and Rényi showed that every finite group has such a generating set of size $\text{poly}(\log |\Gamma|)$.

With $\Gamma = \text{Aut}(G)$, obtain an ER generating set of size $\text{poly}(n)$.

Hardwire the ER set into a circuit sampling the random subproduct.
Other Applications of KT v. Entropy
More theorems and consequences:

- Any ‘explicit’ iso. problem is in $\text{ZPP}^{\text{MKTP}}$
- New proof of $\text{SZK} \subseteq \text{BPP}^{\text{MKTP}}$
- $\text{DET} \subseteq \text{AC}_0^{\text{MKTP}}$. Consequently, $\text{MKTP} \not\subseteq \text{AC}^0[p]$ [Allender–Hirahara]
- Random-3SAT, Planted Clique $\leq \text{MKTP}$ [Hirahara–Santhanam]
Open Problems
Open problem: SZK?

Techniques essentially boil down to estimating entropy by KT-complexity.

Complete problem for SZK: determine whether a given samplable distribution has entropy at least a given threshold.

Entropy estimator theorem can reproduce $\text{SZK} \subseteq \text{BPP}^{\text{MKTP}}$.

$\text{SZK} \subseteq \text{ZPP}^{\text{MKTP}}$?

Obstacle is devising witnesses for non-flat distributions.

There are distributions with low entropy but supported on every string—nontrivial worst-case bound on KT-complexity is impossible.
Open problem: What about MCSP?

The argument should work for MCSP, but fails for annoying technical reasons. This is true even for rigid-GI.

Use KT complexity in two ways:

- Counting argument: \( KT(y) \gtrsim ts + t \) whp
- Encoding: any string of length \( ts \) has \( KT \lesssim ts \)

For circuits, we get:

- Counting argument: \( \text{CSIZE}(y) \gtrsim (ts + t)/\log(ts + t) \) whp
- Encoding: any string of length \( ts \) has \( \text{CSIZE} \lesssim ts/\log(ts) \)

Low-order terms matter: best known bounds require exponentially-large \( t \) to force gap between the isomorphic and nonisomorphic cases
Open problem: What about MCSP?

Resolving these bounds is only so satisfactory: the answer probably depends on the precise measure of circuit complexity.

Better: boost the gap in entropy between the isomorphic and nonisomorphic cases, then use polynomial relationship between $KT$ and circuit size
Summary

- Reviewed old reductions to MCSP/MKTP based on Inversion Lemma
- Showed a different kind of reduction from GI to MKTP based on estimating entropy by KT complexity
- Stated Encoding Lemma and Entropy Estimator Theorem
- Sketched extension to general isomorphism problems
- Listed other uses of estimating entropy by KT complexity
- Open problems: SZK? MCSP?

Questions?
Thank you!
Random-3SAT reduces to MKTP

Random-3SAT (baby version): Given either

- Satisfiable 3-CNF
- Random 3-CNF with many clauses (likely unsatisfiable)

distinguish between the two cases.

Idea: Existence of a satisfying assignment gives information about the 3-CNF, so it should be easier to describe.

For a satisfying assignment $x$, sample a random clause that $x$ satisfies. Entropy: $\log \binom{n}{3} + \log(7)$.

$\implies$ amortized KT-complexity always bounded

For random 3-CNF: random clause has entropy $\log \binom{n}{3} + \log(8)$

$\implies$ amortized KT-complexity typically high
Planted Clique reduces to MKTP

Planted clique: Given either

- Uniformly random graph
- Uniformly random graph union with a random $k$-clique

distinguish between the two cases.

Uniformly random graph has entropy $\binom{n}{2}$

Random graph with clique has entropy at most $\binom{n}{2} + \log\binom{n}{k} - \binom{k}{2}$

[Hirahara–Santhanam] show KT-complexity closely matches entropy