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

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

How hard is MCSP?

How hard is MCSP?

Some known reductions:

- $\text{Factoring} \in \mathsf{ZPP}^{\text{MCSP}}$ [Allender–Buhrman–Koucký–van Melkebeek–Ronneburger]
- $\text{DiscreteLog} \in \mathsf{ZPP}^{\text{MCSP}}$ [Allender–Buhrman–Koucký–van Melkebeek–Ronneburger, Rudow]
- $\text{GI} \in \mathsf{RP}^{\text{MCSP}}$ [Allender–Das]
- $\mathsf{SZK} \subseteq \mathsf{BPP}^{\text{MCSP}}$ [Allender–Das]

where

GI = graph isomorphism

SZK = problems with statistical zero knowledge protocols

Can replace MCSP by $\text{M}\mu\text{P}$ for any complexity measure μ polynomially related to circuit size

KT Complexity

Describe a string x by a program p so that $p(i) = i$-th bit of x

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

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

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

Time-bounded Turing machines with advice $\cong$ Circuits
$\implies$ KT polynomially-related to circuit complexity

How hard is MμP?

Some known reductions:

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

where

$\mathrm{GI} =$ graph isomorphism

$\mathsf{SZK} =$ problems with statistical zero knowledge protocols

$\mathrm{M}\mu\mathrm{P} = \mathrm{MCSP}, \mathrm{MKTP}, \ldots$

Eliminate error in GI and SZK reductions?

A zero-error reduction

Theorem. $\mathrm{GI} \in \mathsf{ZPP}^{\mathrm{MKTP}}$

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 $\mathrm{M}\mu\mathrm{P}$ 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 $\mathrm{M}\mu\mathrm{P}$ 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]

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

M inverts C: if $\sigma(G)$ is a random permutation of G, then $M(C, \sigma(G))$ finds τ s.t. $\tau(G) = \sigma(G)$ with good probability

Example: GI in RP^MCSP

[Allender–Das]

> **Theorem.** $\mathrm{GI} \in \mathsf{RP}^{\mathrm{M}\mu\mathrm{P}}$ [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 τ 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 τ with $\tau(G_0) = \sigma(G_1)$

$\implies \mathrm{GI} \in \mathsf{RP}^{\mathrm{M}\mu\mathrm{P}}$

Eliminating error?

Similar results:

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

How to eliminate error?

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

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

Approach uses MKTP to help with verification

A zero-error reduction

Theorem. $\mathrm{GI} \in \mathsf{ZPP}^{\mathrm{MKTP}}$

Nonisomorphism has $\mathsf{NP}^{\mathrm{MKTP}}$ witnesses.

Key idea: KT complexity is a good estimator for the entropy of samplable distributions

Graph Isomorphism in ZPP^MKTP

Graph Isomorphism

GI = decide whether two given graphs (G_0, G_1) are isomorphic

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

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

To show $\mathrm{GI} \in \mathsf{ZPP}^{\mathrm{MKTP}}$, suffices to show $\mathrm{GI} \in \mathsf{coRP}^{\mathrm{MKTP}}$, *i.e.*, to witness nonisomorphism

KT Complexity

Recall: $\mathrm{KT}(x) =$ smallest $|p| + T$ where p describes x in time T

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

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

- p with advice α describes x
- p runs in at most T steps for all i

KT Complexity

Examples:

1. $\mathrm{KT}(0^n) = \mathsf{polylog}(n)$
   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
   $\mathrm{KT}(G) \leq \binom{n}{2} + \mathsf{polylog}(n)$
3. Let $y = t$ copies of G
   $\mathrm{KT}(y) \leq \mathrm{KT}(G) + \mathsf{polylog}(nt)$
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
   $\mathrm{KT}(y) \leq 2t + \mathsf{polylog}(t)$

Witnessing nonisomorphism: rigid graphs

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 $G_0 \ncong G_1$, there are $2(n!)$.

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

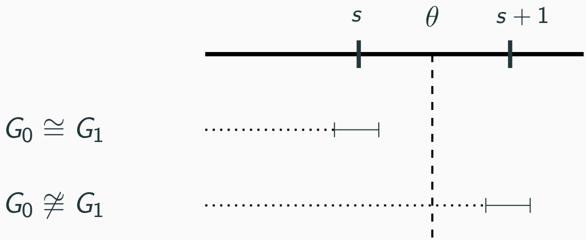
- If $G_0 \cong G_1$, this has entropy $s \doteq \log(n!)$
- If $G_0 \ncong G_1$, this has entropy $s + 1$

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



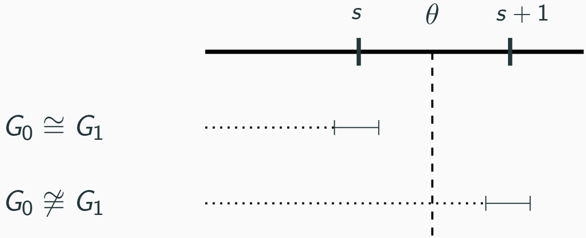
where $s = \log(n!)$

Then $\mathrm{KT}(y) > \theta$ 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: $\mathrm{KT}(y)/t$ is *typically near* the entropy, **never** much larger



where $s = \log(n!)$

Then $\mathrm{KT}(y)/t > \theta$ is a witness of nonisomorphism.

Bounding KT in isomorphic case

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

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

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 τ_i and compute $\tau_i(G_0)$

Suppose each τ_i can be encoded into s bits:

$$\mathrm{KT}(y) < \underbrace{O(1)}_{|p|} + \underbrace{\mathrm{poly}(n, \log t) + ts}_{|\alpha|} + \underbrace{\mathrm{poly}(n, \log t)}_{T}$$
$$= ts + \mathrm{poly}(n, \log t) \ll ts + t \ (t \text{ large})$$

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 poly(n).

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

$\ll ts + t$?

$\not\ll ts + t$

Blocking trick: amortize encoding overhead across samples

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

i.e., $\mathrm{KT}(y)/t \leq s + \mathsf{poly}(n)/t^{\delta}$

Rigid graphs: Recap

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 $\mathrm{KT}(y)/t \leq s + o(1)$ always holds

If $G_0 \ncong G_1$, then as y is t independent samples from a distribution of entropy $s+1$, $\mathrm{KT}(y)/t \geq s + 1 - o(1)$ holds w.h.p.

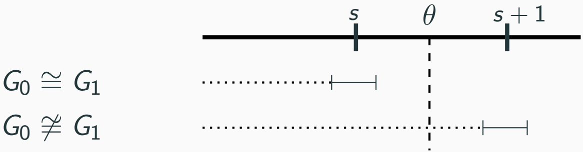
$\implies$ coRP$^{\mathrm{MKTP}}$ algorithm for GI on rigid graphs

General 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: $\log(n!/|\mathrm{Aut}(G_i)|)$

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



If $G_0 \ncong G_1$, $\mathrm{KT}(y)/t > s + 1 - o(1)$ w.h.p.

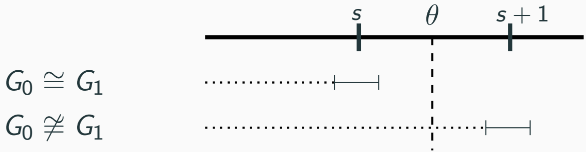
If $G_0 \cong G_1$, y has entropy ts, hope a similar encoding shows $\mathrm{KT}(y)/t \leq s + o(1)$.

General 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: $\log(n!/|\mathrm{Aut}(G_i)|)$

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



Two complications:

- Encoding distinct permutations of G_0 as numbers $1, \ldots, n!$ is too expensive
- Knowing θ requires knowing $|\mathrm{Aut}(G_i)|$

General graphs: encoding permutations of graphs

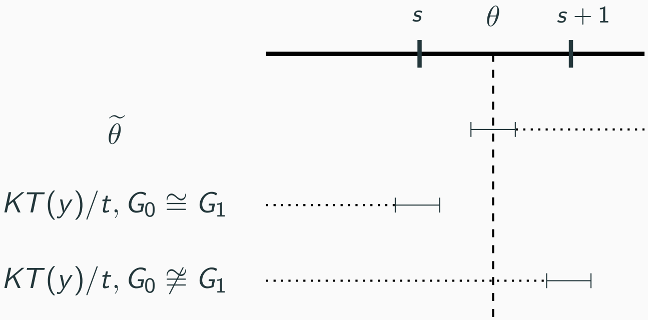
Indexing the various permutations of a non-rigid graph G as numbers $1, \ldots, n!$ is too expensive

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

Such a specific encoding exists, but will see a more general-purpose substitute soon

General graphs: computing θ

It suffices to give a *probably-approximately-correct overestimator* (PAC overestimator) for θ:



Equivalently, it suffices to give a PAC *under*estimator for $\log|\mathrm{Aut}(G_i)|$, since $\theta = (\log n! - \log|\mathrm{Aut}(G_i)|) + 1/2$

General graphs: computing θ

Claim. There is an efficient randomized algorithm using MKTP to PAC underestimate $\log|\mathrm{Aut}(G)|$ when given G.

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

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

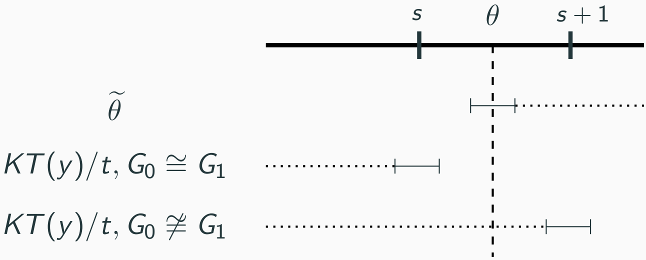
Prune generators of A not in $\mathrm{Aut}(G) \implies A \leq \mathrm{Aut}(G)$

$|A|$ can be computed efficiently from its generators. Output $\log|A|$.

General graphs: Recap

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

$s = \log n!/|\mathrm{Aut}(G_i)|$



Witness of nonisomorphism: $\mathrm{KT}(y)/t > \tilde{\theta}$

Theorem. $\mathrm{GI} \in \mathsf{ZPP}^{\mathrm{MKTP}}$

Generic Encoding Lemma

Encoding outputs of samplable distributions

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 $s = \log(n!/|\mathrm{Aut}(G)|)$

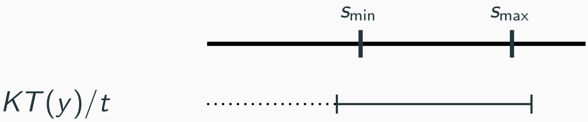
Overhead of $\log s + O(1)$ is worse than $\lceil s \rceil - s$, but can still be amortized out.

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

In general, for any circuit C of max-entropy s, any t samples from C has KT-complexity at most $ts + t^{1-\delta}\mathrm{poly}(|C|)$.

Entropy estimation

Entropy Estimator Theorem.
Let C be any circuit sampling a distribution of max-entropy $s_{\max}$ and min-entropy $s_{\min}$. Let y be the concatenation of t independent samples from C. Then $\mathrm{KT}(y)/t$ is typically between $s_{\min} - o(1)$ and $s_{\max} + o(1)$, and never much larger.



Nice case: $s_{\max} - s_{\min} = o(1)$. C is "almost flat".

Extensions for General Isomorphism Problems

General isomorphism problem

Group H acts on a universe Ω. Given $\omega_0, \omega_1 \in \Omega$, decide whether some $h \in H$ sends ω_0 to ω_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*). $H = \mathsf{GL}_n(\mathbb{F})$, acting by conjugation. Given $\{M_1, M_2, \ldots, M_k\}$ and $\{N_1, N_2, \ldots, N_k\}$, is there T so that $\mathrm{span}\{T^{-1}M_1T, T^{-1}M_2T, \ldots, T^{-1}M_kT\} = \mathrm{span}\{N_1, N_2, \ldots, N_k\}$?

Beyond GI

With the entropy estimator theorem in hand, the techniques for GI mostly generalize

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

Recall, we did this by

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

PAC underestimating log |Aut(G)|

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 $\mathrm{KT}(y')$

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

How to sample Aut(G) with MKTP

Recall that MKTP can be used to invert circuits.

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

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

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

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

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

Conditioned on $\pi(G)$, τ and π are independent, so $\tau^{-1} \circ \pi$ is uniform on $\mathrm{Aut}(G)$.

How to sample Aut(G) with a small circuit

A sequence $\pi_1, \pi_2, \ldots, \pi_k$ of elements of a finite group Γ 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 Γ. $(s_{\max} - s_{\min} = o(1))$

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

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

Hardwire the ER set into a circuit sampling the random subproduct.

Other Applications of KT v. Entropy

KT versus entropy: other applications

More theorems and consequences:

- Any 'explicit' iso. problem is in $\mathsf{ZPP}^{\mathrm{MKTP}}$
- New proof of $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathrm{MKTP}}$
- $\mathsf{DET} \subseteq \mathsf{AC}_0^{\mathrm{MKTP}}$. Consequently, $\mathrm{MKTP} \notin \mathsf{AC}^0[p]$

  [Allender–Hirahara]

- Random-3SAT, Planted Clique $\leq \mathrm{MKTP}$

  [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 SZK $\subseteq$ BPP$^{\mathrm{MKTP}}$

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

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: $\mathrm{KT}(y) \gtrsim ts + t$ whp
- Encoding: any string of length ts has $\mathrm{KT} \lesssim ts$

For circuits, we get:

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

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 $\mathrm{MCSP}/\mathrm{MKTP}$ 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 [Hirahara–Santhanam]

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 [Hirahara–Santhanam]

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