Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 68 (2014)

Zero Knowledge and Circuit Minimization

Eric Allender[1] and Bireswar Das[2]

[1] Department of Computer Science, Rutgers University, USA
allender@cs.rutgers.edu
[2] IIT Gandhinagar, India
bireswar@iitgn.ac.in

Abstract. We show that every problem in the complexity class SZK (Statistical Zero Knowledge) is efficiently reducible to the Minimum Circuit Size Problem (MCSP). In particular Graph Isomorphism lies in $\mathsf{RP}^{\mathsf{MCSP}}$.
This is the first theorem relating the computational power of Graph Isomorphism and MCSP, despite the long history these problems share, as candidate NP-intermediate problems.

1 Introduction

For as long as there has been a theory of NP-completeness, there have been attempts to understand the computational complexity of the following two problems:

- Graph Isomorphism (GI): Given two graphs G and H, determine if there is permutation τ of the vertices of G such that $\tau(G) = H$.
- The Minimum Circuit Size Problem (MCSP): Given a number i and a Boolean function f on n variables, represented by its truth table of size 2^n, determine if f has a circuit of size i. (There are different versions of this problem depending on precisely what measure of "size" one uses (such as counting the number of gates or the number of wires) and on the types of gates that are allowed, etc. For the purposes of this paper, any reasonable choice can be used.)

Cook [Coo71] explicitly considered the graph isomorphism problem and mentioned that he "had not been able" to show that GI is NP-complete. Similarly, it has been reported that Levin's original motivation in defining and studying NP-completeness [Lev73] was in order to understand the complexity of GI [PS03], and that Levin delayed publishing his work because he had hoped to be able to say something about the complexity of MCSP [Lev03]. (Trakhtenbrot has written an informative account, explaining some of the reasons why MCSP held special interest for the mathematical community in Moscow in the 1970s [Tra84].)

For the succeeding four decades, GI and MCSP have been prominent candidates for so-called "NP-Intermediate" status: neither in P nor NP-complete. No connection between the relative complexity of these two problems has been established. Until now.

It is considered highly unlikely that GI is NP-complete. For instance, if the polynomial hierarchy is infinite, then GI is not NP-complete [BHZ87]. Many would conjecture that $\mathsf{GI} \in \mathsf{P}$; Cook mentions this conjecture already in [Coo71]. However this is still very much an open question, and the complexity of GI has been the subject of a great deal of research. We refer the reader to [KST93,AT05] for more details.

ISSN 1433-8092

In contrast, comparatively little was written about MCSP, until Kabanets and Cai revived interest in the problem [KC00], by highlighting its connection to the so-called Natural Proofs barrier to circuit lower bounds [RR97]. Kabanets and Cai provided evidence that MCSP is not in P (or even in P/poly); it is known that $\mathsf{BPP}^{\mathsf{MCSP}}$ contains several problems that cryptographers frequently assume are intractable, including the discrete logarithm, and several lattice-based problems [KC00,ABK$^+$06]. The integer factorization problem even lies in $\mathsf{ZPP}^{\mathsf{MCSP}}$ [ABK$^+$06].

Is MCSP complete for NP? Krajíček discusses this possibility [Kra11], although no evidence is presented to suggest that this is a likely hypothesis. Instead, evidence has been presented to suggest that it will be difficult to reduce SAT to MCSP. Kabanets and Cai define a class of "natural" many-one reductions; after observing that most NP-completeness proofs are "natural" in this sense, they show that any "natural" reduction from SAT to MCSP yields a proof that $\mathsf{EXP} \not\subseteq \mathsf{P/poly}$. Interestingly, Vinodchandran studies a problem called SNCMP, which is similar to MCSP, but defined in terms of strong nondeterministic circuits, instead of deterministic circuits [Var05]. (SNCMP stands for Strong Nondeterministic Circuit Minimization Problem.) Vinodchandran shows that any "natural" reduction from graph isomorphism to SNCMP yields a nondeterministic algorithm for the complement of GI that runs in subexponential time for infinitely many lengths n.

We show that $\mathsf{GI} \in \mathsf{RP}^{\mathsf{MCSP}}$; our proof also shows that $\mathsf{GI} \in \mathsf{RP}^{\mathsf{SNCMP}}$. Thus, although it would be a significant breakthrough to give a "natural" reduction from GI to SNCMP, no such obstacle prevents us from establishing an RP-Turing reduction.

One of the more important results about GI is that GI lies in SZK: the class of problems with statistical zero-knowledge interactive proofs [GMW91]. After giving a direct proof of the inclusion $\mathsf{GI} \in \mathsf{RP}^{\mathsf{MCSP}}$ in Section 3, we give a proof of the inclusion $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathsf{MCSP}}$ in Section 4. We conclude with a discussion of additional directions for research and open questions.

But first, we present the basic connection between MCSP and resource-bounded Kolmogorov complexity, which allows us to use MCSP to invert polynomial-time computable functions.

2 Preliminaries and Technical Lemmas

A small circuit for a Boolean function f on n variables constitutes one form of a short description for the bit string of length 2^n that describes the truth table of f. In fact, as discussed in [ABK$^+$06, Theorem 11], there is a version of time-bounded Kolmogorov complexity (denoted KT) that is roughly equivalent to circuit size. That is, if x is a string of length m representing the truth table of a function f with minimum circuit size s, it holds that

$$\left(\frac{s}{\log m}\right)^{1/4} \leq \mathrm{KT}(x) \leq O(s^2(\log s + \log\log m)).$$

The connection with Kolmogorov complexity is relevant, because of this simple observation: The output of a pseudorandom generator consists of strings with small time-bounded

Kolmogorov complexity. Thus, with an oracle for MCSP, one can take as input a string x and accept iff x has no circuits of size, say, $\sqrt{|x|}$, and thereby ensure that one is accepting a very large fraction of all of the strings of length n (since most x encode functions that require large circuits), and yet accept no strings x such that $\mathrm{KT}(x) \leq n^{\epsilon}$. Such a set is an excellent test to distinguish the uniform distribution from the distribution generated by a pseudorandom generator. Using the tight connection between one-way functions and pseudorandom generators [HILL99], one obtains the following result:

Theorem 1. *[ABK+06, Theorem 45] Let L be a language of polynomial density such that, for some $\epsilon > 0$, for every $x \in L$, $KT(x) \geq |x|^{\epsilon}$. Let $f(y,x)$ be computable uniformly in time polynomial in $|x|$. There exists a polynomial-time probabilistic oracle Turing machine N and polynomial q such that for any n and y*

$$\Pr_{|x|=n,s}[f(y, N^L(y, f(y,x), s)) = f(y,x)] \geq 1/q(n),$$

where x is chosen uniformly at random and s denotes the internal coin flips of N.

Here, "polynomial density" means merely that L contains at least $2^n/n^k$ strings of each length n, for some k. That is, let f_y be a collection of functions indexed by a parameter y, where $f_y(x)$ denotes $f(y,x)$. Then, if one has access to an an oracle L that contains many strings but no strings of small KT-complexity, one can use the probabilistic algorithm N to take as input $f_y(x)$ for a randomly-chosen x, and with non-negligible probability find a $z \in f_y^{-1}(f_y(x))$, that is, a string z such that $f_y(z) = f_y(x)$.

Note that such a set L can be recognized in deterministic polynomial time with an oracle for MCSP, as well as with an oracle for SNCMP. One could also use an oracle for R_{KT}, the KT-random strings: $R_{\mathrm{KT}} = \{x : \mathrm{KT}(x) \geq |x|\}$.

3 Graph Isomorphism and Circuit Size

Theorem 2. $\mathsf{GI} \in \mathsf{RP}^{\mathsf{MCSP}}$.

Proof. We are given as input two graphs G and H, and we wish to determine whether there is an isomorphism from G to H.

Consider the polynomial-time computable function $f(G,\tau)$ that takes as input a graph G on n vertices and a permutation $\tau \in S_n$ and outputs $\tau(G)$. We will use the notation $f_G(\tau)$ to denote $f(G,\tau)$. That is, f_G takes a permutation τ as input, and produces as output the adjacency matrix of the graph obtained by permuting G according to τ. Observe that f_G is uniformly computable in time polynomial in the length of τ.

Thus, by Theorem 1, there is a polynomial-time probabilistic oracle Turing machine N and polynomial q such that for any n and G

$$\Pr_{\tau \in S_n, s}[f_G(N^{\mathsf{MCSP}}(G, f_G(\tau), s)) = f_G(\tau)] \geq 1/q(n),$$

where τ is chosen uniformly at random and s denotes the internal coin flips of N.

Now, given input (G,H) to GI, our $\mathsf{RP}^{\mathsf{MCSP}}$ algorithm does the following for $100q(n)$ independent trials:

1. Pick τ and probabilistic sequence s uniformly at random.
2. Compute $\tau(G)$.
3. Run $N^{\mathsf{MCSP}}(H, \tau(G), s)$ and obtain output π.
4. Report "success" if $\pi(H) = \tau(G)$.

The $\mathsf{RP}^{\mathsf{MCSP}}$ algorithm will accept if at least one of the $100q(n)$ independent trials are successful.

Note that if H and G are not isomorphic, then there is no possibility that the algorithm will succeed.

On the other hand, if H and G are isomorphic, then $\tau(G)$ does appear in the image of f_H. In fact, the distributions $\tau(G)$ and $\tau(H)$ are identical over τ picked uniformly at random. Thus, with probability at least $1/q(n)$ (taken over the choices of τ and s), the algorithm will succeed in any given trial. Thus the expected number of trials that will succeed is at least 100, and hence, by the Chernoff bounds, the probability of having at least one success is well over 1/2. □

Since truth-tables that require large strong nondeterministic circuits also require large deterministic circuits, it is immediate that this reduction can be carried out also with SNCMP.

Corollary 1. $\mathsf{GI} \in \mathsf{RP}^{\mathsf{SNCMP}} \cap \mathsf{RP}^{R_{\mathrm{KT}}}$.

4 Zero Knowledge

In this section, we show $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathsf{MCSP}}$. Note that SZK is best defined not as a class of languages but as a class of "promise problems". A promise problem consists of a pair of disjoint languages (Y, N) where Y consists of "yes-instances" and N consists of "no-instances". Thus the inclusion $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathsf{MCSP}}$ is perhaps more properly stated in terms of "promise" $\mathsf{BPP}^{\mathsf{MCSP}}$. That is, we will show that, for every $(Y, N) \in \mathsf{SZK}$ there is a probabilistic polynomial time oracle Turing machine M with the property that $x \in Y$ implies $M(x)$ accepts with probability at least 2/3 when given oracle MCSP, and $x \in N$ implies $M(x)$ accepts with probability at most 1/3 when given oracle MCSP. M may exhibit any behavior on inputs outside of $N \cup Y$.

It was shown by Chailloux et al. [CCKV08] that SZK is equal to a class that Ben-Or and Gutfreund [BOG03] defined and called $\mathsf{NISZK}|_h$. Importantly for us, Ben-Or and Gutfreund showed that a promise problem they called IID (Image Intersection Density) is complete for $\mathsf{NISZK}|_h$ (and thus, by [CCKV08], IID is also complete for SZK). The yes-instances of IID consist of pairs of circuits (C_0, C_1), each of size n, taking m-bit inputs, such that the distributions $C_0(x)$ and $C_1(x)$ (where x is chosen uniformly at random) have statistical distance at most $1/n^2$. The no-instances of IID consist of pairs of circuits (C_0, C_1) with the property that $\Pr_{|x|=m}[\exists y \; C_1(y) = C_0(x)] < 1/n^2$.

We will not work directly with IID, but rather with a related problem that is shown to be complete for $\mathsf{NISZK}|_h$ in [BOG03, Lemma 20], which is just like IID but with different parameters. Let us call this problem PIID for "polarized IID". The yes-instances of PIID consist of triples (n, D_0, D_1), where each D_i is an m-input circuit of size at most n^k (for some fixed k), such that the distributions $D_0(x)$ and $D_1(x)$ (where x is chosen uniformly at random) have statistical distance at most $1/2^n$. The no-instances of PIID consist of triples (n, D_0, D_1) with the property that $\Pr_{|x|=m}[\exists y\ D_1(y) = D_0(x)] < 1/2^n$.

Furthermore, we need to make use of the fact that we can assume that the length m of the inputs to the circuits D_0 and D_1 may be assumed without loss of generality to be at least n^δ for some fixed $\delta > 0$. This can be accomplished by simply adding dummy input variables. It is easy to check that adding dummy variables to both circuits does not change the statistical difference. Similarly, this does not alter the probability that the output produced by a random input to the first circuit is in the support of the second circuit.

Theorem 3. $\mathsf{SZK} \in \mathsf{BPP}^{\mathsf{MCSP}}$

Proof. It will suffice to show that $\mathsf{PIID} \in \mathsf{BPP}^{\mathsf{MCSP}}$.

Consider the polynomial-time computable function $F(C, x)$ that takes a Boolean circuit C on m-bit inputs, and a string x of length m as input, and outputs $C(x)$. We will use the notation $F_C(x)$ to denote $F(C, x)$. Since the length of x is polynomially-related to the size of C in the instances of PIID that we consider, it follows that F_C is uniformly computable in time polynomial in the length of x.

Thus, by Theorem 1, there is a polynomial-time probabilistic oracle Turing machine N and polynomial q such that for any m and C

$$\Pr_{|x|=m,s}[F_C(N^{\mathsf{MCSP}}(C, F_C(x), s)) = F_C(x)] \geq 1/q(m),$$

where x is chosen uniformly at random and s denotes the internal coin flips of N.

Now, given input (n, D_0, D_1) to PIID, our $\mathsf{BPP}^{\mathsf{MCSP}}$ algorithm does the following for n^ℓ independent trials (for an ℓ to be determined later):

1. Pick m-bit input x and probabilistic sequence s uniformly at random.
2. Compute $z = D_0(x)$.
3. Run $N^{\mathsf{MCSP}}(D_1, z, s)$ and obtain output y.
4. Report "success" if $D_1(y) = z$.

The $\mathsf{BPP}^{\mathsf{MCSP}}$ algorithm will accept if at least $\log n$ of the n^ℓ independent trials are successful.

If (n, D_0, D_1) is a no-instance of PIID, then the probability that any given trial succeeds is at most $1/2^n$. Thus, for all large n the expected number of the n^ℓ trials that will succeed is at most $n^\ell/2^n < 1$. By the Chernoff bounds, the probability that $\log n$ trials will succeed is less than $1/3$.

If (n, D_0, D_1) is a yes-instance of PIID, then $D_0(x)$ and $D_1(x)$ have statistical distance at most $1/2^n$.

Note that

$$\begin{aligned}
&\Pr_{|x|=m,s}[F_{D_1}(N^{\mathsf{MCSP}}(D_1, F_{D_0}(x), s)) = F_{D_0}(x)] \\
&= \sum_z \Pr_{|x|=m,s}[F_{D_1}(N^{\mathsf{MCSP}}(D_1, z, s)) = z | z = F_{D_0}(x)] \Pr[z = F_{D_0}(x)] \\
&= \sum_z \Pr_{|x|=m,s}[F_{D_1}(N^{\mathsf{MCSP}}(D_1, z, s)) = z | z = F_{D_1}(x)] \Pr[z = F_{D_0}(x)]
\end{aligned}$$

Also,

$$\begin{aligned}&\Pr_{|x|=m,s}[F_{D_1}(N^{\mathsf{MCSP}}(D_1, F_{D_1}(x), s)) = F_{D_1}(x)] \\ &= \sum_z \Pr_{|x|=m,s}[F_{D_1}(N^{\mathsf{MCSP}}(D_1, z, s)) = z | z = F_{D_1}(x)] \Pr[z = F_{D_1}(x)]\end{aligned}$$

Thus the difference of these two probabilities is

$$\begin{aligned}&\sum_z \Pr_{|x|=m,s}[F_{D_1}(N^{\mathsf{MCSP}}(D_1, z, s)) = z | z = F_{D_1}(x)] \times \\ &\qquad\qquad (\Pr[z = F_{D_0}(x)] - \Pr[z = F_{D_1}(x)]) \\ &\le \sum_z 1 \cdot (\Pr[z = F_{D_0}(x)] - \Pr[z = F_{D_1}(x)]) \\ &\le 1/2^n\end{aligned}$$

Since $\Pr_{|x|=m,s}[F_{D_1}(N^{\mathsf{MCSP}}(D_1, F_{D_1}(x), s)) = F_{D_1}(x)] > 1/q(m) > 1/q(n^k)$, it follows that each trial has probability at least $1/q(n^k) - 1/2^n$ of success. Thus, the expected number of the n^ℓ trials that will succeed is at least $n^\ell(1/q(n^k) - 1/2^n)$. Picking ℓ so that n^ℓ is enough greater than $q(n^k)$ guarantees that this expected value is at least n. Thus, by the Chernoff bounds the probability that at least $\log n$ trials succeed is greater than $2/3$. □

In the above proof, notice that we obtain one-sided error on those instances (n, D_0, D_1) of PIID where $\Pr_{|x|=m}[\exists y\ D_1(y) = D_0(x)] = 0$, instead of merely being bounded by $1/2^n$. In particular, the promise problem known as $\overline{\mathsf{SD}}^{1,0}$ (consisting of pairs of circuits (D_0, D_1) where, for the `yes`-instances, D_0 and D_1 represent identical distributions, and the `no`-instances have disjoint images) is in $\mathsf{RP}^{\mathsf{MCSP}}$. It was shown in [KMV07] that this problem is complete for the class of problems that have "V-bit" perfect zero knowledge protocols; this class contains most of the problems that are known to have perfect zero-knowledge protocols, including the problems studied in [AD08].

5 Conclusions and Open Problems

We are the first to admit that there appears to be no reason why these results could not have been proved earlier. The techniques involved have been available to researchers for years, and the proofs have much the same flavor as the reductions of factoring, discrete logarithm, and other cryptographic problems to MCSP that were presented in [ABK+06]. Perhaps the only missing ingredient is that the earlier work involved using MCSP (or, equivalently, R_{KT}) to break pseudorandom generators that were constructed from one-way functions that people actually believed *were* cryptographically secure. In contrast, the functions f_G considered here have never seemed like promising candidates to use, in constructing pseudorandom generators.

It is natural to wonder if better reductions are also possible. Is $\mathsf{GI} \in \mathsf{P}^{\mathsf{MCSP}}$? Or in $\mathsf{ZPP}^{\mathsf{MCSP}}$?

Equally temptingly, is it possible to build on these ideas to reduce larger classes to MCSP? The Wikipedia article on "NP-Intermediate Problems" (as of April 10, 2014) says "...MCSP is believed to be NP-complete" [Wik14]. We are unaware of much evidence for this "belief" being very widespread in the complexity theory community, but it is certainly an intriguing possibility.

Alternatively, is it possible to tie MCSP more closely to SZK? For instance, what is the complexity of the promise problem whose yes-instances consist of strings with KT-complexity at most $\sqrt{n}$, and whose no-instances consist of strings with KT-complexity $> n/2$?

Acknowledgments

We acknowledge helpful comments from Lance Fortnow, Valentine Kabanets, Rahul Santhanam, Bruce Kapron, and Salil Vadhan. The first author acknowledges the support of NSF grants CCF-0832787 and CCF-1064785. This work was performed while the second author was a DIMACS postdoctoral fellow at Rutgers University, under support provided by the Indo-US Science and Technology Forum.

References

ABK+06. E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.

AD08. V. Arvind and Bireswar Das. SZK proofs for black-box group problems. *Theory Comput. Syst.*, 43(2):100–117, 2008.

AT05. V. Arvind and J. Torán. Isomorphism testing: Perspective and open problems. *Bulletin of the EATCS*, 86, 2005.

BHZ87. Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

BOG03. Michael Ben-Or and Danny Gutfreund. Trading help for interaction in statistical zero-knowledge proofs. *J. Cryptology*, 16(2):95–116, 2003.

CCKV08. André Chailloux, Dragos Florin Ciocan, Iordanis Kerenidis, and Salil P. Vadhan. Interactive and non-interactive zero knowledge are equivalent in the help model. In *Theory of Cryptography, Fifth Theory of Cryptography Conference (TCC)*, volume 4948 of *Lecture Notes in Computer Science*, pages 501–534. Springer, 2008.

Coo71. S. A. Cook. The complexity of theorem-proving procedures. In *ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 1971.

GMW91. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

HILL99. J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:1364–1396, 1999.

KC00. V. Kabanets and J.-Y. Cai. Circuit minimization problem. In *ACM Symposium on Theory of Computing (STOC)*, pages 73–79, 2000.

KMV07. Bruce M. Kapron, Lior Malka, and Srinivasan Venkatesh. A characterization of non-interactive instance-dependent commitment-schemes (NIC). In *ICALP*, number 4596 in Lecture Notes in Computer Science, pages 328–339. Springer, 2007.

Kra11. Jan Krajíček. *Forcing with Random Variables and Proof Complexity*. Cambridge University Press, 2011.

KST93. Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1993.

Lev73. L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9:265–266, 1973.

Lev03. L. Levin. Personal communication. 2003.

PS03. Sriram Pemmaraju and Steven Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Cambridge University Press, New York, NY, USA, 2003.

RR97. A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:24–35, 1997.

Tra84. B. A. Trakhtenbrot. A survey of Russian approaches to perebor (brute-force searches) algorithms. *IEEE Annals of the History of Computing*, 6(4):384–400, 1984.

Var05. Vinodchandran N. Variyam. Nondeterministic circuit minimization problem and derandomizing Arthur-Merlin games. *Int. J. Found. Comput. Sci.*, 16(6):1297–1308, 2005.

Wik14. Wikipedia. http://en.wikipedia.org/wiki/NP-intermediate, 2014.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de