Information and Computation 256 (2017) 2–8

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Zero knowledge and circuit minimization [☆]

Eric Allender [a,*,1], Bireswar Das [b,2]

[a] Department of Computer Science, Rutgers University, Piscataway, NJ, USA
[b] IIT Gandhinagar, India

ARTICLE INFO

Article history:
Received 5 November 2014
Available online 26 April 2017

Keywords:
Graph Isomorphism
Minimum Circuit Size Problem
NP-intermediate problem
Statistical Zero Knowledge

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 [11] 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 [23] was in order to understand the complexity of GI [25], and that Levin delayed publishing his work because he had hoped to be able to say something about the complexity of MCSP [24]. (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 [28].)

[☆] A preliminary version of this work appeared in Proc. 39th International Symposium on Mathematical Foundations of Computer Science (MFCS '14), Lecture Notes in Computer Science, vol. 8635, 2014, pp. 25–32.

[*] Corresponding author.
E-mail addresses: allender@cs.rutgers.edu (E. Allender), bireswar@iitgn.ac.in (B. Das).
URLs: https://www.cs.rutgers.edu/~allender (E. Allender), http://www.iitgn.ac.in/faculty/comp/bireswar.htm (B. Das).

[1] Supported in part by NSF grants CCF-0832787, CCF-1064785, and CCF-1555409.

[2] Work performed in part while a DIMACS postdoctoral fellow at Rutgers University, under support provided by the Indo-US Science and Technology Forum IUSSTF Fellowship 2013/36.

http://dx.doi.org/10.1016/j.ic.2017.04.004
0890-5401/© 2017 Elsevier Inc. All rights reserved.

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 [7]. Many would conjecture that $\mathsf{GI} \in \mathsf{P}$; Cook mentions this conjecture already in [11]. 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 [22,5] for more details.

In contrast, comparatively little was written about MCSP, until Kabanets and Cai revived interest in the problem [18], by highlighting its connection to the so-called Natural Proofs barrier to circuit lower bounds [26]. 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 [18,2]. The integer factorization problem even lies in $\mathsf{ZPP}^{\mathsf{MCSP}}$ [2]. (More background on complexity classes such as P/poly, BPP, and ZPP can be found in Section 2.)

Is MCSP complete for NP? Krajíček discusses this possibility [21], 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 [30]. (SNCMP stands for Strong Nondeterministic Circuit Minimization Problem. Strong nondeterministic circuits provide a characterization of the complexity class $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. Formally, a strong nondeterministic circuit for inputs of size n has n input gates, and some number of "auxiliary nondeterministic input gates"; in addition to the output gate, there is a second output gate called the *flag*. For any input x, there should be some setting of the auxiliary nondeterministic input gates that causes the flag bit to be turned on. If the flag bit is turned on, then x is accepted iff the output bit is 1 – which means that any two settings of the auxiliary nondeterministic bits that turn on the flag bit must agree on what the output bit is.) 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.

A problem related to MCSP was considered by Ko [20]; Ko studied the set of strings with low time-bounded Kolmogorov complexity, which he called MINKT. (Ko was using a slightly different notion of time-bounded Kolmogorov complexity than the notion that is discussed in Section 2, which is even more closely related to MCSP.) He presented an oracle relative to which MINKT is neither in P nor is NP-complete, even under polynomial-time Turing reducibility.

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 (since this would provide a subexponential-time nondeterministic algorithm[3] for the non-isomorphism problem), no such obstacle prevents us from establishing an RP-Turing reduction. Similarly, our proof implies that $\mathsf{GI} \in \mathsf{RP}^{\mathsf{MINKT}}$.

One of the more important results about GI is that GI lies in SZK: the class of problems with statistical zero-knowledge interactive proofs [14]. 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 $\mathsf{SZK} \subseteq \text{Promise-}\mathsf{BPP}^{\mathsf{MCSP}}$ in Section 4. (We also provide the necessary background regarding "promise problems" at that point.) We conclude with a discussion of additional directions for research and open questions.

But first, in Section 2, 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

We assume that the reader is familiar with elementary computational complexity theory, as presented in standard textbooks such as [1]. Background on P, NP, EXP and all of the other complexity classes that we discuss can be found there.

Here is a quick review of the probabilistic complexity classes that we will be using. A language is in RP if it is accepted by an NP machine with the additional property that, if there is any accepting path at all on input x, then at least half of the $2^{p(|x|)}$ computation paths of length $p(|x|)$ are accepting. Thus $\mathsf{RP} \subseteq \mathsf{NP}$. A language A is in ZPP if both A and its complement are in RP. A probabilistic machine accepting a language $A \in \mathsf{RP}$ will never have an accepting computation path on any input $x \notin A$, but it might have a small number of rejecting paths on inputs $x \in A$. Thus RP is said to consist of problems having probabilistic algorithms with *one-sided error*. The analogous *two-sided error* class is called BPP: A language A is in BPP if there is a probabilistic polynomial-time Turing machine M with the property that, for all x, with probability at least $\frac{3}{4}$, the output of M on input x is the correct answer to the question "Is x in A?".

Complexity classes that are defined by resource-bounded Turing machines (such as RP, BPP, etc.) lend themselves to the definition of reducibilities. Given any set A, the notation RP^A denotes the class of problems that are RP-Turing reducible to A. That is, RP^A denotes the class of problems that can be solved by probabilistic oracle Turing machines running in polynomial time, with one-sided error, using oracle A. BPP^A is defined similarly.

[3] Very recently, precisely such a "significant breakthrough" has been announced. Babai has presented a *deterministic* quasi-polynomial time algorithm for GI [6]. Thus the results of [30] have been superseded, insofar as they apply to GI.