The New Complexity Landscape around Circuit Minimization*

Eric Allender[0000−0002−0650−028X]

Rutgers University, New Brunswick NJ 08854, USA
allender@cs.rutgers.edu
http://www.cs.rutgers.edu/∼allender

Abstract. We survey recent developments related to the Minimum Circuit Size Problem.

Keywords: Complexity Theory · Kolmogorov Complexity · Minimum Circuit Size Problem

1 Introduction

Over the past few years, there has been an explosion of interest in the Minimum Circuit Size Problem (MCSP) and related problems. Thus the time seemed right to provide a survey, describing the new landscape and offering a guidebook so that one can easily reach the new frontiers of research in this area.

It turns out that this landscape is extremely unstable, with new features arising at an alarming rate. Although this makes it a scientifically-exciting time, it also means that this survey is doomed to be obsolete before it appears. It also means that the survey is going to take the form of an "annotated bibliography" with the intent to provide many pointers to the relevant literature, along with a bit of context.

The title of this article is "The *New* Complexity Landscape around Circuit Minimization" (emphasis added). This means that I will try to avoid repeating too many of the observations that were made in an earlier survey I wrote on a related topic [1]. Although that article was written only three years ago, several of the open questions that were mentioned there have now been resolved (and some of the conjectures that were mentioned have been overturned).

2 Meta-complexity, MCSP and Kolmogorov Complexity

The focus of complexity theory is to determine how hard problems are. The focus of *meta-complexity* is to determine how hard it is to determine how hard problems are. Some of the most exciting recent developments in complexity theory have been the result of meta-complexity-theoretic investigations.

* Supported in part by NSF Grant CCF-1909216.

The Minimum Circuit Size Problem (MCSP) is, quite simply, the problem of determining the circuit complexity of functions. The input consists of a pair (f, i), where f is a bit string of length $N = 2^n$ representing the truth-table of a Boolean function, and $i \in \mathbb{N}$, and the problem is to determine if f has a circuit of size at most i. The study of the complexity of MCSP is therefore the canonical meta-complexity-theoretic question. Complexity theoreticians are fond of complaining that the problems they confront (showing that computational problems are hard to compute) are notoriously difficult. But is this really true? Is it hard to show that a particular function is difficult to compute? This question can be made precise by asking about the computational complexity of MCSP. (See also [41] for a different approach.)

A small circuit is a short description of a large truth-table f; thus it is no surprise that investigations of MCSP have made use of the tools and terminology of Kolmogorov complexity. In order to discuss some of the recent developments, it will be necessary to review some of the different notions, and to establish the notation that will be used throughout the rest of the article.

Let U be a Turing machine. We define $K_U(x)$ to be $\min\{|d| : U(d) = x\}$. Those readers who are familiar with Kolmogorov complexity[1] will notice that the definition here is for what is sometimes called "plain" Kolmogorov complexity, although the notation $K_U(x)$ is more commonly used to denote what is called "prefix-free" Kolmogorov complexity. This is intentional. In this survey, the distinctions between these two notions will be blurred, in order to keep the discussion on a high level. Some of the theorems that will be mentioned below are only known to hold for the prefix-free variant, but the reader is encouraged to ignore these finer distinctions here, and seek the more detailed information in the cited references. For some Turing machines U, $K_U(x)$ will not be defined for some x, and the values of $K_U(x)$ and $K_{U'}(x)$ can be very different, for different machines U and U'. But the beauty of Kolmogorov complexity (and the applicability of of the theory it gives rise to) derives from the fact that if U and U' are *universal* Turing machines, then $K_U(x)$ and $K_{U'}(x)$ differ by at most $O(1)$. By convention, we select one particular universal machine U and define $K(x)$ to be equal to $K_U(x)$.

The function K is not computable. The simplest way to obtain a computable function that shares many of the properties of K is to simply impose a time bound, leading to the definition $K^t(x) := \min\{|d| : U(d) = x \text{ in time } t(|x|)\}$ where t is a computable function. Although it is useful in many contexts, $K^t(x)$ does not appear to be closely connected to the circuit size of x (where x is viewed as the truth-table of a function). Thus we will frequently refer to two additional resource-bounded Kolmogorov complexity measures, Kt and KT.

Levin defined $\mathsf{Kt}(x)$ to be $\min\{|d| + \log t : U(d) = x \text{ in time } t\}$ [33]; it has the nice property that it can be used to define the optimal search strategy to use, in searching for accepting computations on a nondeterministic Turing machine. $\mathsf{Kt}(x)$ also corresponds to the circuit size of the function x, but not on "normal"

[1] If the reader is not familiar with Kolmogorov complexity, then we recommend some excellent books on this topic [34, 18].

circuits. As is shown in [3], $\mathsf{Kt}(x)$ is roughly the same as the size of the smallest *oracle* circuit that computes x, where the oracle is a complete set for EXP. (An oracle circuit has "oracle gates" in addition to the usual AND, OR, and NOT gates; an oracle gate for oracle A has k wires leading into it, and if those k wires encode a bitstring y of length k where y is in A, then the gate outputs 1, otherwise it outputs 0.)

It is clearly desirable to have a version of Kolmogorov complexity that is more closely related to "ordinary" circuit size, instead of oracle circuit size. This is accomplished by defining $\mathsf{KT}(x)$ to be $\min\{|d| + t : U(d, i) = x_i \text{ in time } t\}$. (More precise definitions can be found in [3, 11].)

We have now presented a number of different measures $\mathsf{K}_\mu \in \{K, K^t, \mathsf{Kt}, \mathsf{KT}\}$. By analogy with MCSP, we can study K_μ in place of the "circuit size" measure, and thus obtain various problems of the form $\mathsf{MK}_\mu\mathsf{P} = \{(x, i) : \mathsf{K}_\mu(x) \leq i\}$, such as MKTP, $\mathsf{M}K^t\mathsf{P}$ and MKtP. If $t(n) = n^{O(1)}$, then $\mathsf{M}K^t\mathsf{P}$ is in NP, and several theorems about MKTP yield corollaries about $\mathsf{M}K^t\mathsf{P}$ in this case. (See, e.g. [3]). Similarly, if $t(n) = 2^{n^c}$ for some $c > 0$, then $\mathsf{M}K^t\mathsf{P}$ is in EXP, and several theorems about MKtP yield corollaries about $\mathsf{M}K^t\mathsf{P}$ for t in this range [3].

In order to highlight some of the recent developments, let us introduce some notation that is somewhat imprecise and which is not used anywhere else, but which will be convenient for our purposes. Let K^{poly} serve as a shorthand for K^t whenever $t = n^{O(1)}$, and similarly let K^{exp} serve as a shorthand for K^t whenever $t = 2^{n^c}$ for some $c > 0$. We will thus be referring to $\mathsf{M}K^{poly}\mathsf{P}$ and $\mathsf{M}K^{exp}\mathsf{P}$. Doing so will enable us to avoid some confusing notation surrounding the name $MINKT$, which was introduced by Ko [32] to denote the set $\{x, 1^t, 1^i : \exists d\ U(d) = x \text{ in at most } t \text{ steps and } |d| \leq i\}$. That is, $(x, i) \in \mathsf{M}K^{poly}\mathsf{P}$ iff $(x, 1^{n^c}, i) \in MINKT$ (where the time bound $t(n) = n^c$). Hence these sets have comparable complexity and results about $MINKT$ can be rephrased in terms of $\mathsf{M}K^{poly}\mathsf{P}$ with only a small loss of accuracy. In particular, some recent important results [20, 21] are phrased in terms of $MINKT$, and as such they deal with K^{poly} complexity, and they are not really very closely connected with the KT measure; the name $MINKT$ was devised more than a decade before KT was formulated. The reader who is interested in the details should refer to the original papers for the precise formulation of the theorems. However, the view presented here is "probably approximately correct".

Frequently, theorems about MCSP and the various $\mathsf{MK}_\mu\mathsf{P}$ problems are stated not in terms of *exactly* computing the circuit size or the complexity of a string, but in terms of *approximating* these values. This is usually presented in terms of two thresholds $\theta_1 < \theta_2$, where the desired solution is to say *yes* if the complexity of x is less than θ_1, and to say *no* if the complexity of x is greater than θ_2, and any answer is allowed if the complexity of x lies in the "gap" between θ_1 and θ_2. In the various theorems that have been proved in this setting, the choice of thresholds θ_1 and θ_2 is usually important, but in this article those details will be suppressed, and all of these approximation problems will be referred to as GapMCSP, GapMKtP, GapMKTP, etc.

At this point, the reader's eyes may be starting to glaze over. It is natural to wonder if we really need to have all of these different related notions. In particular, there does not seem to be much difference between MCSP and MKTP. Most hardness results for MCSP actually hold for GapMCSP, and if the "gap" is large enough, then MKTP is a solution to GapMCSP (and vice-versa). Furthermore it has frequently been the case that a theorem about MCSP was first proved for MKTP and then the result for MCSP was obtained as a corollary. However, there is no efficient reduction known (in either direction) between MCSP and MKTP, and there are some theorems that are currently known to hold only for MKTP, although they are suspected to hold also for MCSP (e.g., [5, 7, 24]). Similarly, some of the more intriguing recent developments can only be understood by paying attention to the distinction between different notions of resource-bounded Kolmogorov complexity. Thus it is worth making this investment in defining the various distinct notions.

3 Connections to Learning Theory

Certain connections between computational learning theory and Kolmogorov complexity were identified soon after computational learning theory emerged as a field. After all, the goal of computational learning theory is to find a satisfactory (and hence succinct) explanation of a large body of observed data. For instance, in the 1980s and 1990s there was work [42, 43] showing that it is NP-hard to find "succinct explanations" that have size somewhat close to the optimal size, if these "explanations" are required to be finite automata or various other restricted formalisms. Ko studied this in a more general setting, allowing "explanations" to be efficient programs (in the setting of time-bounded Kolmogorov complexity).

Thus Ko studied not only the problem of computing $K^{poly}(x)$ (where one can consider x to be a completely-specified Boolean function), but also the problem of finding the smallest description d such that $U(d)$ agrees with a given list of "yes instances" Y and a list of "no instances" N (that is, x can be considered as a partial Boolean function, with many "don't care" instances). Thus, following [29], we can call this problem Partial-MK^{poly}P. In the setting that is most relevant for computational learning theory, the partial function x is presented compactly as separate lists Y and N, rather than as a string of length 2^n over the alphabet $\{0, 1, *\}$.

Ko showed in [32] that relativizing techniques would not suffice, in order to settle the question of whether MK^{poly}P and Partial-MK^{poly}P are NP-complete. That is, by giving the universal Turing machine U that defines Kolmogorov complexity access to an oracle A, one obtains the problems MK^{poly}P^A and Partial-MK^{poly}P^A, and these sets can either be NP^A-complete or not, depending on the choice of A.

Thus it is noteworthy that it has recently been shown that Partial-MCSP is NP-complete under $\leq^{\mathsf{P}}_{\mathrm{m}}$ reductions [29]. I suspect (although I have not verified) that the proof also establishes that Partial-MKTP is NP-complete under $\leq^{\mathsf{P}}_{\mathrm{m}}$ reductions. One lesson to take from this is that KT and K^{poly} complexity differ

from each other in significant ways. There are other recent examples of related phenomena, which will be discussed below.

There are other strong connections between MCSP and learning theory that have come to light recently. Using MCSP as an oracle (or even using a set that shares certain characteristics with MCSP) one can efficiently learn small circuits that do a good job of explaining the data [12]. For certain restricted classes of circuits, there are sets in P that one can use in place of MCSP to obtain learning algorithms that don't require an oracle [12]. This connection has been explored further [37, 13].

4 Completeness, Hardness, Reducibility

The preceding section mentioned a result about a problem being NP-complete under $\leq^{\mathsf{P}}_{\mathrm{m}}$ reductions. In order to discuss other results about the complexity of MCSP and related problems it is necessary to go into more detail about different notions of reducibility.

Let $\mathcal{C}$ be either a class of functions or a class of circuits. The classes that will concern us the most are the standard complexity classes $\mathsf{L} \subseteq \mathsf{P} \subseteq \mathsf{NP}$ as well as the circuit classes (both uniform and nonuniform):

$$\mathsf{NC}^0 \subsetneq \mathsf{AC}^0 \subsetneq \mathsf{AC}^0[p] \subsetneq \mathsf{NC}^1 \subseteq \mathsf{P/poly}.$$

We refer the reader to the text by Vollmer [47] for background and more complete definitions of these standard circuit complexity complexity classes, as well as for a discussion of uniformity.

We say that $A \leq^{\mathcal{C}}_{\mathrm{m}} B$ if there is a function $f \in \mathcal{C}$ (or f computed by a circuit family in $\mathcal{C}$, respectively) such that $x \in A$ iff $f(x) \in B$. We will make use of $\leq^{\mathsf{L}}_{\mathrm{m}}, \leq^{\mathsf{AC}^0}_{\mathrm{m}}$ and $\leq^{\mathsf{NC}^0}_{\mathrm{m}}$ reducibility. The more powerful notion of Turing reducibility also plays an important role in this work. Here, $\mathcal{C}$ is a complexity class that admits a characterization in terms of Turing machines or circuits, which can be augmented with an "oracle" mechanism, either by providing a "query tape" or "oracle gates". We say that $A \leq^{\mathcal{C}}_{\mathrm{T}} B$ if there is a oracle machine in $\mathcal{C}$ (or a family of oracle circuits in $\mathcal{C}$) accepting A, when given oracle B. We make use of $\leq^{\mathsf{P/poly}}_{\mathrm{T}}, \leq^{\mathsf{RP}}_{\mathrm{T}}, \leq^{\mathsf{ZPP}}_{\mathrm{T}}, \leq^{\mathsf{BPP}}_{\mathrm{T}}, \leq^{\mathsf{P}}_{\mathrm{T}}$, and $\leq^{\mathsf{NC}^1}_{\mathrm{T}}$ reducibility; instead of writing $A \leq^{\mathsf{P/poly}}_{\mathrm{T}} B$ or $A \leq^{\mathsf{ZPP}}_{\mathrm{T}} B$, we will sometimes write $A \in \mathsf{P}^B/\mathsf{poly}$ or $A \in \mathsf{ZPP}^B$. Turing reductions that are "nonadaptive" – in the sense that the list of queries that are posed on input x does not depend on the answers provided by the oracle – are called *truth table reductions*. We make use of $\leq^{\mathsf{P}}_{\mathrm{tt}}$ reducibility.

Not much has changed, regarding what is known about the "hardness" of MCSP, in the three years that have passed since my earlier survey [1]. Here is what I wrote at that time:

> *Table 1 presents information about the consequences that will follow if* MCSP *is* NP*-complete (or even if it is hard for certain subclasses of* NP*). The table is incomplete (since it does not mention the influential theorems of Kabanets and Cai [31] describing various consequences if* MCSP

were complete under a certain restricted type of $\leq^{\mathsf{P}}_{\mathrm{m}}$ reduction). It also fails to adequately give credit to all of the papers that have contributed to this line of work, since – for example – some of the important contributions of [36] have subsequently been slightly improved [26, 8]. But one thing should jump out at the reader from Table 1: All of the conditions listed in Column 3 (with the exception of "FALSE") are widely believed to be true, although they all seem to be far beyond the reach of current proof techniques.

Table 1. Summary of what is known about the consequences of MCSP being hard for NP under different types of reducibility. If MCSP is hard for the class in Column 1 under the reducibility shown in Column 2, then the consequence in Column 3 follows.

class $\mathcal{C}$	reductions $\mathcal{R}$	statement $\mathcal{S}$	Reference
TC^0	$\leq_m^{n^{1/3}}$	FALSE	[36]
TC^0	$\leq^{\mathsf{AC}^0}_{\mathrm{m}}$	$\mathsf{LTH}^2 \not\subseteq \mathsf{io\text{-}SIZE}[2^{\Omega(n)}]$ and $\mathsf{P} = \mathsf{BPP}$	[8, 36]
TC^0	$\leq^{\mathsf{AC}^0}_{\mathrm{m}}$	$\mathsf{NP} \not\subseteq \mathsf{P/poly}$	[8]
P	$\leq^{\mathsf{L}}_{\mathrm{m}}$	$\mathsf{PSPACE} \neq \mathsf{P}$	[8]
NP	$\leq^{\mathsf{L}}_{\mathrm{m}}$	$\mathsf{PSPACE} \neq \mathsf{ZPP}$	[36]
NP	$\leq^{\mathsf{P}}_{\mathrm{tt}}$	$\mathsf{EXP} \neq \mathsf{ZPP}$	[26]

It is significant that neither MCSP nor MKTP is NP-complete under $\leq_m^{n^{1/3}}$ reductions, since SAT and many other well-known problems are complete under this very restrictive notion of reducibility – but it would be more satisfying to know whether these problems can be complete under more widely-used reducibilities such as $\leq^{\mathsf{AC}^0}_{\mathrm{m}}$. These sublinear-time reductions are so restrictive, that even the PARITY problem is not $\leq_m^{n^{1/3}}$-reducible to MCSP or MKTP. In an attempt to prove that PARITY is not $\leq^{\mathsf{AC}^0}_{\mathrm{m}}$-reducible to MKTP, we actually ended up proving the opposite:

Theorem 1. *[7]* MKTP *is hard for* DET *under non-uniform* NC^0 *reductions. This also holds for* MKtP *and* MKP*.*

Here, DET is the class of problems NC^1-Turing-reducible to computing the determinant. It includes the well-known complexity classes L and NL. This remains the only theorem that shows hardness of $\mathsf{MK}_\mu\mathsf{P}$ problems under any kind of $\leq^{\mathcal{C}}_{\mathrm{m}}$ reductions.

As a corollary of this theorem it follows that MKTP is not in $\mathsf{AC}^0[p]$ for any prime p. This was mentioned as an open question in [1] (see footnote 2 of [1]). (An alternate proof was given in [24].) It remained open whether MCSP was in $\mathsf{AC}^0[p]$ until a lower bound was proved in [19].

[2] LTH is the linear-time analog of the polynomial hierarchy. Problems in LTH are accepted by alternating Turing machines that make only $O(1)$ alternations and run for linear time.

It is *still* open whether MCSP is hard for DET. The proof of the hardness result in [7] actually carries over to a version of GapMKTP where the "gap" is quite small. Thus one avenue for proving a hardness result for MCSP had seemed to be to improve the hardness result of [7], so that it worked for a much larger "gap". This avenue was subsequently blocked, when it was shown that PARITY is not AC^0-reducible to GapMCSP (or to GapMKTP) for a moderate-sized "gap" [9]. Thus, although it is still open whether MCSP is NP-complete under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, we now know that GapMCSP is not NP-complete under this notion of reducibility.

When a *much* larger "gap" is considered, it was shown in [7] that, if cryptographically-secure one-way functions exist, then GapMCSP and GapMKTP are NP-intermediate in the sense that neither problem is in P/poly, and neither problem is complete for NP under P/poly-Turing reductions.

The strongest hardness results that are known for the $\mathsf{MK}_\mu\mathsf{P}$ problems in NP remain the results of [4], where it was shown that MCSP, MKTP, and $\mathsf{MK}^{poly}\mathsf{P}$ are all hard for SZK under $\leq_{\mathrm{T}}^{\mathsf{BPP}}$ reductions. SZK is the class of problems that have statistical zero knowledge interactive proofs; SZK contains most of the problems that are assumed to be intractable, in order to build public-key cryptosystems. Thus it is widely assumed that MCSP and related problems lie outside of P/poly, and cryptographers hope that it requires nearly exponential-sized circuits. SZK also contains the Graph Isomorphism problem, which is $\leq_{\mathrm{T}}^{\mathsf{RP}}$-reducible to MCSP and MKTP. In [5], Graph Isomorphism (and several other problems) were shown to be $\leq_{\mathrm{T}}^{\mathsf{ZPP}}$ reducible to MKTP; it remains unknown if this also holds for MCSP. In fact, there is no interesting example of a problem A that is not known to be in $\mathsf{NP} \cap \mathsf{coNP}$ that has been shown to be $\leq_{\mathrm{T}}^{\mathsf{ZPP}}$ reducible to MCSP.

We close this section with a discussion of a very powerful notion of reducibility: SNP reductions. (Informally A is SNP reducible to B means that A is $(\mathsf{NP} \cap \mathsf{coNP})$-reducible to B.) Hitchcock and Pavan have shown, under the very plausible assumption that $\mathsf{NP} \cap \mathsf{coNP}$ contains problems that require large circuits, that if MCSP is NP-complete (under the usual $\leq_{\mathrm{m}}^{\mathsf{P}}$ reductdions), then it is also complete under SNP reductions whose queries avoid asking about very small circuit sizes; they are able to use this as a tool to derive additional interesting consequences from the assumption that MCSP is NP-complete [26]. It is interesting to note that, back in the early 1990's, Ko explicitly considered the possibility that computing $\mathsf{MK}^{poly}\mathsf{P}$ might be NP-complete under SNP reductions [32].

4.1 Completeness in EXP and Other Classes

There are problems "similar" to MCSP that reside in many complexity classes. We can define MCSP^A to be MCSP for oracle circuits with A-oracle gates. That is, $\mathsf{MCSP}^A = \{(f, i) : f \text{ has an } A\text{-oracle circuit of size at most } i\}$. When A is complete for EXP, then MCSP^A is thought of as being quite similar to MKtP. Both of these problems, along with $\mathsf{MK}^{exp}\mathsf{P}$, are complete for EXP under $\leq_{\mathrm{T}}^{\mathsf{P/poly}}$ and $\leq_{\mathrm{T}}^{\mathsf{NP}}$ reductions [3].

It is still open whether either of MKtP or MCSP^A is in P, and it had been open if $\mathsf{M}K^t\mathsf{P}$ is in P for "small" exponential functions t such as $t(n) = 2^{n/2}$. But there is recent progress:

Theorem 2. *[21]* $\mathsf{M}K^{exp}\mathsf{P}$ *is complete for* EXP *under* $\leq^{\mathsf{P}}_{\mathrm{T}}$ *reductions.*

This seems to go a long way toward addressing Open Question 3.6 in [1].

As a corollary, $\mathsf{M}K^{exp}\mathsf{P}$ is not in P. In fact, a much stronger result holds. Let t be any superpolynomial function. Then the set of K^t-random strings $\{x : K^t(x) < |x|\}$ is *immune* to P (meaning: it has no infinite subset in P) [21]. The proof does not seem to carry over to Kt complexity, highlighting a significant difference between Kt and K^{exp}.

Although it remains open whether $\mathsf{MKtP} \in \mathsf{P}$, Hirahara does show that MKtP is not in P-uniform ACC^0, and in fact the set of Kt-random strings is immune to P-uniform ACC^0. Furthermore, improved immunity results for the Kt-random strings are in some sense possible *if and only if* better algorithms for $\mathsf{CircuitSAT}$ can be devised for larger classes of circuits [21].

Oliveira has defined a randomized version of Kt complexity, which is conjectured to be nearly the same as Kt, but for which he is able to prove unconditional intractability results [38].

$\mathsf{MCSP}^{\mathsf{QBF}}$ was known to be complete for PSPACE under $\leq^{\mathsf{ZPP}}_{\mathrm{T}}$ reductions [3]. In more recent work, for various subclasses $\mathcal{C}$ of PSPACE, when A is a suitable complete problem for $\mathcal{C}$, then MCSP^A has been shown to be complete for $\mathcal{C}$ under $\leq^{\mathsf{BPP}}_{\mathrm{T}}$ reductions [30]. Crucially, the techniques used by [30] (and, indeed, by any of the authors who had proved hardness results for MCSP^A previously for various A) failed to work for any A in the polynomial hierarchy. We will return to this issue in the following section.

In related work, it was shown [7] that the question of whether MKTP^A is hard for DET under a type of uniform AC^0 reductions is equivalent to the question of whether $\mathsf{DSPACE}(n)$ contains any sets that require exponential-size A-oracle circuits. Furthermore, this happens if and only if PARITY reduces to MKTP^A. Note that this condition is *more likely* to be true if A is easy, than if A is complex; it is false if A is complete for PSPACE, and it is probably true if $A = \emptyset$. Thus, although $\mathsf{MKTP}^{\mathsf{QBF}}$ is almost certainly more complex than MKTP (the former is PSPACE-complete, and the latter is in NP), a reasonably-large subclass of P probably reduces to MKTP via these uniform AC^0 reductions, whereas hardly anything AC^0-reduces to $\mathsf{MKTP}^{\mathsf{QBF}}$. The explanation for this is that a uniform AC^0 reduction cannot formulate any useful queries to a complex oracle, whereas it (probably) can do so for a simpler oracle.

4.2 NP-Hardness

Recall from the previous section that there were no NP-hardness results known for any problem of the form MCSP^A where A is in the polynomial hierarchy.

This is still true; however, there is some progress to report. Hirahara has shown that computing the "conditional" complexity $K^{poly}(x|y)$ relative to SAT

(i.e., given (x, y), finding the length of the shortest description d such that $U^{\mathsf{SAT}}(d, y) = x$ in time n^c) is NP-hard under $\leq^{\mathsf{P}}_{\mathrm{tt}}$ reductions [21].

It might be more satisfying to remove the SAT oracle, and have a hardness result for computing $K^{poly}(x|y)$ – but Hirahara shows that this can't be shown to be hard for NP (or even hard for ZPP) under $\leq^{\mathsf{P}}_{\mathrm{tt}}$ reductions without first separating EXP from ZPP.

In a similar vein, if one were to show that MCSP or MKTP (or MCSP^A or MKTP^A for any set $A \in \mathsf{EXP}$) is hard for NP under $\leq^{\mathsf{P}}_{\mathrm{tt}}$ reductions, then one will have shown that $\mathsf{ZPP} \neq \mathsf{EXP}$ [21].

A different kind of NP-hardness result for conditional Kolmogorov complexity was proved recently by Ilango [28]. In [3], conditional KT complexity $\mathsf{KT}(x|y)$ was studied by making the string y available to the universal Turing machine U as an "oracle". Thus it makes sense to consider a "conditional complexity" version of MCSP by giving a string y available to a circuit via oracle gates. This problem was formalized and shown to be NP-complete under $\leq^{\mathsf{ZPP}}_{\mathrm{T}}$ reductions [28].

Many of the functions that we compute daily produce more than one bit of output. Thus it makes sense to study the circuit size that is required in order to compute such functions. This problem is called $\mathsf{Multi\text{-}MCSP}$ in [29], where it is shown to be NP-complete under $\leq^{\mathsf{RP}}_{\mathrm{T}}$ reductions. It will be interesting to see how the complexity of this problem varies, as the number of output bits of the functions under consideration shrinks toward one (at which point it becomes MCSP).

It has been known since the 1970's that computing the size of the smallest DNF expression for a given truth-table is NP-complete. (A simple proof, and a discussion of the history can be found in [6].) However, it remains unknown what the complexity is of finding the smallest depth-three circuit for a given truth table. (Some very weak intractability results for minimizing constant-depth circuits can be found in [6], giving subexponential reductions from the problem of factoring Blum integers.) The first real progress on this front was reported in [23], giving an NP-completeness result (under $\leq^{\mathsf{P}}_{\mathrm{m}}$ reductions) for a class of depth three circuits (with MOD gates on the bottom level). Ilango proved that computing the size of the smallest depth-d *formula* for a truth-table lies outside of $\mathsf{AC}^0[p]$ for any prime p [28], and he has now followed that up with a proof that computing the size of the smallest depth-d *formula* is NP-complete under $\leq^{\mathsf{RP}}_{\mathrm{T}}$ reductions [27]. Note that a constant-depth circuit can be transformed into a formula with only a polynomial blow-up; thus in many situations we are able to ignore the distinction between circuits and formulas in the constant-depth realm. However, the techniques employed in [27, 28] are quite sensitive to small perturbations in the size, and hence the distinction between circuits and formulae is important here. Still, this is dramatic progress on a front where progress has been very slow.

5 Average Case Complexity, One-Way Functions

Cai and Kabanets gave birth to the modern study of MCSP in 2000 [31], in a paper that was motivated in part by the study of Natural Proofs [44], and which called attention to the fact that if MCSP is easy, then there are no cryptographically-secure one-way functions. In the succeeding decades, there has been speculation about whether the converse implication also holds. That is, can one base cryptography on assumptions about the complexity of MCSP?

First, it should be observed that, in some sense, MCSP is very easy "on average". For instance the hardness results that we have (such as reducing SZK to MCSP) show that the "hard instances" of MCSP are the ones where we want to distinguish between n-ary functions that require circuits of size $2^n/n^2$ (the "NO" instances) and those that have circuits of size at most $2^{n/3}$ (the "YES" instances). However, an algorithm that simply says "no" on all inputs will give the correct answer more than 99% of the time.

Thus Hirahara and Santhanam [24] chose to study a different notion of heuristics for MCSP, where algorithms must always give an answer in {Yes, No, I don't know}, where the algorithm never gives an incorrect answer, and the algorithm is said to perform well "on average" if it only seldom answers "I don't know". They were able to show unconditionally that MCSP is hard on average in this sense for $\mathsf{AC}^0[p]$ for any prime p, and to show that certain well-studied hypotheses imply that MCSP is hard on average.

More recently, Santhanam [45] has formulated a conjecture (which would involve too big of a digression to describe more carefully here), which – if true – would imply that a version of MCSP is hard on average in this sense if and only if cryptographically-secure one-way functions exist. That is, Santhanam's conjecture provides a framework for believing that one can base cryptography on the average-case complexity of MCSP.

But how does the average-case complexity of MCSP depend on its worst-case complexity? Hirahara [20] showed that GapMCSP has no solution in BPP if and only if a version of MCSP is hard on average. A related result stated in terms of K^{poly} appears in the same paper. These results attracted considerable attention, because prior work had indicated that such worst-case-to-average-case reductions would be impossible to prove using black-box techniques. Additional work has given further evidence that the techniques of [20] are inherently non-black-box [25].

6 Complexity Classes and Noncomputable Complexity Measures

The title of this section is the same as the title of Section 4 of the survey that I wrote three years ago [1]. In that section, I described the work that had been done, studying the classes of sets that are reducible to the (non-computable) set of Kolmogorov-random strings R_K, and to MKP, including the reasons why it

seemed reasonable to conjecture that BPP and NEXP could be characterized in terms of different types of reductions to the Kolmogorov-random strings.

I won't repeat that discussion here, because both of those conjectures have been disproved (barring some extremely unlikely complexity class collapses). Taken together, the papers [25], [22], and [21] give a much better understanding of the classes of languages reducible to the Kolmogorov-random strings.

Previously, it was known that $\mathsf{PSPACE} \subseteq \mathsf{P}^{R_K}$, and $\mathsf{NEXP} \subseteq \mathsf{NP}^{R_K}$. Hirahara [21] has now shown $\mathsf{NEXP} \subseteq \mathsf{EXP}^{\mathsf{NP}} \subseteq \mathsf{P}^{R_K}$.

This same paper also gives a surprising answer to Open Question 4.6 of [1], in showing that Quasipolynomial-time nonadaptive reductions to R_K suffice to capture NP (and also some other classes in the polynomial hierarchy).

7 Magnification

Some of the most important and exciting developments relating to MCSP and related problems deal with the emerging study of "hardness magnification". This is the phenomenon whereby seemingly very modest lower bounds can be "amplified" or "magnified" and thereby be shown to imply superpolynomial lower bounds. I was involved in some of the early work in this direction [10] (which did not involve MCSP), but much stronger work has subsequently appeared.

It is important to note, in this regard, that lower bounds have been proved for MCSP that essentially match the strongest lower bounds that we have for any problems in NP [17]. There is now a significant body of work, showing that slight improvements to those bounds, or other seemingly-attainable lower bounds for GapMKtP or GapMCSP or related problems, would yield dramatic complexity class separations [16, 15, 14, 13, 46, 40, 39, 35].

This would be a good place to survey this work, except that an excellent survey already appears in [13]. Igor Carboni Oliveira has also written some notes entitled "Advances in Hardness Magnification" related to a talk he gave at the Simons Institute in December, 2019, available on his home page. These notes and [13] describe in detail the reasons that this approach seems to avoid the Natural Proofs barrier identified in the work of Razborov and Rudich [44]. But they also describe some potential obstacles that need to be overcome, before this approach can truly be used to separate complexity classes.

Acknowledgments

Thanks are due to Rahul Santhanam, for calling attention to some misstatements in an earlier version of this survey [2].

References

1. Allender, E.: The complexity of complexity. In: Computability and Complexity: Essays Dedicated to Rodney G. Downey on the Occasion of his 60th Birthday. Lecture Notes in Computer Science, vol. 10010, pp. 79–94. Springer (2017). https://doi.org/10.1007/978-3-319-50062-1_6

2. Allender, E.: The new complexity landscape around circuit minimization. In: Proc. 14th International Conference on Language and Automata Theory and Applications (LATA). Lecture Notes in Computer Science, vol. 12038, pp. 3–16. Springer (2020)
3. Allender, E., Buhrman, H., Koucký, M., van Melkebeek, D., Ronneburger, D.: Power from random strings. SIAM Journal on Computing **35**, 1467–1493 (2006). https://doi.org/10.1137/050628994
4. Allender, E., Das, B.: Zero knowledge and circuit minimization. Information and Computation **256**, 2–8 (2017). https://doi.org/10.1016/j.ic.2017.04.004, special issue for MFCS '14
5. Allender, E., Grochow, J., van Melkebeek, D., Morgan, A., Moore, C.: Minimum circuit size, graph isomorphism and related problems. SIAM Journal on Computing **47**, 1339–1372 (2018). https://doi.org/10.1137/17M1157970
6. Allender, E., Hellerstein, L., McCabe, P., Pitassi, T., Saks, M.E.: Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. SIAM Journal on Computing **38**(1), 63–84 (2008). https://doi.org/10.1137/060664537
7. Allender, E., Hirahara, S.: New insights on the (non)-hardness of circuit minimization and related problems. ACM Transactions on Computation Theory (ToCT) **11**(4), 27:1–27:27 (2019). https://doi.org/10.1145/3349616
8. Allender, E., Holden, D., Kabanets, V.: The minimum oracle circuit size problem. Computational Complexity **26**(2), 469–496 (2017). https://doi.org/10.1007/s00037-016-0124-0
9. Allender, E., Ilango, R., Vafa, N.: The non-hardness of approximating circuit size. In: Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia (CSR). Lecture Notes in Computer Science, vol. 11532, pp. 13–24. Springer (2019). https://doi.org/10.1007/978-3-030-19955-5_2
10. Allender, E., Koucký, M.: Amplifying lower bounds by means of self-reducibility. J. ACM **57**, 14:1 – 14:36 (2010). https://doi.org/10.1145/1706591.1706594
11. Allender, E., Koucký, M., Ronneburger, D., Roy, S.: The pervasive reach of resource-bounded Kolmogorov complexity in computational complexity theory. J. Comput. Syst. Sci. **77**, 14–40 (2010). https://doi.org/10.1016/j.jcss.2010.06.004
12. Carmosino, M., Impagliazzo, R., Kabanets, V., Kolokolova, A.: Learning algorithms from natural proofs. In: 31st Conference on Computational Complexity, CCC. LIPIcs, vol. 50, pp. 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016). https://doi.org/10.4230/LIPIcs.CCC.2016.10
13. Chen, L., Hirahara, S., Oliveira, I.C., Pich, J., Rajgopal, N., Santhanam, R.: Beyond natural proofs: Hardness magnification and locality. In: 11th Innovations in Theoretical Computer Science Conference (ITCS). LIPIcs, vol. 151, pp. 70:1–70:48. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2020). https://doi.org/10.4230/LIPIcs.ITCS.2020.70
14. Chen, L., Jin, C., Williams, R.: Hardness magnification for all sparse NP languages. In: Symposium on Foundations of Computer Science (FOCS). pp. 1240–1255 (2019). https://doi.org/10.1109/FOCS.2019.00077
15. Chen, L., Jin, C., Williams, R.: Sharp threshold results for computational complexity (2019), manuscript
16. Chen, L., McKay, D.M., Murray, C.D., Williams, R.R.: Relations and equivalences between circuit lower bounds and Karp-Lipton theorems. In: 34th Computational Complexity Conference (CCC). LIPIcs, vol. 137, pp. 30:1–30:21. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2019). https://doi.org/10.4230/LIPIcs.CCC.2019.30

17. Cheraghchi, M., Kabanets, V., Lu, Z., Myrisiotis, D.: Circuit lower bounds for MCSP from local pseudorandom generators. In: 46th International Colloquium on Automata, Languages, and Programming,(ICALP). LIPIcs, vol. 132, pp. 39:1–39:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.39
18. Downey, R., Hirschfeldt, D.: Algorithmic Randomness and Complexity. Springer (2010)
19. Golovnev, A., Ilango, R., Impagliazzo, R., Kabanets, V., Kolokolova, A., Tal, A.: $AC^0[p]$ lower bounds against MCSP via the coin problem. In: 46th International Colloquium on Automata, Languages, and Programming,(ICALP). LIPIcs, vol. 132, pp. 66:1–66:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.66
20. Hirahara, S.: Non-black-box worst-case to average-case reductions within NP. In: 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS). pp. 247–258 (2018). https://doi.org/10.1109/FOCS.2018.00032
21. Hirahara, S.: Kolmogorov-randomness is harder than expected (2019), manuscript
22. Hirahara, S.: Unexpected power of random strings. In: 11th Innovations in Theoretical Computer Science Conference, ITCS. LIPIcs, vol. 151, pp. 41:1–41:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2020). https://doi.org/10.4230/LIPIcs.ITCS.2020.41
23. Hirahara, S., Oliveira, I.C., Santhanam, R.: NP-hardness of minimum circuit size problem for OR-AND-MOD circuits. In: 33rd Conference on Computational Complexity, CCC. LIPIcs, vol. 102, pp. 5:1–5:31. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/LIPIcs.CCC.2018.5
24. Hirahara, S., Santhanam, R.: On the average-case complexity of MCSP and its variants. In: 32nd Conference on Computational Complexity, CCC. LIPIcs, vol. 79, pp. 7:1–7:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017). https://doi.org/10.4230/LIPIcs.CCC.2017.7
25. Hirahara, S., Watanabe, O.: On nonadaptive reductions to the set of random strings and its dense subsets. Electronic Colloquium on Computational Complexity (ECCC) **26**, 25 (2019)
26. Hitchcock, J.M., Pavan, A.: On the NP-completeness of the minimum circuit size problem. In: Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS). LIPIcs, vol. 45, pp. 236–245. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2015). https://doi.org/10.4230/LIPIcs.FSTTCS.2015.236
27. Ilango, R.: Personal communication (2019)
28. Ilango, R.: Approaching MCSP from above and below: Hardness for a conditional variant and $AC^0[p]$. In: 11th Innovations in Theoretical Computer Science Conference, ITCS. LIPIcs, vol. 151, pp. 34:1–34:26. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2020). https://doi.org/10.4230/LIPIcs.ITCS.2020.34
29. Ilango, R., Loff, B., Oliveira, I.C.: NP-hardness of minimizing circuits and communication (2019), manuscript
30. Impagliazzo, R., Kabanets, V., Volkovich, I.: The power of natural properties as oracles. In: 33rd Conference on Computational Complexity, CCC. LIPIcs, vol. 102, pp. 7:1–7:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/LIPIcs.CCC.2018.7
31. Kabanets, V., Cai, J.Y.: Circuit minimization problem. In: ACM Symposium on Theory of Computing (STOC). pp. 73–79 (2000). https://doi.org/10.1145/335305.335314

32. Ko, K.: On the notion of infinite pseudorandom sequences. Theor. Comput. Sci. **48**(3), 9–33 (1986). https://doi.org/10.1016/0304-3975(86)90081-2
33. Levin, L.A.: Randomness conservation inequalities; information and independence in mathematical theories. Information and Control **61**(1), 15–37 (1984). https://doi.org/10.1016/S0019-9958(84)80060-1
34. Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition. Texts in Computer Science, Springer (2019). https://doi.org/10.1007/978-3-030-11298-1
35. McKay, D.M., Murray, C.D., Williams, R.R.: Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC). pp. 1215–1225 (2019). https://doi.org/10.1145/3313276.3316396
36. Murray, C., Williams, R.: On the (non) NP-hardness of computing circuit complexity. Theory of Computing **13**(4), 1–22 (2017). https://doi.org/10.4086/toc.2017.v013a004
37. Oliveira, I., Santhanam, R.: Conspiracies between learning algorithms, circuit lower bounds and pseudorandomness. In: 32nd Conference on Computational Complexity, CCC. LIPIcs, vol. 79, pp. 18:1–18:49. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017). https://doi.org/10.4230/LIPIcs.CCC.2017.18
38. Oliveira, I.C.: Randomness and intractability in Kolmogorov complexity. In: 46th International Colloquium on Automata, Languages, and Programming,(ICALP). LIPIcs, vol. 132, pp. 32:1–32:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.32
39. Oliveira, I.C., Pich, J., Santhanam, R.: Hardness magnification near state-of-the-art lower bounds. In: 34th Computational Complexity Conference (CCC). LIPIcs, vol. 137, pp. 27:1–27:29. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2019). https://doi.org/10.4230/LIPIcs.CCC.2019.27
40. Oliveira, I.C., Santhanam, R.: Hardness magnification for natural problems. In: 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS). pp. 65–76 (2018). https://doi.org/10.1109/FOCS.2018.00016
41. Pich, J., Santhanam, R.: Why are proof complexity lower bounds hard? In: Symposium on Foundations of Computer Science (FOCS). pp. 1305–1324 (2019). https://doi.org/10.1109/FOCS.2019.00080
42. Pitt, L., Valiant, L.G.: Computational limitations on learning from examples. J. ACM **35**(4), 965–984 (1988). https://doi.org/10.1145/48014.63140
43. Pitt, L., Warmuth, M.K.: The minimum consistent DFA problem cannot be approximated within any polynomial. J. ACM **40**(1), 95–142 (1993). https://doi.org/10.1145/138027.138042
44. Razborov, A., Rudich, S.: Natural proofs. J. Comput. Syst. Sci. **55**, 24–35 (1997). https://doi.org/10.1006/jcss.1997.1494
45. Santhanam, R.: Pseudorandomness and the minimum circuit size problem. In: 11th Innovations in Theoretical Computer Science Conference (ITCS). LIPIcs, vol. 151, pp. 68:1–68:26. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2020). https://doi.org/10.4230/LIPIcs.ITCS.2020.68
46. Tal, A.: The bipartite formula complexity of inner-product is quadratic. Electronic Colloquium on Computational Complexity (ECCC) **23**, 181 (2016)
47. Vollmer, H.: Introduction to Circuit Complexity: A Uniform Approach. Springer-Verlag New York Inc. (1999). https://doi.org/10.1007/978-3-662-03927-4