Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 192 (2023)

A Note On the Universality of Black-box $\mathrm{MK^tP}$ Solvers

Noam Mazor * Rafael Pass †

December 3, 2023

Abstract

The relationships between various meta-complexity problems are not well understood in the *worst-case regime*, including whether the search version is harder than the decision version, whether the hardness scales with the "threshold", and how the hardness of different meta-complexity problems relate to one another, and to the task of function inversion.

In this note, we present resolutions to some of these questions with respect to the *black-box* analog of these problems. In more detail, let $\mathrm{MK^t_MP}[s]$ denote the language consisting of strings x with $\mathrm{K}^t_M(x) < s(|x|)$, where $\mathrm{K}^t_M(x)$ denotes the t-bounded Kolmogorov complexity of x with M as the underlying (Universal) Turing machine, and let search-$\mathrm{MK^t_MP}[s]$ denote the search version of the same problem.

We show that if there for *every* Universal Turing machine U there exists a $2^{\alpha n}\mathrm{poly}(n)$-size U-oracle aided circuit deciding $\mathrm{MK^t_UP}[n - O(1)]$, then for every function s, and every *not necessarily universal* Turing machine M, there exists a $2^{\alpha s(n)}\mathrm{poly}(n)$-size M-oracle aided circuit solving search-$\mathrm{MK^t_MP}[s(n)]$; this in turn yields circuits of roughly the same size for both the Minimum Circuit Size Problem (MCSP), and the function inversion problem, as they can be thought of as instantiating $\mathrm{MK^t_MP}$ with particular choices of (a non-universal) TMs M (the circuit emulator for the case of MCSP, and the function evaluation in the case of function inversion).

As a corollary of independent interest, we get that the complexity of black-box function inversion is (roughly) the same as the complexity of black-box deciding $\mathrm{MK^t_UP}[n - O(1)]$ for any universal TM U; that is, also in the worst-case regime, black-box function inversion is "equivalent" to black-box deciding $\mathrm{MK^t_UP}$.

*Cornell Tech. E-mail: `noammaz@gmail.com`. Research supported by NSF CNS-2149305.

†Tel-Aviv University and Cornell Tech. E-mail: `rafaelp@tau.ac.il`. Supported in part by NSF Award CNS 2149305, AFOSR Award FA9550-18-1-0267, AFOSR Award FA9550-23-1-0387, and an Algorand Foundation grant, and DARPA under Agreement No. HR00110C0086. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, DARPA, AFOSR, or the Algorand Foundation.

ISSN 1433-8092

1 Introduction

We consider the worst-case complexity of solving standard Meta-complexity Programs, notably the the *Time-Bounded Kolmogorov Complexity Problem* [Kol68; Sol64; Cha69; Ko86; Har83; Sip83]—computing the length, denoted $\mathrm{K}^t_{\mathsf{U}}(x)$ of shortest program (evaluated on some particular Universal Turing machine (TM) U) that generates a given string x, within time $t(|x|)$, where t is a polynomial, and the (b) the *Minimimum Circuit Size problem* (MCSP) [KC00; Tra84]—finding the smallest Boolean circuit that computes a given function x. For both of these problem one may also consider *thresholds* versions, $\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[s]$ and $\mathrm{MCSP}[s]$, where $\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[s]$ (resp. $\mathrm{MCSP}[s]$) is the languages of strings x s.t. $\mathrm{K}^t_{\mathsf{U}}(x)$ (resp. the circuit size of x) is less than $s(|x|)$, as well as *search* versions search-$\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[s]$, where the goal is not compute/decide the complexity of a string x but also to *find* a short description that witnesses this complexity.

The relationship between these various meta-complexity problems are not well understood. In particular:

1. **Decision-to-Search:** Solving the search version trivially yields a solver for the decisional (or computational) version with roughly the same complexity. Does the converse hold: Does a $T(n)$-size circuit for solving the decision version imply a, roughly, $T(n)$-size circuit solving the search version?

2. **Hardness Scaling to the Threshold:** Intuitively, the threshold version of the problem, for small thresholds $s(n) \ll n$ ought to be easier than the threshold n version (or computational) version since there exist trivial $2^{s(n)}\mathrm{poly}(n)$ time algorithms for the threshold version (simply doing brute force search). Does this hold more generally: Does a $T(n)$-size circuit solving $\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[n - O(1)]$ imply a roughly $T(s(n))$-size circuit for solving $\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[s(n)]$?

3. **The "Model of Computation" and the Relationship to Function Inversion:** Other meta-complexity problems, such as the MCSP problem, can be stated as an $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}$ problem with respect to a particular *non-universal* underlying TM M (performing circuit emulation). Additionally, a solver for the search-$\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}$ problem with respect to *any* (non-universal) TM M is also equivalent to a solver for the function inversion problem (i.e., the problem of inverting any function on every input). Does a $T(n)$-size solver for $\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[s(n)]$ with respect to any underlying Universal TM U, imply one (of size roughly $T(n)$) that also works with respect non-universal TMs (and thus also for MCSP and function inversion)?

In the *average-case regime*, positive answers to these questions—when restricting to efficient underlying (Universal) TMs—were provided in respectively [LP20] (for question 1), [LP21] (for question 2) and [LP20; RS21] (for question 3), but they remain wide open in the worst-case regime. This is the focus of the current paper, but rather than restricting to efficient underlying TMs, we will consider *arbitrary* TMs (with potentially a large description or running time).

In particular, very recently non-trivial circuits for the various different meta-complexity problems were given. In [MP24], the current authors show that for any efficient Universal TM U, there exists a circuit of size $2^{4n/5}\mathrm{poly}(n, t(n))$ that solves the search version of the $\mathrm{K}^t_{\mathsf{U}}$ (and thus also search-$\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}$). A different, independent, paper by Hirahara, Ilango and Williams [HIW23] focuses on the threshold version of the above meta-complexity problems and presents circuits of size respectively $2^{4/5s(n)} \cdot \mathrm{poly}(n, t(n))$ and $2^{(4/5+o(1))\cdot s(n)\log s(n)}$ for them. In both cases, the core of the technical work consist of providing a circuit implementation of the function inversion algorithm from Fiat and Naor [FN00], and next applying this function inversion algorithm to the one-way function construction of [LP20] (or variants thereof, notably the variant of [RS21] to deal with the MCSP problem) based on the hardness of meta-complexity problems—an approach first envisioned by Ren and Santhanam [RS21].[1] As such, the worst-case complexity bounds obtained are roughly the same for (a) the search and the decisional version (as the function inverter also directly solves the search problem in [LP20]), (b) they naturally scale with the threshold s of $\mathrm{MK^tP}[s]$ (based on an extension of the function inversion attack of Fiat-Naor done in [HIW23]), and (c) are the roughly same for $\mathrm{MK^tP}$, MCSP and function inversion (since the one-way function constructions in [LP20; RS21] are length preserving). These works thus indicate that perhaps the same phenomena that are known in the average-case setting may also hold in the worst-case setting.

In this paper, we demonstrate that this is not a coincidence. Indeed, we provide a positive answer to all the above questions also in the worst-case regiment, when restricting attention to *black-box* solvers and thus all the above result follow from simply obtaining circuit for black-box solving the decisional $\mathrm{MK^tP}$ problem.

Black-box Solvers As noted in [MP24], their algorithm for $\mathrm{K}^t_{\mathsf{U}}$ works for any Universal TM U (as long as the algorithm gets oracle access to U): for *any* (not necessarily efficient) Universal TM U, there exists a U-oracle aided circuit of size $2^{4n/5}\mathrm{poly}(n, t(n))$ that solves the search version of the K^t_{U}. Following [MP24], we say that $\mathrm{MK^tP}[s]$ (resp search- $\mathrm{MK^tP}[s]$) *admits a $T(n)$-size black-box solver* if for every universal TM U, there exists a $T(n)$ size U-oracle aided circuit for solving $\mathrm{MK^t_{\mathsf{U}}P}[s]$ (resp search- $\mathrm{MK^t_{\mathsf{U}}P}[s]$). We additionally say that these problems admit a $T(n)$-size *generalized black-box* solver if the same holds not only with respect to any *universal* TM U but also for non-universal TM M (satisfying the minimal condition that the emulation by M has a unique output: $M(\Pi, 1^{t_1}) = M(\Pi, 1^{t_2})$ if either of those provide some output). (Considering generalized black-box solvers is what will allow us to answer question 3 above, but actually, also from a technical point of view, will also be instrumental also to deal with question 1).

1.1 Our Results

Our main result shows that the existence of a $2^{\alpha n + o(n)}$-size black-box solver for $\mathrm{MK^tP}[n - O(1)]$ implies the existence of a $2^{\alpha s(n) + o(n)}$-size *generalized* black-box solver for search- $\mathrm{MK^tP}[s]$, thus providing a positive answer to all the above questions with respect to black-box solvers.

Theorem 1.1. *Assume the existence of a $2^{\alpha n} \cdot \mathrm{poly}(n)$-size black-box solver for* $\mathrm{MK^tP}[n-4]$ *for $t(n) = n$. Then there exists a $2^{\alpha s(n)} \cdot \mathrm{poly}(n)$-size generalized black-box solver for* search- $\mathrm{MK^{t'}P}[s]$ *for every function $t'(\cdot)$ and every function $s(n) \leq 2n - \lceil \log n \rceil$.*

We highlight that generalized black-box solvers can solve MCSP (since, as implicitly observed in [HIW23] following [RS21; FM05]), the MCSP problem can be stated as an $\mathrm{MK^t_MP}$ problem with respect to a particular *non-universal* underlying TM M (performing circuit emulation)—see Lemma B.2 in the Appendix). As a direct corollary of Theorem 1.1 and Lemma B.2 we thus get:[2]

[1] [RS21] noted that the function inversion algorithm of [FN00] could be applied to the one-way function construction of [LP20] to get a non-trivial non-uniform RAM program that solves the $\mathrm{MK^tP}$ problem, but left open whether a circuit implementation can be given.

[2] When $s(n) \geq 1.1 \cdot n/logn$ then MCSP$[s]$ is the trivial language consisting of all strings due to the result of [Lup58], so the corollary below actually works for all s.

Corollary 1.2. *Let $p \in \text{poly}$ and $\alpha > 0$, and assume that for $t(n) = n$ there exists a black-box* $\text{MK}^{\text{t}}\text{P}[n-4]$ *solver of size $2^{\alpha n} \cdot p(n)$. Then for every $s(n) \leq 1.9n/\log n$,* search-MCSP$[s]$ *can be solved with a circuit family of size $2^{(\alpha+o(1))\cdot s(n)\cdot\log(s(n)+\log n)} \cdot \text{poly}(n)$.*

Additionally, we observe that generalized black-box solvers for search-$\text{MK}^{\text{t}}\text{P}[n]$ can easily be seen to be equivalent to function inversion circuits (for all functions f) of roughly the same size—see Lemmas A.1 and A.2 in the Appendix. As a corollary of Theorem 1.1, we thus get that—in the black-box regime—solving the function inversion problem is not only sufficent (as shown in [MP24; HIW23] for solving $\text{MK}^{\text{t}}\text{P}[n-O(1)]$ but also *necessary*. This matches the converse direction of the *average-case* characterization of one-way functions through the hardness of $\text{MK}^{\text{t}}\text{P}$ from [LP20], and yields a characterization of the *black-box* worst-case hardness of $\text{MK}^{\text{t}}\text{P}$ through the black-box worst-case hardness of one-way functions. In particular, black-box solving just $[n-O(1)]$ is no easier than (black-box) function inversion.

Theorem 1.3. *There exists a black-box* $\text{MK}^{\text{t}}\text{P}[n-O(1)]$ *solver of size $2^{\alpha n} \cdot \text{poly}(n)$ for every polynomial t if and only if every function f can be inverted by an f-oracle aided circuit of size $2^{\alpha n} \cdot \text{poly}(n)$.*

As a consequence of Theorem 1.3, and Impagliazzo's lower bound on the circuit size of black-box one-way function inversion[3], we directly get a lower bound on the complexity of black-box $\text{MK}^{\text{t}}\text{P}$ solvers; such a lower bound was previously proved directly for the $\text{MK}^{\text{t}}\text{P}$ problem in [MP24] but it required a significantly more complicated proof and employing heavier machinery.

Corollary 1.4. *There is no black-box* $\text{MK}^{\text{t}}\text{P}[n-4]$ *solver of size $2^{n/2-o(n)}$.*

1.2 Proof Outline

Theorem 1.1 is proved in two step. The first step is formally stated in Corollary 3.2 and the second in Corollary 4.2.

Step 1: From Black-box to Generalized Black-Box for Small Thresholds . We first that any black-box solver for $\text{MK}^{\text{t}}\text{P}[n-4]$ of size $T(n)$ implies a generalized black box $\text{MK}^{\text{t}}\text{P}[s(n)]$ solver of size $T(s(n)+O(1))\text{poly}(n, t(n))$.[4] The proof follows standard techniques from the literature on hardness magnification (i.e., hashing down the statement x using a pairwise independent hash function h to roughly the threshold size, and then applying the solver of a related language on the smaller instance $h(x)$ and thereby improving the running time) [OS18; CJW19; OPS21]. The key difference with our approach is that by leveraging the black-box property of the algorithm, we can use an algorithm for the *same* problem, but parameterized by a different universal TM M_h, as opposed to a general **NP** problem as in those earlier works—that is, we get "self hardness magnification" [LP21]). (We highlight that [HIW23] also rely on a similar hashing technique to *directly* present an attack on the threshold version of $\text{MK}^{\text{t}}\text{P}$ but do so in a slightly different context: in particular, they use hashing to develop a function inversion algorithm whose circuit complexity only depends on the input size of the function and not the output size, and next function inversion with an input size that depends on the threshold to solve $\text{MK}^{\text{t}}\text{P}[s]$. Nevertheless, our usage of this

[3]Impagliazzo shows that for every large enough $n \in \mathbb{N}$, there exists a permutation $\sigma\colon \{0,1\}^n \to \{0,1\}^n$ such that every σ-oracle aided circuit C of size at most $2^{n/2-2\log^2 n}$ fails to invert f

[4]This reduction also works for the search version of these problems.

approach is inspired by theirs.) Additionally, and perhaps more surprisingly, we show that this technique allows us to solve the orthogonal problem of dealing with *non-universal* Turing machines (so that we can get a generalized black-box solver): in essence, the idea is to define a universal TM M_h that has two tracks: if the first bit of the input "program" Π is 0, it simply runs some Universal TM $\mathsf{U}(\Pi_{>1})$ on the rest of the input $\Pi_{>1}$, and if it is 1, then it outputs $h(M(\Pi_{>1}))$ where M is the non-universal TM that we want a $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}[s]$ solver for. The key point is that due to pairwise independence property of the hash function, $h(x)$ is uniform (for a random choice of h) and thus with high probability $h(x)$ has essentially maximal K^t_{U} complexity, and thus the existence of the first "track" does not disrupt the hardness magnification reduction.

Step 2: From Decision to Search Our next result shows how any generalized $\mathrm{MK}^{\mathrm{t}}\mathrm{P}[s]$ solver of size $2^{\alpha s(n)}$ can be used to solve also the *search version* of the problem with roughly the same running time. In particular, to solve search- $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}[s]$, we will rely on a circuit deciding $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M_n}}\mathrm{P}[s+\lceil\log n\rceil]$ where M_n is defined as a TM that given a program $\Pi = (i, \Pi')$ where i is defined as the first $\lceil\log n\rceil$ bits of Π, checks if Π' generates an output x of exactly n bits, and if so outputs x *concatenated* with the first i bits of Π'. The key point is that for every n-bit length string x, $K^t_{M_n}(x) = K^t_M(x)+\lceil\log n\rceil$ (obtained by letting $i = 0$). Furthermore, this Kolmogorov complexity can be maintained if we concatenate the prefix of any minimum length program Π' that generates x, so the bits of any such minimum length program can be iteratively recovered given an oracle computing $K^t_{M_n}$. The same argument also works if we only have access to a decision oracle for the threshold $s + \lceil\log n\rceil$, but then we only recover a program of length at most s.

2 Definitions

Given some efficient threshold function s, let $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}[s]$ denote the set of strings x s.t. $\mathrm{K}^t_M(x) \le s(|x|)$ (where we let $\mathrm{K}^t_M(x) = \infty$ if there is no Π such that $M(\Pi, 1^t) = x$). Let search- $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}[s]$ denote the search problem in which given a string x with $\mathrm{K}^t_M(x) \le s(|x|)$, the output is a program Π of length at most $s(|x|)$ with $M(\Pi, 1^{t(n)}) = x$.

We start with the definition of a black-box emulator and a black-box universal TM.

Definition 2.1 (Black-box emulator)**.** *A function* $M : \{0,1\}^* \times 1^* \to \{0,1\}^* \cup \{\perp\}$*, is a* black-box TM emulator *if M has "unique outputs": For any* $\Pi \in \{0,1\}^*, t_1, t_2 \in \mathbb{N}, t_1 \le t_2$*, if* $M(\Pi, 1^{t_1}) \ne \perp$*,* $M(\Pi, 1^{t_2}) = M(\Pi, 1^{t_1})$*. A black-box TM emulator* U *is a* black-box universal Turing machine (black-box UTM) *if there exists a universal Turing machine* U_0 *such that for any* $(\Pi, 1^t)$*, if* Π *is a valid description of a Turing machine (w.r.t* U_0*), then* $\mathsf{U}(\Pi, 1^t) = \mathsf{U}_0(\Pi, 1^t)$*.*

We start with the definition of black-box solvers for $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}[s]$. For a TM M, a function $t\colon \mathbb{N} \to \mathbb{N}$, and a number $n \in \mathbb{N}$, we let $f^{M,t}_n\colon \{0,1\}^{\le 2n} \to \{0,1\}^*$ be the function defined by $f^{M,t}_n(\Pi) = M(\Pi, 1^{t(n)})$ for any $\Pi \in \{0,1\}^{\le 2n}$.

Definition 2.2 (Black-box $\mathrm{MK}^{\mathrm{t}}\mathrm{P}$-solver)**.** *For functions* $t, s, T\colon \mathbb{N} \to \mathbb{N}$*, we say that* $\mathrm{MK}^{\mathrm{t}}\mathrm{P}[s]$ *admits a* black-box $\mathrm{MK}^{\mathrm{t}}\mathrm{P}[s]$-solver *of size* $T(n)$ *if the following holds for every black-box universal TM* U*. There exists a circuit family* $\mathcal{C} = \{C_n\}_{n\in\mathbb{N}}$ *of size at most* $T(n)$*, such that for every* $n \in \mathbb{N}$*,* C_n *is a* $f^{\mathsf{U},t}_n$*-oracle aided circuit with* $\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[s]$ *on inputs of length* n*.*

We define generalized black-box solver in exactly the same way except that we quantify over all black-box TM emulator (as opposed to just universal ones).

Definition 2.3 (Generalized black-box $\mathrm{MK^tP}$-solver)**.** *For functions $t, s, T\colon \mathbb{N} \to \mathbb{N}$, we say that* $\mathrm{MK^tP}[s]$ *admits a* generalized black-box $\mathrm{MK^tP}[s]$-solver *of size $T(n)$ if the following holds for every black-box TM M. There exists a circuit family $\mathcal{C} = \{C_n\}_{n\in\mathbb{N}}$ of size at most $T(n)$, such that for every $n \in \mathbb{N}$, C_n is a $f_n^{M,t}$-oracle aided circuit that decides* $\mathrm{MK^t_M P}[s]$ *on inputs of length n.*

We similarly define black-box solvers and generalized black box solvers for search- $\mathrm{MK^tP}$.

3 Generalized Solvers Scaling with the Threshold

We show how to turn a black-box solver into a *generalized* black-box solver where the circuit size *scales with the threshold*. As mentioned before, proof follows standard techniques from the literature on hardness magnification (i.e., hashing down the statement to roughly the threshold size, and then applying the solver on the smaller instance and thereby improving the running time) [OS18; CJW19; OPS21].

Theorem 3.1. *There exists $q \in \mathrm{poly}$ such that the following holds. Let $T\colon \mathbb{N} \to \mathbb{N}$ be a function, and assume that for $t(n) = n$ there exists a black-box $\mathrm{MK^tP}[n-4]$ solver of size $T(n)$. Then, there exists a generalized black-box $\mathrm{MK^{t'}P}[s]$ solver of size $T(s(n)+5) \cdot q(n)$ for every function $s(\cdot)$ with $s(n) \leq 2n$ and for every function $t'(\cdot)$.*[5]

Proof of Theorem 3.1. Fix an efficient universal TM U, and let $p \in \mathrm{poly}$ be such that $p(n,t)$ bounds the size of a circuit implementing $\mathsf{U}(\Pi, 1^t)$ for inputs $(\Pi, 1^t)$ with $|\Pi| = n$. Let M, $s(n)$, and $t'(n)$ be the TM, time function and threshold for which we want to solve $\mathrm{MK^{t'}_M P}[s]$. Let $t(n) = n$. For every $n \in \mathbb{N}$, let $\mathcal{H}_n = \left\{h\colon \{0,1\}^n \to \{0,1\}^{s(n)+5}\right\}$ be a pairwise independent hash family, such that there exists $m \in \mathrm{poly}$ for which $m(n + s(n))$ bounds the circuit size evaluating h for every $h \in \mathcal{H}_n$. Fix $n \in \mathbb{N}$. We start by showing a distribution over circuits that solves $\mathrm{MK^{t'}_M P}[s]$ with good probability.

For every $h \in \mathcal{H}_n$, we define U_h to be the following black-box universal TM:

$$\mathsf{U}_h(\Pi, 1^t) = \begin{cases} \mathsf{U}(\Pi_{>1}, 1^t) & \text{if } \Pi_1 = 0 \\ h(M(\Pi_{>1}, 1^{t'(n)})) & \text{if } |\Pi| \leq 2n+1,\ \Pi_1 = 1 \text{ and } \left|M(\Pi_{>1}, 1^{t'(n)})\right| = n \\ \perp & \text{Otherwise} \end{cases}$$

By the assumption that there exists a black-box solver of size $T(n)$ for every black-box universal TM, there exists a circuit of size C_h^s of size $T(s(n)+5)$ that solves $\mathrm{MK^t_{\mathsf{U}_h}P}[n-4]$ on input of length $n' = s(n) + 5$, and using oracle to the function $f_n^h\colon \{0,1\}^{\leq 2n'} \to \{0,1\}^*$ defined by $f_n^h(\Pi) = \mathsf{U}_h(\Pi, 1^{t(n')})$.

Let C_h be the circuit that given input $x \in \{0,1\}^n$, computes $h(x)$ and outputs $C_h^s(h(x))$. We claim that for every $x \in \{0,1\}^n$,

1. if $\mathrm{K}^{t'}_M(x) \leq s(n)$, $C_h(x)$ outputs Yes for every h, and,
2. if $\mathrm{K}^{t'}_M(x) > s(n)$, it holds that for $h \leftarrow \mathcal{H}_n$, $C_h(x)$ outputs No with probability at least $3/4$.

[5] We remark that the theorem extends also to the search versions of the same problems with essentially identically the same proof. Since we later will show a generic decision-to-search reduction, we omit the details.

To see (1), consider any $x \in \{0,1\}^n$ s.t. $\mathrm{K}^{t'}_M(x) \le s(n)$. Then there exists a program Π of length at most $s(n)$ such that $M(\Pi, 1^{t'(n)}) = x$. Therefore $\mathsf{U}_h(1||\Pi, 1^{t(n)}) = h(M(\Pi, 1^{t'(n)})) = h(x)$ and thus $\mathrm{K}^t_{\mathsf{U}_h}(h(x)) \le s(n) + 1 \le n' - 4$, so $C_h(x)$ will always answer Yes.

For (2), consider any $x \in \{0,1\}^n$ s.t. $\mathrm{K}^{t'}_M(x) > s(n)$. We claim that with probability at least $3/4$ over the choice of a random $h \leftarrow \mathcal{H}_n$, it holds that $\mathrm{K}^t_{\mathsf{U}_h}(h(x)) > n' - 4$, which implies that $C_h(x)$ outputs No. To see the above, assume that for some h, $\mathrm{K}^t_{\mathsf{U}_h}(h(x)) \le n' - 4$. Then there exists Π such that $|\Pi| \le n' - 4$, and $\mathsf{U}_h(\Pi, 1^{t(n)}) = h(x)$. By the definition of U_h, it either holds that $\Pi_1 = 0$, and then $\mathrm{K}^t_{\mathsf{U}}(h(x)) \le n' - 5$, or $\Pi_1 = 1$, which means that $h(x) = h(x')$ for some x' with $\mathrm{K}^{t'}_M(x') \le n' - 5 = s(n)$. In the following we show that the probability that one of the above happens is at most $1/4$ (over a random choice of $h \leftarrow \mathcal{H}_n$). Indeed, since $\mathcal{H}_n$ is a pairwise independent family, $h(x)$ uniformly distributed when $h \leftarrow \mathcal{H}_n$. Therefore,

$$\Pr_{h \leftarrow \mathcal{H}_n}\left[\mathrm{K}^t_{\mathsf{U}}(h(x)) \le n' - 4\right] = \Pr_{y \leftarrow \{0,1\}^{n'}}\left[\mathrm{K}^t_{\mathsf{U}}(y) \le n' - 4\right] \le 2^{-3}.$$

Moreover, for every $x' \ne x$, it holds that $\Pr_{h \leftarrow \mathcal{H}_n}[h(x) = h(x')] \le 2^{-s(n)-5}$. By a union bound over all x' with $\mathrm{K}^{t'}_M(x') \le s(n)$, we get that the probability of collision $h(x) = h(x')$ with such x' is at most 2^{-4}. Using the union bound again, it holds that with probability at least $1 - 2^{-3} - 2^{-4} > 3/4$, both $\mathrm{K}^t_{\mathsf{U}}(h(x)) > n' - 4$ and there is no $x' \ne x$ with $\mathrm{K}^{t'}_M(x') \le s(n)$ such that $h(x) = h(x')$. In this case, $\mathrm{K}^t_{\mathsf{U}_h}(h(x)) > n' - 4$, and $C_h(h(x))$ answers No.

The proof now follows by simple amplification: for $h_1, \ldots, h_n \in \mathcal{H}_n$, let $C_{h_1,\ldots,h_n}$ be the circuit that computes $C_{h_1}(x), \ldots, C_{h_n}(x)$ and outputs No if one of the execution output No. It follows using a standard Union bound, that with positive probability over the random choice of $h_1, \ldots, h_n \leftarrow \mathcal{H}_n$, $C_{h_1,\ldots,h_n}$ outputs the right answer for all $x \in \{0,1\}^n$; thus, there exists a fixed choice of $h_1, \ldots, h_n$ that works for every input.

We finally bound the size of $C_{h_1,\ldots,h_n}$. We start with bounding the size of a circuit with f^h_n oracle, for every $h \in \{h_1, \ldots, h_n\}$. In this case,

$$\left|C_{h_1,\ldots,h_{n+1}}\right| \le n \cdot m(n + s(n)) + n \cdot |C^s_h| + O(n) \le n \cdot m(n + s(n)) + n \cdot T(s(n) + 5) + O(n).$$

Next, observe that each f^h_n oracle can be implemented using a circuit of size $m(n+s(n))+p(2n', 2n')$ using oracle to the function $f^{M,t}_n \colon \{0,1\}^{\le 2n} \to \{0,1\}^*$ defined by $f^{M,t'}_n(\Pi) = M(\Pi, 1^{t'(n)})$. Thus, the size of a $f^{M,t'}_n$-oracle aided circuit computing $C_{h_1,\ldots,h_{n+1}}$ is at most $T(s(n) + 5) \cdot q(n)$ for $q(n) = (m(n + s(n)))^2 + p(2(s(n) + 5), 2(s(n) + 5))$. $\square$

By taking $T(n) = 2^{\alpha \cdot n \cdot \mathrm{poly}(n)}$, we get the following corollary.

Corollary 3.2. *Assume the existence of a $2^{\alpha n} \cdot \mathrm{poly}(n)$-size black-box solver for $\mathrm{MK^tP}[n-4]$ for $t(n) = n$. Then there exists a generalized black box $\mathrm{MK^{t'}P}[s]$ solver of size $2^{\alpha \cdot s(n)} \cdot \mathrm{poly}(n)$ for all functions $t'(\cdot)$ and $s = s(n)$ with $s(n) \le 2n$.*

4 From Decision to Search

In this section we show that if there exists a non-trivial black box solver to $\mathrm{MK^tP}$, then such a solver (with roughly the same efficiency) exists also for search-$\mathrm{MK^tP}$.

Theorem 4.1. *There exists $q \in \text{poly}$ such that the following holds. Let $p\colon \mathbb{N} \to \mathbb{N}$ be a monotone function, and let $T\colon \mathbb{N} \to \mathbb{N}$ and $t\colon \mathbb{N} \to \mathbb{N}$ be functions. Assume that for every $s\colon \mathbb{N} \to \mathbb{N}$ with $s(n) \leq 2n$ there exists a generalized black-box $\mathrm{MK^tP}[s]$ solver of size $T(s(n)) \cdot p(n)$. Then, there exists a generalized black-box search-$\mathrm{MK^tP}[s]$ solver of size $T(s(n) + \lceil \log n \rceil) \cdot p(n + s(n)) \cdot q(n)$ for every $s\colon \mathbb{N} \to \mathbb{N}$ such that $s(n) \leq 2n - \lceil \log n \rceil$.*

Proof. Let M be a black-box TM emulator, and for every $n \in \mathbb{N}$, let M_n be the following black-box TM emulator. Given Π and $1^{t'}$, M_n interprets $\Pi = i||\Pi'$, where the first $\lceil \log n \rceil$ bits of Π interpreted as an index $i \in [n]$, and the rest of the bits interpreted as a program Π'. Then, M_n acts as follows:

$$M_n(i, \Pi', 1^{t'}) = \begin{cases} M(\Pi', 1^{t(n)})||\Pi'_{\leq i} & \text{if } i \leq n,\ |\Pi'| \leq 2n \text{ and } \left|M(\Pi', 1^{t(n)})\right| = n \\ \perp & \text{Otherwise} \end{cases}$$

We observe that for every x, for every $i \in [n]$, and for every program Π' of length $\ell \leq 2n$ such that $M(\Pi', 1^t) = x$, it holds that $\mathrm{K}^t_M(x) + \lceil \log n \rceil \leq \mathrm{K}^t_{M_n}(x||\Pi'_{\leq i}) \leq \ell + \lceil \log n \rceil$. In particular, assuming that $\mathrm{K}^t_M(x) \leq 2n$, for the minimal-length program Π' such that $M(\Pi', 1^t) = x$ it holds that $\mathrm{K}^t_{M_n}(x||\Pi'_{\leq i}) = \mathrm{K}^t_M(x) + \lceil \log n \rceil$. Moreover, for every $z \in \{0,1\}^*$ such that z is not a prefix of a program of length at most ℓ that outputs x, it holds that $\mathrm{K}^t_{M_n}(x||z) > \ell$. We can thus use an algorithm that decides $\mathrm{MK^t_{M_n}P}$ to find a program Π of length at most s such that $M(\Pi, 1^t) = x$. This can be done by the following process:

1. Check if $K^t_{M_n}(x) \leq s(n) + \lceil \log n \rceil$. If not output $\perp$.
2. Let $z = \perp$.
3. For every $i \in [s(n)]$:
   (a) Check if $M(z, 1^t) = x$. If it does, output z.
   (b) Check if $\mathrm{K}^t_{M_n}(x||z||0) \leq s(n) + \lceil \log n \rceil$, let $z = z||0$. Otherwise let $z = z||1$.
4. Output z.

Since $\mathrm{K}^t_{M_n}(x||z||0) \leq s(n) + \lceil \log n \rceil$ if and only if z is a prefix of a program Π of length at most s such that $M(\Pi, 1^t) = x$, the above process always finds such a program. We left to show that the above process can be implemented using a circuit of size $T(s(n) + \lceil \log n \rceil) \cdot p(n + s(n)) \cdot \text{poly}(n)$.

Let s' be the function defined by $s'(k) = 1$ for every $k < n$, and $s'(k) = s(n) + \lceil \log n \rceil$ otherwise. Then if $s(n) \leq 2n - \lceil \log n \rceil$, it holds that $s'(k) \leq 2k$. By our assumption, for every n' there exists a $f^{M_n,t}_{n'}$-oracle aided circuit $C_{n'}$ of size $T(s'(n)) \cdot p(n')$ that decides $\mathrm{MK^t_{M_n}P}[s']$ on inputs of lenght n'. We observe that the above process can be implemented with one call to each of $C_{n'}$, for $n' \in \{n, \ldots, n + s(n)\}$. Moreover, the $f^{M_n,t}_{n'}$-oracle can be implemented by a poly-size circuit using an $f^{M,t}_n$-oracle. Thus, the above process can be implemented using a circuit of size at most $s(n) \cdot T(s(n) + \lceil \log n \rceil) \cdot p(n + s(n)) \cdot \text{poly}(n)$, as required. □

By taking $T(s(n)) = 2^{\alpha \cdot s(n)}$, we get the following corollary.

Corollary 4.2. *Assume there exists a generalized black box $\mathrm{MK^tP}[s]$ solver of size $2^{\alpha \cdot s(n)} \cdot \text{poly}(n)$ for all functions $t(\cdot)$ and $s = s(n)$ with $s(n) \leq 2n$. Then there exists a $2^{\alpha s(n)} \cdot \text{poly}(n)$-size generalized black-box solver for search-$\mathrm{MK^{t'}P}[s]$ for every function $t'(\cdot)$ and every function $s(n) \leq 2n - \lceil \log n \rceil$.*

References

[Cha69] Gregory J. Chaitin. "On the Simplicity and Speed of Programs for Computing Infinite Sets of Natural Numbers". In: *J. ACM* 16.3 (1969), pp. 407–422 (cit. on p. 2).

[CJW19] Lijie Chen, Ce Jin, and R Ryan Williams. "Hardness magnification for all sparse NP languages". In: *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2019, pp. 1240–1255 (cit. on pp. 4, 6).

[FM05] Gudmund Skovbjerg Frandsen and Peter Bro Miltersen. "Reviewing bounds on the circuit size of the hardest functions". In: *Information processing letters* 95.2 (2005), pp. 354–357 (cit. on pp. 3, 11).

[FN00] Amos Fiat and Moni Naor. "Rigorous time/space trade-offs for inverting functions". In: *SIAM Journal on Computing* 29.3 (2000), pp. 790–803 (cit. on p. 3).

[Har83] J. Hartmanis. "Generalized Kolmogorov complexity and the structure of feasible computations". In: *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*. 1983, pp. 439–445. DOI: `10.1109/SFCS.1983.21` (cit. on p. 2).

[HIW23] Shuichi Hirahara, Rahul Ilango, and Ryan Williams. *Beating Brute Force for Compression Problems*. Tech. rep. TR23-171. Electronic Colloquium on Computational Complexity, 2023 (cit. on pp. 2–4, 11).

[KC00] Valentine Kabanets and Jin-yi Cai. "Circuit minimization problem". In: *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*. 2000, pp. 73–79 (cit. on p. 2).

[Ko86] Ker-I Ko. "On the Notion of Infinite Pseudorandom Sequences". In: *Theor. Comput. Sci.* 48.3 (1986), pp. 9–33. DOI: `10.1016/0304-3975(86)90081-2`. URL: `https://doi.org/10.1016/0304-3975(86)90081-2` (cit. on p. 2).

[Kol68] A. N. Kolmogorov. "Three approaches to the quantitative definition of information". In: *International Journal of Computer Mathematics* 2.1-4 (1968), pp. 157–168 (cit. on p. 2).

[LP20] Yanyi Liu and Rafael Pass. "On one-way functions and Kolmogorov complexity". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2020, pp. 1243–1254 (cit. on pp. 2–4).

[LP21] Yanyi Liu and Rafael Pass. "Cryptography from sublinear-time average-case hardness of time-bounded Kolmogorov complexity". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 722–735 (cit. on pp. 2, 4).

[Lup58] Oleg B Lupanov. "On a method of circuit synthesis". In: *Izvestia VUZ* 1 (1958), pp. 120–140 (cit. on p. 3).

[MP24] Noam Mazor and Rafael Pass. "The Non-Uniform Perebor Conjecture for Time-Bounded Kolmogorov Complexity is False". In: *15th Innovations in Theoretical Computer Science* (2024) (cit. on pp. 2–4, 10).

[OPS21] Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. "Hardness magnification near state-of-the-art lower bounds". In: *Theory OF Computing* 17.CCC 2019 Special Issue (2021) (cit. on pp. 4, 6).

[OS18] Igor Carboni Oliveira and Rahul Santhanam. "Hardness magnification for natural problems". In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2018, pp. 65–76 (cit. on pp. 4, 6).

[RS21] Hanlin Ren and Rahul Santhanam. "Hardness of KT Characterizes Parallel Cryptography". In: *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2021 (cit. on pp. 2, 3, 11).

[Sip83] Michael Sipser. "A Complexity Theoretic Approach to Randomness". In: *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*. 1983, pp. 330–335 (cit. on p. 2).

[Sol64] R.J. Solomonoff. "A formal theory of inductive inference. Part I". In: *Information and Control* 7.1 (1964), pp. 1 –22. ISSN: 0019-9958. DOI: `https://doi.org/10.1016/S0019-9958(64)90223-2` (cit. on p. 2).

[Tra84] Boris A Trakhtenbrot. "A survey of Russian approaches to perebor (brute-force searches) algorithms". In: *Annals of the History of Computing* 6.4 (1984), pp. 384–400 (cit. on p. 2).

A search- $\mathrm{MK^tP}$ and Function Inversion

We observe that generalized black-box solvers for search- $\mathrm{MK^tP}$ directly yield function inverters with roughly the same complexity, and vice versa.

Lemma A.1. *There exists $p \in \mathrm{poly}$ such that the following holds. Assume that for some $t = t(n)$ there exists a generalized black-box search- $\mathrm{MK^tP}[n]$ solver of size $T = T(n)$. Then for every function $\pi\colon \{0,1\}^n \to \{0,1\}^n$ there exists a π-oracle aided circuit of size $T(n) \cdot p(n)$ that inverts π.*

Proof. Let M be the black box TM that on input $x \in \{0,1\}^n, 1^t$ outputs $y = \pi(x)$. By assumption, there exists a $f_n^{M,t}$-oracle aided circuit of size at most $T(n)$, that given an input $y \in \{0,1\}^n$ finds an input x of length at most n, such that $M(x, 1^{t(n)}) = y$, if such exists. Since $M(x, 1^{t(n)}) = \pi(x)$, such an x is a pre-image of y. Moreover, by the definition of M, the $f_n^{M,t}$-oracle can be implemented efficiently using a π-oracle. $\square$

The converse of Lemma A.1 was implicitly proven in [MP24]; we repeat the proof for the convenience of the reader.

Lemma A.2. *There exists $p \in \mathrm{poly}$ such that the following holds. Assume that for every function $\pi\colon \{0,1\}^n \to \{0,1\}^n$ there exists a π-oracle aided circuit with of size $T(n)$ that inverts π with probability 1 (for every $y = \pi(x)$, $f(C(y)) = y$). Then there exists a black-box $\mathrm{MK^tP}[s]$ solver of size $T(n + \lceil \log n \rceil) \cdot p(n)$ for every $t\colon \mathbb{N} \to \mathbb{N}$ and every $s\colon \mathbb{N} \to \mathbb{N}$ with $s(n) \le n$.*

Proof. Let U be a black-box universal TM. Let ${f'}_n\colon \{0,1\}^n \times [n] \to \{0,1\}^n \times [n]$ be defined as

$$
{f'}_n(\Pi, i) = \begin{cases} (\mathsf{U}(\Pi_{\le i}, 1^{t(n)}), i) & \left|\mathsf{U}(\Pi_{\le i}, 1^{t(n)})\right| = n \\ 0^n & \text{Otherwise} \end{cases}
$$

Let $n' = n + \lceil \log n \rceil$. In the following, we assume that both the domain and the range of f'_n is $\{0,1\}^{n'}$, by the use of appropriate encoding and padding. By assumption, there is a circuit family $\widehat{\mathcal{C}} = \left\{\widehat{C}_n\right\}_{n\in\mathbb{N}}$ with ${f'}_n$ oracle, of size $T(n')$ that inverts ${f'}_n$ with probability 1.

Given a circuit $\widehat{C}_n$ that inverts ${f'}_n$, we can construct a (${f'}_n$-oracle aided) circuit C_n that computes the $\mathrm{K}^t_{\mathsf{U}}$ complexity of any string x of length n with $\mathrm{K}^t_{\mathsf{U}}(x) \le n$. This can be done by computing ${f'}_n^{-1}(x, 1), \ldots, {f'}_n^{-1}(x, n)$ and outputing Yes if there exists (Π, i) such that $\mathsf{U}(\Pi_{<i}, 1^{t(n)}) = x$ and $i \le s(n)$ (the t-bounded Kolmogorov complexity of the string 0^n can be hardcoded in the circuit).

Observe that the size of C_n is $n' \cdot \left|\widehat{C}_n\right| + \mathrm{poly}(n)$. Thus, there exists a circuit family of size $n' \cdot T(n') + \mathrm{poly}(n) = T(n') \cdot \mathrm{poly}(n)$, with ${f'}_n$ oracle, that solves $\mathrm{MK}^{\mathrm{t}}_{\mathsf{U}}\mathrm{P}[s]$. Lastly, observe that f'_n can be efficiently computed from $f^{\mathsf{U},t}_n$, thus we can replace the f'_n oracle with a small circuit using an $f^{\mathsf{U},t}_n$-oracle, to get a circuit of size $T(n + \lceil \log n \rceil)\mathrm{poly}(n)$. ☐

B $\mathrm{MCSP}[s]$ as a special case of $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}$

We note that any generalized black-box $\mathrm{MK}^{\mathrm{t}}\mathrm{P}[s]$ solver can be used to solve $\mathrm{MCSP}[s]$. In fact, we observe that the $\mathrm{MCSP}[s]$ problem can be formulated as a $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}[s']$ instance for a particular choice of an (efficient but non-universal) TM M, and for a function $s'(n) \approx s(n)$.

Towards this, we will rely on the fact that circuits can be *succinctly* encoded as bit strings from which the circuit can be efficiently decoded. In particular, as observed in [RS21; HIW23], the encoding from [FM05] satisfies this requirement.

Lemma B.1 (Implicit in [FM05], see also [RS21; HIW23])**.** *There exists an efficiently computable function $\ell(s, k) \in (1 + o(1))(s \cdot \log(s + k))$ such that ℓ is monotone in s and the following holds. There exists an efficient algorithm Dec, such that for every circuit $C \colon \{0,1\}^k \to \{0,1\}$ of size s, there exists $x \in \{0,1\}^{\ell(s,k)}$ such that $Dec(x)$ is a circuit of size s that computes the same function as C. Moreover, for every x such that $Dec(x)$ outputs a circuit $C \colon \{0,1\}^k \to \{0,1\}$ of size s, it holds that $|x| = \ell(s, k)$.*[6]

We now observe that the MCSP is a special-case of the $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}$ problem for a specific choice of the TM M.

Lemma B.2. *There exists an efficient TM M such that the following holds for every $s \colon \mathbb{N} \to \mathbb{N}$ and every $t \colon \mathbb{N} \to \mathbb{N}$ with $t(n) \ge n$. Deciding $\mathrm{MCSP}[s]$ is equivalent to deciding $\mathrm{MK}^{\mathrm{t}}_{\mathrm{M}}\mathrm{P}[s']$, for $s'(n) = \ell(s(n), \lfloor \log n \rfloor)$.*

Proof. Let Dec be the function from Lemma B.1, and let M be the TM that given an input $x, 1^t$, computes $Dec(x)$ to get a circuit $C \colon \{0,1\}^k \to \{0,1\}$. If x is not valid encoding of a circuit, or $2^k \ne |x|$, M outputs $\perp$. If $t \le 2^k$, M also outputs $\perp$. Otherwise, M outputs the truth table of C. Since ℓ is monotone in s, there exists a program of length less then $s'(n)$ if and only if there exists a circuit of size less than $s(n)$ for x. ☐

[6] Note that we here requires the length of an encoding of a circuit of size s to be *exactly* $\ell(s, k)$ (in contrast to bounded by $\ell(s, k)$). As far as we can tell, this property has not been previously stated but it can be assumed without loss of generality using padding, and by assuming that given an input x, *Dec* only outputs a circuit C of size s if it holds that $|x| = \ell(s, k)$, or outputs $\perp$ otherwise.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il