Electronic Colloquium on Computational Complexity, Report No. 148 (2017)

The Choice and Agreement Problems of a Random Function

Or Meir[*] Avishay Tal[†]

October 6, 2017

Abstract

The direct-sum question is a classical question that asks whether performing a task on m independent inputs is m times harder than performing it on a single input. In order to study this question, Beimel et. al [BBKW14] introduced the following related problems:

- **The choice problem:** Given m distinct instances, choose one of them and solve it.
- **The agreement problem:** Given m distinct instances, output a solution that is correct for at least one of them.

It is easy to see that these problems are no harder than performing the original task on a single instance, and it is natural to ask *whether it is strictly easier or not.* In particular, proving that the choice problem is not easier is necessary for proving a direct-sum theorem, and is also related to the KRW composition conjecture [KRW95].

In this note, we observe that in a variety of computational models, if f is a random function then with high probability its corresponding choice and agreement problem are not asymptotically easier than computing f on a single instance (as long as m is noticeably smaller than 2^n).

1 Introduction

The direct-sum question is a classical question that asks whether performing a task on m independent inputs is m times harder than performing it on a single input. More generally, one may ask whether performing multiple independent tasks in parallel is as hard as performing each of them separately. It will be convenient to use the following notation.

Definition 1. Let $T_1, \ldots, T_m$ be computational tasks. The direct-sum problem $\mathsf{sum}_{T_1,\ldots,T_m}$ is the following task: given m inputs $x_1, \ldots, x_m$ for $T_1, \ldots, T_m$ respectively, output a vector $(y_1, \ldots, y_m)$ such that y_i is a correct solution for T_i on x_i for every i.

The direct-sum question asks whether the complexity of $\mathsf{sum}_{T_1,\ldots,T_m}$ is the sum of the individual complexities of $T_1, \ldots, T_m$. This natural question was studied in a variety of computational models (see, e.g., [Uhl74, Pau76, GF81, Sto86, Bsh89, Bsh98]), and the answer turns out to be positive in some models and negative in others. In order to study this question, Beimel et. al. [BBKW14] (following Ambainis et. al. [ABG$^+$01]) considered the following related problems.

Definition 2 ([BBKW14])**.** Let $T_1, \ldots, T_m$ be computational tasks.

[*] Department of Computer Science, Haifa University, Haifa 31905, Israel. ormeir@cs.haifa.ac.il. Partially supported by the Israel Science Foundation (grant No. 1445/16).

[†] Department of Computer Science, Stanford University, Stanford, CA 94305. avishay.tal@gmail.com. Partially supported by a Motwani Postdoctoral Fellowship and by NSF grant CCF-1749750.

ISSN 1433-8092

- The *choice problem* $\mathsf{choose}_{T_1,\ldots,T_m}$ is the following task: Given m inputs $x_1, \ldots, x_m$ for $T_1, \ldots, T_m$ respectively, output a pair (i, y) such that y is a correct solution for T_i on x_i.
- The *agreement problem* $\mathsf{agree}_{T_1,\ldots,T_m}$ is the following task: Given m inputs $x_1, \ldots, x_m$ for $T_1, \ldots, T_m$ respectively, output a value y such that y is a correct solution for T_i on x_i for some $i \in [m]$.

It is easy to see that the agreement task is not harder than the choice task, and that both tasks are not harder than performing the easiest task T_i on a single input. [BBKW14] asked whether we can prove that the choice and agreement problem are *not strictly easier than the easiest task* T_i. In addition to being interesting in its own right, this question has the following motivations:

- Proving that that the choice problem $\mathsf{choose}_{T_1,\ldots,T_m}$ is *not* strictly easier than the easiest task T_i is *necessary for proving a direct-sum theorem*: To see it, observe that if the choice problem $\mathsf{choose}_{T_1,\ldots,T_m}$ is strictly easier than the easiest task T_i, then the complexity of $\mathsf{sum}_{T_1,\ldots,T_m}$ is strictly less than the sum of the individual complexities of $T_1, \ldots, T_m$ (since one can solve one of the tasks using the best algorithm for $\mathsf{choose}_{T_1,\ldots,T_m}$, and then solve each of the remaining tasks individually).
- Sine the agreement problem $\mathsf{agree}_{T_1,\ldots,T_m}$ is not harder than the choice problem $\mathsf{choose}_{T_1,\ldots,T_m}$, one can prove lower bounds for $\mathsf{choose}_{T_1,\ldots,T_m}$ by proving lower bounds for $\mathsf{agree}_{T_1,\ldots,T_m}$.

In this note, we consider the special case where all the tasks $T_1, \ldots, T_m$ are the same task T. Observe that in this case special case, it is trivial to prove that the choice and agreement problems are not strictly easier than solving T for the following reason: Solving T on an input x reduces to solving $\mathsf{choose}_{T,\ldots,T}$ and $\mathsf{agree}_{T,\ldots,T}$ on m copies of x. In order to avoid this trivial case, we require the m inputs to be distinct, resulting in the following definition.

Definition 3. Let T be a computational task.

- The *m-fold choice problem* choose_T^m is the following task: Given m distinct inputs $x_1, \ldots, x_m$ for T, output a pair (i, y) such that y is a correct solution for T on x_i.
- The *m-fold agreement problem* agree_T^m is the following task: Given m distinct inputs $x_1, \ldots, x_m$ for T, output a value y such that y is a correct solution for T on x_i for some $i \in [m]$.

It is again natural to ask whether choose_T^m and agree_T^m are strictly easier than T on its own. In particular, note that the foregoing motivation still holds: proving that choose_T^m is not easier than T is necessary for proving a direct-sum theorem for T (i.e., that the complexity of $\mathsf{sum}_{T,\ldots,T}$ is m times the complexity of T).

In this note, we observe that in a variety of computational models, the answer to this question is negative when T is the task of computing a *random function* f, with high probability over f. Intuitively, this result holds in every model in which the hardness of a random function can be proved using a counting argument, including Boolean circuits, formulas, decision trees, etc.

In order to make this intuition more precise, consider a computational model that comes with some size measure (e.g., number of wires for circuits, depth for decision trees, etc.). We use the term *computer* to refer to a specific instantiation of this model (e.g. a specific circuit, a specific decision tree, etc.). Let $N(s, n)$ denote the number of distinct Boolean functions over n bits that are computed by a computer of size at most s. Then, the standard counting argument says the probability that a random function over n bits can be computed by a computer of size at most s is at most

$$\frac{N(s,n)}{2^{2^n}}.$$

We prove the following observation.

Theorem 4. *Fix a computational model that comes with some size measure, and let $N(s,n)$ be defined as above. Let $n,m,s \in \mathbb{N}$, and let $f : \{0,1\}^n \to \{0,1\}$ be a uniformly distributed function. Then, the probability that choose_f^m or agree_f^m can be decided by a computer of size at most s is at most*

$$\frac{N(s,n) \cdot \binom{2^n}{\leq 2m-2}}{2^{2^n}}$$

where $\binom{a}{\leq b} \stackrel{\text{def}}{=} \binom{a}{0} + \binom{a}{1} + \binom{a}{2} + \ldots + \binom{a}{b}$ for non-negative integers $a \geq b$.

Observe that the expression $\binom{2^n}{\leq 2m-2}$ in the latter probability is negligible compared to 2^{2^n} unless m is very close to 2^n. Thus, as long as m is not too large, choose_f^m and agree_f^m will have asymptotically the same complexity as that of a random function. The following corollary lists the immediate consequences of Theorem 4 for some important computational models. Essentially, it says that in all those models choose_f^m and agree_f^m are as hard as a random function.

Corollary 5. *There exists a universal constant $\varepsilon > 0$ such that the following holds. Let $f : \{0,1\}^n \to \{0,1\}$ be a uniformly distributed function and let $m \leq \varepsilon \cdot 2^n$. Then, each of the following events occurs with probability $1 - o(1)$ (where the $o(1)$ is a decreasing function of n):*

- *The circuit complexity of choose_f^m and agree_f^m is $\Omega(\frac{2^n}{n})$.*
- *The formula complexity of choose_f^m and agree_f^m is $\Omega(\frac{2^n}{\log n})$.*
- *The depth complexity of choose_f^m and agree_f^m is $n - O(\log\log n)$ (for circuits with fan-in 2).*
- *The decision tree complexity of choose_f^m and agree_f^m is $n - O(\log\log n)$.*

The rest of this note is organized as follows: In Section 2, we prove Theorem 4. Then, in Section 3, we discuss a connection of this result to the KRW conjecture [KRW95], which provides an additional motivation for our result.

2 Proof of the Theorem 4

In this section we prove Theorem 4. We first observe that it suffices to prove the theorem for the agreement problem agree_f^m, since the choice problem choose_f^m is not easier than agree_f^m. Following [BBKW14], we also observe that when the task T is a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the agreement problem can be rephrased as follows:

- Given $x_1, \ldots, x_m \in \{0,1\}^n$ that satisfy the promise that $f(x_1), \ldots, f(x_m)$ are all equal, decide whether they are equal to 0 or 1.

The reason is that if $f(x_1), \ldots, f(x_m)$ are not all equal, then both 0 and 1 are correct outputs of agree_f^m. This observation is the reason for the name "agreement problem".

Fix a computational model with some associated size measure, and let $N(s,n)$ denote the number of distinct functions that are computed by a computer of size at most s. Let $m,n,s \in \mathbb{N}$.

We would like to prove that if $f : \{0,1\}^n \to \{0,1\}$ is a uniformly distributed function, then the probability that its agreement problem agree_f^m is decided by a computer of size at most s is at most

$$\frac{N(s,n) \cdot \binom{2^n}{\le 2m-2}}{2^{2^n}}.$$

To this end, it suffice to prove the following result.

Proposition 6. *Every computer decides the m-fold agreement problem of at most $\binom{2^n}{\le 2m-2}$ functions from $\{0,1\}^n \to \{0,1\}$.*

The rest of this section is devoted to proving the latter proposition. Let C be a computer, and let $\mathcal{F}$ be the family of functions from $\{0,1\}^n$ to $\{0,1\}$ whose agreement problem is decided by C. We wish to upper bound $|\mathcal{F}|$. The crux of the proof is the following observation.

Claim 7. *Every two functions $f, g \in \mathcal{F}$ differ on at most $2m-2$ inputs.*

Proof. We prove the contrapositive. Suppose that f and g differ on more than $2m-2$ inputs, and let C be a computer. We will prove that the computer C fails to solve either agree_f^m or agree_g^m.

Our assumption implies that there must be at least m inputs $x_1, \ldots x_m$ and some bit $b \in \{0,1\}$ such that $f(x_1) = \cdots = f(x_m) = b$ and $g(x_1) = \cdots = g(x_m) = 1-b$ — this follows from the pigeonhole principle, by mapping each input x on which f and g differ to the pigeonhole $(f(x), g(x))$. Thus, the output of agree_f^m on $x_1, \ldots, x_m$ is b and the output of agree_g^m on $x_1, \ldots, x_m$ is $1-b$. Now, suppose we give $x_1, \ldots, x_m$ to the computer C. If C outputs b, then it errs with respect to agree_g^m, and otherwise it errs with respect to agree_f^m. Either way, it fails to solve one of these problems. ■

We turn to upper bound the size of $\mathcal{F}$. We start by fixing an arbitrary function $f \in \mathcal{F}$. Then, in order to choose any function $g \in \mathcal{F}$, we only have to choose on which inputs it disagrees with f. There are at most

$$\binom{2^n}{0} + \binom{2^n}{1} + \binom{2^n}{2} + \ldots + \binom{2^n}{2m-2} = \binom{2^n}{\le 2m-2}$$

such possibilities, as required.

3 Connection to the KRW composition conjecture

The KRW conjecture. One of the major challenges in the quest for proving circuit lower bounds is to find an explicit function that requires circuits of super-logarithmic depth (here, we consider Boolean circuits with fan-in 2). Karchmer, Raz, and Wigderson [KRW95] proposed to attack this question by considering the composition of Boolean functions: given two functions $f : \{0,1\}^m \to \{0,1\}$, $g : \{0,1\}^n \to \{0,1\}$, their composition $f \diamond g : (\{0,1\}^n)^m \to \{0,1\}$ is defined by

$$(f \diamond g)(x_1, \ldots, x_m) = f(g(x_1), \ldots, g(x_m)).$$

The KRW conjecture says that the depth complexity of $f \diamond g$ is roughly the sum of the depth complexities of f and g, provided that f and g are non-constant functions. [KRW95] showed that if proved, this conjecture will imply the desired super-logarithmic depth lower-bounds. In fact, the conclusion will follow even if the conjecture is proved only for the case where f is a random function, or for the case where g is a random function. We will see that the question of whether the KRW conjecture holds for a random g is related to the complexity of agree_g^m,

Karchmer-Wigderson relations. [KRW95] suggested to study their conjecture using the framework of Karchmer-Wigderson relations: Karchmer and Wigderson [KW90] observed that for every Boolean function f there is a related communication problem KW_f, called the *Karchmer-Wigderson relation of* f, whose (deterministic) communication complexity is exactly equal to the depth complexity of f. Hence, one can prove depth lower-bounds for f by proving communication-complexity lower-bounds for KW_f. In particular, an equivalent formulation of the KRW conjecture says that the communication complexity of $KW_{f\diamond g}$ is roughly equal to the sum of the communication complexities of KW_f and KW_g.

The relation KW_f is defined as follows: Alice gets an input $x \in f^{-1}(0)$, and Bob gets as input $y \in f^{-1}(1)$. Clearly, it holds that $x \neq y$. The goal of Alice and Bob is to find a coordinate i such that $x_i \neq y_i$. Note that there may be more than one possible choice for i, which means that KW_f is a relation rather than a function.

The Karchmer-Wigderson relation of $f \diamond g$. In the relation $KW_{f\diamond g}$, Alice and Bob's inputs are conveniently viewed as $m \times n$ matrices X, Y respectively, such that $g(X) \in f^{-1}(0)$ and $g(Y) \in f^{-1}(1)$, where $g(X) \in \{0,1\}^m$ is obtained by applying g to each row of X and similarly $g(Y)$. Their goal is to find an entry (i,j) such that $X_{i,j} \neq Y_{i,j}$. A naive protocol for Alice and Bob is as follows. Alice computes $a = g(X)$ and Bob computes $b = g(Y)$. In the first stage they solve KW_f on a, b and find an index $i \in [m]$ where $a_i \neq b_i$. Then, in the second stage, they solve KW_g on inputs X_i, Y_i to find j as required. Note that the communication complexity of this protocol is indeed the sum of the communication complexities of KW_f and KW_g, so the KRW conjecture says that this protocol is essentially optimal.

The intuition for why the above protocol should be optimal[1] is the following: It appears that Alice and Bob must solve KW_g on some row in order to find an entry on which X and Y differ, and it seems that the only way to find a row on which they can solve KW_g is to solve KW_f on a and b.

The choice problem of KW_g. Dinur and Meir [DM16] pointed out one potential weakness in this intuition: when Alice and Bob solve KW_f, they may end up finding *multiple* indices i for which $a_i \neq b_i$. This means that there are now multiple rows on which they can solve KW_g, and they only need to solve KW_g on one of these rows. In other words, Alice and Bob *now face the choice problem of* KW_g. Thus, if we are to prove that the foregoing intuition is correct, it seems that we should prove that the choice problem $\mathsf{choose}^m_{KW_g}$ is not easier than solving KW_g on a single instance.

The connection to our result. We are therefore interested in the communication complexity of $\mathsf{choose}^m_{KW_g}$. Beimel et. al. [BBKW14] observed that the communication complexity of the choice problem $\mathsf{choose}^m_{KW_g}$ is exactly equal to the depth complexity of the *agreement* problem agree^m_g (this follows directly from the reasoning of [KW90]). Now, Theorem 4 says that if g is a random function, then agree^m_g is not significantly easier than g. In particular, Corollary 5 implies the following result.

Corollary 8. *There exists a universal constant $\varepsilon > 0$ such that the following holds. Let $g : \{0,1\}^n \to \{0,1\}$ be a uniformly distributed function and let $m \leq \varepsilon \cdot 2^n$. Then, with probability at least $1 - o(1)$, the deterministic communication complexity of $\mathsf{choose}^m_{KW_g}$ is at least $n - O(\log\log n)$.*

[1] More precisely, close to the optimal protocol, since there are functions for which some savings is possible. One such example is the function T^n_2, the threshold function checking if there are at least 2 ones in an input string of length n. The communication complexity of $T^n_2 \diamond T^n_2$ is strictly smaller than twice the communication complexity of T^n_2, but the saving is rather small (an additive $O(\log\log n)$ saving). See [KRW95] for more details.

This corollary is interesting, since even if we restrict the KRW conjecture to the case where g is a random function and where $m \leq \varepsilon \cdot 2^n$, the conjecture still implies the desired lower bounds on depth complexity. We therefore hope that Corollary 8 constitutes a small step toward new lower bounds.

Acknowledgement. We would like to thank Sajin Koroth for useful ideas and discussions. Part of this work was done during a workshop at Schloss Dagstuhl Seminar 17121 – we wish to thank Schloss Dagstuhl and the workshop organizers for creating a great atmosphere for performing research.

References

[ABG+01] Andris Ambainis, Harry Buhrman, William I. Gasarch, Bala Kalyanasundaram, and Leen Torenvliet. The communication complexity of enumeration, elimination, and selection. *J. Comput. Syst. Sci.*, 63(2):148–185, 2001.

[BBKW14] Amos Beimel, Sebastian Ben Daniel, Eyal Kushilevitz, and Enav Weinreb. Choosing, agreeing, and eliminating in communication complexity. *Computational Complexity*, 23(1):1–42, 2014.

[Bsh89] Nader H. Bshouty. On the extended direct sum conjecture. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 177–185, 1989.

[Bsh98] Nader H. Bshouty. On the direct sum conjecture in the straight line model. *J. Complexity*, 14(1):49–62, 1998.

[DM16] Irit Dinur and Or Meir. Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 3:1–3:51, 2016.

[GF81] Giulia Galbiati and Michael J. Fischer. On the complexity of 2-output boolean networks. *Theor. Comput. Sci.*, 16:177–185, 1981.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5(3/4):191–204, 1995.

[KW90] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discrete Math.*, 3(2):255–265, 1990.

[Pau76] Wolfgang J. Paul. Realizing boolean functions on disjoint sets of variables. *Theor. Comput. Sci.*, 2(3):383–396, 1976.

[Sto86] Quentin F. Stout. Meshes with multiple buses. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 264–273, 1986.

[Uhl74] D Uhlig. On the synthesis of self-correcting schemes from functional elements with a small number of reliable elements. *Matematicheskie Zametki*, 15(6):937–944, 1974.

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