Symmetry and Quantum Query-to-Communication Simulation

Sourav Chakraborty
Indian Statistical Institute, Kolkata, India

Arkadev Chattopadhyay
TIFR, Mumbai, India

Peter Høyer
Department of Computer Science, University of Calgary, Canada

Nikhil S. Mande [1]
CWI, Amsterdam, the Netherlands

Manaswi Paraashar
Indian Statistical Institute, Kolkata, India

Ronald de Wolf
QuSoft, CWI and University of Amsterdam, the Netherlands

Abstract

Buhrman, Cleve and Wigderson (STOC'98) showed that for every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$, the bounded-error quantum communication complexity of the composed function $f \circ G$ equals $O(\mathsf{Q}(f) \log n)$, where $\mathsf{Q}(f)$ denotes the bounded-error quantum query complexity of f. This is achieved by Alice running the optimal quantum query algorithm for f, using a round of $O(\log n)$ qubits of communication to implement each query. This is in contrast with the classical setting, where it is easy to show that $\mathsf{R}^{cc}(f \circ G) \leq 2\mathsf{R}(f)$, where R^{cc} and R denote bounded-error communication and query complexity, respectively. Chakraborty et al. (CCC'20) exhibited a total function for which the $\log n$ overhead in the BCW simulation is required. This established the somewhat surprising fact that quantum reductions are in some cases inherently more expensive than classical reductions. We improve upon their result in several ways.

- We show that the $\log n$ overhead is *not* required when f is symmetric (i.e., depends only on the Hamming weight of its input), generalizing a result of Aaronson and Ambainis for the Set-Disjointness function (Theory of Computing'05). Our upper bound assumes a shared entangled state, though for most symmetric functions the assumed number of entangled qubits is less than the communication and hence could be part of the communication.
- In order to prove the above, we design an efficient distributed version of noisy amplitude amplification that allows us to prove the result when f is the OR function. This also provides a different, and arguably simpler, proof of Aaronson and Ambainis's $O(\sqrt{n})$ communication upper bound for Set-Disjointness.
- In view of our first result above, one may ask whether the $\log n$ overhead in the BCW simulation can be avoided even when f is transitive, which is a weaker notion of symmetry. We give a strong negative answer by showing that the $\log n$ overhead is still necessary for some transitive functions even when we allow the quantum communication protocol an error probability that can be arbitrarily close to $1/2$ (this corresponds to the unbounded-error model of communication).
- We also give, among other things, a general recipe to construct functions for which the $\log n$ overhead is required in the BCW simulation in the bounded-error communication model, even if the parties are allowed to share an arbitrary prior entangled state for free.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Oracles and decision trees; Theory of computation → Quantum complexity theory

[1] Part of this work was done while the author was a postdoc at Georgetown University.

© Sourav Chakraborty, Arkadev Chattopadhyay, Peter Høyer, Nikhil S. Mande, Manaswi Paraashar and Ronald de Wolf;
licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022).
Editors: Petra Berenbrink and Benjamin Monmege; Article No. 37; pp. 37:1–37:22

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Classical and quantum communication complexity, query-to-communication-simulation, quantum computing.

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.37

Related Version A full version can be found at https://arxiv.org/abs/2012.05233.

Funding *Arkadev Chattopadhyay*: Partially supported by a MATRICS grant of the Science and Enginnering Research Board, DST, India.
Nikhil S. Mande : Supported by the Dutch Research Council (NWO) through QuantERA project QuantAlgo 680-91-034
Ronald de Wolf: Partially supported by ERC Consolidator Grant 615307-QPROGRESS (which ended February 2019), and by the Dutch Research Council (NWO/OCW), as part of the Quantum Software Consortium programme (project number 024.003.037), and through QuantERA ERA-NET Cofund project QuantAlgo (680-91-034).

1 Introduction

1.1 Motivation and main results

The classical model of communication complexity was introduced by Yao [24], who also subsequently introduced its quantum analogue [25]. Communication complexity has important applications in several disciplines, in particular for lower bounds on circuits, data structures, streaming algorithms, and many other complexity measures (see, for example, [16] and the references therein).

A natural way to derive a communication problem from a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is via composition. Let $f : \{-1,1\}^n \to \{-1,1\}$ be a function and let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a "two-party function". Then $F = f \circ G : \{-1,1\}^{nj} \times \{-1,1\}^{nk} \to \{-1,1\}$ denotes the function corresponding to the communication problem in which Alice is given input $X = (X_1, \ldots, X_n) \in \{-1,1\}^{nj}$, Bob is given $Y = (Y_1, \ldots, Y_n) \in \{-1,1\}^{nk}$, and their task is to compute $F(X,Y) = f(G(X_1,Y_1), \ldots, G(X_n,Y_n))$. Many well-known functions in communication complexity are derived in this way, such as Set-Disjointness ($\mathsf{DISJ}_n := \mathsf{NOR}_n \circ \mathsf{AND}_2$), Inner Product ($\mathsf{IP}_n := \mathsf{PARITY}_n \circ \mathsf{AND}_2$) and Equality ($\mathsf{EQ}_n := \mathsf{NOR}_n \circ \mathsf{XOR}_2$). A natural approach to obtain efficient quantum communication protocols for $f \circ G$ is to "simulate" a quantum query algorithm for f, where a query to the ith input bit of f is simulated by a communication protocol that computes $G(X_i, Y_i)$. Buhrman, Cleve and Wigderson [7] observed that such a simulation is indeed possible if G is bitwise AND or XOR.

▶ **Theorem 1** ([7]). *For every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and $\Box \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$, we have*

$$\mathsf{Q}^{cc}(f \circ \Box) = O\left(\mathsf{Q}(f) \log n\right).$$

Here $\mathsf{Q}(f)$ denotes the bounded-error quantum query complexity of f, and $\mathsf{Q}^{cc}(f \circ \Box)$ denotes the bounded-error quantum communication complexity for computing $f \circ \Box$. Throughout this paper, we refer to Theorem 1 as the BCW simulation. [7] used this, for instance, to show that the bounded-error quantum communication complexity of the Set-Disjointness function is $O(\sqrt{n} \log n)$, using Grover's $O(\sqrt{n})$-query search algorithm [11] for the NOR_n function.

It is folklore in the classical world that the analogous simulation does not incur a $\log n$ factor overhead. That is,

$$\mathsf{R}^{cc}(f \circ \square) \leq 2\mathsf{R}(f), \tag{1}$$

where $\mathsf{R}(f)$ denotes the bounded-error randomized query complexity of f and $\mathsf{R}^{cc}(f \circ \square)$ denotes the bounded-error randomized communication complexity for computing $f \circ \square$. Thus, a natural question is whether the multiplicative $\log n$ blow-up in the communication cost in the BCW simulation is necessary. Chakraborty et al. [9] answered this question and exhibited a total function for which the $\log n$ blow-up is indeed necessary when XOR_2 is the inner function.

▶ **Theorem 2** ([9, Theorem 2]). *There exists a function $f : \{-1,1\}^n \to \{-1,1\}$ such that*

$$\mathsf{Q}^{cc,*}(f \circ \mathsf{XOR}_2) = \Omega(\mathsf{Q}(f) \log n).$$

Here $\mathsf{Q}^{cc,*}(F)$ denotes the bounded-error quantum communication complexity of two-party function F when Alice and Bob shared an entangled state at the start of the protocol for free. Comparing Theorem 2 with Equation 1 we see the somewhat surprising fact that quantum reductions can in some cases be more expensive than classical reductions.

This gives rise to the following basic question: is there a natural class of functions for which the $\log n$ overhead in the BCW simulation is *not* required? Improving upon Høyer and de Wolf [13], Aaronson and Ambainis [1] showed that for the canonical problem of Set-Disjointness, the $\log n$ overhead in the BCW simulation can be avoided. Since the outer function NOR_n is symmetric (i.e., it only depends on the Hamming weight of its input, its number of -1s), a natural question is whether the $\log n$ overhead can be avoided whenever the outer function is symmetric. Our first result gives a positive answer to this question.

▶ **Theorem 3.** *For every symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and two-party function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{0,1\}$, we have*

$$\mathsf{Q}^{cc,*}(f \circ G) = O(\mathsf{Q}(f)\mathsf{Q}_E^{cc}(G)).$$

Here $\mathsf{Q}_E^{cc}(G)$ denotes the *exact* quantum communication complexity of G, where the error probability is 0. In particular, if $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$ then $\mathsf{Q}_E^{cc}(G) = 1$ and hence $\mathsf{Q}^{cc,*}(f \circ G) = O(\mathsf{Q}(f))$.

▶ Remark 4. If $\mathsf{Q}(f) = \Theta(\sqrt{tn})$, then our protocol in the proof of Theorem 3 starts from a shared entangled state of $O(t \log n)$ EPR-pairs. Note that if $t \leq n\mathsf{Q}_E^{cc}(G)^2/(\log n)^2$ (this condition holds for instance if $\mathsf{Q}_E^{cc}(G) \geq \log n$) then this number of EPR-pairs is no more than the amount of communication and hence might as well be established in the first message, giving asymptotically the same upper bound $\mathsf{Q}^{cc}(f \circ G) = O(\mathsf{Q}(f)\mathsf{Q}_E^{cc}(G))$ for the model without prior entanglement.

The next question one might ask is whether one can weaken the notion of symmetry required in Theorem 3. A natural generalization of the class of symmetric functions is the class of *transitive-symmetric* functions. A function $f : \{-1,1\}^n \to \{-1,1\}$ is said to be transitive-symmetric if for all $i, j \in [n]$, there exists $\sigma \in S_n$ such that $\sigma(i) = j$, and $f(x) = f(\sigma(x))$ for all $x \in \{-1,1\}^n$. Here, and in the rest of the paper, by $\sigma(x)$ we mean the n-bit string $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$. Henceforth we refer to transitive-symmetric functions as simply transitive functions. Can the $\log n$ overhead in the BCW simulation be avoided whenever the outer function is transitive? We give a negative answer to this question in a strong sense: the $\log n$ overhead is still necessary even when we allow the quantum communication protocol an error probability that can be arbitrarily close to $1/2$.

▶ **Theorem 5.** *There exists a transitive and total function* $f : \{-1,1\}^n \to \{-1,1\}$*, such that*

$$\mathsf{UPP}^{cc}(f \circ \Box) = \Omega(\mathsf{Q}(f) \log n)$$

for every $\Box \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Here $\mathsf{UPP}^{cc}(f \circ \Box)$ denotes the unbounded-error quantum communication complexity of $f \circ \Box$ (adding "quantum" here only changes the communication complexity by a constant factor). The unbounded-error model of communication was introduced by Paturi and Simon [21] and is the strongest communication complexity model against which we know how to prove explicit lower bounds. This model is known to be strictly stronger than the bounded-error quantum model. For instance, the Set-Disjointness function on n inputs requires $\Omega(n)$ bits or $\Omega(\sqrt{n})$ qubits of communication in the bounded-error model, but only requires $O(\log n)$ bits of communication in the unbounded-error model. In fact, it follows from a recent result of Hatami, Hosseini and Lovett [12] that there exists a function $F : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$ with $\mathsf{Q}^{cc,*}(F) = \Omega(n)$ while $\mathsf{UPP}^{cc}(F) = O(1)$.

Theorem 3 and Theorem 5 clearly demonstrate the role of symmetry in determining the presence of the $\log n$ overhead in the BCW query-to-communication simulation: this overhead is absent for symmetric functions (Theorem 3), but present for a transitive function even when the model of communication under consideration is as strong as the unbounded-error model (Theorem 5). We also give a general recipe to construct functions for which the $\log n$ overhead is required in the BCW simulation in the bounded-error communication model (see Theorem 6).

1.2 Overview of our approach and techniques

In this section we discuss the ideas that go into the proofs of Theorem 3 and Theorem 5.

1.2.1 Communication complexity upper bound for symmetric functions

To prove Theorem 3 we use the well-known fact that every symmetric function f has an interval around Hamming weight $n/2$ where the function is constant; for NOR_n the length of this interval would be essentially n, while for PARITY_n it would be 1. To compute f, it suffices to either determine that the Hamming weight of the input lies in that interval (because the function value is the same throughout that interval) or to count the Hamming weight exactly.

For two-party functions of the form $f \circ G$, we want to do this type of counting on the n-bit string $z = (G(X_1, Y_1), \ldots, G(X_n, Y_n)) \in \{-1,1\}^n$. We show how this can be done with $O(\mathsf{Q}(f)\, \mathsf{Q}_E^{cc}(G))$ qubits of communication if we had a quantum protocol that can find -1s in the string z at a cost of $O(\sqrt{n}\, \mathsf{Q}_E^{cc}(G))$ qubits. Such a protocol was already given by Aaronson and Ambainis for the special case where $G = \mathsf{AND}_2$ for their optimal quantum protocol for Set-Disjointness, as a corollary of their quantum walk algorithm for search on a grid [1]. In this paper we give an alternative $O(\sqrt{n}\, \mathsf{Q}_E^{cc}(G))$-qubit protocol. This implies the result of Aaronson and Ambainis as a special case, but it is arguably simpler and may be of independent interest.

Our protocol can be viewed as an efficient distributed implementation of amplitude amplification with faulty components. In particular, we replace the usual reflection about the uniform superposition by an imperfect reflection about the n-dimensional maximally entangled state ($= \log n$ EPR-pairs if n is a power of 2). Such a reflection would require $O(\log n)$ qubits of communication to implement perfectly, but can be implemented with small

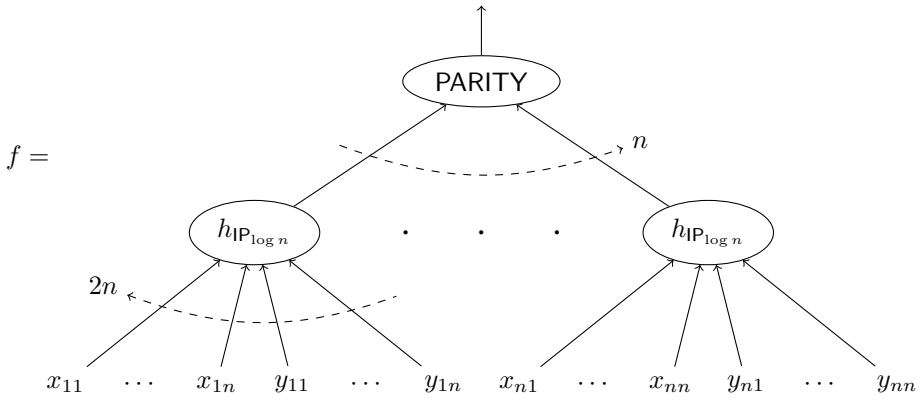

Figure 1 If for all $j \in [n]$ and some $s_j, t_j \in \{-1,1\}^{\log n}$, the inputs to the j-th $h_{\mathsf{IP}_{\log n}}$ are Hadamard codewords in $\pm H(s_j)$ and $\pm H(t_j)$, then $f = \mathsf{PARITY}(\mathsf{IP}_{\log n}(s_1,t_1),\ldots,\mathsf{IP}_{\log n}(s_n,t_n))$. If there exists at least one $j \in [n]$ for which either $x_{j1},\ldots,x_{jn}$ or $y_{j1},\ldots,y_{jn}$ is not a Hadamard codeword, then f outputs -1. This function f equals $\mathsf{PARITY}_n \mathbin{\widetilde{\circ}} h_{\mathsf{IP}_{\log n}}$ (see Definition 26 and Definition 28).

error using only $O(1)$ qubits of communication, by invoking the efficient protocol of Aharonov et al. [2, Theorem 1] that tests whether a given bipartite state equals the n-dimensional maximally entangled state. Still, at the start of this protocol we need to assume (or establish by means of quantum communication) a shared state of $\log n$ EPR-pairs. If $\mathsf{Q}(f) = \Theta(\sqrt{tn})$ then our protocol for $f \circ G$ will run the -1-finding protocol $O(t)$ times, which accounts for our assumption that we share $O(t \log n)$ EPR-pairs at the start of the protocol.

1.2.2 Communication complexity lower bound for transitive functions

For proving Theorem 5, we exhibit a transitive function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ whose bounded-error quantum query complexity is $O(n)$ and the unbounded-error communication complexity of $f \circ \square$ is $\Omega(n \log n)$ for $\square \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Function construction and transitivity: For the construction of f we first require the definition of Hadamard codewords. The Hadamard codeword of $s \in \{-1,1\}^{\log n}$, denoted by $H(s) \in \{-1,1\}^n$, is a list of all parities of s. That is, $(H(s))_t = \prod_{i:s_i=-1} t_i$ for all $t \in \{-1,1\}^{\log n}$. See Figure 1 for a graphical visualization of f.

Using properties of IP and Hadamard codewords, and the symmetry of PARITY_n, we are able to show that f is transitive (see Claim 32).

Query upper bound: The query upper bound of $O(n)$ follows along the lines of [9], using the Bernstein-Vazirani algorithm to decode the Hadamard codewords, and Grover's algorithm to check that they actually are Hadamard codewords. This approach was in turn inspired by a query upper bound due to Ambainis and de Wolf [3]. See the proof of Theorem 29 for the query algorithm and its analysis.

Communication lower bound: Towards the unbounded-error communication lower bound, we first recall that each input block of f equals $\mathsf{IP}_{\log n}$ if the inputs to each block are promised to be Hadamard codewords. Hence f equals $\mathsf{IP}_{n \log n}$ under this promise, since $\mathsf{PARITY}_n \circ \mathsf{IP}_{\log n} = \mathsf{IP}_{n \log n}$. Thus by setting certain inputs to Alice and Bob suitably, $f \circ \square$ is at least as hard as $\mathsf{IP}_{n \log n}$ for $\square \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$ (for a formal statement, see Lemma 31 with $r = \mathsf{PARITY}_n$ and $g = \mathsf{IP}_{\log n}$). It is known from a seminal result of Forster [10] that

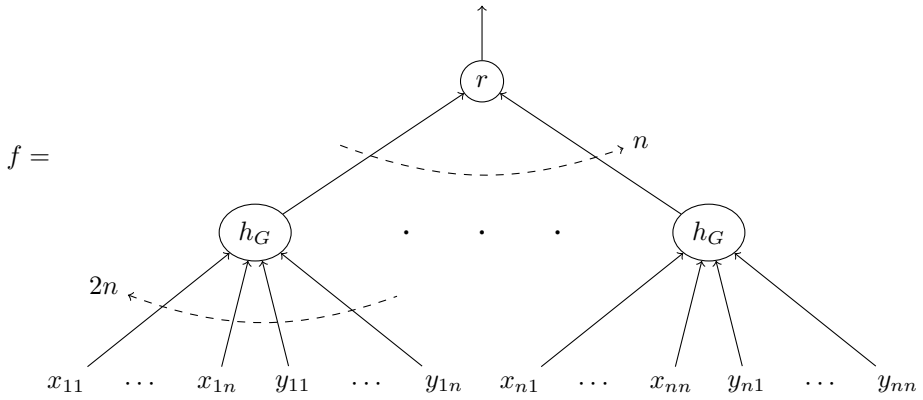

Figure 2 In this figure, $G : \{-1,1\}^{\log} \times \{-1,1\}^{\log n} \to \{-1,1\}$. If for all $j \in [n]$ and some $s_j, t_j \in \{-1,1\}^{\log n}$, the inputs to the j-th h_G are Hadamard codewords in $\pm H(s_j)$ and $\pm H(t_j)$, then $f = r(G(s_1,t_1),\ldots,G(s_n,t_n))$. If there exists at least one $j \in [n]$ for which either $x_{j1},\ldots,x_{jn}$ or $y_{j1},\ldots,y_{jn}$ is not a Hadamard codeword, then f outputs -1. This function f equals $r \mathbin{\widetilde{\circ}} h_G$ (see Definition 26 and Definition 28).

the unbounded-error communication complexity of $\mathsf{IP}_{n\log n}$ equals $\Omega(n \log n)$, completing the proof of the lower bound. This proof is more general than and arguably simpler than the proof of the lower bound for bounded-error quantum communication complexity in [9, Theorem 2].

1.3 Other results

We give a general recipe for constructing a class of functions that witness tightness of the BCW simulation where the inner gadget is either AND_2 or XOR_2. However, the communication lower bound we obtain here is in the bounded-error model in contrast to Theorem 5, where the communication lower bound is proven in the unbounded-error model.

The functions f constructed for this purpose are composed functions similar to the construction in Figure 1, except that we are able to use a more general class of functions in place of the outer PARITY function, and also a more general class of functions in place of the inner $\mathsf{IP}_{\log n}$ functions. See Figure 2 and its caption for an illustration and a more precise definition.

We require some additional constraints on the outer and inner functions. First, the approximate degree of r should be $\Omega(n)$. Second, the discrepancy of G should be small with respect to some "balanced" probability distribution (see Definition 17 and Definition 16 for formal definitions of these notions).

▶ **Theorem 6.** *Let $r : \{-1,1\}^n \to \{-1,1\}$ be such that $\widetilde{\deg}(r) = \Omega(n)$ and let $G : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$ be a total function. Define $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ as in Figure 2. If there exists $\mu : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \mathbb{R}$ that is a balanced probability distribution with respect to G and $\mathrm{disc}_\mu(G) = n^{-\Omega(1)}$, then for every $\square \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$,*

$$\mathsf{Q}(f) = O(n), \qquad \text{and} \qquad \mathsf{Q}^{cc,*}(f \circ \square) = \Omega(n \log n).$$

The query upper bound follows along similar lines as that of Theorem 5. For the lower bound, we first show via a reduction that for f as described in Figure 2 and $\square \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$, the communication problem $f \circ \square$ is at least as hard as $r \circ G$ (see Lemma 31).

This part of the lower bound proof is the same as in the proof of Theorem 5. For the hardness of $r \circ G$ (which in the case of Theorem 5 turned out to be $\mathsf{IP}_{n \log n}$, for which Forster's theorem yields an unbounded-error communication lower bound), we are able to use a theorem implicit in a work of Lee and Zhang [17]. This theorem gives a lower bound on the bounded-error communication complexity of $r \circ G$ in terms of the approximate degree of r and the discrepancy of G under a balanced distribution. Due to space constraints we defer the proof of Theorem 6 to the full version of our paper [8].

We recover the result of Chakraborty et al. (Theorem 2) using a more general technique, and additionally show that $\mathsf{Q}^{cc,*}(f \circ \mathsf{AND}_2) = \Omega(\mathsf{Q}(f) \log n)$, where f is as in Theorem 2. We refer the reader to the full version [8] for details.

1.4 Organization

Section 2 gives notations and preliminaries. In Section 3 we prove Theorem 3, which shows that the $\log n$ overhead in the BCW simulation can be avoided when the outer function is symmetric. This proof relies on our new one-sided error protocol for finding solutions in the string $z = (G(X_1, Y_1), \ldots, G(X_n, Y_n)) \in \{-1,1\}^n$, as a corollary of our distributed version of amplitude amplification. We give this protocol in Appendix A.

We prove Theorem 5 in Section 4. This is our result regarding necessity of the $\log n$ overhead in the BCW simulation in the unbounded-error model of communication.

2 Notation and preliminaries

Without loss of generality, we assume n to be a power of 2 in this paper, unless explicitly stated otherwise. All logarithms in this paper are base 2. Let S_n denote the symmetric group over the set $[n] = \{1, \ldots, n\}$. For a string $x \in \{-1,1\}^n$ and $\sigma \in S_n$, let $\sigma(x)$ denote the string $x_{\sigma(1)}, \ldots, x_{\sigma(n)} \in \{-1,1\}^n$. Consider an arbitrary but fixed bijection between subsets of $[\log n]$ and elements of $[n]$. For a string $s \in \{-1,1\}^{\log n}$, we abuse notation and also use s to denote the equivalent element of $[n]$. The view we take will be clear from context. For a string $x \in \{-1,1\}^n$ and set $S \subseteq [n]$, define the string $x_S \in \{-1,1\}^S$ to be the restriction of x to the coordinates in S. Let 1^n and $(-1)^n$ denote the n-bit string $(1, 1, \ldots, 1)$ and $(-1, -1, \ldots, -1)$, respectively.

2.1 Boolean functions

For two bits $b_1, b_2 \in \{-1,1\}$, let $b_1 \wedge b_2$ be defined to be -1 if $b_1 = b_2 = -1$, and 1 otherwise. For strings $x, y \in \{-1,1\}^n$, let $\langle x, y \rangle$ denote the inner product (mod 2) of x and y. That is, $\langle x, y \rangle = \prod_{i=1}^n (x_i \wedge y_i)$. For every positive integer n, let $\mathsf{PARITY}_n : \{-1,1\}^n \to \{-1,1\}$ be defined as:

$$\mathsf{PARITY}_n(x_1, \ldots, x_n) = \prod_{i \in [n]} x_i.$$

▶ **Definition 7** (Symmetric functions). *A function $f : \{-1,1\}^n \to \{-1,1\}$ is symmetric if for all $\sigma \in S_n$ and for all $x \in \{-1,1\}^n$ we have $f(x) = f(\sigma(x))$.*

▶ **Definition 8** (Transitive functions). *A function $f : \{-1,1\}^n \to \{-1,1\}$ is transitive if for all $i, j \in [n]$ there exists a permutation $\sigma \in S_n$ such that*

- *$\sigma(i) = j$, and*
- *$f(x) = f(\sigma(x))$ for all $x \in \{-1,1\}^n$.*

▶ **Definition 9** (Approximate degree). *For every $\varepsilon \geq 0$, the ε-approximate degree of a function $f : \{-1,1\}^n \to \{-1,1\}$ is defined to be the minimum degree of a real polynomial $p : \{-1,1\}^n \to \mathbb{R}$ that uniformly approximates f to error ε. That is,*

$$\widetilde{\deg}_\varepsilon(f) = \min\{\deg(p) : |p(x) - f(x)| \leq \varepsilon \text{ for all } x \in \{-1,1\}^n\}.$$

Unless specified otherwise, we drop ε from the subscript and assume $\varepsilon = 1/3$.

We assume familiarity with quantum computing [19], and use $\mathsf{Q}_\varepsilon(f)$ to denote the ε-error query complexity of f. Unless specified otherwise, we drop ε from the subscript and assume $\varepsilon = 1/3$.

▶ **Theorem 10** ([4]). *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a function. Then $\mathsf{Q}(f) \geq \widetilde{\deg}(f)/2$.*

2.2 Communication complexity

We assume familiarity with communication complexity [16].

▶ **Definition 11** (Two-party function). *We call a function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ a* two-party function *to indicate that it corresponds to a communication problem in which Alice is given input $x \in \{-1,1\}^j$, Bob is given input $y \in \{-1,1\}^k$, and their task is to compute $G(x,y)$.*

▶ Remark 12. Throughout this paper, we use uppercase letters to denote two-party functions, and lowercase letters to denote functions which are not two-party functions.

▶ **Definition 13** (Composition with two-party functions). *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a function and let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a two-party function. Then $F = f \circ G : \{-1,1\}^{nj} \times \{-1,1\}^{nk} \to \{-1,1\}$ denotes the two-party function corresponding to the communication problem in which Alice is given input $X = (X_1, \ldots, X_n) \in \{-1,1\}^{nj}$, Bob is given $Y = (Y_1, \ldots, Y_n) \in \{-1,1\}^{nk}$, and their task is compute $F(X,Y) = f(G(X_1,Y_1), \ldots, G(X_n,Y_n))$.*

▶ **Definition 14** (Inner Product function). *For every positive integer n, define the function $\mathsf{IP}_n : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$ by $\mathsf{IP}_n(x,y) = \langle x,y\rangle$. In other words, $\mathsf{IP}_n = \mathsf{PARITY}_n \circ \mathsf{AND}_2$.*

▶ **Observation 15.** *For all positive integers k, t, $\mathsf{PARITY}_k \circ \mathsf{IP}_t = \mathsf{IP}_{kt}$.*

We also assume familiarity with quantum communication complexity [23]. We use $\mathsf{Q}^{cc}_\varepsilon(G)$ and $\mathsf{Q}^{cc,*}_\varepsilon(G)$ to represent the ε-error quantum communication complexity of a two-party function G in the models without and with unlimited shared entanglement, respectively. Unless specified otherwise, we drop ε from the subscript and assume $\varepsilon = 1/3$.

▶ **Definition 16** (Balanced probability distribution). *We call a probability distribution $\mu : \{-1,1\}^n \to \mathbb{R}$ balanced w.r.t. a function $f : \{-1,1\}^n \to \{-1,1\}$ if $\sum_{x \in \{-1,1\}^n} f(x)\mu(x) = 0$.*

▶ **Definition 17** (Discrepancy). *Let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a function and λ be a distribution on $\{-1,1\}^j \times \{-1,1\}^k$. For every $S \subseteq \{-1,1\}^j$ and $T \subseteq \{-1,1\}^k$, define*

$$\mathrm{disc}_\lambda(S \times T, G) = \left| \sum_{x,y \in S \times T} G(x,y)\lambda(x,y) \right|.$$

The discrepancy of G under the distribution λ is defined to be

$$\mathrm{disc}_\lambda(G) = \max_{S \subseteq \{-1,1\}^j, T \subseteq \{-1,1\}^k} \mathrm{disc}_\lambda(S \times T, G),$$

and the discrepancy of f is defined to be $\mathrm{disc}(G) = \min_\lambda \mathrm{disc}_\lambda(G)$.

2.3 Additional concepts from quantum computing

The Bernstein-Vazirani algorithm [5] is a quantum query algorithm that takes an n-bit string as input and outputs a $(\log n)$-bit string. The algorithm has the following properties:

- the algorithm makes one quantum query to the input and
- if the input $x \in \{-1,1\}^n$ satisfies $x \in \pm H(s)$ for some $s \in \{-1,1\}^{\log n}$, then the algorithm returns s with probability 1.

Consider a symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$. Define the quantity

$$\Gamma(f) = \min\{|2k - n + 1| : f(x) \neq f(y) \text{ if } |x| = k \text{ and } |y| = k+1\}$$

from [20]. One can think of $\Gamma(f)$ as essentially the length of the interval of Hamming weights around $n/2$ where f is constant (for example, for the majority and parity functions this would be 1, and for OR_n this would be $n-1$).

▶ **Theorem 18** ([4, Theorem 4.10]). *For every symmetric function $f : \{-1,1\}^n \to \{-1,1\}$, we have $\mathsf{Q}(f) = \Theta(\sqrt{(n - \Gamma(f))n})$.*

The upper bound follows from a quantum algorithm that exactly counts the Hamming weight $|x|$ of the input if $|x| \leq t$ or $|x| \geq n - t$ for $t = \lceil (n - \Gamma(f))/2 \rceil$, and that otherwise learns $|x|$ is in the interval $[t+1, n-t-1]$ (which is an interval around $n/2$ where $f(x)$ is constant). By the definition of $\Gamma(f)$, this information about $|x|$ suffices to compute $f(x)$. In Section 3 we use this observation to give an efficient quantum communication protocol for a two-party function $f \circ G$.

We will need a unitary protocol that allows Alice and Bob to implement an approximate reflection about the n-dimensional maximally entangled state

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{i \in \{0,1\}^{\log n}} |i\rangle|i\rangle.$$

Ideally, such a reflection would map $|\psi\rangle$ to itself, and put a minus sign in front of all states orthogonal to $|\psi\rangle$. Doing this perfectly would requires $O(\log n)$ qubits of communication. Fortunately we can derive a cheaper protocol from a test that Aharonov et al. [2, Theorem 1] designed, which uses $O(\log(1/\varepsilon))$ qubits of communication and checks whether a given bipartite state equals $|\psi\rangle$, with one-sided error probability ε. By the usual trick of running this protocol, applying a Z-gate to the answer qubit, and then reversing the protocol, we can implement the desired reflection approximately.[2] A bit more precisely:

▶ **Theorem 19.** *Let $R_\psi = 2|\psi\rangle\langle\psi| - I$ be the reflection about the maximally entangled state shared between Alice and Bob. There exists a protocol that uses $O(\log(1/\varepsilon))$ qubits of communication and that implements a unitary R_ψ^ε such that $\left\| R_\psi^\varepsilon - R_\psi \right\| \leq \varepsilon$ and $R_\psi^\varepsilon |\psi\rangle = |\psi\rangle$.*

We use $\mathsf{UPP}^{cc}(F)$ to denote unbounded-error quantum communication complexity of two-party function F. It is folklore (see for example [15]) that the unbounded-error quantum communication complexity[3] of F equals its classical counterpart up to a factor of at most 2

[2] Possibly with some auxiliary qubits on Alice and Bob's side which start in $|0\rangle$ and end in $|0\rangle$, except in a part of the final state that has norm at most ε.

[3] The unbounded-error model does not allow shared randomness or prior shared entanglement (which yields shared randomness by measuring) between Alice and Bob, since any two-party function F would have constant communication complexity in that setting.

so it does not really matter much whether we use UPP^{cc} for classical unbounded-error communication complexity (as it is commonly used) or for quantum unbounded-error complexity. Crucially, for both the complexity of IP_n is linear in n:

▶ **Theorem 20** ([10]). *Let n be a positive integer. Then $\mathsf{UPP}^{cc}(\mathsf{IP}_n) = \Omega(n)$.*

3 No log-factor needed for symmetric functions

We present a version of quantum amplitude amplification that still works if the reflections involved are not perfectly implemented. In particular, the usual reflection about the uniform superposition will be replaced in the communication setting by an imperfect reflection about the n-dimensional maximally entangled state, based on the communication-efficient protocol of Aharonov et al. [2, Theorem 1] for testing whether Alice and Bob share that state. This allows us to avoid the $\log n$ factor that would be incurred if we instead used a BCW-style distributed implementation of standard amplitude amplification, with $O(\log n)$ qubits of communication to implement each query. Our main technical contribution for proving Theorem 3 is the following general theorem, which allows us to search among a sequence of two-party instances $(X_1, Y_1), \ldots, (X_n, Y_n)$ for an index $i \in [n]$ where $G(X_i, Y_i) = -1$, for any two-party function G.

▶ **Theorem 21.** *Let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a two-party function, $X = (X_1, \ldots, X_n) \in \{-1,1\}^{nj}$ and $Y = (Y_1, \ldots, Y_n) \in \{-1,1\}^{nk}$. Define $z = (G(X_1, Y_1), \ldots, G(X_n, Y_n)) \in \{-1,1\}^n$. Assume Alice and Bob start with $\lceil \log n \rceil$ shared EPR-pairs.*

- *There exists a quantum protocol using $O(\sqrt{n}\, \mathsf{Q}_E^{cc}(G))$ qubits of communication that finds (with success probability ≥ 0.99) an $i \in [n]$ such that $z_i = -1$ if such an i exists, and says "no" with probability 1 if no such i exists.*
- *If the number of -1s in z is within a factor of 2 from a known integer t, then the communication can be reduced to $O(\sqrt{n/t}\, \mathsf{Q}_E^{cc}(G))$ qubits.*

We prove Theorem 21 in Appendix A. Consider a symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$. As we explained in Section 2.3, there is an integer $t = \lceil (n - \Gamma(f))/2 \rceil$ such that we can compute f if we learn the Hamming weight $|z|$ of the input $z \in \{-1,1\}^n$ or learn that $|z| \in [t+1, n-t-1]$. The bounded-error quantum query complexity is $\mathsf{Q}(f) = \Theta(\sqrt{tn})$ (Theorem 18). We now prove Theorem 3 assuming Theorem 21.

For a given two-party function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$, we have an induced two-party function $F : \{-1,1\}^{nj} \times \{-1,1\}^{nk} \to \{-1,1\}$ defined as $F(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = f(G(X_1, Y_1), \ldots, G(X_n, Y_n))$. Define

$$z = (G(X_1, Y_1), \ldots, G(X_n, Y_n)) \in \{-1,1\}^n .$$

Then $F(X, Y) = f(z)$ only depends on the number of -1s in z. The following theorem allows us to count this number using $O(\mathsf{Q}(f)\, \mathsf{Q}_E^{cc}(G))$ qubits of communication.

▶ **Theorem 22.** *For every t between 1 and $n/2$, there exists a quantum protocol that starts from $O(t \log n)$ EPR-pairs, communicates $O(\sqrt{tn}\, \mathsf{Q}_E^{cc}(G))$ qubits, and tells us $|z|$ or tells us that $|z| > t$, with error probability $\leq 1/8$.*

Proof. Abbreviate $q = \mathsf{Q}_E^{cc}(G)$. Our protocol has two parts: the first filters out the case $|z| \geq 2t$, while the second finds all solutions if $|z| < 2t$.

Part 1. First Alice and Bob decide between the case (1) $|z| \geq 2t$ and (2) $|z| \leq t$ (even though $|z|$ might also lie in $\{t+1, \ldots, 2t-1\}$) using $O(\sqrt{n}q)$ qubits of communication,

as follows. They use shared randomness to choose a uniformly random subset $S \subseteq [n]$ of $\lceil n/(2t) \rceil$ elements. Let E be the event that $z_i = -1$ for at least one $i \in S$. By standard calculations there exist $p_1, p_2 \in [0,1]$ with $p_1 = p_2 + \Omega(1)$ such that $\Pr[E] \geq p_1$ in case (1) and $\Pr[E] \leq p_2$ in case (2). Alice and Bob use the distributed-search protocol from the first bullet of Theorem 21 to decide E, with $O(\sqrt{|S|}\, q) = O(\sqrt{n}\, q)$ qubits of communication (plus a negligible $O(\log n)$ EPR-pairs) and error probability much smaller than $p_1 - p_2$. By repeating this a sufficiently large constant number of times and seeing whether the fraction of successes was larger or smaller than $(p_1 + p_2)/2$, they can distinguish between cases (1) and (2) with success probability $\geq 15/16$. If they conclude they're in case (1) then they output "$|z| > t$" and otherwise they proceed to the second part of the protocol.

Note that if $|z| \in \{t+1, \ldots, 2t-1\}$ (the "grey zone" in between cases (1) and (2)), then we can't give high-probability guarantees for one output or the other, but concluding (1) leads to the correct output "$|z| > t$" in this case, while concluding (2) means the protocol proceeds to Part 2. So either course of action is fine if $|z| \in \{t+1, \ldots, 2t-1\}$.

By Newman's theorem [18] the shared randomness used for choosing S can be replaced by $O(\log n)$ bits of private randomness on Alice's part, which she can send to Bob in her first message, so Part 1 communicates $O(\sqrt{n}\, q)$ qubits in total.

Part 2. We condition on Part 1 successfully filtering out case (1), so from now on assume $|z| < 2t$. Our goal in this second part of the protocol is to find all indices i such that $z_i = -1$ (we call such i "solutions"), with probability $\geq 15/16$, using $O(\sqrt{tn}\, q)$ qubits of communication. This will imply that the overall protocol is correct with probability $1 - 1/16 - 1/16 = 7/8$, and uses $O(\sqrt{tn}\, q)$ qubits of communication in total. For an integer $k \geq 1$, consider the following protocol P_k.

Algorithm 1 Protocol P_k

Input: An integer $k \geq 1$

repeat

1. Run the protocol from the last bullet of Theorem 21 with $t = 2^{k-1}$.
   (suppressing some constant factors, assume for simplicity that this uses $\sqrt{n/2^k}\, q$ qubits of communication, $\log n$ shared EPR-pairs at the start, and has probability $\geq 1/100$ to find a solution if the actual number of solutions is in $[t/2, 2t]$).
2. Alice measures and gets outcome $i \in [n]$ and Bob measures and gets outcome $j \in [n]$, respectively.
3. Alice sends i to Bob, Bob sends j to Alice.
4. If $i = j$ then they verify that $G(X_i, Y_i) = -1$ by one run of the protocol for G, and if so then they replace X_i, Y_i by some pre-agreed inputs X_i', Y_i', respectively, such that $G(X_i', Y_i') = 1$ (this reduces the number of -1s in z by 1)

until $200\sqrt{2^k n}\, q$ qubits have been sent;

▷ Claim 23. Suppose $|z| \in [2^{k-1}, 2^k)$. Then protocol P_k uses $O(\sqrt{2^k n}\, q)$ qubits of communication, assumes $O(2^k \log n)$ EPR-pairs at the start of the protocol, and finds at least $|z| - 2^{k-1} + 1$ solutions, except with probability $\leq 1/2$.

Proof. The upper bound on the communication is obvious from the stopping criterion of P_k.

As long as the remaining number of solutions is $\geq 2^{k-1}$, each run of the protocol has probability $\geq 1/100$ to find another solution. Hence the expected number of runs of the protocol of Theorem 21 to find at least $|z| - 2^{k-1} + 1$ solutions, is $\leq 100(|z| - 2^{k-1} + 1)$. By Markov's inequality, the probability that we haven't yet found $|z| - 2^{k-1} + 1$ solutions

after $\leq 200(|z| - 2^{k-1} + 1) \leq 100 \cdot 2^k$ runs, is $\leq 1/2$. The communication cost of so many runs is $100 \cdot 2^k(\sqrt{n/2^k}\, q + \log n) \leq 200\sqrt{2^k n}\, q$ qubits. Hence by the time that the number of qubits of the stopping criterion have been communicated, we have probability $\geq 1/2$ of having found at least $|z| - 2^{k-1} + 1$ solutions. The assumed number of EPR-pairs at the start is $\log n$ per run, so $O(2^k \log n)$ in total. ◁

Note that if we start with a number of solutions $|z| \in [2^{k-1}, 2^k)$, and P_k succeeds in finding at least $|z| - 2^{k-1} + 1$ new solutions, then afterwards we have $< 2^{k-1}$ solutions left. The following protocol runs these P_k in sequence, pushing down the remaining number of solutions to 0.

Algorithm 2 Protocol P

for $k = \lceil \log_2(2t) \rceil$ downto 1 **do**
  1. Run P_k a total of $r_k = \lceil \log_2(2t) \rceil - k + 5$ times (replacing all -1s found by $+1$s in z).
  2. Output the total number of solutions found.

end

▷ Claim 24. If $|z| < 2t$ then protocol $\mathcal{P}$ uses $O(\sqrt{tn}\, q)$ qubits of communication, assumes $O(t \log n)$ EPR-pairs at the start of the protocol, and outputs $|z|$, except with probability $\leq 1/16$.

Proof. First, by Claim 23, the total number of qubits communicated is

$$\sum_{k=1}^{\lceil \log_2(2t) \rceil} r_k \cdot O(\sqrt{2^k n}\, q) = O(\sqrt{tn}\, q) \cdot \sum_{\ell=0}^{\lceil \log_2(2t) \rceil - 1} (\ell + 5)/\sqrt{2^\ell} = O(\sqrt{tn}\, q),$$

where we used a variable substitution $k = \lceil \log_2(2t) \rceil - \ell$. Second, the number of EPR-pairs we're starting from is

$$\sum_{k=1}^{\lceil \log_2(2t) \rceil} r_k \cdot O(2^k \log n) = O(t \log n) \cdot \sum_{\ell=0}^{\lceil \log_2(2t) \rceil - 1} (\ell + 5)/2^\ell = O(t \log n).$$

Third, by Claim 23 and the fact that we are performing r_k repetitions of P_k, if the kth round of $\mathcal{P}$ starts with a remaining number of solutions that is in the interval $[2^{k-1}, 2^k)$ then that round ends with $< 2^{k-1}$ remaining solutions, except with probability at most $1/2^{r_k}$. By the union bound, the probability that any one of the $\lceil \log_2(2t) \rceil$ rounds does not succeed at this, is at most

$$\sum_{k=1}^{\lceil \log_2(2t) \rceil} \frac{1}{2^{r_k}} = \sum_{\ell=0}^{\lceil \log_2(2t) \rceil - 1} \frac{1}{2^{\ell+5}} \leq \frac{1}{16}.$$

Since $2^{\lceil \log_2(2t) \rceil} \geq 2t$ and we start with $|z| < 2t$, if each round succeeds, then by the end of $\mathcal{P}$ there are no remaining solutions left. Thus, the protocol $\mathcal{P}$ finds all solutions and learns $|z|$ with probability at least $15/16$. ◁

Part 1 and Part 2 each have error probability $\leq 1/16$, so by the union bound the protocol succeeds except with probability $1/8$. If $|z| \geq 2t$ then Part 1 outputs the correct answer "$|z| > t$"; if $|z| \leq t$ then all solutions (and hence $|z|$) are found by Part 2; and if $|z| \in \{t+1, \ldots, 2t-1\}$ then either Part 1 already outputs the correct answer "$|z| > t$" or the protocol proceeds to Part 2 which then finds all solutions. ◀

We can use the above theorem twice: once to count the number of -1s in z (up to t) and once to count the number of 1s in z (up to t). This uses $O(\sqrt{tn}\,\mathsf{Q}^{cc}_E(G)) = O(\mathsf{Q}(f)\,\mathsf{Q}^{cc}_E(G))$ qubits of communication, assumes $O(t \log n)$ shared EPR-pairs at the start of the protocol, and gives us enough information about $|z|$ to compute $f(z) = F(X, Y)$. This concludes the proof of Theorem 3 from the introduction, restated below.

▶ **Theorem 25** (Restatement of Theorem 3). *For every symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and two-party function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{0,1\}$, we have $\mathsf{Q}^{cc,*}(f \circ G) = O(\mathsf{Q}(f)\mathsf{Q}^{cc}_E(G))$.*

If $\mathsf{Q}(f) = \Theta(\sqrt{tn})$, then our protocol in the proof of Theorem 3 assumes a shared state of $O(t \log n)$ EPR-pairs at the start. We remark that for the special case where $G = \mathsf{AND}_2$, our upper bound matches the lower bound proved by Razborov [22], except for symmetric functions f where the first switch of function value happens at Hamming weights very close to n. In particular, if $f = \mathsf{AND}_n$ and $G = \mathsf{AND}_2$, then $\mathsf{Q}^{cc}(f \circ G) = 1$ but $\mathsf{Q}(f) = \Theta(\sqrt{n})$.

4 Necessity of the log-factor overhead in the BCW simulation

In this section we prove Theorem 5. We exhibit a function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ for which $\mathsf{Q}(f) = O(n)$ and $\mathsf{UPP}(f \circ \square) = \Omega(n \log n)$ for $\square \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

The proofs of Theorem 5 and Theorem 6 each involve proving a query complexity upper bound and a communication complexity lower bound. The proofs of the query complexity upper bounds are along similar lines and follow from Theorem 29 and Corollary 30 (see Section 4.1). The proofs of the communication complexity lower bounds each involve a reduction from a problem whose communication complexity is easier to analyze (see Lemma 31 in Section 4.2). We complete the proof of Theorem 5 in Section 4.2.1. See the full version of our paper [8] for a proof of Theorem 6.

4.1 Quantum query complexity upper bound

For total functions f, g, let $f \circ g$ denote the standard composition of the functions f and g. We also require the following notion of composition of a total function f with a partial function g.

▶ **Definition 26** (Composition with partial functions). *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a total function and let $g : \{-1,1\}^m \to \{-1,1,\star\}$ be a partial function. Let $f \mathbin{\widetilde{\circ}} g : \{-1,1\}^{nm} \to \{-1,1\}$ denote the total function that is defined as follows on input $(X_1, \ldots, X_n) \in \{-1,1\}^{nm}$, where $X_i \in \{-1,1\}^m$ for all $i \in [n]$.*

$$f \mathbin{\widetilde{\circ}} g(X_1, \ldots, X_n) = \begin{cases} f(g(X_1), \ldots, g(X_n)) & \text{if } g(X_i) \in \{-1,1\} \text{ for all } i \in [n], \\ -1 & \text{otherwise.} \end{cases}$$

That is, we use $f \mathbin{\widetilde{\circ}} g$ to denote the total function that equals $f \circ g$ on inputs when each copy of g outputs a value in $\{-1,1\}$, and equals -1 otherwise.

Recall that we index coordinates of n-bit strings by integers in $[n]$, and also interchangeably by strings in $\{-1,1\}^{\log n}$ via the natural correspondence. For $x \in \{-1,1\}^n$, let $-x \in \{-1,1\}^n$ be defined as $(-x)_i = -x_i$ for all $i \in [n]$. We use the notation $\pm x$ to denote the set $\{x, -x\}$.

▶ **Definition 27** (Hadamard Codewords). *For every positive integer n and $s \in \{-1,1\}^{\log n}$, let $H(s) \in \{-1,1\}^n$ be defined as $(H(s))_t = \prod_{i:s_i=-1} t_i$ for all $t \in \{-1,1\}^{\log n}$. If $x \in \{-1,1\}^n$ is such that $x = H(s)$ for some $s \in \{-1,1\}^{\log n}$, we say x is a Hadamard codeword corresponding to s.*

That is, for every $s \in \{-1,1\}^{\log n}$, there is an n-bit Hadamard codeword corresponding to s. This represents the enumeration of all parities of s.

We now define how to encode a two-party total function G on $(\log j + \log k)$ input bits to a partial function h_G on $(j+k)$ input bits, using Hadamard encoding.

▶ **Definition 28** (Hadamardization of functions). *Let $j,k \geq 1$ be powers of 2, and let $G : \{-1,1\}^{\log j} \times \{-1,1\}^{\log k} \to \{-1,1\}$ be a function. Define a partial function $h_G : \{-1,1\}^{j+k} \to \{-1,1,\star\}$ by*

$$h_G(x,y) = \begin{cases} G(s,t) & \text{if } x \in \pm H(s), y \in \pm H(t) \text{ for some } s \in \{-1,1\}^{\log j}, t \in \{-1,1\}^{\log k} \\ \star & \text{otherwise.} \end{cases}$$

We next prove the following theorem. (See Figure 2 for a visual description of h_G.)

▶ **Theorem 29.** *Let $G : \{-1,1\}^{\log j} \times \{-1,1\}^{\log k} \to \{-1,1\}$ and $r : \{-1,1\}^n \to \{-1,1\}$. Then the quantum query complexity of the function $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{n(j+k)} \to \{-1,1\}$ is given by $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = O(n + \sqrt{n(j+k)})$.*

Proof. Recall from Definition 26 that the function $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{n(j+k)} \to \{-1,1\}$ is defined as $r \mathbin{\widetilde{\circ}} h_G((X_1,Y_1),\ldots,(X_n,Y_n)) = r \circ h_G((X_1,Y_1),\ldots,(X_n,Y_n))$ if $h_G((X_i,Y_i)) \in \{-1,1\}$ for all $i \in [n]$, and -1 otherwise.

Quantum query algorithm:

View inputs to $r \mathbin{\widetilde{\circ}} h_G$ as $(X_1,Y_1,\ldots,X_n,Y_n)$, where $X_i \in \{-1,1\}^j$ for all $i \in [n]$ and $Y_i \in \{-1,1\}^k$ for all $i \in [n]$. We give a quantum algorithm and its analysis below.

1. Run $2n$ instances of the Bernstein-Vazirani algorithm: 1 instance on each X_i and 1 instance on each Y_i, to obtain $2n$ strings $x_1,\ldots,x_n,y_1,\ldots,y_n$, where each x_i is a $(\log j)$-bit string and each y_i is a $(\log k)$-bit string.
2. For each X_i and Y_i, query $(X_i)_{1^{\log j}}$ and $(Y_i)_{1^{\log k}}$ to obtain bits $b_i, c_i \in \{-1,1\}$ for all $i \in [n]$.
3. Run Grover's search [11, 6] to check equality of the following two $(nj+nk)$-bit strings: $(b_1 H(x_1),\ldots,b_n H(x_n), c_1 H(y_1),\ldots,c_n H(y_n))$ and $(X_1,\ldots,X_n,Y_1,\ldots,Y_n)$.
4. If the step above outputs that the strings are equal, then output $r(G(x_1,y_1),\ldots,G(x_n,y_n))$. Else, output -1.

Analysis of the algorithm:

- If the input is indeed of the form $(X_1,Y_1),\ldots,(X_n,Y_n)$ where each $X_i \in \pm H(x_i)$ and $Y_i \in \pm H(y_i)$ for some $x_i \in \{-1,1\}^{\log j}$ and $y_i \in \{-1,1\}^{\log k}$, then Step 1 outputs the correct strings $x_1,\ldots,x_n,y_1,\ldots,y_n$ with probability 1 by the properties of the Bernstein-Vazirani algorithm. Step 2 then implies that $X_i = b_i H(x_i)$ and $Y_i = c_i H(y_i)$ for all $i \in [n]$. Next, Step 3 outputs that the strings are equal with probability 1 (since the strings whose equality are to be checked are equal). Hence the algorithm is correct with probability 1 in this case, since $(r \mathbin{\widetilde{\circ}} h_G)(X_1,Y_1,\ldots,X_n,Y_n) = r(G(x_1,y_1),\ldots,G(x_n,y_n))$.
- If the input is such that there exists an index $i \in [n]$ for which $X_i \notin \pm H(x_i)$ for every $x_i \in \{-1,1\}^{\log j}$ or $Y_i \notin \pm H(y_i)$ for every $y_i \in \{-1,1\}^{\log k}$, then the two strings for which equality is to be checked in the Step 3 are not equal. Grover's search catches a discrepancy with probability at least $2/3$. Hence, the algorithm outputs -1 (as does $r \mathbin{\widetilde{\circ}} h_G$), and is correct with probability at least $2/3$ in this case.

Cost of the algorithm:

Step 1 accounts for $2n$ quantum queries. Step 2 accounts for $2n$ quantum queries. Step 3 accounts for $O(\sqrt{n(j+k)})$ quantum queries. Thus, $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = O(n + \sqrt{n(j+k)})$. ◀

As a corollary to Theorem 29, we obtain the following on instantiating $j = k = n$ and r as a Boolean function with quantum query complexity $\Theta(n)$.

▶ **Corollary 30.** *Let $G : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$ be a non-constant function and let $r : \{-1,1\}^n \to \{-1,1\}$ be a total function with $\mathsf{Q}(r) = \Theta(n)$. Then the quantum query complexity of the total function $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = \Theta(n)$.*

Proof. The upper bound $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = O(n)$ follows by plugging in parameters in Theorem 29.

For the lower bound, we show that $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) \geq \mathsf{Q}(r)$. Since G is non-constant, there exist $x_1, y_1, x_2, y_2 \in \{-1,1\}^{\log n}$ such that $G(x_1, y_1) = -1$ and $G(x_2, y_2) = 1$. Let $X_1 = H(x_1), Y_1 = H(y_1)$, $X_2 = H(x_2)$ and $X_2 = H(y_2)$. Consider $r \mathbin{\widetilde{\circ}} h_G$ only restricted to inputs where the inputs to each copy of h_G are either (X_1, Y_1) or (X_2, Y_2). Under this restriction, $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is the same as $r : \{-1,1\}^n \to \{-1,1\}$. Thus $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) \geq \mathsf{Q}(r) = \Omega(n)$. ◀

4.2 On the tightness of the BCW simulation

In this section we first state a communication lower bound (under some model) on $(r \mathbin{\widetilde{\circ}} h_G) \circ \square$ in terms of the communication complexity of $r \circ G$ (in the same model of communication). We state the lemma below (Lemma 31) for the case where the models under consideration are the bounded-error and unbounded-error quantum models, since these are the models of interest to us.

▶ **Lemma 31.** *Let $r : \{-1,1\}^n \to \{-1,1\}$, $G : \{-1,1\}^{\log j} \times \{-1,1\}^{\log k} \to \{-1,1\}$, $\square \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$ and $CC \in \{\mathsf{Q}^{cc,*}, \mathsf{UPP}^{cc}\}$. Then $CC((r \mathbin{\widetilde{\circ}} h_G) \circ \square) \geq CC(r \circ G)$.*

The proof of this lemma follows by a simple reduction. We refer the reader to the full version [8] for a formal proof.

4.2.1 Proof of Theorem 5

The total function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ that we use to prove Theorem 5 is $f = r \mathbin{\widetilde{\circ}} h_G$, where $r = \mathsf{PARITY}_n$ and $G = \mathsf{IP}_{\log n}$. The following claim shows that f is transitive.

▷ Claim 32. Let $n > 0$ be a power of 2. Let $r = \mathsf{PARITY}_n : \{-1,1\}^n \to \{-1,1\}$ and $G = \mathsf{IP}_{\log n} : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$. The function $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is transitive.

Proof. We first show that $h_G : \{-1,1\}^{2n} \to \{-1,1\}$ is transitive. We next observe that $s \mathbin{\widetilde{\circ}} t$ is transitive whenever s is symmetric and t is transitive. The theorem then follows since PARITY_n is symmetric.

Towards showing transitivity of h_G, let $\pi \in S_{2n}$, and $(\sigma_\ell, \sigma_\ell) \in S_{2n}$ for $\ell \in \{-1,1\}^{\log n}$ be defined as follows. (Here $\sigma_\ell \in S_n$; the first copy acts on the first n coordinates, and the second copy acts on the next n coordinates.)

$$\pi(k) = \begin{cases} k+n & k \leq n \\ k-n & k > n. \end{cases}$$

That is, on every string $(x,y) \in \{-1,1\}^{2n}$, the permutation π maps (x,y) to (y,x).

For every $\ell \in \{-1,1\}^{\log n}$, the permutation $\sigma_\ell \in S_n$ is defined as

$$\sigma_\ell(i) = i \oplus \ell, \tag{2}$$

where $i \oplus \ell$ denotes the bitwise XOR of the strings i and ℓ. That is, for every input $(x,y) \in \{-1,1\}^{2n}$ and every $k \in \{-1,1\}^{\log n}$, the input bit x_k is mapped to $x_{k\oplus\ell}$ and y_k is mapped to $y_{k\oplus\ell}$.

For every $(x,y) \in \{-1,1\}^{2n}$ and $i,j \in \{-1,1\}^{\log n}$, the permutation $\sigma_{i\oplus j}(x,y)$ swaps x_i and x_j, and also swaps y_i and y_j. If for $i,j \in \{-1,1\}^{\log n}$, our task was to swap the i'th index of the first n variables with the j'th index of the second n variables, then the permutation $\sigma_{i\oplus j} \circ \pi$ does the job. That is, for every $(x,y) \in \{-1,1\}^{2n}$ and $i,j \in \{-1,1\}^{\log n}$, the permutation $\sigma_{i\oplus j} \circ \pi$ maps x_i to y_j. Thus the set of permutations $\{\pi, \{\sigma_\ell : \ell \in \{-1,1\}^{\log n}\}\}$ acts transitively on S_{2n}.

Now we show that for all $x,y \in \{-1,1\}^{2n}$ and all $\ell \in \{-1,1\}^{\log n}$, we have $h_G(\sigma_\ell(x), \sigma_\ell(y)) = h_G(x,y)$. Fix $\ell \in \{-1,1\}^{\log n}$.

- If $x \in \pm H(s)$ and $y \in \pm H(t)$ are Hadamard codewords, then $x_k = \langle k,s\rangle$ and $y_k = \langle k,t\rangle$ for all $k \in \{-1,1\}^{\log n}$, and $G(x,y) = \langle s,t\rangle$. Thus, for every $k \in \{-1,1\}^{\log n}$ we have $\sigma_\ell(x_k) = x_{k\oplus\ell} = \langle k\oplus\ell, s\rangle = \langle \ell, s\rangle \cdot \langle k,s\rangle$. Hence $\sigma_\ell(x) \in \pm H(s)$ (since $\langle \ell, s\rangle$ does not depend on k, and takes value either 1 or -1). Similarly, $\sigma_\ell(y) \in \pm H(t)$. Thus $h_G(\sigma_\ell(x,y)) = h_G(x,y)$.
- If x (y, respectively) is not a Hadamard codeword, then a similar argument shows that for all $\ell \in [n]$, $\sigma_\ell(x)$ ($\sigma_\ell(y)$, respectively) is also not a Hadamard codeword.

Using the fact that $\langle s,t\rangle = \langle t,s\rangle$ for every $s,t \in \{-1,1\}^{\log n}$, one may verify that $h_G(\pi(x,y)) = h_G(x,y)$ for all $x,y \in \{-1,1\}^{2n}$.

Along with the observation that PARITY_n is a symmetric function, we have that $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is transitive under the following permutations:

- S_n acting on the inputs of PARITY_n, and
- The group generated by $\{\pi\} \cup \{(\sigma_\ell, \sigma_\ell) : \ell \in [n]\}$ acting independently on the inputs of each copy of h_G, where σ_ℓ is as in Equation (2).

◁

Proof of Theorem 5. Let $n > 0$ be a power of 2. Let $r = \mathsf{PARITY}_n : \{-1,1\}^n \to \{-1,1\}$ and $G = \mathsf{IP}_{\log n} : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$. Let $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$. By Claim 32, f is transitive. By Corollary 30 we have $\mathsf{Q}(f) = \Theta(n)$. For the communication lower bound we have

$$\begin{aligned}
\mathsf{UPP}^{cc}(f \circ \square) &= \mathsf{UPP}^{cc}((r \mathbin{\widetilde{\circ}} h_G) \circ \square) \\
&\geq \mathsf{UPP}^{cc}(\mathsf{PARITY}_n \circ \mathsf{IP}_{\log n}) && \text{by Lemma 31} \\
&= \mathsf{UPP}^{cc}(\mathsf{IP}_{n \log n}) && \text{Observation 15} \\
&= \Omega(n \log n). && \text{by Theorem 20}
\end{aligned}$$

◀

References

1 Scott Aaronson and Andris Ambainis. Quantum search of spatial regions. *Theory of Computing*, 1(1):47–79, 2005. Earlier version in FOCS'03. quant-ph/0303041.

2 Dorit Aharonov, Aram W. Harrow, Zeph Landau, Daniel Nagaj, Mario Szegedy, and Umesh V. Vazirani. Local tests of global entanglement and a counterexample to the generalized area law. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 246–255, 2014. doi:10.1109/FOCS.2014.34.

3 Andris Ambainis and Ronald de Wolf. How low can approximate degree and quantum query complexity be for total Boolean functions? *Computational Complexity*, 23(2):305–322, 2014. Earlier version in CCC'13.

4 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. quant-ph/9802049.

5 Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

6 Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055.

7 Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 63–68, 1998. doi:10.1145/276698.276713.

8 Sourav Chakraborty, Arkadev Chattopadhyay, Peter Høyer, Nikhil S. Mande, Manaswi Paraashar, and Ronald de Wolf. Symmetry and quantum query-to-communication simulation. *CoRR*, abs/2012.05233, 2020. URL: https://arxiv.org/abs/2012.05233.

9 Sourav Chakraborty, Arkadev Chattopadhyay, Nikhil S. Mande, and Manaswi Paraashar. Quantum query-to-communication simulation needs a logarithmic overhead. In *Proceedings of the 35th Computational Complexity Conference (CCC)*, pages 32:1–32:15, 2020. doi:10.4230/LIPIcs.CCC.2020.32.

10 Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and Systems Sciences*, 65(4):612–625, 2002. doi:10.1016/S0022-0000(02)00019-3.

11 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 212–219, 1996.

12 Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Sign rank vs discrepancy. In *Proceedings of the 35th Computational Complexity Conference (CCC)*, pages 18:1–18:14, 2020. doi:10.4230/LIPIcs.CCC.2020.18.

13 Peter Høyer and Ronald de Wolf. Improved quantum communication complexity bounds for disjointness and equality. In *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 299–310, 2002.

14 Peter Høyer, Michele Mosca, and Ronald de Wolf. Quantum search on bounded-error inputs. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2719 of *Lecture Notes in Computer Science*, pages 291–299. Springer, 2003. quant-ph/0304052.

15 Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Shigeru Yamashita. Unbounded-error one-way classical and quantum communication complexity. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 110–121. Springer, 2007.

16 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

17 Troy Lee and Shengyu Zhang. Composition theorems in communication complexity. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming, (ICALP)*, pages 475–489, 2010. doi:10.1007/978-3-642-14165-2_41.

18 Ilan Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 1991.

19 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

20 Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 468–474, 1992. doi:10.1145/129712.129758.

21 Ramamohan Paturi and Janos Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986.

22 Alexander Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya of the Russian Academy of Sciences, mathematics*, 67(1):159–176, 2003. quant-ph/0204025.

23 Ronald de Wolf. Quantum communication and complexity. *Theoretical Computer Science*, 287(1):337–353, 2002.

24 Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.

25 Andrew Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science (FOCS)*, pages 352–361. IEEE, 1993.

A Noisy amplitude amplification and a new distributed-search protocol

In this section we prove Theorem 21, restated below.

▶ **Theorem 33** (Restatement of Theorem 21). *Let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a two-party function, $X = (X_1, \ldots, X_n) \in \{-1,1\}^{nj}$ and $Y = (Y_1, \ldots, Y_n) \in \{-1,1\}^{nk}$. Define $z = (G(X_1, Y_1), \ldots, G(X_n, Y_n)) \in \{-1,1\}^n$. Assume Alice and Bob start with $\lceil \log n \rceil$ shared EPR-pairs.*

- *There exists a quantum protocol using $O(\sqrt{n}\, \mathsf{Q}_E^{cc}(G))$ qubits of communication that finds (with success probability ≥ 0.99) an $i \in [n]$ such that $z_i = -1$ if such an i exists, and says "no" with probability 1 if no such i exists.*
- *If the number of -1s in z is within a factor of 2 from a known integer t, then the communication can be reduced to $O(\sqrt{n/t}\, \mathsf{Q}_E^{cc}(G))$ qubits.*

▶ Remark 34. The $\log n$ shared EPR-pairs that we assume Alice and Bob share at the start could also be established by means of $\log n$ qubits of communication at the start of the protocol. For the result in the first bullet, this additional communication does not change the asymptotic bound. For the result of the second bullet, if $t \leq n\mathsf{Q}_E^{cc}(G)^2/(\log n)^2$, then this additional communication does not change the asymptotic bound either. However, if $t = \omega(n/(\log n)^2)$ and $\mathsf{Q}_E^{cc}(G) = O(1)$ then the quantum communication $O(\sqrt{n/t}\, \mathsf{Q}_E^{cc}(G))$ is $o(\log n)$ and establishing the $\log n$ EPR-pairs by means of a first message makes a difference.

As a corollary, we obtain a new $O(\sqrt{n})$-qubit protocol for the distributed search problem composed with $G = \mathsf{AND}_2$ (whose decision version is the Set-Disjointness problem).

A.1 Amplitude amplification with perfect reflections

We first describe basic amplitude amplification in a slightly unusual recursive manner, similar to [14]. We are dealing with a search problem where some set $\mathcal{G}$ of basis states are deemed "good" and the other basis states are deemed "bad." Let $P_{\mathcal{G}} = \sum_{g \in \mathcal{G}} |g\rangle\langle g|$ be the projector onto the span of the good basis states, and $O_{\mathcal{G}} = I - 2P_{\mathcal{G}}$ be the reflection that puts a '$-$' in front of the good basis states: $O_{\mathcal{G}}|g\rangle = -|g\rangle$ for all basis states $g \in \mathcal{G}$, and $O_{\mathcal{G}}|b\rangle = |b\rangle$ for all basis states $b \notin \mathcal{G}$.

Suppose we have an initial state $|\psi\rangle$ which is a superposition of a good state and a bad state:

$$|\psi\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle,$$

where $|G\rangle = P_{\mathcal{G}}|\psi\rangle / \|P_{\mathcal{G}}|\psi\rangle\|$ and $|B\rangle = (I - P_{\mathcal{G}})|\psi\rangle / \|(I - P_{\mathcal{G}})|\psi\rangle\|$. For example in Grover's algorithm, with a search space of size n containing t solutions, the initial state $|\psi\rangle$ would be the uniform superposition, and its overlap (inner product) with the good subspace spanned by the t "good" (sometimes called "marked") basis states would be $\sin(\theta) = \sqrt{t/n}$.

We'd like to increase the weight of the good state, i.e., move the angle θ closer to $\pi/2$. Let R_ψ denote the reflection about the state $|\psi\rangle$, i.e., $R_\psi|\psi\rangle = |\psi\rangle$ and $R_\psi|\phi\rangle = -|\phi\rangle$ for every $|\phi\rangle$ that is orthogonal to $|\psi\rangle$. Then the algorithm $A_1 = R_\psi \cdot O_{\mathcal{G}}$ is the product of two reflections, which (in the 2-dimensional space spanned by $|G\rangle$ and $|B\rangle$) corresponds to a rotation by an angle 2θ, thus increasing our angle from θ to 3θ. This is the basic amplitude amplification step. It maps

$$|\psi\rangle \mapsto A_1|\psi\rangle = \sin(3\theta)|G\rangle + \cos(3\theta)|B\rangle.$$

We can now repeat this step recursively, defining

$$A_2 = A_1 R_\psi A_1^* \cdot O_\mathcal{G} \cdot A_1.$$

Note that $A_1 R_\psi A_1^*$ is a reflection about the state $A_1|\psi\rangle$. Thus A_2 triples the angle between $A_1|\psi\rangle$ and $|B\rangle$, mapping

$$|\psi\rangle \mapsto A_2|\psi\rangle = \sin(9\theta)|G\rangle + \cos(9\theta)|B\rangle.$$

Continuing recursively in this fashion, define the algorithm

$$A_{j+1} = A_j R_\psi A_j^* \cdot O_\mathcal{G} \cdot A_j. \tag{3}$$

The last algorithm A_k will map

$$|\psi\rangle \mapsto A_k|\psi\rangle = \sin(3^k\theta)|G\rangle + \cos(3^k\theta)|B\rangle.$$

Hence after k recursive amplitude amplification steps, we have angle $3^k\theta$. Since we want to end up with angle $\approx \pi/2$, if we know θ then we can choose

$$k = \lfloor \log_3(\pi/(2\theta)) \rfloor. \tag{4}$$

This gives us an angle $3^k\theta \in (\pi/6, \pi/2]$, so the final state $A_k|\psi\rangle$ has overlap $\sin(\theta_k) > 1/2$ with the good state $|G\rangle$.

Let C_k denote the "cost" (in whatever measure, for example query complexity, or communication complexity, or circuit size) of algorithm A_k. Looking at its recursive definition (Equation (3)), C_k is 3 times C_{k-1}, plus the cost of R_ψ plus the cost of $O_\mathcal{G}$. If we just count applications of $O_\mathcal{G}$ ("queries"), considering R_ψ to be free, then $C_{k+1} = 3C_k + 1$. This recursion has the closed form $C_k = \sum_{i=0}^{k-1} 3^i < 3^k$. With the above choice of k we get $C_k = O(1/\theta)$. In the case of Grover's algorithm, where $\theta = \arcsin(\sqrt{t/n}) \approx \sqrt{t/n}$, the cost is $C_k = O(\sqrt{n/t})$.

A.2 Amplitude amplification with imperfect reflections

Now we consider the situation where we do not implement the reflections R_ψ perfectly, but instead implement another unitary R_ψ^ε at operator-norm distance $\left\|R_\psi^\varepsilon - R_\psi\right\| \leq \varepsilon$ from R_ψ, with the additional property that $R_\psi^\varepsilon|\psi\rangle = |\psi\rangle$ (this one-sided error property will be important for the proof). We can control this error ε, but smaller ε will typically correspond to higher cost of R_ψ^ε. The reflection $O_\mathcal{G}$ will still be implemented perfectly below.

We again start with the initial state

$$|\psi\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle.$$

For errors $\varepsilon_1, \ldots, \varepsilon_k$ that we will specify later, recursively define the following algorithms.

$$A_1 = R_\psi^{\varepsilon_1} \cdot O_\mathcal{G} \quad \text{and} \quad A_{j+1} = A_j R_\psi^{\varepsilon_{j+1}} A_j^* \cdot O_\mathcal{G} \cdot A_j.$$

These algorithms will map the initial state as follows:

$$|\psi\rangle \mapsto |\psi_j\rangle = A_j|\psi\rangle = \sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle + |E_j\rangle, \tag{5}$$

where $|E_j\rangle$ is some unnormalized error state defined by the above equation; its norm η_j quantifies the extent to which we deviate from perfect amplitude amplification. Our goal here

is to upper bound this η_j. In order to see how η_j can grow, let us see how $A_j R_\psi^{\varepsilon_{j+1}} A_j^* \cdot O_{\mathcal{G}}$ acts on $\sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle$ (we'll take into account the effects of the error term $|E_j\rangle$ later). If $R_\psi^{\varepsilon_{j+1}}$ were equal to R_ψ, then we would have one perfect round of amplitude amplification and obtain $\sin(3^{j+1}\theta)|G\rangle + \cos(3^{j+1}\theta)|B\rangle$; but since $R_\psi^{\varepsilon_{j+1}}$ is only ε_{j+1}-close to R_ψ, additional errors can appear. First we apply $O_{\mathcal{G}}$, which flips the phase of $|G\rangle$ and hence changes the state to

$$-\sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle = |\psi_j\rangle - |E_j\rangle - 2\sin(3^j\theta)|G\rangle.$$

Second we apply $V = A_j R_\psi^{\varepsilon_{j+1}} A_j^*$. Let $V' = A_j R_\psi A_j^*$, and note that $V|\psi_j\rangle = V'|\psi_j\rangle = |\psi_j\rangle$ and $\|V' - V\| = \left\|R_\psi - R_\psi^{\varepsilon_{j+1}}\right\| \le \varepsilon_{j+1}$. The new state is

$$\begin{aligned} V(|\psi_j\rangle - |E_j\rangle - 2\sin(3^j\theta)|G\rangle) &= V'(|\psi_j\rangle - |E_j\rangle - 2\sin(3^j\theta)|G\rangle) + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle) \\ &= V'(-\sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle) + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle) \\ &= \sin(3^{j+1}\theta)|G\rangle + \cos(3^{j+1}\theta)|B\rangle + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle). \end{aligned}$$

Putting back also the earlier error term $|E_j\rangle$ from Equation (5) (to which the unitary $VO_{\mathcal{G}}$ is applied as well), it follows that the new error state is

$$|E_{j+1}\rangle = |\psi_{j+1}\rangle - (\sin(3^{j+1}\theta)|G\rangle + \cos(3^{j+1}\theta)|B\rangle) = VO_{\mathcal{G}}|E_j\rangle + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle).$$

Its norm is

$$\begin{aligned} \eta_{j+1} &\le \|VO_{\mathcal{G}}|E_j\rangle\| + \left\|(V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle)\right\| \\ &\le \eta_j + \varepsilon_{j+1}(\eta_j + 2\sin(3^j\theta)) = (1+\varepsilon_{j+1})\eta_j + 2\varepsilon_{j+1}\sin(3^j\theta). \end{aligned}$$

Since $\eta_0 = 0$, we can "unfold" the above recursive upper bound to the following, which is easy to verify by induction on k:

$$\eta_k \le \sum_{j=1}^{k} \prod_{\ell=j+1}^{k} (1+\varepsilon_\ell) 2\varepsilon_j \sin(3^{j-1}\theta). \tag{6}$$

For each $1 \le j \le k$, choose

$$\varepsilon_j = \frac{1}{100 \cdot 4^j}. \tag{7}$$

Note that $\sigma = \sum_{j=1}^{k} \varepsilon_k \le 1/300$. With this choice of ε_j's, and the inequalities $1 + x \le e^x$, $e^\sigma \le 1.5$ and $\sin(x) \le x$ for $x \le \pi/2$ (which is the case here), we can upper bound the norm of the error term $|E_k\rangle$ after k iterations (see Equation (5)) as

$$\eta_k \le \sum_{j=1}^{k} e^\sigma 2\varepsilon_j 3^{j-1}\theta \le \frac{3\theta}{400}\sum_{j=1}^{k}(3/4)^{j-1} \le \frac{3\theta}{100}. \tag{8}$$

Accordingly, up to very small error we have done perfect amplitude amplification.

A.3 Distributed amplitude amplification with imperfect reflection

We will now instantiate the above scheme to the case of *distributed* search, where our measure of cost is communication, that is, the number of qubits sent between Alice and Bob. Specifically, consider the *intersection problem* where Alice and Bob have inputs $x \in \{-1,1\}^n$

and $y \in \{-1,1\}^n$, respectively. Assume for simplicity that n is a power of 2, so $\log n$ is an integer. Alice and Bob want to find an $i \in \{0,\ldots,n-1\} = \{0,1\}^{\log n}$ such that $x_i = y_i = -1$, if such an i exists.

The basis states in this distributed problem are $|i\rangle|j\rangle$, and we define the set of "good" basis states as

$$\mathcal{G} = \{|i\rangle|j\rangle \mid x_i = y_j = -1\},$$

even though we are only looking for i,j where $i=j$ (it's easier to implement $O_{\mathcal{G}}$ with this more liberal definition of $\mathcal{G}$). Our protocol will start with the maximally entangled initial state $|\psi\rangle$ in n dimensions, which corresponds to $\log n$ EPR-pairs:

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{i\in\{0,1\}^{\log n}} |i\rangle|i\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle,$$

where we assume there are t i's where $x_i = y_i = -1$, i.e., t solutions to the intersection problem, so

$$\theta = \arcsin(\sqrt{t/n}). \tag{9}$$

and

$$|G\rangle = \frac{1}{\sqrt{t}} \sum_{(i,i)\in\mathcal{G}} |i\rangle|i\rangle.$$

It costs $\lceil \log n \rceil$ qubits of communication between Alice and Bob to establish this initial shared state, or it costs nothing if we assume pre-shared entanglement. Our goal is to end up with a state that has large inner product with $|G\rangle$.

In order to be able to use amplitude amplification, we would like to be able to reflect about the above state $|\psi\rangle$. However, in general this perfect reflection R_ψ costs a lot of communication: Alice would send her $\log n$ qubits to Bob, who would unitarily put a -1 in front of all states orthogonal to $|\psi\rangle$, and then sends back Alice's qubits. This has a communication cost of $O(\log n)$ qubits, which is too much for our purposes. Fortunately, Theorem 19 gives us a way to implement a one-sided ε-error reflection protocol R^ε_ψ that only costs $O(\log(1/\varepsilon))$ qubits of communication.

The reflection $O_{\mathcal{G}}$ puts a '$-$' in front of the basis states $|i\rangle|j\rangle$ in $\mathcal{G}$. This can be implemented perfectly using only 2 qubits of communication, as follows. For the variables $x_i \in \{-1,1\}$, let $\widehat{x_i}$ denote their $\{0,1\}$-valued counterparts. That is, $\widehat{x_i} = 1$ if $x_i = -1$ and $\widehat{x_i} = 0$ if $x_i = 1$. To implement the reflection $O_{\mathcal{G}}$ on her basis state $|i\rangle$, Alice XORs $|\widehat{x_i}\rangle$ into a fresh auxiliary $|0\rangle$-qubit and sends this qubit to Bob. Bob receives this qubit and applies the following unitary map:

$$|b\rangle|j\rangle \mapsto y_j^b|b\rangle|j\rangle, \quad b\in\{0,1\}, j\in[n].$$

He sends back the auxiliary qubit. Alice sets the auxiliary qubit back to $|0\rangle$ by XOR-ing $\widehat{x_i}$ into it. Ignoring the auxiliary qubit (which starts and ends in state $|0\rangle$), this maps $|i\rangle|j\rangle \mapsto (-1)^{[x_i=y_j=-1]}|i\rangle|j\rangle$. Hence we have implemented $O_{\mathcal{G}}$ correctly: a minus sign is applied exactly for the good basis states, the ones where $x_i = y_j = -1$.

Now consider the algorithms (more precisely, communication protocols):

$$A_1 = R^{\varepsilon_1}_\psi \cdot O_{\mathcal{G}} \quad \text{and} \quad A_{j+1} = A_j R^{\varepsilon_{j+1}}_\psi A_j^* \cdot O_{\mathcal{G}} \cdot A_j$$

with the choice of ε_j's from Equation (7). If we pick $k = \lfloor \log_3(\pi/(2\theta)) \rfloor$, like in Equation (4), then $3^k\theta \in (\pi/6, \pi/2]$. Hence by Equation (5) and Equation (8), the inner product of our final state with $|G\rangle$ will be between $\sin(3^k\theta) - 3\theta/100 \geq 0.4$ and 1.

At this point Alice and Bob can measure, and with probability $\geq 0.4^2$ they will each see the same i, with the property that $x_i = y_i = -1$.

From Equation (3) and Theorem 19, the recursion for the communication costs of these algorithms is

$$C_{j+1} = 3C_j + O(\log(1/\varepsilon_{j+1})) + 2.$$

Solving this recurrence with our ε_j's from Equation (7) and the value of θ from Equation (9) we obtain

$$C_k = \sum_{j=1}^{k} 3^{k-j}(O(\log(1/\varepsilon_j)) + 2) = \sum_{j=1}^{k} 3^{k-j}O(j) = O(3^k) = O(\sqrt{n/t}).$$

Thus, using $O(\sqrt{n/t})$ qubits of communication we can find (with constant success probability) an intersection point i. This also allows us to solve the Set-Disjointness problem (the decision problem whose output is 1 if there is no intersection between x and y). Note that if the t we used equals the actual number of solutions only up to a factor of 2, the above protocol still has $\Omega(1)$ probability to find a solution, and $O(1)$ repetitions will boost this success probability to 0.99. In case we do not even know t approximately, we can use the standard technique of trying exponentially decreasing guesses for t to find an intersection point with communication $O(\sqrt{n})$.

Note that there is no log-factor in the communication complexity, in contrast to the original $O(\sqrt{n}\log n)$-qubit Grover-based quantum protocol for the intersection problem of Buhrman et al. [7]. Aaronson and Ambainis [1] earlier already managed to remove the log-factor, giving an $O(\sqrt{n})$-qubit protocol for Set-Disjointness as a consequence of their local version of quantum search on a grid graph (which is optimal [22]). We have just reproved this result of [1] in a different and arguably simpler way.

The above description is geared towards the intersection problem, where the "inner" function is $G = \mathsf{AND}_2$: we called a basis state $|i\rangle|j\rangle$ "good" if $x_i = y_j = -1$. However, this can easily be generalized to the situation where Alice and Bob's respective inputs are $X = (X_1, \ldots, X_n)$ and $Y = (Y_1, \ldots, Y_n)$ and we want to find an $i \in [n]$ where $G(X_i, Y_i) = -1$ for some two-party function G, and define the set of "good" basis states as $\mathcal{G} = \{|i\rangle|j\rangle \mid G(X_i, Y_j) = -1\}$.[4] The only thing that changes in the above is the implementation of the reflection $O_{\mathcal{G}}$, which would now be computed by means of an exact quantum communication protocol for $G(X_i, Y_j)$, at a cost of $2\mathsf{Q}_E^{cc}(G)$ qubits of communication.[5] Note that because we can check (at the expense of another $\mathsf{Q}_E^{cc}(G)$ qubits of communication) whether the output index i actually satisfies $G(X_i, Y_i) = -1$, we may assume the protocol has one-sided error: it always outputs "no" if there is no such i. This concludes the proof of Theorem 21.

[4] We intentionally use the letter 'G' to mean "good" in $\mathcal{G}$ and and to refer to the two-party function G, since G determines which basis states $|i\rangle|j\rangle$ are "good."

[5] The factor of 2 is to reverse the protocol after the phase $G(X_i, Y_j)$ has been added to basis state $|i\rangle|j\rangle$, in order to set any workspace qubits back to $|0\rangle$.