Quantum Communication Complexity
Ronald de Wolf

Communication complexity has been studied extensively in the area of theoretical computer science and has deep connections with seemingly unrelated areas, such as VLSI design, circuit lower bounds, lower bounds on branching programs, size of data structures, and bounds on the length of logical proof systems, to name just a few.

1 Classical communication complexity

First we sketch the setting for classical communication complexity. Alice and Bob want to compute some function $f : D \rightarrow \{0, 1\}$, where $D \subseteq X \times Y$.\footnote{If the domain $D$ equals $X \times Y$ then $f$ is called a total function, otherwise it is called a partial or promise function.} Alice receives input $x \in X$, Bob receives input $y \in Y$, with $(x, y) \in D$. A typical situation, illustrated in Fig. 1, is where $X = Y = \{0, 1\}^n$, so both Alice and Bob receive an $n$-bit input string. As the value $f(x, y)$ will generally depend on both $x$ and $y$, some communication between Alice and Bob is required in order for them to be able to compute $f(x, y)$. We are interested in the \textit{minimal} amount of communication they need.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{communication_complexity}
\caption{Alice and Bob solving a communication complexity problem}
\end{figure}

A communication \textit{protocol} is a distributed algorithm where first Alice does some individual computation, and then sends a message (of one or more bits) to Bob, then Bob does some computation and sends a message to Alice, etc. Each message is called a \textit{round}. After one or more rounds the protocol terminates and one of the parties (let’s say Bob) outputs some value that should be $f(x, y)$. The \textit{cost} of a protocol is the total number of bits communicated on the worst-case input. A \textit{deterministic} protocol for $f$ always has to output the right value $f(x, y)$ for all $(x, y) \in D$. In a \textit{bounded-error} protocol, Alice and Bob may flip coins and the protocol has to output the right value $f(x, y)$ with probability $\geq 2/3$ for all $(x, y) \in D$. We could either allow Alice and Bob to toss
coins individually (local randomness, or “private coin”) or jointly (shared randomness, or “public coin”). A public coin can simulate a private coin and is potentially more powerful. However, Newman’s theorem [8] says that having a public coin can save at most $O(\log n)$ bits of communication, compared to a protocol with a private coin.

To illustrate the power of randomness, let us give a simple yet efficient bounded-error protocol for the equality problem, where the goal for Alice is to determine whether her $n$-bit input is the same as Bob’s or not: $f(x, y) = 1$ if $x = y$, and $f(x, y) = 0$ otherwise. Alice and Bob jointly toss a random string $r \in \{0, 1\}^n$. Alice sends the bit $a = x \cdot r$ to Bob (where ‘$\cdot$’ is inner product mod 2). Bob computes $b = y \cdot r$ and compares this with $a$. If $x = y$ then $a = b$, but if $x \neq y$ then $a \neq b$ with probability $1/2$. Repeating this a few times, Alice and Bob can decide equality with small error using $O(n)$ public coin flips and a constant amount of communication. This protocol uses public coins, but note that Newman’s theorem implies that there exists an $O(\log n)$-bit protocol that uses a private coin.

2 The quantum question

Now what happens if we give Alice and Bob a quantum computer and allow them to send each other qubits and/or to make use of EPR-pairs that they share at the start of the protocol?

Formally speaking, we can model a quantum protocol as follows. The total state consists of 3 parts: Alice’s private space, the channel, and Bob’s private space. The starting state is $|x\rangle |0\rangle |y\rangle$: Alice gets $x$, the channel is initialized to 0, and Bob gets $y$. Now Alice applies a unitary transformation to her space and the channel. This corresponds to her private computation as well as to putting a message on the channel (the length of this message is the number of channel-qubits affected by Alice’s operation). Then Bob applies a unitary transformation to his space and the channel, etc. At the end of the protocol Alice or Bob makes a measurement to determine the output of the protocol. This model was introduced by Yao [11].

In the second model, introduced by Cleve and Buhrman [5], Alice and Bob share an unlimited number of EPR-pairs at the start of the protocol, but now they communicate via a classical channel: the channel has to be in a classical state throughout the protocol. We only count the communication, not the number of EPR-pairs used. Protocols of this kind can simulate protocols of the first kind with only a factor 2 overhead: using teleportation, the parties can send each other a qubit using an EPR-pair and two classical bits of communication. Hence the qubit-protocols that we describe below also immediately yield protocols that work with entanglement and a classical channel. Note that an EPR-pair can simulate a public coin toss: if Alice and Bob each measure their half of the pair of qubits, they get the same random bit.

The third variant combines the strengths of the other two: here Alice and Bob start out with an unlimited number of EPR-pairs and they are allowed to communicate qubits. This third kind of communication complexity is in fact equivalent to the second, up to a factor of 2, again by teleportation.

Before continuing to study this model, we first have to face an important question: is there anything to be gained here? At first sight, the following argument seems to rule out any significant gain. Suppose that in the classical world $k$ bits have to be communicated in order to compute $f$. Since Holevo’s theorem says that $k$ qubits cannot contain more information than $k$ classical bits, it seems that the quantum communication complexity should be roughly $k$ qubits as well (maybe $k/2$ to account for superdense coding, but not less). Surprisingly (and fortunately for us), this argument
is false, and quantum communication can sometimes be much less than classical communication complexity. The information-theoretic argument via Holevo’s theorem fails, because Alice and Bob do not need to communicate the information in the \( k \) bits of the classical protocol; they are only interested in the value \( f(x, y) \), which is just 1 bit. Below we will go over four of the main examples that have so far been found of differences between quantum and classical communication complexity.

### 3 Example 1: Distributed Deutsch-Jozsa

The first impressively large gaps between quantum and classical communication complexity were exhibited by Buhrman, Cleve, and Wigderson [4]. Their protocols are distributed versions of known quantum query algorithms, like the Deutsch-Jozsa and Grover algorithms. Let us start with the first one. It is actually explained most easily in a direct way, without reference to the Deutsch-Jozsa algorithm (though that is where the idea came from). The problem is a promise version of the equality problem. Suppose the \( n \)-bit inputs \( x \) and \( y \) are restricted to the following case:

Distributed Deutsch-Jozsa: either \( x = y \), or \( x \) and \( y \) differ in exactly \( n/2 \) positions

Note that this promise only makes sense if \( n \) is an even number, otherwise \( n/2 \) would not be integer. In fact it will be convenient to assume \( n \) a power of 2. Here is a simple quantum protocol to solve this promise version of equality using only \( \log n \) qubits:

1. Alice sends Bob the \( \log n \)-qubit state \( \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (-1)^{x_i} |i\rangle \), which she can prepare unitarily from \( x \) and \( \log n |0\rangle \)-qubits.

2. Bob applies the unitary map \(|i\rangle \mapsto (-1)^{y_i} |i\rangle\) to the state, applies a Hadamard transform to each qubit (for this it is convenient to view \( i \) as a \( \log n \)-bit string), and measures the resulting \( \log n \)-qubit state.

3. Bob outputs 1 if the measurement gave \( |0^\log n\rangle \) and outputs 0 otherwise.

It is clear that this protocol only communicates \( \log n \) qubits, but why does it work? Note that the state that Bob measures is

\[
H^{\otimes \log n} \left( \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (-1)^{x_i+y_i} |i\rangle \right) = \frac{1}{n} \sum_{i=1}^{n} (-1)^{x_i+y_i} \sum_{j \in \{0,1\}^{\log n}} (-1)^{i\cdot j} |j\rangle
\]

This superposition looks rather unwieldy, but consider the amplitude of the \( |0^{\log n}\rangle \) basis state. It is \( \frac{1}{n} \sum_{i=1}^{n} (-1)^{x_i+y_i} \), which is 1 if \( x = y \) and 0 otherwise because the promise now guarantees that \( x \) and \( y \) differ in exactly \( n/2 \) of the bits! Hence Bob will always give the correct answer.

What about efficient classical protocols (without entanglement) for this problem? Proving lower bounds on communication complexity often requires a very technical combinatorial analysis. Buhrman, Cleve, and Wigderson used a deep combinatorial result from [6] to prove that every classical errorless protocol for this problem needs to send at least 0.007\( n \) bits.

This \( \log n \)-qubits-vs.-0.007\( n \)-bits example was the first exponentially large separation of quantum and classical communication complexity. Notice, however, that the difference disappears if we move to the bounded-error setting, allowing the protocol to have some small error probability. We can
use the randomized protocol for equality discussed above or even simpler: Alice can just send a few 
\((i, x_i)\) pairs to Bob, who then compares the \(x_i\)'s with his \(y_i\)'s. If \(x = y\) he will not see a difference, 
but if \(x\) and \(y\) differ in \(n/2\) positions, then Bob will probably detect this. Hence \(O(\log n)\) classical 
bits of communication suffice in the bounded-error setting, in sharp contrast to the errorless setting.

4 Example 2: The Intersection problem

Now consider the Intersection function, which is 1 if \(x_i = y_i = 1\) for at least one \(i\). Buhrman, Cleve, 
and Wigderson [4] also presented an efficient quantum protocol for this, based on Grover’s search 
algorithm. We can solve Intersection if we can solve the following search problem: find some \(i\) such 
that \(x_i = y_i = 1\) (if such an \(i\) exists).\(^2\) We want to find a solution to the search problem on the 
string \(z = x \land y\) (which is the bit-wise AND of \(x\) and \(y\)), since \(z_i = 1\) whenever both \(x_i = 1\) and 
\(y_i = 1\). The idea is now to let Alice run Grover’s algorithm to search for such a solution. Clearly, 
she can prepare the uniform starting state herself. She can also apply \(H\) and \(O_G\) herself. The only 
thing where she needs Bob’s help, is in implementing \(O_z\). This they do as follows. Whenever Alice 
want to apply \(O_z\) to a state 
\[
|\phi\rangle = \sum_{i=1}^{n} \alpha_i |i\rangle,
\]
she tags on her \(x_i\) in an extra qubit and sends Bob the state 
\[
\sum_{i=1}^{n} \alpha_i |i\rangle |x_i\rangle.
\]
Bob applies the unitary map 
\[
|i\rangle |x_i\rangle \mapsto (-1)^{x_i \land y_i} |i\rangle |x_i\rangle
\]
and sends back the result. Alice sets the last qubit back to |0\rangle (which she can do unitarily because 
she has \(x\)), and now she has the state \(O_z|\phi\rangle\). Thus we can simulate \(O_z\) using 2 messages of \(\log(n) + 1\) 
qubits each. Thus Alice and Bob can run Grover’s algorithm to find an intersection, using \(O(\sqrt{n})\) 
messages of \(O(\log n)\) qubits each, for total communication of \(O(\sqrt{n} \log n)\) qubits. Later Aaronson 
and Ambainis [1] gave a more complicated protocol that uses \(O(\sqrt{n})\) qubits of communication.

What about lower bounds? It is a well-known result of classical communication complexity that 
classical bounded-error protocols for the Intersection problem need about \(n\) bits of communication. 
Thus we have a quadratic quantum-classical separation for this problem. Could there be a quantum 
protocol that uses much less than \(\sqrt{n}\) qubits of communication? This question was open for quite 
a few years after [4] appeared, until finally Razborov [10] showed that any bounded-error quantum 
protocol for Intersection needs to communicate about \(\sqrt{n}\) qubits.

5 Example 3: The vector-in-subspace problem

Notice the contrast between the examples of the last two sections. For the Distributed Deutsch- 
Jozsa problem we get an exponential quantum-classical separation, but the separation only holds

---

\(^2\)This is sometimes called the appointment-scheduling problem: view \(x\) and \(y\) as Alice’s and Bob’s agendas, 
respectively, with a 1 indicating a free timeslot. Then the goal is to find a timeslot where Alice and Bob are both 
free.
if we require the classical protocol to be errorless. On the other hand, the gap for the disjointness function is only quadratic, but it holds even if we allow classical protocols to have some error probability.

Here is a function where the quantum-classical separation has both features: the quantum protocol is exponentially better than the classical protocol, even if the latter is allowed some error:

Alice receives a unit vector $v \in \mathbb{R}^m$
Bob receives two $m$-dimensional projectors $P_0$ and $P_1$ such that $P_0 + P_1 = I$
Promise: either $P_0 v = v$ or $P_1 v = v$.
Question: which of the two?

As stated, this is a problem with continuous input, but it can be discretized in a natural way by approximating each real number by $O(\log m)$ bits. Alice and Bob’s input is now $n = O(m^2 \log m)$ bits long. There is a simple yet efficient 1-round quantum protocol for this problem: Alice views $v$ as a log $m$-qubit state and sends this to Bob; Bob measures with operators $P_0$ and $P_1$, and outputs the result. This takes only $\log m = O(\log n)$ qubits of communication.

The efficiency of this protocol comes from the fact that an $m$-dimensional unit vector can be “compressed” or “represented” as a log $m$-qubit state. Similar compression is not possible with classical bits, which suggests that any classical protocol will have to send the vector $v$ more or less literally and hence will require a lot of communication. This turns out to be true but the proof is quite hard [7]. It shows that any bounded-error protocol needs to send at least roughly $n^{1/3}$ bits.

6 Simultaneous communication: Quantum fingerprinting

The previous examples were either exponential quantum improvements for promise problems (Deutsch-Jozsa and vector-in-subspace) or polynomial improvements for total problems (disjointness). We will now give an exponential improvement for the total problem of equality-testing, but in a restricted setting called the simultaneous message passing (SMP) model. Alice and Bob receive $n$-bit input $x$ and $y$, respectively. They do not have any shared resources like shared randomness or an entangled state, but they do have local randomness. They don’t communicate with each other directly, but instead send a single message to a third party, called the Referee. The Referee, upon receiving message $m_A$ from Alice and $m_B$ from Bob, should output the value $f(x, y)$. The goal is to compute $f(x, y)$ with a minimum amount of communication from Alice and Bob to the Referee.

We will see that for the equality problem there is an exponential savings in communication when qubits are used instead of classical bits. Classically, the problem of the bounded-error communication complexity of equality in the SMP model was open for almost twenty years, until Newman and Szegedy [9] exhibited a lower bound of about $\sqrt{n}$ bits. This is tight, since Ambainis [2] constructed a bounded-error protocol for this problem where the messages are $O(\sqrt{n})$ bits long (see homework). In contrast, in the quantum setting this problem can be solved with very little communication: only $O(\log n)$ qubits suffice [3].

The quantum idea is to associate each $x \in \{0, 1\}^n$ with a short quantum state $|\phi_x\rangle$, called the quantum fingerprint of $x$. Let us consider an error-correcting code $C : \{0, 1\}^n \rightarrow \{0, 1\}^N$. There exist codes where $N = O(n)$ and any two codewords $C(x)$ and $C(y)$ have Hamming distance close to $N/2$, say $d(C(x), C(y)) \in [0.49N, 0.51N]$ (for instance, a random linear code will work). Define
the quantum fingerprint of $x$ as follows:

$$|\phi_x\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} (-1)^{C(x)_j} |j\rangle.$$ 

This is a unit vector in an $N$-dimensional space, so it corresponds to only $\lceil \log N \rceil = \log(n) + O(1)$ qubits. For distinct $x$ and $y$, the corresponding fingerprints will have small inner product:

$$\langle \phi_x | \phi_y \rangle = \frac{1}{N} \sum_{j=1}^{N} (-1)^{C(x)_j + C(y)_j} = \frac{N - 2d(C(x), C(y))}{N} \in [-0.02, 0.02].$$

![Figure 2: Quantum fingerprinting protocol for the equality problem](image)

The quantum protocol is very simple (see Figure 2): Alice and Bob send quantum fingerprints of $x$ and $y$ to the Referee, respectively. The referee now has to determine whether $x = y$ (which corresponds to $\langle \phi_x | \phi_y \rangle = 1$) or $x \neq y$ (which corresponds to $\langle \phi_x | \phi_y \rangle \in [-0.02, 0.02]$). The following test (Figure 3), sometimes called the SWAP-test, accomplishes this with small error probability.

![Figure 3: Quantum circuit to test if $|\phi_x\rangle = |\phi_y\rangle$ or $|\langle \phi_x | \phi_y \rangle|$ is small](image)

This circuit first applies a Hadamard transform to a qubit that is initially $|0\rangle$, then SWAPs the other two registers conditioned on the value of the first qubit being $|1\rangle$, then applies another Hadamard transform to the first qubit and measures it. Here SWAP is the operation that swaps the two registers: $|\phi_x\rangle|\phi_y\rangle \mapsto |\phi_y\rangle|\phi_x\rangle$. The Referee receives $|\phi_x\rangle$ from Alice and $|\phi_y\rangle$ from Bob and applies the test to these two states. An easy calculation reveals that the outcome of the measurement is 1 with probability $(1 - |\langle \phi_x | \phi_y \rangle|^2)/2$. Hence if $|\phi_x\rangle = |\phi_y\rangle$ then we observe a 1 with probability 0, but if $|\langle \phi_x | \phi_y \rangle|$ is close to 0 then we observe a 1 with probability close to 1/2. Repeating this procedure with several individual fingerprints can make the error probability arbitrarily close to 0.
References


