# Quantum Computing (Fall 2013) Instructor: Shengyu Zhang.

# Lecture 7 Quantum communication complexity: upper bounds

## Classical communication complexity: a brief introduction

### 1.1 models and definitions

In *communication complexity*, we consider the problem of computing a function with input variables distributed to two (or more) parties. Since each party only holds part of the input, they need to communicate in order to compute the function value. The communication complexity measures the minimum communication needed.

The most common communication models include

* *two-way*: two parties, Alice and Bob, holding inputs and , respectively, talk to each other back and forth.
* *one-way*: two parties, Alice and Bob, holding inputs and , respectively, but only Alice sends a message to Bob.
* *Simultaneous Message Passing (SMP)*: Three parties: Alice, Bob and Referee. Alice and Bob hold inputs and , respectively, and they each send one message to Referee, who then outputs an answer.
* *multiparty*: parties, each holding some input variables and they communicate in a certain way.

Some example functions are:

* **Equality ()**: To decide whether , where .
* **Inner** **Product ()**: To compute , where .
* **Disjointness ()**: To decide whether s.t. for .

The computation modes include

* *deterministic*: all messages are deterministically specified in the protocol. Namely, each message of any party depends only on her input and the previous messages that she saw.
* *randomized*: parties can use randomness to decide the messages they send.
* *quantum*: the parties have quantum computers and they send quantum messages.

We say that *a protocol computes the function* if on all possible input instances, at the end of the protocol, the party who receives the last message knows the function value on the input. (Sometimes for two-way communication model we assume that both parties know the function value. This doesn’t change the complexity much since for Boolean functions, one party can always send the answer to the other party by just one bit.) For randomized protocols, we sometimes allow a small error probability. The *cost* of a protocol is measured by the number of communication bits. The *communication complexity* of a function is the minimum cost of any protocol that computes the function. We ignore the issue of local computation cost by assuming that all parties are computationally all-powerful. Though all the computation factors are gone and only *communication* is focused on, it turns out to be very helpful for communication complexity to be exploited to prove impossibility results for other *computational* complexity.

### 1.2 An efficient deterministic protocol

Let’s consider the basic case of two parties. Note that if and are both at most -bit long, then at the very least Alice can send her input to Bob, who then can compute since he has both inputs. Finally he tells Alice the answer by 1 bit communication. Thus the communication complexity of any -bit function is at most .

What makes communication complexity interesting on its own is that some functions enjoy much more efficient protocols.

**Example 1. Max.** Alice and Bob hold , respectively. Interpret as a subset of ; the subset contains the positions s.t. . Similarly for . The task is to compute , the maximum number in .

If Alice sends her entire input , it takes communication bits. But she can do something better: She merely needs to send the maximal number in her set , which only takes bits. Bob then compares this value with his maximal number in . Thus the protocol only takes bits of communication. (If we want both parties to know the answer, then Bob can send the answer back to Alice which takes extra bits of communication.) This is much more efficient than sending bits.

This protocol seems trivial, but some others need more thoughts.

**Example 2**. **Median**. Alice and Bob again hold subsets , respectively. But this time they wish to compute the median of the multi-set . Recall that the median of numbers is defined as the -th smallest number.

An efficient protocol is as follows. At any point, they maintain an interval which guarantees to contain the median. Initially and . They halve the interval in each round, so it takes only rounds to finish. How do they halve the interval? Actually very simple: Alice sends to Bob the number of her elements that are at least , and the number of elements that are less than . This information is enough for Bob to know whether the median is above or not. (Verify this!) Bob uses 1 bit to indicate the answer, and then the protocol goes to the next round.

Since each round has only bits, and there are only rounds, the total communication cost is , much more efficient than the trivial protocol of cost .

### 1.3 Hardness for deterministic and saving by randomness

We've seen examples of efficient protocols. On the other hand, for some problems, there isn't any protocol with communication less than bits.

First, let’s observe that a deterministic communication protocol partitions the communication matrix into monochromatic rectangles. Here for a function , the *communication matrix* is defined by , namely the rows are indexed by and the columns are indexed by , and the -entry is . A *rectangle* is a pair of subsets where is a set of rows and is a set of columns. A rectangle is *monochromatic* if restricted on it is a submatrix containing only one value. (Namely, all entries in the submatrix are of the same value.) For any value in , a *-rectangle* is a monochromatic rectangle in which all entries of are of value . By considering how the protocol goes round-by-round, we can observe the following fact.

**Fact 1**. *A -bit deterministic communication protocol for partitions the communication matrix into monochromatic rectangles*.

**Example 3**. **Equality**.

The above fact enables us to show that there is no efficient protocol for solving the Equality function . Indeed, the communication matrix for is , the identity matrix of size . Notice that for this matrix, all 1-rectangles are of size 11. So no matter how one partitions the matrix into monochromatic rectangles, there are always 1-rectangles. Considering that there are also 0-rectangles, we have , so . We have thus proved the following theorem.

**Theorem 2**. Any deterministic protocol computing needs bits of communication.

Despite the above negative result, next we’ll see that the Equality function does enjoy a very efficient (and one-way!) protocol if we are allowed to use some randomness in the protocol. Let’s assume that Alice and Bob have shared randomness; we call such protocols *public-coin*.

**Theorem 3**. There is a one-way public-coin randomized protocol computing with only bits of communication. (On any input , the protocol outputs the correct with probability 99%.)

Let’s first design a protocol with 50% 1-sided error probability, and later boost the success probability. The protocol is as follows.

* **Alice**:
  + Compute , where is a public random string of length ,
  + Send the -bit result to Bob.
* **Bob**:
  + Compute , where is the same random string that Alice used.
  + Compare and Output “” if and “” if 

Note that the communication cost of the protocol is only bit. Let’s see what the protocol does. To compare and using little communication, Alice tries to give a very short summary of her input , and send only this summary to Bob, who computes the same summary of his input . The summary is so short that it’s only 1 bit, so it contains very very little information of the input string. So … could this possibly work? Next we’ll see that though it’s a small summary, it contains quite some information about the identity of the string. For this reason, the summary is usually called a *fingerprint*.

Let’s analyze it case by case. If , then of course , and the protocol is correct with certainty. If , we claim that with probability exactly 1/2! (Namely, if , then with half probability, the 1-bit summary can catch this distinction.) Actually, when , they differ at at least one position. Say it’s position i. then

.

Now notice that takes value 0 and 1 each with half probability, so regardless of the value of the second item, the summation is 1 always with probability half. Thus with probability half the protocol detects that .

To make the error probability smaller, one can simply repeat the above protocol k times, dropping the error probability to .

In the above protocol we assumed that Alice and Bob have shared randomness. There is a theorem (by Newman) saying that in two-way or one-way models, allowing public-coin randomness doesn’t change the complexity by an additive amount, compared to *private-coin randomness* where each party has her own and private randomness.

In the SMP model, however, things are different. The above procedure works if Alice and Bob share public randomness ---They just send and respectively to Referee, who then compares these two fingerprints. What if they do not share public randomness? Then it turns out that the communication complexity for in the SMP model is . Both upper and lower bounds are nontrivial, and we’ll skip the proofs here. (It’s in a reading project.)

## Quantum communication complexity

### Models

We define two party two-way model, and others are similar. A protocol goes as follows. Alice performs a unitary operation on her register A and the channel register C, and then sends the channel register C to Bob. Bob then performs a unitary operation on his register B and the channel register C, and sends the channel register C back to Alice. And the protocol goes on. The communication cost is the number of qubits of C times the number of times C is sent (in either direction). The -error quantum communication complexity is the minimum communication cost of any quantum protocol that computes the function with error probability at most .

We will use to denote the -error quantum communication complexity for . We usually omit the subscript if .

### Protocol for Disjointness

A classic result for is that the randomized communication complexity (bounded error, two-way communication) is . Namely,

**Theorem 4**.

*Proof*. The protocol is actually pretty simple. Just run Grover’s search to search for a position s.t. . The Grover’s search algorithm is a query algorithm, which uses an oracle  
Note that is nothing but where . So it is enough to simulate the oracle in order to invoke Grover’s search. Can we use efficient communication to simulate ? Yes. Actually Alice can do and send the working space (which has only qubits) to Bob, who can do . Then Bob does . Finally they erase by doing the first two steps again:

## SMP communication and quantum fingerprint

Recall that the communication complexity for in the private-coin SMP model is . We’ll show that if we use quantum communication, then the complexity can be reduced to .

First, let’s review the notion of error correcting code. Suppose that a sender A wishes to send an -bit string to a receiver B through a noisy channel. Suppose the message is . The channel flips each bit of independently with probability at most a small constant . To cope with the noise, A needs to add some redundancy. What’s the minimum length of the message s.t. B can recover the correct with probability 99%? The answer is . Check: what if we add and the recovery probability in? Fix such an error correcting code .

We’ll need another tool called Swap-Test, which measures how close two *unknown* states are. Suppose that we are given two states, in register A and in register B, and we want to estimate . Here is the test.

Swap-Test:

* Attach an extra qubit in register C.
* Conditioned on in register C, swap content in A and B. Here swap is the operation .
* Measure the register C in basis, and the test is considered to pass if the outcome is .

To get some intuition of the test, let’s consider a special case first. If , then the second step has no effect, and thus the measurement in the third step passes with probability 1. In general, note that after Step 2, the state becomes

So the measurement in Step 3 gives a with probability

Now we give a quantum protocol in private-coin SMP model for . Actually, Alice and Bob do not even use any randomness.

* Alice: Send .
* Bob: Send .
* Referee: Apply Swap-Test on . Output “” if the test passed and output “” if the test didn’t pass.

The analysis of the protocol is easy. If , then and the Swap-Test passes with probability 1. So Referee outputs . If , then not only but they are far apart:

as guaranteed by the error correcting code. Thus the Swap-Test fails with probability. (As always, one can repeat the protocol to reduce the error probability exponentially.)

The communication cost is .

We have thus proved the following theorem.

**Theorem 5**. The quantum communication complexity of in the private-coin SMP model is .

## Note

Communication complexity is a very interesting subject on its own and has numerous applications to other algorithm and complexity theories. For a general introduction, see [KN97].

## Reference

[KN97] Eyal Kushilevitz and Noam Nisan, **Communication Complexity**, *Cambridge University Press*, 1997.

## Exercise