IAS/Park City Mathematics Series
Volume 3, 1993

Circuit and Communication Complexity

Ran Raz

LECTURE 4
Communication complexity

A communication problem arises when two or more parties need to carry out a task which could not be carried out by any of them alone, because of lack of information. Each of the parties has is own input string. In order to achieve some common goal, defined as a function of all input strings, the parties need to communicate. Communication complexity is concerned with analyzing the amount of information that must be communicated between the different parties in order to perform the intended task. Many times, the formulation of a problem as a communication problem serves merely as a convenient abstraction.

4.1. Basic Model and Some Examples

The basic model of communication complexity was first introduced by Yao, and was found out to be relevant to many other complexity issues. The model assumes the following:

(1) There are only two parties, referred to as, Player 1 and Player 2. Each one of the two players is given an input. At the beginning, non of the players has any information about the input of the other player.
(2) Each player has unlimited computational power. We are only concerned with the number of bits that have to be communicated between the two players.
(3) The task is to compute a predefined function of the two inputs.

[1]Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel.
E-mail address: `ranraz@wisdom.weizmann.ac.il`.

November 15, 2000

Based on lecture notes taken by Amiel Ferman, Noam Sadot and Yoad Lustig.

More formally, we have 3 finite sets X, Y, Z, and a function $f : X \times Y \to Z$ (known to both players). In most cases, we will have $X = Y = \{0,1\}^n$ and $Z = \{0,1\}$. Player 1 gets as an input $x \in X$. Player 2 gets as an input $y \in Y$. At the beginning, none of the players has any information about the input of the other player. Their (common) goal is to compute the value of $f(x, y)$.

In each step, one of the players sends one bit of information to the other player. This bit depends on the input of the player who sends it, as well as on all previous bits communicated between the two players. A communication protocol specifies in each step which player sends the next bit, as well as the value of this bit as a function of the input of the player who sends it and the sequence of previous bits communicated between the two players. In most cases, we will assume for simplicity that the players send the bits alternately. At the end of the protocol, both players have to know the value of $f(x, y)$. In most cases, $f(x, y)$ consists of only one bit, and we can assume that this bit is communicated by one of the players as the last bit in the protocol. The issue here is the communication between the two players. That is, the number of bits that the players have to exchange between them in order to compute $f(x, y)$.

Let us see a few examples for interesting communication complexity problems:

(1) Equality (denoted EQ):
The inputs are: $x, y \in \{0,1\}^n$.
The function f is defined by:
$f(x, y) = 1$ if $x = y$
$f(x, y) = 0$ if $x \neq y$

(2) Set Disjointness (denoted $Disj$):
The inputs are two subsets: $x, y \subseteq \{1, \ldots, n\}$.
The function f is defined by:
$f(x, y) = 1$ if $x \cap y \neq \emptyset$
$f(x, y) = 0$ if $x \cap y = \emptyset$

(3) Inner Product (denoted IP):
The inputs are: $x, y \in \{0,1\}^n$.
The function f is defined by:
$f(x, y) = \sum_1^n x_i \cdot y_i (\mathrm{mod} 2)$

4.2. Deterministic versus Probabilistic Complexity

A deterministic protocol P specifies in each step the value of the next (communication) bit, as a function of the input of the player who sends it and the sequence of previous bits communicated by the two players. At the end, both players have to know a certain value, denoted by $P(x, y)$. We refer to $P(x, y)$ as the output of the protocol on input (x, y). We say that the protocol P is a protocol for f (or, computes f) if

$$\forall x \in X, y \in Y \quad : \quad P(x, y) = f(x, y).$$

The communication complexity of the protocol is defined to be the maximal number of bits communicated between the two players (where the maximum is taken over all possible inputs). The deterministic communication complexity of a function f, denoted by $CC(f)$, is defined to be the minimal communication complexity of a deterministic protocols for f.

In the probabilistic case, we allow the two players to use random bits. That is, the bit communicated by a player may depend on a random string. More formally, we can present a probabilistic protocol as a distribution over deterministic protocols. That is, each player has an access to a random string, and the protocol that is carried out depends on that random string. However, once the strings are chosen, the protocol is completely deterministic. The output of the protocol, $P(x, y)$, is now a random variable. We require that

$$\forall x \in X, y \in Y \qquad Pr[P(x, y) = f(x, y)] \geq 1 - \epsilon,$$

where the probability is over the random strings, and ϵ is some small constant (referred to as the probability of error). That is, we allow a small probability of error, denoted by ϵ. The exact value of ϵ is not too important, as it can be efficiently reduced by repetitions of the same protocol.

We distinguish between two models of probabilistic communication complexity. The first model is the public random string model, where all random bits are seen by both players. That is, the two players share a common (public) random string. The second model is the private random string model, where each player has his own private string (invisible by the other player). How are these two models related ? First, it is obvious that any private random string protocol can be simulated as a public random string protocol: The private strings of the two players are just concatenated and serve as a public string. It turns out that a weaker reduction exists in the other direction as well: Any public random string protocol can be simulated as a private random string protocol, with a small increase in the probability of error and an addition of $O(\log n)$ bits of communication. The idea of the proof is to first show that any public random string protocol can be converted into an equivalent public random string protocol that uses only $O(\log n)$ random bits. The conversion to a private random string protocol is now easy: One of the players samples a string of that length (using is private random string) and sends it to the other player. The two players can then run the public random string protocol. This increases the communication complexity by only $O(\log n)$ bits. The two models are hence almost equivalent. Here, we will work mainly with the public random string model (which is the more general one). That is, we assume that the two players share a common random string.

Note that in a probabilistic protocol, the number of bits communicated by the two players may also be a random variable (as it may depend on the random string). We define the communication complexity of a probabilistic protocol as the maximal number of bits communicated by the two players, where the maximum is taken over all inputs and over all assignments to the random string. The probabilistic communication complexity (with error ϵ) of a function f, denoted by $CC_\epsilon(f)$, is the minimal communication complexity of a protocol P that computes f (with probability of error ϵ).

4.3. Communication Complexity of Equality

As mentioned above, the Equality problem is the following: Player 1 gets an n-bits string x and Player 2 gets an n-bits string y. The goal is to determine whether or not $x = y$. How many communication bits are needed ? The deterministic communication complexity of the Equality problem is

$$CC(EQ) = n + 1.$$

The upper bound is trivial. The lower bound will be proved later on in this lecture. In the probabilistic model, much less communication bits are needed. We will see that in the public random string model

$$CC_\epsilon(EQ) = O(1),$$

and in the private random string model

$$CC_\epsilon(EQ) = O(\log n).$$

Let us first present the public random string protocol of communication complexity $O(1)$ that solves the Equality problem with a constant error probability: Let $z \in \{0,1\}^n$ be part of the common random string. Player 1 computes the inner product of x and z (modulo 2), and Player 2 computes the inner product of y and z (modulo 2). The players exchange the results and output 1 iff the two results are equal. If $x = y$ then the two results are always equal. At the other hand, if $x \neq y$ then with probability of one half the two results differ. Thus, the error probability of a single iteration of the above protocol is one half. To reduce the probability of error, we can just iterate the protocol. In each iteration, we will use a different (independently chosen) random value z. The players output 1 if the two results are equal in every iteration of the protocol. If in at least one iteration the two results differ then the players output 0. Note that if $x = y$ then the protocol always outputs 1. At the other hand, if $x \neq y$ then the protocol outputs 1 with probability 2^{-C}, where C is the number of iterations. Hence, the error probability of the protocol is 2^{-C}. The communication complexity of the protocol is $O(C)$. In particular, if C is a constant, we get both, a constant error probability and a constant communication complexity. If we choose C to be $\log n$ then the error probability and the communication complexity will be, respectively, $\frac{1}{n}$ and $O(\log n)$.

As mentioned above, there is a general method to convert any public random string protocol of communication complexity $O(1)$ to a private random string protocol of communication complexity $O(\log n)$. Hence, there exist private random string protocols for Equality with communication complexity $O(\log n)$. Let us explicitly present such a protocol:

Player 1 chooses a random prime p of length smaller than $2\log n$ (using his private random string) and sends p to Player 2. Player 1 computes $x \bmod p$, and Player 2 computes $y \bmod p$. The players exchange the results and output 1 iff the two results are equal. As before, if $x = y$ the output is always 1. If $x \neq y$ then since the number of primes that divide $|x - y|$ is small, the probability that the output is 1 is small. It is not hard to prove that this probability is lower than $1/n$. Thus, the communication complexity of the protocol is $O(\log n)$, while the probability of error is at most $1/n$.

4.4. Input Matrix and Monochromatic Covers

It is sometimes convenient to describe the function $f : X \times Y \to Z$ as a matrix A. Each row of the matrix corresponds to $x \in X$ and each column corresponds to $y \in Y$. The entries of the matrix are defined by $A_{x,y} = f(x, y)$. The matrix A is referred to as the input matrix. Since the input matrix is just another way to describe the function f, we sometimes talk about the communication complexity of the matrix A, rather than the communication complexity of the underlying function f. We denote the deterministic communication complexity of A by $CC(A)$

and the probabilistic communication complexity (with error ϵ) of A by $CC_\epsilon(A)$. Examples for interesting input matrices are the identity matrix which corresponds to the Equality function, and the Hadamard matrix (i.e., the Fourier transform over $(GF_2)^n$) which corresponds to the Inner Product function.

A subset $R \subseteq X \times Y$ is called a rectangle if $R = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$. Equivalently, we can define a rectangle as a subset $R \subseteq X \times Y$, such that, if $(x_1, y_1) \in R$ and $(x_2, y_2) \in R$ then $(x_1, y_2) \in R$. We think of a rectangle R also as a submatrix of A. A rectangle $R \subseteq X \times Y$ is called monochromatic if for some z and for every $(x, y) \in R$, we have $A_{x,y} = z$ (i.e., if all the entries of the submatrix R are identical).

Let P be a deterministic communication protocol. Assume that Player 1 sends the first bit. Since this bit depends only on the input $x \in X$, there is a partition $X_0 \cup X_1 = X$, such that, for every $x \in X_0$ Player 1 sends the bit 0 and for every $x \in X_1$ Player 1 sends the bit 1. More generally, in step t, the protocol P defines a partition of $X \times Y$ into (up to 2^t) disjoint subsets, such that, each subset includes all input pairs (x, y) that cause P to communicate the same first t bits. At the end, we will get a partition of $X \times Y$ into disjoint subsets, such that, each subset includes all input pairs (x, y) that cause P to communicate exactly the same bits (i.e., all bits communicated during the protocol are the same for any two input pairs in the same subset). What can we say about each subset in the partition ? It is not hard to prove that each of these subsets is a rectangle. Let R be one of these rectangles. For all $(x, y) \in R$ the same bits are communicated by the protocol P. Since the output $P(x, y)$ depends only on these bits, we conclude that for any $(x, y) \in R$ the output $P(x, y)$ is the same. In particular, if the protocol P is a protocol for the input matrix A then $P(x, y) = A_{x,y}$ and we conclude that the rectangle R is monochromatic. To summarize, we have shown the following lemma.

Lemma 4.4.1. A deterministic protocol P of communication complexity k partitions the input matrix A into at most 2^k monochromatic rectangles.

A partition of a matrix into monochromatic rectangles is called a monochromatic cover. The logarithm of the minimal number of rectangles in a monochromatic cover of a matrix A is hence a lower bound for the deterministic communication complexity of A.

Let us show how to use Lemma 4.4.1 to prove a lower bound for the communication complexity of the Equality problem. Recall that the input matrix of the Equality problem is the identity matrix. For the identity matrix, a monochromatic rectangle of ones is of size 1×1. Hence, any partition of the identity matrix into monochromatic rectangles must contain at least 2^n rectangles of ones, and at least one additional rectangle (of zeros). That is, the total number of rectangles is larger than 2^n. Hence, there is no deterministic protocol of communication complexity n for the Equality problem.

Corollary 1. $CC(EQ) = n + 1$.

4.5. The Rank Lower Bound

We already saw one method for proving lower bounds for the deterministic communication complexity of a matrix. We will now state and prove a useful lemma that bounds the deterministic communication complexity of a matrix as a function of its rank.

Lemma 4.5.1. Let F be a field and let A be a matrix with entries in F, then $CC(A) \geq \log_2(r_A)$, where r_A is the rank of A over F.

Proof. The proof is by induction on $CC(A)$.

Induction Base: If $CC(A) = 0$ then both players are able to compute the value $A_{x,y}$ without any communication. Hence, $A_{x,y}$ is the same for all (x, y). Hence, r_A is either 0 or 1.

Induction Step: Suppose that the claim is true for $CC(A) \leq k - 1$. We will prove the claim for $CC(A) = k$. Consider the first bit sent by one of the players. This bit partitions A into two submatrices A_0 and A_1, such that the rest of the protocol can be viewed as a protocol for each one of these matrices. Hence,

$$CC(A_0), CC(A_1) \leq k - 1.$$

Therefore,

$$CC(A) \geq 1 + Max\{CC(A_0), CC(A_1)\} \geq 1 + Max\{\log_2(r_{A_0}), \log_2(r_{A_1})\},$$

where the second inequality is by the inductive hypothesis. Since $r_A \leq r_{A_0} + r_{A_1}$, we have that $r_A \leq 2 \cdot Max\{r_{A_0}, r_{A_1}\}$. That is,

$$Max\{\log_2(r_{A_0}), \log_2(r_{A_1})\} \geq \log_2(r_A) - 1.$$

Hence,

$$CC(A) \geq 1 + \log_2(r_A) - 1 = \log_2(r_A).$$

■

Applying Lemma 4.5.1 to the identity matrix, we obtain the lower bound $CC(EQ) \geq n$ (recall however that we already proved that $CC(EQ) = n + 1$). A linear lower bound for the deterministic communication complexity of the Inner Product function and the Set Disjointness function can also be proved by applying this lemma. In the next lecture, we will prove a linear lower bound for the probabilistic communication complexity of the Inner Product function.

Lemma 4.5.1 states that the communication complexity of a matrix is bounded from below by the logarithm of the rank of the matrix. The *log rank conjecture* states that the communication complexity of a matrix is bounded from above by a certain power of that logarithm. This would be interesting since the rank of a matrix is a relatively simple object. In particular, the rank can be computed in polynomial time (in the size of the matrix). If the log rank conjecture is true then the deterministic communication complexity of a matrix can be approximated (up to a constant power) in polynomial time.

Lemma 4.5.1 is stated for an arbitrary field F. It is not hard to prove, however, that the rank of a $\{0, 1\}$ matrix is maximal over fields of characteristic zero.