BOUNDS ON TRADEOFFS BETWEEN RANDOMNESS AND COMMUNICATION COMPLEXITY

Ran Canetti and Oded Goldreich

Abstract. The power of randomness in improving the efficiency (or even possibility) of computations has been demonstrated in numerous contexts. A fundamental question is *how much* randomness is required for these improvements, or how does the improvement grow as a function of the amount of randomness allowed. This quantitative question, restricted to the context of communication complexity, is the focus of our paper.

We prove general lower bounds on the amount of randomness used in randomized protocols for computing a function f, the input of which is split between two parties. The bounds depend on the number of bits communicated and the deterministic communication complexity of f. Four models for communication complexity are considered: the random input of the parties may be public or private, and the communication may be one-way or two-way. (Unbounded advantage is allowed.)

The bounds are shown to be tight; i.e., we demonstrate functions and protocols for these functions which meet the above bounds up to a constant factor. We do this for all the models, for all values of the deterministic communication complexity, and for all possible quantities of bits communicated.
Key words. Communication Complexity, Randomized Computation, Complexity-Randomness tradeoff, Lower bounds, protocols.
Subject classifications. 68Q22

1. Introduction

The power of randomness in computation is a major issue in all aspects of computer-science, and is yet to be fully understood. There are many cases in which there are tremendous gaps between the complexities, or even possibilities of deterministic and randomized computations (e.g., routing: Valiant 1982, Aleliunas 1982, Borodin and Hopcroft 1985; agreement problems: Fischer *et al.* 1985, Rabin 1983, Chor and Dwork 1989; communication complexity: Yao 1979, Melhorn and Schmidt 1982, Paturi and Simon 1986).

A method of 'smoothing' these gaps is to measure the 'degree of randomization' of an algorithm, thus substituting the qualitative question "Is the algorithm deterministic or randomized?" by the quantitative question "How much randomness does the algorithm use?". We initiate a quantitative study of randomness in the context of communication complexity.

The communication complexity of a boolean function f, as was initially defined and studied by Yao (1979), measures the number of bits that have to be transferred between two computationally unlimited parties in order to compute $f(x, y)$ (with probability greater than $\frac{1}{2}$), when one party has x and the other has y.

Although for most functions randomization does not help in reducing the communication complexity (Alon *et al.* 1985, Chor and Goldreich 1988), for some functions there exists a tremendous gap in the communication complexity in the two models of computation. For instance, Yao (1979) showed that n bits of communication are needed in order to deterministically compute the identity function ($ID(x, y) = 1$ iff $x = y$), and Paturi and Simon (1986) gave a randomized protocol that uses only *two* bits of communication and computes ID (with probability greater than $\frac{1}{2}$). In view of these gaps, this model seems to be a convenient test-field for a quantitative investigation of the power of randomness in computation.

We show a tradeoff between the maximum number of coin tosses[1] used, and the number of bits exchanged by parties executing a protocol for computing a given function. This tradeoff may be interpreted in two alternative ways. One interpretation is a lower bound on the number of bits exchanged by parties using a given number of coin tosses. The other interpretation (which is used in this paper) is a lower bound on the number of coins tossed by the parties, as a function of the number of bits communicated. It can be seen that the lower bound on the number of coins tossed increases gradually from zero up to a maximum value (which is at most the length of the input to the parties), as the number of bits communicated decreases from the deterministic communication complexity of the function to the randomized one. The bounds hold for any protocol that computes a function with probability greater than (but not necessarily bounded away from) one half, and do not depend heavily on the advantage over $\frac{1}{2}$ achieved. We consider the following variants of the model. The random input may be accessible to both parties, or split into two parts, each available to one party only. The communication may be *one-way,* or *two-way.* (In one-way communication, one party sends a message to the second party, and the second party decides on an output. In two-way communication, the parties exchange messages until one of them outputs the function value.)

The bounds are tight in the following strong sense. We construct a sequence of functions that cover the range of all possible deterministic communication complexities. For each function, computation model and a given number of bits to be exchanged, we show a corresponding protocol that meets the lower bound on the

[1] Throughout this paper, the term 'coin tosses' means unbiased, independent random bits.

maximum number of coins tossed, up to a multiplicative constant.

In the above discussion, we considered only protocols using coin tosses as their random input, and measured the maximum number of coin tosses used. However, protocols may use a random source other than the uniform one. A general measure for the amount of randomness used in algorithms is the *entropy* of the random source used. We thus extend our lower bounds to bound also the entropy of any random source used by the parties (executing a protocol to compute a function).

It turns out that the bounds on the entropy of the random source of the parties depend heavily on the advantage over $\frac{1}{2}$ achieved by the protocol. (This stands in contrast to the situation with respect to the more 'conservative' measure of randomness discussed above, i.e. the maximum number of coins tossed.) On one hand, we show a lower bound on the entropy of the random source as a function of the advantage of the protocol and the corresponding lower bound on the maximum number of coins tossed. In particular, for constant advantage the lower bound on entropy equals the conservative bound, up to a multiplicative factor. On the other hand, we show that the entropy of the random source used in *any* protocol can be reduced up to a constant, without increasing the number of bits communicated, but with the advantage of the protocol becoming possibly smaller than any constant.

Newman (1991) showed that the number of coin-tosses used in *any* coin tossing protocol (in the public coins model) can be reduced, without increasing the number of bits communicated, up to a value depending on the advantage over $\frac{1}{2}$ achieved by the protocol. We extend this result to the private coins model. This extended result, combined with the lower bounds on the number of coins tossed, gives an upper bound on the advantage achieved by the protocol, depending on the number of bits communicated.

Relation to other work. A tradeoff between randomness and communication complexity was independently investigated by Fleischer *et al.* (1990). They consider the expected communication complexity of 'Las Vegas' protocols (i.e., no error allowed, as opposed to our 'Monte Carlo' model), in the two-way communication, private random input model. In their setting, they show a tight tradeoff similar to the one presented in this paper. Note that lower bounds for protocols in the 'Monte Carlo' model (i.e., the model considered in this paper) apply to the 'Las Vegas' model as well.

A quantitative study of randomness was carried out in the contexts of oblivious routing (Krizank *et al.* 1988, Peleg and Upfal 1990, Karloff and Raghavan 1988) and cashing algorithms (Raghavan and Snir 1989).

Reducing randomness in various computational models has been the focus of much attention recently. Typical settings include pseudorandomness (Blum and Micali 1984, Yao 1982, Nisan 1990), efficient error reduction (Sipser 1986, Impagliazzo and Zuckerman 1989, Cohen and Wigderson 1989, Bellare *et al.* 1990), using weak sources of randomness (Vazirani and Vazirani 1985, Chor and Goldreich 1988, Zuckerman 1991), and computational number theory (Bach and Shoup 1990). However, unlike our work and works quoted in previous paragraphs, these works do not claim to achieve the best saving possible (i.e., match lower bounds).

A preliminary version of this work has appeared in the proceedings of the 31st FOCS (1990).

Organization. In Section 2 we define the models and the parameters to be discussed. Section 3 contains the lower bounds on the *maximum* number of coin tosses used by the parties, in the four models. In Section 4 we demonstrate the tightness of these bounds: we present functions, and protocols for these functions, matching the bounds. In Section 5 we generalize the results of Section 3 to show bounds on other measures of randomness.

2. Preliminaries

Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and let P_1 and P_2 be two parties having inputs $x, y \in \{0,1\}^n$ respectively, and communicating according to a randomized protocol π in order to compute $f(x,y)$. Denote the output of the protocol on input x, y as $\pi(x,y)$. (I.e., $\pi(x,y)$ is a random variable determined by the random choices of the parties.)

Consider the following two parameters.

- The communication may be *one-way* or *two-way.* In the one-way communication model, party P_1 sends a message to P_2, and party P_2 decides on the output. In the two-way communication model, the parties take turns sending messages, until a party sends a special symbol *'halt'* and outputs the protocol answer. (We assume, as usual, that the messages transferred between the parties are prefix-free.)
- The random input of the parties may be *private* or *public.* A private random input is known to one party only, while a public random input is known to both parties without need of communication.

Clearly, a one-way protocol is a special case of a two-way protocol. Private random inputs can be emulated by a public one, by splitting the public input to two parts and letting each party read only one part.

In the sequel, we consider protocols that use coin tosses as their random input, unless otherwise stated.

In the public random input models, let $r(\pi)$ denote the *maximum* number of coin-tosses used during the execution of protocol π on the worst input pair. In the private random input models, let $r_1(\pi)$ and $r_2(\pi)$ denote the maximum number of coin-tosses used by P_1 and P_2 respectively, and let $r(\pi) = r_1(\pi) + r_2(\pi)$.

Let $\epsilon(\pi, f)$ denote the advantage achieved by protocol π, with respect to function f:

$$\epsilon(\pi, f) = \min_{x,y \in \{0,1\}^n} \mathrm{Prob}(\pi(x,y) = f(x,y)) - \frac{1}{2}$$

We say that a protocol π *computes* function f, if $\epsilon(\pi, f) > 0$. (In the sequel, whenever it will be clear which function is being computed, we will write $\epsilon(\pi)$ instead of $\epsilon(\pi, f)$.)

Let $m(\pi)$ denote the maximum, over the inputs and coin tosses, of the number of bits communicated by the parties during the execution of protocol π.

Let $R_m(f)$ ($R_{i,m}(f)$, for $i \in \{1,2\}$) denote the minimum of $r(\pi)$ ($r_i(\pi)$) over all protocols π that compute f (with any advantage), such that $m(\pi) \leq m$.

As our bounds depend on the *deterministic* communication complexity of a function f, we recall the standard definition. Let the two-way (one-way) deterministic communication complexity of a function f, denoted $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f)$ ($\mathrm{C}_{\mathrm{D}}^{1\Rightarrow 2}$), be defined as the minimum of $m(\pi)$ over all *deterministic* two-way (one-way) protocols π that compute f. (Note that $\mathrm{C}_{\mathrm{D}}^{1\Rightarrow 2}(f)$ is the logarithm of the number of distinct rows in the matrix representation of f.[2])

We say that a function is *non-degenerate* if all the rows (columns) in the table of f are distinct. (Let $\hat{x}$ denote the integer whose binary representation is x. The table of f is the $2^n \times 2^n$ table whose $(\hat{x}, \hat{y})^{\mathrm{th}}$ entry is $f(x,y)$.) For simplicity of presentation we consider only *non-degenerate* functions. However, our results can be easily extended to all f.

3. Lower bounds

In Sections 3.1 through 3.3, we show lower bounds on the maximum number of coins tossed in probabilistic protocols for computing functions, in the different models of computation. The proofs of the bounds use either combinatorial arguments (i.e., counting the number of vectors of a particular type), or a simulation argument (i.e., deterministically going over the possible random inputs of a randomized protocol). In Section 3.4 we give an overview of the different bounds, and discuss the relations between them.

3.1. The one-way, private coins model.

THEOREM 3.1. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a non-degenerate function, and let π be a* **one-way, private coins** *protocol that computes f with advantage ϵ. Let*

[2] All the logarithms in this paper are of base 2.

$r = r_1(\pi)$ and $m = m(\pi)$. Then,

$$\frac{\binom{2^r+2^m-1}{2^r}}{\binom{\epsilon 2^r+2^m-1}{\epsilon 2^r}} \geq 2^n. \tag{3.1}$$

(Remark: For Theorem 3.1, function f need not be non-degenerate. We only need that all the *rows* in the table of f are distinct.)

Approximating (1), we have

$$\begin{aligned} 2^n &\leq \frac{\binom{2^r+2^m-1}{2^r}}{\binom{\epsilon 2^r+2^m-1}{\epsilon 2^r}} = \frac{((2^r+2^m-1)\cdot\ldots\cdot 2^m)\cdot(\epsilon 2^r\cdot\ldots\cdot 1)}{(2^r\cdot\ldots\cdot 1)\cdot((\epsilon 2^r+2^m-1)\cdot\ldots\cdot 2^m)} \\ &= \frac{(2^r+2^m-1)\cdot\ldots\cdot(\epsilon 2^r+2^m)}{2^r\cdot\ldots\cdot(\epsilon 2^r+1)} \leq \left(\frac{\epsilon 2^r+2^m}{\epsilon 2^r+1}\right)^{(1-\epsilon)2^r} \leq 2^{m(1-\epsilon)2^r}. \end{aligned} \tag{3.2}$$

Using the same method, we also have

$$2^n \leq \frac{\binom{2^r+2^m-1}{2^r}}{\binom{\epsilon 2^r+2^m-1}{\epsilon 2^r}} \leq \binom{2^r+2^m-1}{2^r} \leq (2^r+1)^{2^m}. \tag{3.3}$$

Thus, the following two inequalities hold simultaneously (inequality (a) (respectively, (b)) follows from (2) (respectively, (3))):

$$\begin{array}{lcl} \text{(a)} & r & \geq \frac{n}{2^m} - 1 \\ \text{(b)} & r & \geq \log(\frac{n}{(1-\epsilon)m}). \end{array}$$

The first inequality is stronger for $0 < m < \log n - \log\log n$, whereas the second is stronger for $\log n < m < n$. Note that the advantage ϵ has little effect on the bounds.

Proof. We prove this using a combinatorial argument. Consider an enumeration of all possible (up to 2^m) messages. Let p_i^x denote the probability that party P_1 sends the i^{th} message on input x, and q_i^y the probability that P_2, on input y and having received the i^{th} message from P_1, outputs 1. Then, $\text{Prob}(\pi(x,y)=1) = \sum_{i=1}^{2^m} p_i^x \cdot q_i^y$. (This method for analyzing protocols was introduced by Yao 1979.)

Consider two different inputs $x, x' \in \{0,1\}^n$ to party P_1. Since f is non-degenerate, there exists an input y to party P_2 such that $f(x,y) \neq f(x',y)$. Since π computes f with advantage ϵ, we have

$$\mid \text{Prob}(\pi(x,y)=1) - \text{Prob}(\pi(x',y)=1) \mid \geq 2\epsilon.$$

Let $\bar{p}^x \stackrel{\text{def}}{=} (p_1^x, \cdots, p_{2^m}^x)$ and $\bar{q}^y \stackrel{\text{def}}{=} (q_1^y, \cdots, q_{2^m}^y)$. Since $0 \leq p_i^x, q_i^y \leq 1$, we have

$$||\bar{p}^x - \bar{p}^{x'}||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{2^m} \mid p_i^x - p_i^{x'} \mid \geq \sum_{i=1}^{2^m} \mid q_i^y p_i^x - q_i^y p_i^{x'} \mid \geq \mid \sum_{i=1}^{2^m} p_i^x \cdot q_i^y - \sum_{i=1}^{2^m} p_i^{x'} \cdot q_i^y \mid \geq 2\epsilon.$$

(Using the fact that $\sum_{i=1}^{2^m} p_i^x = \sum_{i=1}^{2^m} p_i^{x'} = 1$, it can be shown that $||\bar{p}^x - \bar{p}^{x'}||_1 \geq 4\epsilon$. This was pointed out by the anonymous referee. The derivation is presented in Proposition 3.1 below.)

Thus, the protocol π defines a set $S = \{\bar{p}^x | x \in \{0,1\}^n\}$ of 2^n distinct probability vectors, so that for every x, x', we have $||\bar{p}^x - \bar{p}^{x'}||_1 \geq 4\epsilon$.

On the other hand, we bound from above the maximum possible size of set S. As party P_1 tosses only r coins, each probability p_i^x may be assigned only $2^r + 1$ different values: $\{i \cdot 2^{-r} | 0 \leq i \leq 2^r\}$. Moreover, for every input x, we have $\sum_{i=1}^{2^m} p_i^x = 1$. Thus, the number of distinct such vectors is the number of ways to partition 2^r 'probability elements' among 2^m messages, namely $\binom{2^r+2^m-1}{2^r}$. This alone implies a simplified form of the bound (1), namely $\binom{2^r+2^m-1}{2^r} \geq 2^n$.

However, the bound can be slightly improved by considering the distance between each pair of vectors $\bar{p}, \bar{p}' \in S$. Let S' be the set of probability vectors that were produced by partitioning 2^r probability elements among 2^m messages (hence, $|S'| = \binom{2^r+2^m-1}{2^r}$ and $S \subseteq S'$). For every vector $\bar{p} \in S$, the number of vectors $\bar{p}' \in S'$ such that $||\bar{p} - \bar{p}'||_1 < 2\epsilon$ is at least the number of ways to partition $\epsilon 2^r$ elements among 2^m cells, namely $\binom{\epsilon 2^r+2^m-1}{\epsilon 2^r}$. Thus, for every probability vector $\bar{p} \in S$ there exist a set $N_{\bar{p}} \subseteq S'$ of $\binom{\epsilon 2^r+2^m-1}{\epsilon 2^r}$ distinct vectors that may not be in S. Furthermore, for every $\bar{p}, \bar{p}' \in S$, we have that $N_{\bar{p}}$ and $N_{\bar{p}'}$ are disjoint. The theorem follows. □

Note that Theorem 1 does not give a lower bound on the number of coins tossed by party P_2. This bound is implied by Theorem 2, as one-way communication is a special case of two-way communication. (In Section 4.1.1 we describe one-way, private coins protocols that achieve the bound of Theorem 2 on the number of coins tossed by P_2. Thus, the bound of Theorem 2 is tight even for the special case of one-way protocols.)

Proposition 3.2. *Let $\bar{p}^x$ and $\bar{p}^{x'}$ be defined as in the proof of Theorem 3.1. Then*

$$||\bar{p}^x - \bar{p}^{x'}||_1 \geq 4\epsilon.$$

Proof. Let $\alpha \stackrel{\text{def}}{=} \{i | p_i^x \geq p_i^{x'}\}$, and let $\beta \stackrel{\text{def}}{=} \{1, \ldots, 2^m\} - \alpha$. Since $\sum_{i=1}^{2^m} p_i^x = \sum_{i=1}^{2^m} p_i^{x'} = 1$, we have

$$\sum_{i \in \alpha} p_i^x - p_i^{x'} = -\sum_{i \in \beta} p_i^x - p_i^{x'} = \frac{1}{2} \sum_{i=1}^{2^m} |p_i^x - p_i^{x'}|.$$

Assume, without loss of generality, that

$$|\sum_{i \in \alpha} q_i^y (p_i^x - p_i^{x'})| \geq |\sum_{i \in \beta} q_i^y (p_i^x - p_i^{x'})| \qquad (3.4)$$

(clearly, the left hand sum is positive and the right hand sum is negative).

We now have:

$$||\bar{p}^x - \bar{p}^{x'}||_1 \stackrel{\rm def}{=} \sum_{i=1}^{2^m} \mid p_i^x - p_i^{x'} \mid = 2\sum_{i\in\alpha} p_i^x - p_i^{x'} \geq 2\sum_{i\in\alpha} q_i^y (p_i^x - p_i^{x'}) \stackrel{(*)}{\geq}$$

$$2|\sum_{i\in\alpha} q_i^y (p_i^x - p_i^{x'}) + \sum_{i\in\beta} q_i^y (p_i^x - p_i^{x'})| = 2|\sum_{i=1}^{2^m} q_i^y (p_i^x - p_i^{x'})| \geq 4\epsilon$$

(inequality $(*)$ is derived using (4)). $\square$

3.2. The two-way, private coins model.

THEOREM 3.3. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a non-degenerate function, and let π be a* **two-way, private coins** *protocol that computes f with advantage ϵ. Let $r_i = r_i(\pi)$, let $r = r_1 + r_2$, and let $m = m(\pi)$. Then the following inequalities hold simultaneously:*

$$\text{(a)} \quad r_i \ \geq \ \frac{n}{2^m} - 1 \quad \text{for } i \in \{1,2\}$$

$$\text{(b)} \quad r \ \geq \ \log\left(\frac{\frac{\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f)}{m} - 1}{1-2\epsilon}\right).$$

PROOF. Part (a): Proof is by a combinatorial argument. Consider a conversation $con = u_1 \circ v_1 \circ \ldots u_k \circ v_k$ between the parties, where u_j is the message sent by P_1 in round j of the conversation, v_j is the subsequent message of P_2, and k is the number of rounds. (The parsing is unique since the messages are prefix-free.) Let $p_j^x(con)$ denote the probability that P_1 sends u_j on input x, given that the conversation so far is $u_1 \circ v_1 \circ \ldots u_{j-1} \circ v_{j-1}$ (the corresponding prefix of con), and let $p^x(con) \stackrel{\rm def}{=} \prod_{j=1}^k p_j^x(con)$. (Intuitively, $p^x(con)$ is the probability that the messages of P_1 are consistent with con on input x.) Let $q^y(con)$ be defined similarly, with respect to P_2. Let $C_{x,y}$ be the set of the conversations with output 1, on input x,y. Using a cut-and-paste argument, we get

$$\mathrm{Prob}(\pi(x,y)=1) = \sum_{con\in C_{x,y}} \mathrm{Prob}(\text{the conversation on inputs } x,y \text{ is } con)$$

$$= \sum_{con\in C_{x,y}} p_1^x(con)\cdot q_1^y(con)\cdot\ldots\cdot p_k^x(con)\cdot q_k^y(con) = \sum_{con\in C_{x,y}} p^x(con)\cdot q^y(con).$$

Consider an enumeration of all the (up to 2^m) conversations. Let $\bar{p}^x \stackrel{\rm def}{=} (p^x(con_1),\ldots,p^x(con_{2^m}))$, and $\bar{q}^y \stackrel{\rm def}{=} (q^y(con_1),\ldots,q^y(con_{2^m}))$. Following the lines of the first part of proof of Theorem 1, we conclude that the sets $S_1 = \{\bar{p}^x | x \in \{0,1\}^n\}$ and $S_2 = \{\bar{q}^y | y \in \{0,1\}^n\}$ are each of size 2^n. (I.e., for every two inputs x,x' to party P_1, the vectors $\bar{p}^x$ and $\bar{p}^{x'}$ must be different, and for every two inputs y,y' to party P_2, the vectors $\bar{q}^y$ and $\bar{q}^{y'}$ must be different.)

However, since party P_1 tosses only r_1 coins, each probability element in a vector $\bar{p}^x$ may be assigned only $2^{r_1}+1$ different values. Thus, $|S_1| \leq (2^{r_1}+1)^{2^m}$. Similarly, $|S_2| \leq (2^{r_2}+1)^{2^m}$. Part (a) follows.

(Note that in contrast to the setting of Theorem 1 (one-way communication), the sum $\sum_i p^x(con_i)$ does not necessarily amount to 1.)

Part (b): Proof is by a simulation argument. We simulate a randomized protocol π by a deterministic one, going over the coin tosses of π. Since protocol π computes f with advantage ϵ, it suffices to find a value $\sigma \in \{0,1\}$ such that π outputs σ on $1+(\frac{1}{2}-\epsilon)2^r$ different coin-toss sequences. Consequently, it suffices to go over $1+(1-2\epsilon)2^r$ of the coin-toss sequences of π. Thus, $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f) \leq m \cdot (1+(1-2\epsilon)2^r)$. □

3.3. The public coins models.

THEOREM 3.4. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a non-degenerate function, and let π be a* **one-way, public coins** *protocol that computes f with advantage ϵ. Let $r = r(\pi)$ and $m = m(\pi)$. Then*

$$r \geq \log\left(\frac{\frac{n}{m}-1}{1-2\epsilon}\right).$$

PROOF. The proof uses the same simulation technique as in Part (b) of Theorem 2. However, here the resulting deterministic protocol is one-way, and $\mathrm{C}_{\mathrm{D}}^{1\Rightarrow 2}(f) = n$. □

(A similar result can be achieved using a combinatorial argument.)

THEOREM 3.5. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a non-degenerate function, and let π be a* **two-way, public coins** *protocol that computes f with advantage ϵ. Let $r = r(\pi)$ and $m = m(\pi)$. Then the following inequalities hold simultaneously.*

$$\text{(a)} \quad \frac{2^{2^{m+r}}}{2^{(2^m-1)\epsilon 2^r} \cdot \binom{2^r}{\epsilon 2^r}} \geq 2^n$$

$$\text{(b)} \quad r \geq \log\left(\frac{\frac{\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f)}{m}-1}{1-2\epsilon}\right).$$

Approximating part (a), we have

$$2^n \leq \frac{2^{2^{m+r}}}{2^{(2^m-1)\epsilon 2^r} \cdot \binom{2^r}{\epsilon 2^r}} \leq 2^{2^{m+r}-(2^m-1)\epsilon 2^r - \epsilon(r+\log\epsilon^{-1})2^r} \leq 2^{(1-\epsilon)2^{r+m}}.$$

Thus,

$$r \geq \log\left(\frac{n}{(1-\epsilon)2^m}\right).$$

Note that the first bound is stronger for smaller values of m, and the second for larger values of m. The crossing point of the bounds is determined by the deterministic, two-way communication complexity $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f)$. In particular, for functions f with $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f) = n$, the first bound is higher for all values of m, whereas for functions f with $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f) = \log n$ (the lowest possible, see Yao 1979), the second bound is higher for all values of m.

Proof. Part (b) is identical to Part (b) in the proof of Theorem 2.

Part (a): Proof is by a combinatorial argument similar to the one used to prove Part (a) of Theorem 2. Consider a conversation $con = u_1 \circ v_1 \circ \ldots \circ u_k \circ v_k$. Let $p_r^x(con)$ be the indicator of the event "party P_1 'goes along' con, on input x and coin tosses r". I.e., $p_r^x(con) = 1$ if, for each j, on input x and coin tosses r, party P_1 sends u_j, given that the conversation so far is $u_1 \circ v_1 \circ \ldots u_{j-1} \circ v_{j-1}$ (the corresponding prefix of con). Consider an enumeration of all the possible conversations and coin tosses. For each input x, let T^x be the $2^r \times 2^m$ table where $T^x_{i,j} = p^x_{r_i}(con_j)$. Using the same arguments as in the proofs of Theorems 1 and 2, we get that all 2^n tables are distinct. Furthermore, for every two different inputs x, x', the corresponding tables $T^x, T^{x'}$ differ by at least $2\epsilon \cdot 2^r$ rows (namely, the random inputs).

However, there are only $2^{2^m \cdot 2^r}$ distinct such tables T. Moreover, the number of distinct tables that differ from a given table by at most $\epsilon \cdot 2^r$ rows is at least the number of ways to alter $\epsilon 2^r$ given rows, times the number of ways to choose a set of $\epsilon 2^r$ rows, namely $2^{(2^m-1)\cdot\epsilon 2^r} \cdot \binom{2^r}{\epsilon 2^r}$. Part (a) follows. □

3.4. A summary of the lower bounds. We give a short overview of the bounds, and the relations between them. Table 1 presents the lower bounds for the four models, when the advantage ϵ approaches 0. Note that this is the 'weakest' form of the lower bounds. However, when ϵ is bounded away from $\frac{1}{2}$ (i.e., when there exists a constant $c > 0$ such that $\epsilon < c < \frac{1}{2}$), the dependence on ϵ adds only a constant factor to the bounds.

For three of the four models we have two different bounds, which hold simultaneously. However, in each model there exists a value m^* such that for $m < m^*$ the first bound is higher, and for $m > m^*$ the second bound is higher. In the following remarks we state some observations concerning the bounds.

- Clearly, the bounds for the two-way communication models hold for the corresponding one-way models as well. Also, the bounds on the public coins models hold also for the corresponding private coins models.

- In general, the bounds depend both on n and on the deterministic communication complexity of the function being computed. (The distinction makes sense only in the two-way models, since only in these models does the deterministic communication complexity have values different from n.) However,

Table 1: A summary of the bounds, for $\epsilon \to 0$

	one-way	two-way
private coins	$R_{i,m}(f) \geq \frac{n}{2^m} - 1,\ i = 1,2$ $R_{1,m}(f) \geq \log \frac{n}{m}$	$R_{i,m}(f) \geq \frac{n}{2^m} - 1,\ i = 1,2$ $R_m(f) \geq \log(\frac{\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f)}{m})$
public coins	$R_m(f) \geq \log \frac{n}{m}$	$R_m(f) \geq \log n \ - \ m$ $R_m(f) \geq \log(\frac{\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f)}{m})$

the dependence on the deterministic communication complexity holds only for large numbers of bits communicated. I.e., for every function f, there exists a limit m^* such that for every $m < m^*$, the corresponding bounds are independent of $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f)$. In this range (i.e., $m < m^*$), the bounds are similar also to those in the one-way communication models. In the private coins, two-way communication model, and for every function, this limit (i.e., m^*) is $\Theta(\log n)$.

- For functions with two-way deterministic communication complexity $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2} = n$, the bounds for the one-way models are similar to the bounds for the two-way models, for every value of m.

- The relation between the bounds on public and private coins varies with m. For $m > \log n$, the bounds are similar; as m becomes smaller, the difference between these bounds grows. In particular, for $m = O(1)$, the bounds in the public coins models are logarithmic in the input length n, while the bounds in the private coins models are linear in n.

4. Tightness of the bounds

In Sections 4.1 and 4.2, we demonstrate the tightness of the lower bounds presented in Section 3. We do this by showing protocols for non-degenerate functions that match the corresponding bounds on the maximum number of coin tosses used. We do this separately for each of the models.

In Section 4.3 we show a general method for reducing the number of coin tosses used by *any* protocol (computing a function with advantage ϵ) by up to $\log n + O(\log \frac{1}{\epsilon})$, without increasing the number of bits communicated, Recall that $\hat{x}$ denotes the integer whose binary representation is x, and let $e \in_R D$ denote that element e is chosen at random from domain D, with uniform probability distribution.

4.1. Tightness of the bounds for one-way communication. The tightness of the bounds for the one-way cases is demonstrated by showing two families of one-way protocols for the identity function (i.e., $ID(x,y) = 1$ iff $x = y$), one family in the private coins model and the other in the public coins model. Both families consist of protocols parameterized by the maximum number of bits exchanged by the parties, denoted m_0. For each possible value of m_0, there exists a protocol that communicates up to m_0 bits, and meets the corresponding bound on the number of coin tosses used, up to a multiplicative factor.

4.1.1. Private coins. In this section we demonstrate the tightness of the lower bound given by Theorem 1. We use two known protocols for ID as building blocks:

- The following protocol is due to Rabin and Yao (1990), and is denoted RY. On input $x, y \in \{0,1\}^n$, party P_1 chooses at random a prime p in $[n, 4n]$, and sends $(p, \hat{x} \bmod p)$ to P_2. Party P_2 outputs 1 iff $(\hat{x} \bmod p) = (\hat{y} \bmod p)$.
  It is easy to verify that if $x \neq y$ then $\mathrm{Prob}(\pi(x,y) = 0) \geq \frac{2}{3} - o(1)$. (There are $(3 - o(1))\frac{n}{\log n}$ primes in $[n, 4n]$ and if $x \equiv y$ modulo more than $\frac{n}{\log n}$ of the primes then $x = y$.) If $x = y$ then $\pi(x,y) = 1$ always holds. Note that $m(\mathrm{RY}) = 2\log n - \log\log n + 4$, and $r_1(\mathrm{RY}) = \log n - \log\log n + 2$, and $r_2(\mathrm{RY}) = 0$.

- The following protocol is a slight modification of a protocol presented by Paturi and Simon (1986), and is denoted PS. The parties use three different messages. Let p_i^x denote the probability that party P_1 sends the i^{th} message on input x, and q_i^y the probability that party P_2, on input y and having received the i^{th} message, outputs 1. On inputs $x, y \in \{0,1\}^n$, let
$$\bar{p}^x = c_x(\cos\left(\frac{\hat{x}\pi}{2^{n+1}}\right), \sin\left(\frac{\hat{x}\pi}{2^{n+1}}\right), 1)$$
  where c_x is a normalization constant, and
$$\bar{q}^y = \frac{1}{2}(1 + \cos\left(\frac{\hat{y}\pi}{2^{n+1}}\right), 1 + \sin\left(\frac{\hat{y}\pi}{2^{n+1}}\right), 1 - \cos\left(\frac{\pi}{2^{n+2}}\right)).$$
  It is easy to verify that
$$\begin{array}{lll} x \neq y & \Rightarrow & \mathrm{Prob}(\pi(x,y) = 1) = \bar{p}^x \cdot \bar{q}^y \leq \frac{1}{2} - 2^{-2n-4} \\ x = y & \Rightarrow & \frac{1}{2} + 2^{-2n-6} \leq \mathrm{Prob}(\pi(x,y) = 1) \leq \frac{1}{2} + 2^{-2n-4}. \end{array} \tag{4.5}$$
  (Use the fact that $\frac{1}{3} \leq c_x \leq \frac{1}{2}$ for all x, and that $\frac{\alpha^2}{4} \leq 1 - \cos\alpha \leq \frac{\alpha^2}{2}$ for $0 \leq \alpha \leq \frac{\pi}{2}$.) Note that we may approximate these probabilities by multiples of 2^{-2n-8}, using only $2n + 8$ coin tosses. Thus, $r_1(\mathrm{PS}) = r_2(\mathrm{PS}) = 2n + 8$.

The combined protocol operates as follows on inputs $x, y \in \{0,1\}^n$. Let m_0 be the maximum number of bits to be communicated. We distinguish three cases.

Case 1: $m_0 \geq 2\log n - \log\log n + 4$. Partition the input into blocks of length k (to be computed) and execute RY on each block, using the same prime for all the blocks. Namely:

Let $x = x_1 \ldots x_{\frac{n}{k}}$, and $y = y_1 \ldots y_{\frac{n}{k}}$, where each x_i, y_i is of length k.
Party P_1**:** Choose a prime $p \in_R [k, 4k]$. compute $x^p = (\hat{x}_1 \bmod p) \circ \ldots \circ (\hat{x}_{\frac{n}{k}} \bmod p)$. Send (p, x^p) to P_2.
Party P_2**:** Compute $y^p = (y_1 \bmod p) \circ \ldots \circ (y_{\frac{n}{k}} \bmod p)$. Output 1 iff $y^p = x^p$.

It can be verified that for these values of m_0,

$$\begin{array}{l} \epsilon(\pi) > \frac{1}{7} \\ m(\pi) = (\frac{n}{k}+1)\log 4k - \log\log k \\ r_1(\pi) = \log 4k - \log\log k. \end{array}$$

Set k to be the smallest integer such that $m(\pi) \leq m_0$. (This value of k satisfies $\frac{k}{\log k} \approx \frac{n}{m+0}$.) [3] We get that for every m in this range,

$$R_{1,m}(ID) \leq r_1(\pi) \leq \log\left(\frac{4k}{\log k}\right) \leq \log\left(\frac{n}{m}\right) + 3.$$

Recall that for this case, Part (b) of Theorem 1 gives the higher bound, namely $\log(\frac{n}{m}) \leq R_{1,m}(f)$, for every non-degenerate function f. Hence, in this case, the bound given by Theorem 1 is tight up to an additive constant.

Case 2: $m_0 \leq \log n - \log\log n + 4$. As in Case 1, partition the input into blocks of length k (to be computed). For these values of m_0, the parties execute PS on (x^p, y^p) instead of having party P_1 send x^p to P_2. Namely:

Party P_1**:** Choose a prime $p \in_R [k, 4k]$. Compute $x^p = (\hat{x}_1 \bmod p) \circ \ldots \circ (\hat{x}_{\frac{n}{k}} \bmod p)$. Send p to P_2 and execute PS on x^p.
Party P_2**:** compute $y^p = (\hat{y}_1 \bmod p) \circ \ldots \circ (\hat{y}_{\frac{n}{k}} \bmod p)$. Execute PS on y^p.

Let us compute the advantage achieved by the protocol in this range. Let $l = \frac{n}{k}\log 4k$, and note that $|x^p| \leq l$. If $x = y$, then $x^p = y^p$ always holds. Thus, by (5)

$$\mathrm{Prob}(\pi(x,y)=1) = \mathrm{Prob}(\mathrm{PS}(x^p,y^p)=1) \geq 2^{-2l-6}.$$

If $x \neq y$, we have

$$\begin{aligned} \mathrm{Prob}(\pi(x,y)=1)) = \; & \mathrm{Prob}(\mathrm{PS}(x^p,y^p)=1 \mid x^p \neq y^p)\cdot\mathrm{Prob}(x^p \neq y^p) + \\ & \mathrm{Prob}(\mathrm{PS}(x^p,y^p)=1 \mid x^p = y^p)\cdot\mathrm{Prob}(x^p = y^p) \\ & \leq (\tfrac{1}{2} - 2^{-2l-4})\cdot\tfrac{2}{3} + (\tfrac{1}{2} + 2^{-2l-4})\cdot\tfrac{1}{3} \leq \tfrac{1}{2} - 2^{-2l-6}. \end{aligned}$$

[3] Assume that k divides n. Otherwise, the last block will be shorter.

(Note that we used both lower *and upper* bounds (5) on the advantage of protocol PS.)

Consequently, for these values of m_0, we have

$$\begin{aligned}
&\epsilon(\pi) \geq 2^{-6-\frac{2n}{k}\log 4k} \\
&r_1(\pi) = \log 4k - \log\log k + \tfrac{2n}{k}\log 4k + 8 \leq \tfrac{3n}{k}\log 4k + 8 \\
&m(\pi) = \log k - \log\log k + 4.
\end{aligned}$$

Set k to be the *largest* integer such that $m(\pi) \leq m_0$. (This value of k satisfies $\frac{k}{\log k} \approx 2^{m_0-4}$.) We get that for every m in this range,

$$R_{1,m}(ID) \leq r_1(\pi) \leq \frac{3n}{k}\log 4k + 8 \leq \frac{48n}{2^m} + 8.$$

Recall that for this case, Part (a) of Theorem 1 gives the higher bound, namely $\frac{n}{2^m} - 1 \leq R_{1,m}(f)$, for every non-degenerate function f. Hence, in this case, the bound given by Theorem 1 is tight up to a multiplicative constant.

Case 3: $\log n - \log\log n + 4 < m_0 < 2\log n - \log\log n + 4$. For these values of m_0, we use the construction of the previous case with $k = n$ (i.e., one block). Thus, for every m in this range,

$$R_{1,m}(ID) \leq r_1(\pi) = 3\log 4n + 8 \leq 4\log\left(\frac{n}{2\log n}\right) \leq 4\log\left(\frac{n}{m}\right).$$

Using part (b) of Theorem 1 once again, we get that the bound is tight up to a multiplicative constant in this case as well.

4.1.2. Public coins. In this section we demonstrate the tightness of the bound given by Theorem 3. Let us first remark that $n+1$ public coins are sufficient for computing *any* function f with advantage 2^{-n}, exchanging one bit. (If the first n coin tosses are equal to the input of P_1, it sends '1' and P_2 computes the function value. Otherwise P_1 sends '0' and P_2 outputs the value of the remaining coin.) However, the lower bound on $R_1(f)$ (i.e., the number of coins used in protocols communicating a single bit) in the public coins models is only $\log n$. The following family of protocols for *ID* matches the bound for every number of bits communicated.

We show a family of one-way, public coins protocols for *ID*. Given a limit of m_0 bits to be exchanged, the parties use $\log(\frac{n}{m_0}) + 2$ coin tosses, and interpret the first $\log(\frac{n}{m_0})$ coins as an integer $i \in \{1 \ldots \frac{n}{m_0}\}$. Let the inputs be partitioned to $\frac{n}{m_0}$ blocks $x = x_1 \ldots x_{\frac{n}{m_0}}$ and $y = y_1 \ldots y_{\frac{n}{m_0}}$, such that $|y_j| = |x_j| = m_0$.

Party P_1: Send the block x_i to P_2.

Party P_2: If $x_i \neq y_i$ output 0. Otherwise ($x_i = y_i$), if $i = 1$ output 1 with probability $\frac{3}{4}$, and if $i \neq 1$, output 1 with probability $\frac{1}{2}$. These random selections are implemented using the two remaining public coins.

We have

$$x = y \Rightarrow \mathrm{Prob}(\pi(x,y)=1) = \begin{array}{l}\mathrm{Prob}(\pi(x,y)=1|i=0)\cdot\mathrm{Prob}(i=0)+\\ \mathrm{Prob}(\pi(x,y)=1|i\neq 0)\cdot\mathrm{Prob}(i\neq 0)\\ =\frac{3}{4}\frac{m_0}{n}+\frac{1}{2}(1-\frac{m_0}{n})=\frac{1}{2}+\frac{m_0}{4n},\ \text{ and}\end{array}$$

$$x \neq y \Rightarrow \mathrm{Prob}(\pi(x,y)=1) \leq \frac{3}{4}\frac{m_0}{n}+\frac{1}{2}(1-\frac{2m_0}{n})=\frac{1}{2}-\frac{m_0}{4n}.$$

Thus, $\epsilon(\pi) \geq \frac{m_0}{4n}$. (Note that a more 'natural' way to design the above protocol is to let party P_2 output 1 with probability $\frac{1}{2}+\frac{1}{4n}$, in the case where $x_i = y_i$, for every i. However, this 'natural' design is wasteful in coin tosses, as it requires $\log n$ additional coins.)

We get that for every value of m, $R_m(ID) \leq r(\pi) = \log(\frac{n}{m_0}) + 2$. Recall that Theorem 3 states that $\log(\frac{n}{m}) \leq R_{1,m}(f)$, for every non-degenerate function f. Hence the bound given by Theorem 3 is tight up to an additive constant for every value of m.

We remark that a public coins, one-way protocol presented by Feder *et al.* (1991) achieves *constant* advantage with $O(1)$ communication and $O(\log n)$ coin tosses. (However, this protocol tosses more than $2\log n$ coins. Thus, it is tight with the bound only up to a multiplicative constant.)

4.2. Tightness of the bounds for two-way communication. In this section, we demonstrate the tightness of the bounds given by Theorems 2 and 4. In the two-way communication models, the bounds depend also on the deterministic communication complexity of the function being computed. We thus demonstrate the tightness of the bounds using a sequence of functions, one function for each possible value of the deterministic communication complexity. For each function, model, and maximum number of bits to be communicated (denoted m_0), we give a protocol computing the function, communicating up to m_0 bits, and using a number of coin tosses that matches the corresponding lower bound.

Note that for every non-degenerate function f, we have $\log n \leq \mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f) \leq n$. (The lower bound appears in Yao (1979). It is also a special case of Theorem 2.)

The functions we use are hybrids of two functions. One is the Identity function, with $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(ID) = n$, and the other is the Pointer function PO (defined below), with $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(PO) = \log n$. The protocols for the hybrid functions use the protocols for the Identity and the Pointer functions as primitives.

In Section 4.2.1 we describe PO, and two-way protocols computing PO. In Section 4.2.2 we describe the hybrid functions and their protocols.

All protocols use a number of coin tosses that is up to a constant times the corresponding bounds.

4.2.1. The Pointer function. The Pointer function consists of three regions:

$$PO(x,y) = \begin{cases} \text{the } \hat{y}\text{-th bit in } x & \text{if } \hat{y} < n \text{ and } \hat{x} \geq n \\ \text{the } \hat{x}\text{-th bit in } y & \text{if } \hat{x} < n \text{ and } \hat{y} \geq n \\ 0 & \text{otherwise.} \end{cases}$$

The table of PO can be described as follows. The first n columns are constructed by writing all 2^n binary words of length n, one below the other, in the standard lexicographical ordering. The first n rows are constructed in a similar manner. The rest of the entries in this table (as well as the entries in the intersection of the first n columns with the first n rows) are 1.

It can be seen that $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(PO) = 1 + \log n$. A lower bound follows from the existence of $2n$ entries in the PO table, no two of which belong to the same generalized monochromatic rectangle. (These $2n$ entries are the single '0' entries in rows (columns) that are otherwise all '1's. See Yao (1979) for details about lower bounds on the deterministic communication complexity using decompositions to monochromatic rectangles.) An upper bound is due to the deterministic protocol described below.

We show two-way protocols for PO. These protocols are used as primitives in the protocols for the hybrid functions. We treat separately the public coins and the private coins models.

Public coins. We show a family of two-way, public coins protocols for PO. Let $x, y \in \{0,1\}^n$ be the inputs, and let m_0 be the maximum number of bits to be communicated. Let $k = \lceil \frac{n}{2^{m_0-3}} \rceil$. The parties use $\log k + 1$ coin tosses, in the following way. Each party first sends one bit indicating whether the value of its input is at least n. Without loss of generality, assume that $\hat{y} < n$ and $\hat{x} \geq n$. Then, party P_1 partitions $x = x_1 \ldots x_k$, so that $|x_i| = \frac{n}{k}$. Let the parties interpret the first $\log k$ public coins as $i \in \{0 \ldots k-1\}$.
Party P_2: If $\frac{in}{k} \leq \hat{y} < \frac{(i+1)n}{k}$ (namely if $\hat{y}$ is in the i^{th} block) then send $1 \circ (\hat{y} - \frac{in}{k})$ to P_1. Otherwise send 0. (Note that $\hat{y} - \frac{in}{k}$ is at most $2^{m_0-3} = \frac{n}{k}$, and thus can be communicated using only $m_0 - 3$ bits.)
Party P_1: If received $1 \circ z$, output the $\hat{z}^{\text{th}}$ bit in x_i (namely, the $\hat{y}^{\text{th}}$ bit in x). Otherwise (received 0), output the value of the remaining coin.
Clearly, m_0 communication-bits are sufficient, and it can be verified that $\epsilon(\pi) = \frac{1}{2k}$. Thus, for every value of m, we have that $R_m(PO) \leq \log n - m + 4$. Recall that according to Part (b) in Theorem 4, we have that $\log n - m \leq R_m(f)$, for every non-degenerate function f and for every possible value of m. Thus, for $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2} = 1 + \log n$, the bound of Theorem 4 is tight up to an additive constant.

Private coins. In the (two-way) private coins model, we use a *deterministic* protocol for PO. This protocol is a special case of the above public coins protocol. Namely, party P_1 first sends a bit specifying if the value of its input is at least n. Assume, without loss of generality, that $\hat{y} < n$ and $\hat{x} \geq n$. Then, party P_2 sends the first $\log n$ bits of y, and party P_1 outputs the function value. Thus, at most $1 + \log n$ bits are communicated. Note that our lower bound on the number of coin tosses in this case (i.e., Theorem 2, with $m = 1 + \log n$ and $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2} = 1 + \log n$) is indeed zero.

We note that randomization does not reduce the communication complexity of computing PO in the two-way, private coins model. Theorem 4 in Paturi and Simon (1986) implies that every two-way, private coins protocol for PO, tossing any number of coins, communicates at least $\log n$ bits.

4.2.2. The hybrid functions. In order to show the tightness of the bounds for all possible values of the deterministic communication complexity, we define a series of hybrid functions. For $\log n \leq i \leq n$, let

$$H_i(x,y) = \begin{cases} ID(x,y) & \text{if } \hat{x} < 2^i \text{ and } \hat{y} < 2^i \\ PO(x,y) & \text{otherwise.} \end{cases}$$

It can be seen that $i \leq \mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(H_i) \leq i+2$. The main contributor to the communication complexity of each hybrid is the ID block, while the PO block is used to keep the function non-degenerate.

Protocols for H_i, parameterized by the number of bits to be communicated, are an immediate combination of the corresponding protocols for ID and PO, either in the private coins or the public coins models. Namely, the parties first learn, using two bits of communication, the region their inputs are in. Then the parties execute the appropriate protocol. (For the ID region we use the one way protocols described in Section 4.1, and for the PO region we use the protocols described in Subsection 4.2.1.)

We show that for each H_i, and for every number of bits communicated, these combined protocols (using either public or private coins) match the corresponding lower bounds on the number of coins tossed (namely, the bounds of Theorems 2 and 4), up to a multiplicative constant.

Fix a computation model (either public or private coins), and let $\pi_{i,m}$ be the combined protocol computing H_i that communicates up to m bits. For analyzing the protocol, we distinguish two cases.

Case 1: The inputs are in the PO region of H_i. In this case, the number of coins tossed by $\pi_{i,m}$ is the same as in the corresponding protocol for the PO function. However, we have that $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(H_i) \geq \mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(PO)$. Thus, the lower bound on the number of coins tossed in protocols for H_i is at least the corresponding lower bound for protocols computing the PO function. Since the protocol for PO used by $\pi_{i,m}$ matches the bound for PO, we have that, in this case, protocol $\pi_{i,m}$ matches the bound for H_i.

Case 2: The inputs are in the ID region of H_i. Let $ID^{(i)}$ denote the ID function for inputs of length i. If the input of $\pi_{i,m}$ is in the ID region of function H_i, then the parties execute the corresponding protocol for $ID^{(i)}$. Recall that for functions

f with $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(f) = n$ (e.g., ID), the one-way and the two-way bounds are similar up to a multiplicative constant for every number of bits communicated. Therefore, the *one-way* protocols for $ID^{(i)}$ used by $\pi_{i,m}$ meet also the corresponding *two-way* lower bounds for protocols computing $ID^{(i)}$. However, we have that $\mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(H_i) \geq i = \mathrm{C}_{\mathrm{D}}^{1\Leftrightarrow 2}(ID^{(i)})$. Thus, the lower bound on the number of coins tossed in protocols for H_i is at least the corresponding lower bound for protocols computing $ID^{(i)}$. Consequently, protocol $\pi_{i,m}$ matches the corresponding lower bound in this case as well.

4.3. Reducing the number of coin tosses. We have constructively demonstrated the tightness of the bounds (for all the computation models, and all possible numbers of bits communicated). I.e., we showed that there exist *specific* functions, and protocols for these functions, that use a minimal number of coin tosses.

The following theorem gives a more general result. It shows that the number of coins tossed in *any* protocol computing a function (with advantage ϵ) can be reduced by up to $\log n + O(\log \frac{1}{\epsilon})$, without increasing the number of bits communicated.

THEOREM 4.1. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and let π be a protocol that computes f with advantage ϵ. Then there exists a protocol π' (of the same model) such that $m(\pi') = m(\pi)$ and*

(a) $r(\pi') \leq \log n + 2\log\frac{1}{\epsilon} + 3$ *and* $\epsilon(\pi') = \frac{\epsilon}{2}$ *(for the public coins models)*

(b) $r_i(\pi') \leq \log n + 2\log\frac{1}{\epsilon} + 5$ *and* $\epsilon(\pi') = \frac{\epsilon}{4}$ *(for the private coins models, and $i = 1, 2$)*

The upper bound of Theorem 4.3 on r as a function of $\frac{1}{\epsilon}$, combined with the lower bounds of Table 3.4 on r as a function of m, yields upper bounds on ϵ as a function of m. In the private coins models, the resulting bound is:

COROLLARY 4.2. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and let π be a private coins, two-way protocol that computes f with advantage ϵ. Let $m = m(\pi)$. Then*

$$\epsilon \leq \sqrt{32n} \cdot 2^{-\frac{n}{2^{m+1}}}.$$

(Note that a slightly weaker bound can be achieved using known simulation techniques.)

PROOF. A proof of part (a) appears in (Newman 1991). For sake of self-containment we present a proof of both parts.

Part a (public coins). Consider the $2^{2n} \times 2^{r(\pi)}$ table where a row corresponds to an input pair, and a column to a sequence of coin tosses. A table-entry $((x,y),r)$ is 1 iff the output of protocol π on the input (x,y) and coin tosses r is $f(x,y)$. Since protocol π computes f with advantage ϵ, the fraction of '1' entries in each row is at least a $(\frac{1}{2}+\epsilon)$.

We use a probabilistic argument to show that there exists a subset K of the columns (i.e., coin tosses), of size $k=\frac{8n}{\epsilon^2}$, with the following property: in the reduction of the table to the columns in the subset K, the fraction of '1' entries in each row is at least $(\frac{1}{2}+\frac{\epsilon}{2})$.

Protocol π' is thus the following. Use $\log k$ (public) coin tosses to choose a sequence $r \in_{\mathrm{R}} K$, and run protocol π with r as a random input. Due to the above property of subset K, protocol π computes f with advantage $\frac{\epsilon}{2}$.

We show the existence of such a subset, K, in the following way. Consider a random selection of k columns (i.e., k independent selections of a column are made, where each selection is made with uniform probability). Say that a row is *bad* for the selection if less than $(\frac{1}{2}+\frac{\epsilon}{2})k$ entries of value 1 were chosen in this row. In order to estimate the probability of a row being bad for the selection, we use the Hoefding inequality (Hoefding 1963):

Hoefding inequality. *Let $x_1, \ldots x_n$ be identically distributed, independent random variables over the interval $[a,b]$. Let $\bar{x}=\sum_{i=1}^{n} x_i$. Then*

$$\mathrm{Prob}(|\bar{x}-E(\bar{x})| \geq \delta) \leq 2 \cdot \mathrm{e}^{-\frac{2n\delta^2}{b-a}}.$$

Consequently, the probability of a row being bad for the selection is less than $2e^{-\frac{k\epsilon^2}{2}}$. Thus the probability that there *exists* a bad row for the selection is at most $2^{2n} \cdot 2e^{-\frac{k\epsilon^2}{2}}$.

For $k=\frac{8n}{\epsilon^2}$, we have that the probability (over the selections of k columns) that there exists a bad row for the selection is less than 1. We conclude that there exists a subset of $\frac{8n}{\epsilon^2}$ coin toss sequences (columns) that is good for all the input pairs (rows).

Part b (private coins). We use a technique similar to the one used in Part (a). We apply the technique twice: first to reduce r_1, the number of coin tosses used by party P_1, and then to reduce r_2, the number of coin tosses used by party P_2.

Consider the $2^{2n} \times 2^{r_1(\pi)}$ table T where a row corresponds to an input pair, and a column to a sequence of coin tosses of party P_1. The value of a table-entry is $T((x,y),r)=\mathrm{Prob}(\pi(x,y)=f(x,y))$, where r is a fixed random input of party P_1, and the probability is taken only over the coin tosses of party P_2.

Since protocol π computes f with advantage ϵ, we have that for each row (input (x,y)), the average of $T((x,y),r)$ over all $r \in \{0,1\}^{r_1(\pi)}$ is at least $\frac{1}{2}+\epsilon$.

Consider a random selection of k columns, and let K denote the set of chosen columns. We say that a row (x,y) is *bad* for the selection K, if the average of $T((x,y),r)$, over all $r \in K$, is less than $(\frac{1}{2}+\frac{\epsilon}{2})$. Using the Hoefding inequality, we get that the probability of a row being bad for the selection is less than $2e^{-\frac{k\epsilon^2}{2}}$. Following the lines of the proof of Part (a), we conclude that there exists a protocol π' such that $r_1(\pi')=\log n+2\log\frac{1}{\epsilon}+3$, and $\epsilon(\pi')=\frac{\epsilon}{2}$.

Consider now the $2^{2n} \times 2^{r_2(\pi')}$ table T', where a row corresponds to an input pair, and a column to a sequence of coin tosses of party P_2. The value of a table-entry is $T'((x,y),r) = \text{Prob}(\pi'(x,y) = f(x,y))$, where r is a fixed random input of party P_2 and the probability is taken only over the coin tosses of party P_1 during the execution of protocol π'.

Since protocol π' computes f with advantage $\frac{\epsilon}{2}$, we have that for each row (input (x,y)), the average of $T'((x,y),r)$, over all $r \in \{0,1\}^{r_2(\pi')}$, is at least $\frac{1}{2} + \frac{\epsilon}{2}$.

Following the same line of proof, we conclude that there exists a protocol π'' such that $r_2(\pi'') = \log n + 2\log\frac{1}{\epsilon} + 5$, and $r_1(\pi'') = r_1(\pi') = \log n + 2\log\frac{1}{\epsilon} + 3$, and $\epsilon(\pi'') = \frac{\epsilon}{4}$. $\square$

5. Other measures of randomness

In the previous sections, we considered only protocols using coin tosses as their source of randomness, and measured the *maximum* number of coin tosses used. However, protocols may use a random source other than coin tosses. A general measure for the amount of randomness used in algorithms is the *entropy* of the random source used.

In Section 5.1 we consider the *expected* number of coins tossed by parties executing a protocol to compute a function. We derive a lower bound on the *expected* number of coins tossed, depending on the advantage achieved by the protocol and on the corresponding lower bound on the *maximum* number of coins tossed. We also show that the expected number of coins tossed in *any* protocol can be reduced up to a constant, with the advantage of the protocol becoming smaller in the process.

In Section 5.2 we show that the lower bound on the expected number of coins tossed yields an identical (up to an additive constant) lower bound on the entropy of the random source of the parties, and vice versa.

In the sequel, we do not specify the model used (private or public random input, one- or two-way communication). However, the results described in this section hold for each one of the four models of computation.

5.1. The expected number of coin tosses. Let $\bar{r}(\pi)$ denote the expected number of coin tosses used in protocol π on the worst input pair:

$$\bar{r}(\pi) \stackrel{\text{def}}{=} \max_{x,y\in\{0,1\}^n} \bar{r}(\pi(x,y)),$$

where $\bar{r}(\pi(x,y))$ is the expected number of coin tosses used on input (x,y):

$$\bar{r}(\pi(x,y)) = \sum_{r>0} r \cdot \text{Prob}(\text{exactly } r \text{ coin tosses are used on input } (x,y)).$$

For a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, let $\bar{R}_{m,\epsilon}(f)$ denote the minimum of $\bar{r}(\pi)$ over all coin tossing protocols π that compute f with up to m communication bits and advantage at least ϵ.

Recall that $R_m(f)$ denotes the minimum $r(\pi)$, over all protocols π that compute function f with any positive advantage, and using up to m communication bits. We use a standard argument to show that the lower bound on $\bar{R}_{m,\epsilon}(f)$ is related to $R_m(f)$ times the advantage ϵ achieved by the protocol.

THEOREM 5.1. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, let m be an integer and let $\epsilon > 0$. Then,*

$$\bar{R}_{m,\epsilon}(f) \geq \frac{\epsilon}{2} \cdot R_m(f).$$

PROOF. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and let π be a protocol computing f with advantage ϵ, using up to m communication bits. Let π' be the following protocol. The parties run protocol π, using at most $2\bar{r}_\pi \epsilon^{-1}$ coin tosses. If more coin tosses are required, their outcome is set to 0. We have, for every pair of inputs $x, y \in \{0,1\}^n$,

$$\begin{aligned}&|\text{Prob}(\pi'(x,y) = 1) - \text{Prob}(\pi(x,y) = 1)|\\ &\leq \text{Prob}(\pi \text{ uses more than } 2\bar{r}\epsilon^{-1} \text{ coins}) \leq \tfrac{\bar{r}}{2\bar{r}\epsilon^{-1}} = \tfrac{\epsilon}{2}.\end{aligned}$$

Thus, protocol π' computes f (with advantage $\frac{\epsilon}{2}$), using up to m communication bits and a maximum of $2\bar{r}_\pi \epsilon^{-1}$ coin tosses. □

Note that Theorem 5.1 allows the expected number of coin tosses $\bar{r}$ used by a protocol computing a function f to be arbitrarily small, provided that the advantage ϵ achieved by the protocol is small enough (namely, $\epsilon \leq 2\bar{r}/R_{m(\pi)}(f)$). Let us show that it is indeed possible to arbitrarily reduce the expected number of coins tossed in a protocol, up to a value of 4, at the cost of a linear reduction of the advantage.

THEOREM 5.2. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and let π be a protocol that computes f with advantage ϵ, and uses at most r coin tosses. Then, for every $k \geq 1$, there exists a protocol π_k that computes f with advantage $\frac{\epsilon}{k}$, with $m(\pi_k) = m(\pi) + 1$ and $\bar{r}_{\pi_k} \leq \frac{r}{k} + 3$.*

PROOF. Protocol π_k operates as follows. Party P_1 first tosses $\log k$ coins, one by one. If the outcome of a coin is 0, party P_1 outputs the value of an additional coin and halts. If the outcome of all $\log k$ coin tosses is 1, then the parties run protocol π. □

5.2. The entropy of the random source. For a probability distribution $P = (p_1, p_2, \ldots)$, let the entropy $H(P) \stackrel{\text{def}}{=} -\sum_i p_i \log p_i$. Let π be a probabilistic protocol using any random source. Let h_π denote the maximum, over all input pairs (x, y), of the entropy of the random source of the parties executing protocol π on input (x, y). For a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, let $H_{m,\epsilon}(f)$ denote the minimum value of h_π over all randomized protocols π that compute f with up to m communication bits and advantage at least ϵ.

We show that in each of the computation models, the lower bounds on the entropy of the random input of a protocol are equal, up to an additive constant 2, to the corresponding lower bounds on the expected number of coin tosses used.

THEOREM 5.3. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and let m be a natural number and $\epsilon > 0$. Then*

$$\bar{R}_{m,\epsilon}(f) - 2 \le H_{m,\epsilon}(f) \le \bar{R}_{m,\epsilon}(f).$$

PROOF. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Clearly, for every protocol π using coin tosses as its random input, we have $h_\pi = \bar{r}_\pi$. Thus, $H_{m,\epsilon}(f) \le \bar{R}_{m,\epsilon}(f)$.

We use a construction by Knuth and Yao (1976) to show that for every protocol π using a random source P, there exists a protocol π' using coin tosses as its random source, and $\bar{r}_{\pi'} \le H(P) + 2$. Protocol π' computes f with the same parameters as π (namely, $m(\pi') = m(\pi)$, and $\epsilon(\pi') = \epsilon(\pi)$). Thus $\bar{R}_{m,\epsilon}(f) - 2 \le H_{m,\epsilon}(f)$.

Knuth and Yao showed that for every discrete probability distribution $P = (p_1, p_2, \ldots)$ there exists an algorithm A_P with the following properties.

- Algorithm A_P reads the input z bit by bit until it stops and outputs $A_P(z)$.
- Let the input z be a sequence of independent, unbiased random bits. Then
  1. The expected number of bits A_P reads until it stops is bounded above by $H(P) + 2$.
  2. For every possible output s_i of algorithm A_P, we have that $\text{Prob}(A_P(z) = s_i) = p_i$.

Thus, in protocol π', each party executes algorithm A_P using coin tosses for input, and then protocol π with the output of A_P as a random input. $\square$

Acknowledgements

This work was initiated and inspired by a course on communication complexity given by Benny Chor, to whom we are greatly indebted. We also wish to thank Guy Even, Hugo Krawczyk, Eyal Kushilevitz, Benny Pinkas and Avi Wigderson for helpful discussions. Finally, we thank the anonymous referees for their helpful criticism and modifications.

This research was supported by grant No. 89-00312 from the United States - Israel Binational Science Foundation (BSF), Jerusalem, Israel.

References

R. ALELIUNAS, Randomized parallel communication, in *Proc. of 1st PODC* (1982), 60-72.

N. ALON, P. FRANKL AND V. RÖDL, Geometrical realization of set systems and probabilistic communication complexity, *Proc. of 26th FOCS* (1982), 277-280.

E. BACH AND V. SHOUP, Factoring polynomials using fewer random Bits, *Journal of Symbolic Computation no.* **9** (1990), 229-239.

M. BELLARE, O. GOLDREICH AND S. GOLDWASSER, Randomness in interactive proofs, *Proc. of 31st FOCS* (1990), 563-571.

A. BORODIN AND J. E. HOPCROFT, Routing, merging, and sorting on parallel models of computing, *Journal of Computer and System Science no.* **30** (1985), 130-145.

M. BLUM AN S. MICALI, How to generate cryptographically strong sequences of pseudo-random bits, *SIAM J. Comput.* **13** (Nov. 1984), 850-864.

R. CANETTI AND O. GOLDREICH, Bounds on tradeoffs between randomness and communication complexity, *Proc. of 31st FOCS* (1990), 766-775.

B. CHOR AND C. DWORK, Randomization in byzantine agreements, *Advances in Computing Research,* Vol. **5** (1989), 443-497.

B. CHOR AND O. GOLDREICH, Unbiased bits from sources of weak randomness and probabilistic communication complexity, *SIAM J. Comp., Vol* **17**, *No.* **2** (1988), 230-261.

A. COHEN AND A. WIGDERSON, Dispersers, deterministic amplification, and weak random sources, *30th FOCS* (1989), 14-19.

T. FEDER, E. KUSHILEVITZ AND M. NAOR, Amortized communication complexity, *32nd FOCS* (1991), 239-248.

M. J. FISCHER, N. A. LYNCH AND N. PATERSON, "Impossibility of distributed consensus with one faulty process", *JACM, Vol.* **32**, *no.* **2** (1985), 374-382.

R. FLEISCHER, H. JUNG AND K. MELHORN, "A time-randomness tradeoff for communication complexity", *4th International Workshop on Distributed Algorithms* (1990).

W. HOEFDING, "Probability inequalities for sums of bounded random variables", *Journal of the American Statistical Association,* Vol. **58** (1963), 13-30.

R. IMPAGLIAZZO AND D. ZUCKERMAN, "How to recycle random bits", *Proc. of 30th FOCS* (1989), 248-253.

H. J. KARLOFF AND P. RAGHAVAN, "Randomized algorithms and pseudorandom generators", *Proc. of 20th STOC* (1988), 310-321.

D. E. KNUTH AND A. C. YAO, The complexity of non uniform random number generation, Algorithms and Complexity, ed. J.Traub, AC press, New York (1976), 357-428.

D. KRIZANC, D. PELEG, AND E. UPFAL, A time-randomness tradeoff for oblivious routing, *Proc. of 20th STOC* (1988), 93-102.

K. MELHORN AND E. SCHMIDT, Las-Vegas is better than determinism in VLSI and distributed computing, *Proc. of 14th STOC* (1982), 330-337.

I. NEWMAN, Private vs. common random bits in communication complexity, *Information Processing Letters* **39** (1991), 67-71.

N. NISAN, Pseudorandom generator for space bounded computation *Proc. of 22nd STOC* (1990), 204-212.

R. PATURI AND J. SIMON, Probabilistic communication complexity , *Journal of Computer and System Science, Vol.* **33** (1986), 106-123.

D. Peleg and E. Upfal, A time-randomness tradeoff for oblivious routing, *SIAM J. on Computing no.* **19** (1990), 256-266.

M. Rabin, Randomized Byzantine generals, *Proc. of 24th FOCS* (1983), 403-409.

M. Rabin and A. C. Yao, private communication via M. Rabin (1990).

P. Raghavan and M. Snir, Memory versus randomization in on-line algorithms, *Proc. of 16th ICALP* (1989), 687-703.

M. Sipser, Expanders, randomness, or time versus space, *Proc. of Structure in Complexity Theory, in Lecture Notes in Computer Science no.* **223**, ed. Springer-Verlag (1989), 325-329.

L. G. Valiant, A scheme for fast parallel communication, *SIAM Journal of Computing,* Vol. **11**, No. **2** (1982), 350-361.

U. Vazirani and V. Vazirani, Random polynomial time is equal to slightly-random polynomial time',' *Proc. of 26th FOCS* (1985), 417-428.

A. C. Yao, Probabilistic complexity: towards a unified measure of complexity, *Proc. of 18th STOC* (1977), 222-227.

A. C. Yao, Some complexity questions related to distributive computing, *Proc. of 11th STOC* (1979), 209-213.

A. C. Yao, Theory and applications of trapdoor functions, *Proc. of 23rd FOCS* (1982), 80-91.

D. Zuckerman, Simulating BPP using a general weak random source, *Proc. of 32nd FOCS* (1991), 79-89.

Manuscript received 4 December 1991

Ran Canetti
Department of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel, 32000
`canetti@tx.technion.ac.il`

Oded Goldreich
Department of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel, 32000
`oded@cs.technion.ac.il`