Amortized Communication Complexity[*]

Tomàs Feder

IBM Research Division
Almaden Research Center

Eyal Kushilevitz[‡]

Computer Science Department
Technion - Israel Institute of Technology

Moni Naor[†]

Dept. of Applied Math and Computer Science
Weizmann Institute

Noam Nisan[§]

Computer Science Department
Hebrew University

September 7, 1995

Abstract

In this work we study the *direct-sum* problem with respect to communication complexity: Consider a relation f defined over $\{0,1\}^n \times \{0,1\}^n$. Can the communication complexity of simultaneously computing f on ℓ instances $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ be smaller than the communication complexity of computing f on the ℓ instances, separately?

Let the *amortized* communication complexity of f be the communication complexity of simultaneously computing f on ℓ instances, divided by ℓ. We study the properties of the amortized communication complexity. We show that the amortized communication complexity of a relation can be smaller than its communication complexity. More precisely, we present, a *partial function* whose (deterministic) communication complexity is $\Theta(\log n)$ and its amortized (deterministic) communication complexity is $O(1)$. Similarly, for *randomized* protocols, we present a function whose randomized communication complexity is $\Theta(\log n)$ and its amortized randomized communication complexity is $O(1)$.

We also give a general lower bound on the amortized communication complexity of any *function* f in terms of its communication complexity $C(f)$: for every function f the amortized communication complexity of f is $\Omega\left(\sqrt{C(f)} - \log n\right)$.

[*] An early version of this paper appeared in *Proc. of 32nd IEEE Conference on the Foundations of Computer Science*, October 1991, pp. 239-248.

[†] Most of this work was done while the author was at the IBM Almaden Research center. e-mail: naor@wisdom.weizmann.ac.il

[‡] Research was supported in part by US-Israel BSF grant 88-00282. e-mail: eyalk@cs.technion.ac.il. Some of this work was done while the author was at the Aiken Computation Lab., Harvard University.

[§] Research was supported by the Wolfson Research awards administered by the Israel academy of Sciences and Humanities and by US-Israel BSF 89-00126. e-mail: noam@cs.huji.ac.il

1 Introduction

A very basic question in the theory of computation is the *direct-sum* question: Can the cost of solving ℓ independent instances of a problem simultaneously be smaller than the cost of independently solving the ℓ problems, say, sequentially? In this work we study the direct-sum question in the context of communication complexity. This question was recently raised by Karchmer, Raz and Wigderson [7] as part of a new approach for proving lower bounds on Boolean circuits using communication complexity arguments (as in [8, 18]). For a general survey on communication complexity, see [11]. Different scenarios where the direct-sum question was investigated are [4, 6, 17, 20].

Let f be a *relation* defined on $\{0,1\}^n \times \{0,1\}^n$.[1] Let $f^{(\ell)}$ be the extension of f to ℓ instances. The communication complexity problem associated with $f^{(\ell)}$ is the following: Party P_1 receives ℓ inputs $x_1, \ldots, x_\ell$ and party P_2 receives ℓ inputs $y_1, \ldots, y_\ell$ (each of x_i and y_i is an n bit string). They need to find values $z_1, \ldots, z_\ell$ such that for each i, the value z_i satisfies the relation $f(x_i, y_i)$. Denote by $C(f)$ the communication complexity of f. Namely, the number of bits that the parties need to exchange, on the worst-case input, in the best protocol for computing f. Similarly, denote by $\overline{C}(f)$ the amortized communication complexity of f. Namely,

$$\overline{C}(f) = \limsup_{\ell \to \infty} \frac{1}{\ell} C(f^{(\ell)}).$$

Clearly, $\overline{C}(f) \leq C(f)$ for every relation f. It was observed in [7] that when (non-partial) functions are considered, an upper bound on $\overline{C}(f)$ which is significantly smaller than $C(f)$, implies that the *rank* lower-bound on $C(f)$ [12] is not tight. This is because the rank of the matrix representing $f^{(\ell)}$ equals the rank of the matrix representing f, to the power of ℓ.

We present a *partial* function f, such that $C(f) = \Theta(\log n)$ and $\overline{C}(f) = O(1)$. This proves that computing a relation f on ℓ instances simultaneously may be easier than computing f on the ℓ instances separately. In [7], it was conjectured that $\overline{C}(f)$ can not be smaller than $C(f)$ by more than an *additive* factor of $O(\log n)$. We prove two weaker versions of this conjecture:

- If *one-way* communication protocols are considered then *any* (partial) function f over $\{0,1\}^n \times \{0,1\}^n$ satisfies $\overline{C}_1(f) \geq C_1(f) - \log n - O(1)$.
- For *general* (two-way) protocols, any (non-partial) function f over $\{0,1\}^n \times \{0,1\}^n$ satisfies $\overline{C}(f) \geq \sqrt{C(f)/2} - \log n - O(1)$.

The proof of the first lower bound is via a reduction to an appropriate graph-coloring problem, and then applying the results of Linial and Vazirani [10] on the chromatic number of product graphs. The lower bound for general protocols is achieved by considering *non-deterministic* protocols and proving that $\overline{C}_N(f) \geq C_N(f) - \log n - O(1)$, and then applying a result of Aho, Ullman and Yannakakis

[1] A *relation* defines for every input pair (x, y) a subset $f(x, y)$ of a domain $\mathcal{D}$. We will be interested in particular in two special cases of relations: *functions* – where for each input pair (x, y) there is a unique value in $f(x, y)$, and *partial functions* where for each input pair (x, y) either there is a unique possible value or all values in $\mathcal{D}$ are possible.

[1] which relates the non-deterministic communication complexity of a function with its deterministic communication complexity.

We also study the direct-sum question with respect to *randomized* protocols. The only trivial upper bound on $C_R(f^{(\ell)})$ in this case is that for any (partial or non-partial) function f, $C_R(f^{(\ell)}) = O(\ell \cdot \log \ell \cdot C_R(f))$ (the $\log \ell$ factor seems to be needed, since we are required to have a "good" probability of success on *all* ℓ instances simultaneously). For explicit functions we can do much better: We consider the *identity* function (i.e., $ID : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ defined by $ID(x,y) = 1$ iff $x = y$). It is well known that $C_R(ID) = \Theta(\log n)$ [23]. We prove that $\overline{C}_R(ID) = O(1)$.

Organization: In section 2 the various notions of communication complexity and amortized communication complexity are defined. In section 3 we exhibit a partial function whose amortized communication complexity is smaller than its communication complexity. In section 4 we discuss the special case of one-way communication protocols. In section 5 we prove our lower bound on the amortized communication complexity, for the case of general protocols. In section 6 we present a function whose amortized communication complexity is smaller than its communication complexity, when randomized protocols are considered. Finally, in section 7 we mention some open problems.

2 Preliminaries

In this section we give formal definitions for the various notions of communication protocols and communication complexity used in this work.

Let $\mathcal{D}$ be a set, and let f be a *relation* defined over $\{0,1\}^n \times \{0,1\}^n$ such that for every $(x,y) \in \{0,1\}^n \times \{0,1\}^n$ it satisfies $\emptyset \neq f(x,y) \subseteq \mathcal{D}$. We say that f is *Boolean* if $\mathcal{D} = \{0,1\}$. We say that f is a *function*, if for every (x,y), $|f(x,y)| = 1$, and it is a *partial function* if for every (x,y) either $|f(x,y)| = 1$ or $f(x,y) = \mathcal{D}$.

Given a relation f and an integer $\ell \geq 1$, we define the relation $f^{(\ell)}$ over $(\{0,1\}^n)^\ell \times (\{0,1\}^n)^\ell$, with range $\mathcal{D}^\ell$ as follows:

$$f^{(\ell)}((x_1,\ldots,x_\ell),(y_1,\ldots,y_\ell)) \stackrel{\triangle}{=} \{(z_1,\ldots,z_\ell) \mid z_1 \in f(x_1,y_1),\ldots,z_\ell \in f(x_\ell,y_\ell)\}.$$

In what follows we define the communication complexity of relations of the form $f^{(\ell)}$. Note however that this covers the special case of $f^{(1)} \equiv f$.

Two parties P_1 and P_2 wish to compute a possible value of $f^{(\ell)}$ on their input. The party P_1 is given a $n\ell$-bit input x and the party P_2 is given a $n\ell$-bit input y. We interpret x (resp. y) as consisting of ℓ pieces (or *instances*) $x_1,\ldots,x_\ell$ (resp. $y_1,\ldots,y_\ell$) each of n bits. The parties exchange messages in rounds according to a *deterministic* protocol. That is, each message sent by a party P_i depends on its input, and the messages it received in previous rounds. The last message in the protocol is an ℓ-tuple $z = (z_1,\ldots,z_\ell)$ called the *output* of the protocol. We say that a protocol $\mathcal{F}$ computes the relation $f^{(\ell)}$ if for all inputs x and y the output z satisfies $z \in f^{(\ell)}(x,y)$.

The concatenation of all the messages exchanged in the protocol $\mathcal{F}$ on input (x,y) is denoted $\mathcal{F}(x,y)$. The *(deterministic)* communication complexity of the protocol $\mathcal{F}$, denoted $C(\mathcal{F})$, is the maximum

$|\mathcal{F}(x,y)|$ over all (x,y). The *(deterministic)* communication complexity of the relation $f^{(\ell)}$, denoted $C(f^{(\ell)})$, is the minimum of $C(\mathcal{F})$, over all deterministic protocols $\mathcal{F}$ computing $f^{(\ell)}$.

The *amortized communication complexity* of the relation f is defined as

$$\overline{C}(f) = \limsup_{\ell\to\infty} \frac{1}{\ell} C(f^{(\ell)}).$$

We sometimes restrict the discussion to *one-way* protocols. In such protocols the communication consists of a single message: P_1 sends a message to P_2 and P_2 has to compute the output. We denote by $C_1(\mathcal{F}), C_1(f)$ and $\overline{C}_1(f)$ the analogous of $C(\mathcal{F}), C(f)$ and $\overline{C}(f)$ for the case that only one-way protocols are considered.

We also consider *randomized* protocols, in which each of the parties has, in addition to its input, a string of random coins (the random strings of the two parties are independent). A randomized protocol $\mathcal{F}$ computes the relation $f^{(\ell)}$ if for every input (x,y) the output z of $\mathcal{F}$ satisfies $z \in f^{(\ell)}(x,y)$ with probability $\geq \frac{3}{4}$. The notions of $C_R(\mathcal{F}), C_R(f^{(\ell)})$ and $\overline{C}_R(f)$ are defined in a similar way, with respect to randomized protocols. That is, $C_R(\mathcal{F})$ is the maximal length of communication (over all inputs and all strings of random coins) in the protocol $\mathcal{F}$; $C_R(f^{(\ell)})$ is the minimum of $C_R(\mathcal{F})$ over all randomized protocols that compute the relation $f^{(\ell)}$; and $\overline{C}_R(f)$ equals $\limsup_{\ell\to\infty} \frac{1}{\ell} C_R(f^{(\ell)})$. We emphasize that the meaning of this definition is that when computing $f^{(\ell)}$ we require that with probability at least 3/4 the output is correct for *all* ℓ instances simultaneously.

It is also useful to consider a variant of the randomized model in which both parties have access to a *public* random string. The quantities $C_{pub}(f^{(\ell)})$ and $\overline{C_{pub}}(f)$ are defined in a similar way.

Finally, we give the definitions for the *nondeterministic* case. In a nondeterministic protocol for computing $f^{(\ell)}$ the parties are allowed to make "guesses" while choosing their messages. In any computation, the protocol gives either a correct value of $f^{(\ell)}(x,y)$ or "fail". The protocol is required to output a correct value of $f^{(\ell)}(x,y)$ in at least one computation on (x,y) (i.e., in this computation the output is correct for *all* ℓ instances). The nondeterministic complexity of a protocol $\mathcal{F}$, $C_N(\mathcal{F})$, is defined as the maximum over all (x,y) and over all computations ("guesses") of $\mathcal{F}(x,y)$ (note that for nondeterministic protocols $\mathcal{F}(x,y)$ is not unique). The measures $C_N(f^{(\ell)})$ and $\overline{C}_N(f)$ are defined with respect to nondeterministic protocols.

3 A Partial Function With a Low Amortized Complexity

In this section we prove that (deterministic) amortized communication complexity can be substantially lower than the corresponding communication complexity. We present a partial function f such that $C(f) = \Theta(\log n)$, while $\overline{C}(f) = O(1)$.

We start with the definition of f: Let $M = \{0,1,2,\ldots,m-1\}$. Let $t \geq 2$, be a parameter. The input of P_1 is S, a subset of M of size t (the length of this input is $n = t \cdot \log m$ bits). The input of P_2 is $x \in S$ (the length of this input is $\log m < n$ bits). The parties wish to compute the rank of x in the subset S (a number in the range $0,\ldots,t-1$). If $x \notin S$ then any output (in the range $0,\ldots,t-1$) is allowed. Orlitsky [16] showed that the communication complexity of this function is $C(f) = \Theta(\log t + \log\log m)$.

The protocols we present make use of the following set of hash-functions suggested by Fredman, Komlòs and Szemerèdi [5]: Let $p \simeq t^2 \log m$ be a prime. Define

$$H = \left\{ h : M \to \{0, 1, \ldots, 2t^2 - 1\} || \; h(x) = (ax \bmod p) \bmod 2t^2, \quad 1 \leq a \leq p-1 \right\}.$$

We say that $h \in H$ is *good* for a set $S \subset M$ if h is $1-1$ with respect to the elements of S. Otherwise, we say that h is *bad* for S. Fredman, Komlòs and Szemerèdi [5] proved the following property of these hash-functions:

Lemma 1: Let H be as above and let S be any subset of M of size t. Then, at least $\frac{1}{2}$ of the functions in H are good for S.

We start by presenting the following protocol from [16] that meets the lower bound for computing f on a *single* instance (S, x). This protocol (which uses the above H) has the advantage that an appropriate generalization of it gives the amortized result.

- P_1 finds a function $h \in H$ which is good with respect to S. It sends its name ($O(\log t + \log \log m)$ bits) to P_2.
- P_2 computes $h(x)$ and sends this value ($O(\log t)$ bits) to P_1.
- Since h is good with respect to S, then if $x \in S$ the value $h(x)$ determines x. (If $x \notin S$ then either $h(x) = h(s)$ for some $s \in S$ or not. For the correctness of the protocol it does not matter which is the case.) Now P_1 computes the value $f(S, x)$ and sends it to P_2 ($O(\log t)$ bits).

We now show how to generalize the protocol in order to efficiently compute the values $f(S_1, x_1)$, $f(S_2, x_2), \ldots, f(S_\ell, x_\ell)$ simultaneously. The main idea is formalized by the following claim:

Claim 1: Let H be as above and let $S_1, \ldots, S_\ell$ be any ℓ subsets of M of size t. Then, there exists a set L of $\log \ell + 1$ hash-functions $h_1, h_2, \ldots, h_{\log \ell + 1} \in H$ such that:

- For every j $(1 \leq j \leq \log \ell + 1)$, h_j is good with respect to at least $\frac{1}{2}$ of the S_i's for which $h_1, \ldots, h_{j-1}$ are all bad.

In particular, it follows that for every S_i $(1 \leq i \leq \ell)$ there exists at least one hash-function in L, denoted $h_{j(i)}$, such that $h_{j(i)}$ is good for S_i. The proof uses Lemma 1 and a simple counting argument:
Proof: We show how to construct L iteratively. In the j^{th} iteration we consider a matrix with all the subsets S_i for which $h_1, \ldots, h_{j-1}$ are bad as rows, and the hash functions in H as columns. The (S, h) entry in this matrix is 1 if h is good with respect to S, and 0 otherwise. By Lemma 1, at least half of the entries in every row are 1's. Therefore, there exists a column in which at least half of the entries are 1's. We take the corresponding hash-function to be h_j. □

The following protocol computes f on ℓ instances simultaneously:

- P_1 finds a set L of $\log \ell + 1$ hash functions as above, and sends the names of functions in L to P_2. In addition, for every $1 \leq i \leq \ell$, it sends the index $j(i)$.
- P_2 computes $h_{j(i)}(x_i)$, for every i, and sends it to P_1.
- Since $h_{j(i)}$ is good with respect to S_i, the party P_1 knows the value of x_i for every $1 \leq i \leq \ell$ and thus can compute $f(S_1, x_1), \ldots, f(S_\ell, x_\ell)$.

The correctness of the protocol is obvious. For every i such that $x_i \in S_i$ it computes the correct answer (and if $x_i \notin S_i$ then any answer is good). We now analyze its complexity:

Claim 2: The above protocol can be implemented so that the number of bits exchanged is $O(\ell \cdot \log t + \log \ell \cdot (\log t + \log \log m))$.

Proof: To specify the names of functions in L, P_1 uses $O(\log \ell \cdot (\log t + \log \log m))$ bits. In addition, for specifying *all* the indices $j(i)$, P_1 needs only $O(\ell)$ bits (which is better than the obvious $O(\ell \log \ell)$ bits). This is because h_1 is good for about $\frac{1}{2}$ of the sets, h_2 is good for about $\frac{1}{4}$ of the sets etc. Therefore, by using, say Huffman coding, we get that $O(\ell)$ bits are enough. In the second step P_2 sends the results of applying $h_{j(i)}$ on x_i, for every i, which requires $O(\ell \cdot \log t)$ bits. □

Take, for example, $t = 2$ and recall that in this case the length of the input satisfies $n = 2 \log m$, we get that the number of bits exchanged in this protocol is $O(\ell + \log \ell \cdot \log n)$. Thus, we proved the following theorem:

Theorem 1: There exists a (partial) function f with communication complexity $C(f) = \Theta(\log n)$, and amortized communication complexity $\overline{C}(f) = \limsup_{\ell \to \infty} \frac{1}{\ell} C(f^{(\ell)}) = O(1)$.

4 One-Way Communication

In this section we deal with one-way communication protocols. We show that if we restrict the discussion to the computation of relations using one-way protocols then we can still "save" bits by computing f on many instances simultaneously. In fact, the partial function f of the previous section yields such an example: take $t = 2$ and assume that $S_i = \{y_1^i, y_2^i\}$ where $0 \leq y_1^i < y_2^i \leq m - 1$. As stated before, $C(f) = \Theta(\log n)$ (and clearly $C_1(f) \geq C(f)$). On the other hand, a slight modification of the previous protocol gives $\overline{C}_1(f) = O(1)$: P_1 sends together with the list L of hash functions also $h_{j(i)}(y_1^i)$ and $h_{j(i)}(y_2^i)$ for $1 \leq i \leq \ell$. Now P_2 can decide whether $x_i = y_1^i$ or $x_i = y_2^i$.

On the other hand, we can prove that for every (partial) function f no more than $\log n$ bits can be saved: $\overline{C}_1(f) \geq C_1(f) - \log n - O(1)$. We start with a simple theorem, which claims that if f is a *non-partial* function then essentially nothing can be saved. That is, $\overline{C}_1(f) \cong C_1(f)$.

Theorem 2: Let f be a (non-partial) function defined on $\{0,1\}^n \times \{0,1\}^n$. Then, $C_1(f) - 1 \leq \overline{C}_1(f) \leq C_1(f)$.

Proof: Define the following relation on the inputs of P_1: $x_1 \sim x_2$ if $f(x_1, y) = f(x_2, y)$ for every y. Clearly $\sim$ is an equivalence relation. Denote by $Class(f)$ the number of equivalence classes of the $\sim$ relation. It can be easily verified that for computing f the party P_1 must use $Class(f)$ different messages (i.e, $C_1(f)$ is exactly $\lceil \log Class(f) \rceil$). This is true, since P_1 can send on input x the index of equivalence class for which x belongs. From this information P_2 can easily compute $f(x, y)$ (by choosing arbitrary x' from that equivalence class and computing $f(x', y)$). On the other hand, if for two inputs x, x' in different equivalence classes P_1 sends the same string then by the definition of the relation $\sim$ there exists y such that $f(x, y) \neq f(x', y)$. If P_2 holds y as his input then clearly the protocol is wrong for at least one of $f(x, y)$ or $f(x', y)$. Similar arguments show that for computing $f^{(\ell)}$ the party P_1 must use $Class(f^{(\ell)}) = Class(f)^{\ell}$ different messages. As this number of strings is enough, the theorem follows. □

The above example shows that this result cannot be extended to *partial* functions. The key point is that for partial functions $\sim$ is not necessarily an equivalence relation. However, in the following we show that this example is optimal in a sense. More precisely, we prove for every partial function f that $\overline{C}_1(f)$ cannot be smaller than $C_1(f)$ by more than an additive factor of $O(\log n)$.

Theorem 3: Let f be a (partial) function defined over $\{0,1\}^n \times \{0,1\}^n$. Then $C_1(f^{(2)}) \geq 2C_1(f) - \log n - O(1)$.

Proof: The idea of the proof is to reduce the problem of the one-way communication complexity of a function to an appropriate graph-coloring problem,[2] and then to use results of Linial and Vazirani [10] on this problem.

We construct a graph $G_f = (V, E)$ as follows: Each vertex corresponds to $x \in \{0,1\}^n$. There is an edge between x and x' if there exists y such that $f(x, y) \cap f(x', y) = \emptyset$ (this happens if and only if $|f(x, y)| = |f(x', y)| = 1$ and $f(x, y) \neq f(x', y)$). Intuitively, there is an edge between x and x' if P_2 should be able to distinguish between these two inputs in order to compute the output correctly when it holds input y. Similarly, we define a graph $G_{f^{(2)}}$; its vertices correspond to pairs $(x_1, x_2) \in \{0,1\}^n \times \{0,1\}^n$. There is an edge between $x = (x_1, x_2)$ and $x' = (x'_1, x'_2)$ if there exists $y = (y_1, y_2)$ such that $f^{(2)}(x, y) \cap f^{(2)}(x', y) = \emptyset$ (this happens if and only if either $|f(x_1, y_1)| = |f(x'_1, y_1)| = 1$ and $f(x_1, y_1) \neq f(x'_1, y_1)$, or if $|f(x_2, y_2)| = |f(x'_2, y_2)| = 1$ and $f(x_2, y_2) \neq f(x'_2, y_2)$).

The number of different messages used by the optimal one-way communication protocol for f is exactly the *chromatic number* of G_f (denoted $\chi(G_f)$): If we have a legal coloring of G_f then this coloring defines a one-way communication protocol for computing f: P_1 sends the color c of its input x. This color together with P_2's input y determine $z \in f(x, y)$. To see this, fix a y and consider all the vertices colored by c. If for all these vertices, the corresponding x satisfies $f(x, y) = \mathcal{D}$ then any $z \in \mathcal{D}$ will do. If for some x, $|f(x, y)| = 1$ then we take $z = f(x, y)$. For any other x' colored by c since there is no edge between x and x' it follows from the construction that $z \in f(x', y)$. On the other hand, every

[2] Similar reductions appear in [16, 21]. In these works the two parties have an input (x, y) in some domain $\mathcal{A}$ and P_1 has to transmit its input x to P_2. This problem corresponds in our setting to the problem of computing the specific function f which is defined as $f(x, y) = x$ if $(x, y) \in \mathcal{A}$ and $f(x, y) = \mathcal{D}$ otherwise.

protocol induces a legal coloring of G_f where the color of every x is the message P_1 sends on it. This is because for ever x, x' on which the same message m is sent by P_1 and for every y, there is a z that P_2 outputs. The correctness of the protocol guarantees that $z \in f(x,y)$ and $z \in f(x',y)$ and therefore $f(x,y) \cap f(x',y) \neq \emptyset$. Hence, there is no edge between x and x' so the coloring is legal. Similarly, the number of different messages used by the optimal one-way communication protocol for $f^{(2)}$ is exactly $\chi(G_{f^{(2)}})$ (again, fix (y_1, y_2) and argue about each coordinate separately the existence of z_1 and z_2 as needed).

Now, we define the *product* operation on graphs: Given $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ the vertices set of the product $G_1 \times G_2$ is $V_1 \times V_2$. The edge set includes all the edges $((v_1, v_2), (u_1, u_2))$ such that $(v_1, u_1) \in E_1$ *or* $(v_2, u_2) \in E_2$. (In the terminology of [10] this is called *inclusive*-product). It is easy to verify that $G_{f^{(2)}} = G_f \times G_f$.

Using this reduction to the graph-coloring problem we can now prove the theorem: it is enough to prove that for every f, $\chi(G_{f^{(2)}}) \geq \frac{\chi^2(G_f)}{cn}$, for some constant c. This is proved in [10, Theorem 1]. □

The statement of [10, Theorem 1] is more general than what we used and allows not only products of a graph G by itself but products of any two graphs. In particular, it says that for any two graphs G_1, G_2 such that $|V_1| \leq |V_2|$, the chromatic number satisfies $\chi(G_1 \times G_2) \geq \frac{\chi(G_1)\chi(G_2)}{c \log |V_1|}$. Thus, by the same proof as above, we get:

Theorem 4: Let $n \leq m$. Let f be a (partial) function defined over $\{0,1\}^n \times \{0,1\}^n$, and let g be a (partial) function defined over $\{0,1\}^m \times \{0,1\}^m$. Let $f \times g$ be defined in the obvious way over $(\{0,1\}^n \times \{0,1\}^m) \times (\{0,1\}^n \times \{0,1\}^m)$ (each party receives two instances; one is an n-bit string and the other is an m-bit string). Then, $C_1(f \times g) \geq C_1(f) + C_1(g) - \log n - O(1)$.

Therefore, we have

Corollary 5: Let f be a (partial) function defined over $\{0,1\}^n \times \{0,1\}^n$. Then $C_1(f) \geq \overline{C}_1(f) \geq C_1(f) - \log n - O(1)$.

Proof: The first inequality is obvious. For the second inequality, we will prove (by induction) that $C_1(f^{(\ell)}) \geq \ell C_1(f) - (\ell-1)\log n - (\ell-1)c$ (for some constant c), which implies the corollary. This is certainly true for $\ell = 1$. For a general ℓ we can write $C_1(f^{(\ell)}) = C_1(f \times f^{(\ell-1)})$. By Corollary 4 this is at least $C_1(f) + C_1(f^{(\ell-1)}) - \log n - c$. Now, by the induction hypothesis $C_1(f^{(\ell-1)}) \geq (\ell-1)C_1(f) - (\ell-2)\log n - (\ell-2)c$ which gives us what we need. □

For additional examples of partial functions with $\overline{C}_1(f)$ significantly smaller than $C_1(f)$, we show that for every graph G with 2^n vertices there exists a (partial) function f such that $G = G_f$. Label the vertices of G by strings in $\{0,1\}^n$ and define a function f as follows: for every x, $f(x,x) = 1$. For every edge $(x,y) \in E$ define $f(x,y) = 0$. For all the other pairs $f(x,y) = \mathcal{D}$. It can be easily verified that $G = G_f$. This implies that from every graph G with 2^n vertices, such that $\chi(G \times G) \cong \frac{\chi^2(G)}{cn}$, we can construct a partial function f such that $C_1(f^{(2)}) \cong 2C_1(f) - \log n - O(1)$. Examples of such graphs are given in [10, Theorem 2].

5 Lower Bound for General Protocols

In order to prove lower bounds on $\overline{C}(f)$ for a specific relation f, we may use traditional techniques. For example, consider the *identity* function (i.e., $ID(x,y)$ equals 1 if $x=y$, and 0 otherwise). It is easy to verify that $\overline{C}(ID) = C(ID) = n$ (as in [23]). In this section we give a *general* lower bound on $\overline{C}(f)$ in terms of $C(f)$, for any *(non-partial) boolean function* f.

To this end, we first discuss the amortized *non-deterministic* communication complexity of relations. We start with some definitions and notations that are used in the proof. Given a relation f defined over $\{0,1\}^n \times \{0,1\}^n$, and $\ell \geq 1$, we denote by $M_{f(\ell)}$ the matrix representing the relation $f^{(\ell)}$. That is, each row of $M_{f(\ell)}$ corresponds to an input $x = (x_1, x_2, \ldots, x_\ell)$ of P_1, and each column corresponds to an input $y = (y_1, y_2, \ldots, y_\ell)$ of P_2. The entry (x,y) of $M_{f(\ell)}$ contains the set $f(x,y)$ (a subset of $\mathcal{D}^\ell$). A *monochromatic rectangle* of $M_{f(\ell)}$ is a set $R = R_x \times R_y \subseteq \{0,1\}^n \times \{0,1\}^n$ such that we can associate with R an output vector $z_R \in \mathcal{D}^\ell$, in a way that every input $(x,y) \in R$ satisfies $z_R \in f(x,y)$. We denote by $N(f^{(\ell)})$ the minimal number of monochromatic rectangles needed to cover (possibly with overlaps) all the entries of $M_{f(\ell)}$. Since any nondeterministic protocol for computing $f^{(\ell)}$ induces such a cover, $\log N(f^{(\ell)}) \leq C_N(f^{(\ell)})$. The next theorem claims that $N(f^{(2)})$ cannot be much smaller than $N^2(f)$.

Theorem 6: Let f be a relation defined over $\{0,1\}^n \times \{0,1\}^n$. Then, for some constant c,

$$N(f^{(2)}) \geq \frac{N^2(f)}{c \cdot n}.$$

For the proof of this theorem, we need the following claim, provided by the proof of [10, Theorem 1]:

Claim 3: Let A be an $\ell \times d$ matrix whose entries assume k values and such that $\ell \leq d$. Let k_1 be the minimal size of a set $T \subseteq \{1,2,\ldots k\}$ that covers all the rows of A. That is, for every row i there exists a column j such that the value $A_{i,j}$ belongs to T. Similarly, let k_2 be the minimal size of a set that covers all the columns. Then $k_1 \cdot k_2 \leq c' \cdot \log \ell \cdot k$.

Proof: Consider an optimal cover of $M_{f(2)}$, with $k = N(f^{(2)})$ monochromatic rectangles, denoted by $R_1, R_2, \ldots, R_k$. We show how to cover M_f with m monochromatic rectangles, where $m^2 \leq c \cdot n \cdot N(f^{(2)})$ for some constant c. This implies that $N^2(f) \leq c \cdot n \cdot N(f^{(2)})$.

Consider the following $2^{2n} \times 2^{2n}$ matrix A (this is *not* $M_{f(2)}$): each row of A corresponds to an input (x_1, y_1) and each column to an input (x_2, y_2). Every entry $((x_1,y_1),(x_2,y_2))$ of A contains an element t in $\{1,2,\ldots k\}$ such that $((x_1,x_2),(y_1,y_2))$ belongs to R_t. (If $((x_1,x_2),(y_1,y_2))$ belongs to more than one rectangle, then we choose one of them arbitrarily). Apply Claim 3 to the matrix A described above, and assume without loss of generality that $k_1 \leq k_2$; we get that $k_1^2 \leq c \cdot n \cdot k$. Let T be a set of k_1 values that covers the rows. We now prove that this implies that M_f can be covered with k_1 monochromatic rectangles.

Associate with every entry (x,y) in M_f an element of T that appears in the row (x,y) of A (if there is more than one possibility, then choose one arbitrarily). Now we extend this to (possibly overlapping) rectangles in the obvious way. Namely, for every $t \in T$ the rectangle R'_t includes every (x, y) with value t, and if (x, y) and (x', y') are in R'_t then also (x', y) and (x, y') are in R'_t.

Clearly, these are k_1 rectangles and they cover M_f. What we still have to prove is that any such rectangle R'_t is monochromatic. That is, there exists a z such that for all $(x, y) \in R'_t$ it satisfies $z \in f(x, y)$. By the construction, if (x, y) and (x', y') both have the value t, then there exist x_2, y_2, x'_2 and y'_2 such that both $((x, x_2), (y, y_2))$ and $((x', x'_2), (y', y'_2))$ belong to R_t. Since R_t is monochromatic, we can associate with R_t a vector (z_1, z_2) with whom all pairs in R_t "agree". This, in particular, implies that $z_1 \in f(x, y)$ and $z_1 \in f(x', y')$. In addition, since R_t is a rectangle it also contains $((x, x_2), (y', y'_2))$ and $((x', x'_2), (y, y_2))$ which implies that also $z_1 \in f(x, y')$ and $z_1 \in f(x', y)$. Therefore R'_t is monochromatic.

To conclude, we can cover M_f with no more than $\sqrt{c \cdot n \cdot N(f^{(2)})}$ monochromatic rectangles, which completes the proof of the theorem. □

Again, the above theorem (using [10]) can be generalized to prove the following:

Theorem 7: Let $n \leq m$. Let f be a relation defined over $\{0,1\}^n \times \{0,1\}^n$, and let g be a relation defined over $\{0,1\}^m \times \{0,1\}^m$. Then,

$$N(f \times g) \geq \frac{N(f) \cdot N(g)}{c \cdot n}.$$

It follows that $N(f^{(\ell)}) \geq \frac{N^\ell(f)}{(cn)^{\ell-1}}$. We now focus our attention on the case where f is a (non-partial) function. For this case we can apply known relations between deterministic and nondeterministic communication complexity [1]:

Claim 4: Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, be a (non-partial) function. Then, $C(f) \leq 2 \log^2 N(f)$.

Using Theorem 6 and Claim 4 we get the desired lower bound:

Corollary 8: Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, be a (non-partial) function. Then, $C(f) \geq \overline{C}(f) \geq \sqrt{C(f)/2} - \log n - O(1)$.

Proof: Clearly, $C(f) \geq \overline{C}(f)$. For the other inequality we write

$$\begin{aligned} C(f^{(\ell)}) &\geq \log N(f^{(\ell)}) \\ &\geq \ell \log N(f) - \ell \log n - O(\ell) \\ &\geq \ell \cdot \left(\sqrt{C(f)/2} - \log n - O(1)\right) \end{aligned}$$

By the definition of $\overline{C}(f)$ the result follows. □

We do not know how to extend the above result to general relations or even to partial functions. Our proof method fails in these cases as the gap between deterministic and nondeterministic complexity may be exponential (examples of such partial functions can be constructed based on results is [19]).

6 A Function With Low Amortized Randomized Complexity

In this section we consider amortized *randomized* communication complexity. Clearly, for every relation f, $\overline{C}_R(f) \leq \overline{C}(f) \leq n$. However, unlike the deterministic case, we do not know whether $\overline{C}_R(f) \leq C_R(f)$ for all relations f. If f is a (partial) function then $\frac{1}{\ell} \cdot C_R(f^{(\ell)})$ is $O(C_R(f) \cdot \log \ell)$, as we can compute f separately for each instance. We do this $O(\log \ell)$ times and take the majority as the output (the $O(\log \ell)$ factor seems to be needed, since we require the protocols for computing $f^{(\ell)}$ to be correct with high probability on *all* ℓ instances simultaneously). For specific relations we can do much better. We consider the *identity* function $ID(x, y)$. It is known that $C_R(ID) = \Theta(\log n)$ (see [23]). We show that the amortized complexity of ID, with respect to randomized protocols, is $\overline{C}_R(ID) = O(1)$. Moreover, the probability of error in our protocol for ID is much less than a constant: it goes down exponential with $\sqrt{\ell}$. (This can actually be improved to exponential in ℓ.)

For simplifying the presentation of the protocols we first assume that the two parties have a way of agreeing on a random string with no cost in communication. This can be thought as protocols in the *public-coins* model. After presenting the protocols we describe how the parties can agree on such strings while preserving both the communication complexity and the correctness of the protocols.
The following protocol computes the identity function on a *single* pair of inputs, (x, y):

- The parties agree on a random string $b \in \{0,1\}^n$.
- P_1 computes $\langle b, x\rangle$, the inner product of b and $x \pmod 2$, and P_2 computes $\langle b, y\rangle$.
- The parties exchange the bits $\langle b, x\rangle$ and $\langle b, y\rangle$. If the bits are equal they output "equal" $(x = y)$, otherwise they output "not-equal" $(x \neq y)$.

The number of bits exchanged in the protocol is $O(1)$. If $x = y$ it is always correct, while if $x \neq y$ it is correct with probability $\frac{1}{2}$ (which can be improved to any other constant advantage while preserving the $O(1)$ complexity).

Suppose now that the two parties P_1 and P_2 wish to compute the identity function on ℓ input pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)$. Consider the protocol where P_1 and P_2 amortize the first step in the above protocol while exchanging the bits $\langle b, x_i\rangle$ and $\langle b, y_i\rangle$, for all $1 \leq i \leq \ell$. Such a protocol gives a "good" success probability for computing each of the $f(x_i, y_i)$ separately, while what we want is a "good" probability of computing f on all ℓ instances simultaneously. A possible idea is to decrease the error probability on each (x_i, y_i) to $\frac{1}{poly(\ell)}$ by choosing $k = O(\log \ell)$ vectors b_i's. Formally,

Protocol *multi_compare*:

1. The parties agree on k random strings $b_1, b_2, \ldots, b_k \in \{0,1\}^n$.
2. For $i = 1, 2, \ldots, k$:
   (a) P_1 computes $u_i = \langle b_i, x_1\rangle, \langle b_i, x_2\rangle, \ldots, \langle b_i, x_\ell\rangle$.
   P_2 computes $v_i = \langle b_i, y_1\rangle, \langle b_i, y_2\rangle, \ldots, \langle b_i, y_\ell\rangle$.

(b) The parties exchange the vectors u_i and v_i (each of them is an ℓ-bit string) using a procedure *exchange*(u_i, v_i).

(c) For $1 \leq j \leq \ell$, if the j-th bits of u_i and v_i are different then the parties P_1 and P_2 replace x_j and y_j (respectively) by $x_j = y_j = 0^n$, where 0^n denotes a string of n zeros. (The motivation for this step will become clear while making the analysis below.)

3. The output for the j-th pair (x_j, y_j) is "equal" $(x_j = y_j)$ if and only if for every $1 \leq i \leq k$ the j-th bits of u_i and v_i are equal.

The probability that the protocol will err on any pair is at most $\ell 2^{-k}$. The only problem with this protocol is that if $k = O(\log \ell)$, and if the procedure *exchange*, in step (2b), is implemented in a naive way (i.e., P_1 sends u_i to P_2, and P_2 sends v_i to P_1) then the communication complexity of the protocol is $O(\ell \log \ell)$ (i.e., $O(\log \ell)$ invocations of the procedure *exchange*, each requires $O(\ell)$ bits). This complexity is more than what we are aiming for.

The main idea for reducing the communication complexity is the following: even if a vector b_i does not recognize all the pairs such that $x_j \neq y_j$, we expect that it does recognize a constant fraction of them. At each time that the parties recognize such a pair, they replace it by $x_j = y_j = 0^n$ (step (2c)), therefore the expected Hamming distance between the vectors u_i and v_i in the above protocol decreases from round to round. We present an implementation of the procedure *exchange*(u, v) that uses this property: It enables the parties to exchange u_i and v_i (step (2b)) in a cost that depends on the Hamming distance between the vectors; Namely, the smaller the Hamming distance, the lower the communication complexity. This will give us the desired complexity.

We start with a simple case where the parties P_1 and P_2 receive, in addition to the input vectors $u, v \in \{0,1\}^\ell$ respectively, a bound d such that u and v are promised to be at Hamming distance at most d. The following *deterministic* protocol *exchange*$_d(u, v)$ enables each party to learn the value of the other party, by exchanging $O(\log \binom{\ell}{d})$ bits (we assume that $d \leq \ell/4$, otherwise the parties simply exchange their inputs). The protocol is due to Brandman, El-Gamal and Orlitsky (in [15]), Witsenhausen and Wyner [22] and Karchmer and Wigderson [9]:

Protocol *exchange*$_d(u, v)$**:**

- The parties consider the graph with 2^ℓ nodes corresponding to the strings in $\{0,1\}^\ell$ and edges between nodes which are at Hamming distance at most $2d$. The parties fix a coloring of the graph. (An effective coloring can be constructed using linear error correcting codes such as BCH.)

- P_1 sends P_2 the color of u and P_2 sends the color of v under the coloring. Since the Hamming distance between u and v is bounded by d and since there is at most one member of every color class at distance d from v (as we have a legal coloring of vectors with Hamming distance $\leq 2d$) then P_2 can identify u. Similarly, P_1 can identify v.

The degree of every node in this graph is less than $2d \cdot \binom{\ell}{2d}$. Therefore there exists a coloring of the graph with that many colors. Since the communication in this protocol consists of names of colors then $O(\log \binom{\ell}{2d}) = O(\log \binom{\ell}{d})$ bits are communicated.

The protocol *exchange*$_d$ above assumes that we have an upper bound on the Hamming distance between u and v. In our case (step (2b) of the protocol *multi_compare*), a good bound on the distance between u_i and v_i is not knows. If we use the protocol *exchange*$_d$ with the wrong bound d then it may fail. Therefore, we generalize the protocol *exchange*$_d$ to a (randomized) protocol *exchange* (which in fact uses *exchange*$_d$ as a procedure). This generalized protocol can work in the case that a good bound d is not known. The expected number of bits exchanged is still only $O(\log \binom{\ell}{\Delta})$ bits, where Δ is the *actual* Hamming distance between u and v. We use this protocol to implement step (2b) of the protocol *multi_compare*.

Protocol *exchange*(u, v):

1. The parties agree on k random "test strings" $c_1, c_2, \ldots, c_k \in \{0,1\}^\ell$.
2. For $d = 2^1, 2^2, 2^3, \ldots, 2^{\log \ell}$
   (a) P_1 and P_2 engage in *exchange*$_d(u, v)$. Denote the output of P_1 by v' and the output of P_2 by u'.
   (b) *Test step:* P_1 and P_2 test whether $u' = u$ by comparing the inner product of the "test strings" $c_1, c_2, \ldots c_k$ with u and u'; this is done in a bit by bit manner, quitting early if they discover an error and going to the next d. If all the k bits are equal the protocol terminates (i.e., the parties assume that d is correct, and therefore $u' = u$ and $v' = v$).

By the analysis of the protocol *exchange*$_d$ made above, the number of bits required in step (2a) is $O(\log \binom{\ell}{d})$ if $d \leq \ell/4$ and $O(\ell)$ otherwise. If $u' \neq u$ then the *expected* number of bits exchanged in the test step is $O(1)$. If $u' = u$ then the number of bits exchanged in the test step is k, however this happens only once (note that once we reach d such that $d \geq \Delta$ then the deterministic sub-protocol *exchange*$_d$ (step (2a)) always stops and with the correct values). Therefore, the expected number of bits communicated is $O(k + \sum_{i=1}^{\log \Delta} \log \binom{\ell}{2^i})$. For computing the overall number of bits communicated we need the following technical claim:

Claim 5: For any $D \leq \ell/2$, $\displaystyle\sum_{i=1}^{\log D} \log \binom{\ell}{2^i}$ is $O(\log \binom{\ell}{D})$.

Proof: We claim that for all $1 \leq k \leq \ell/8$ we have $\binom{\ell}{2k} \geq \binom{\ell}{k}^{3/2}$: we know that

$$\frac{\binom{\ell}{2k}}{\binom{\ell}{k}} = \frac{(\ell-k)(\ell-k-1)\cdots(\ell-2k+1)}{2k(2k-1)\cdots(k+1)} \geq \frac{\ell(\ell-1)\cdots(\ell-k+1)}{2k(2k-1)\cdots(k+1)2^k} \geq \frac{\ell(\ell-1)\cdots(\ell-k+1)}{k!\cdot 2^k \cdot 2^k} = \frac{\binom{\ell}{k}}{4^k}.$$

In addition, we have that $\binom{\ell}{k} \geq (\frac{\ell}{k})^k \geq 8^k$ (for the last inequality we use the assumption $k \leq \ell/8$) and hence $\binom{\ell}{2k} \geq \binom{\ell}{k}^2/4^k \geq \binom{\ell}{k}^{3/2}$. Therefore every term (except perhaps the last two) in the sum $\sum_{i=1}^{\log D} \log \binom{\ell}{2^i}$ is at least at 3/2 times the preceding term, and the sum is bounded by some constant times the largest term which is $\log \binom{\ell}{D}$. □

Therefore the expected number of bits exchanged is $O(k+\log\binom{\ell}{\Delta})$ if $\Delta \leq \frac{\ell}{2}$ and $O(k+\log\binom{\ell}{\ell/2})$ otherwise. The error probability in each round is bounded by 2^{-k} and therefore the total error probability is bounded by $\log \ell \cdot 2^{-k}$.

As mentioned, we now use the procedure *exchange* described above to implement step (2b) of the protocol *multi_compare*. The analysis of the protocol *multi_compare* is as follows: let D_i be the random variable counting the number of indices $1 \leq j \leq \ell$ such that $\langle b_i, x_j\rangle \neq \langle b_i, y_j\rangle$ but $\langle b_1, x_j\rangle = \langle b_1, y_j\rangle, \ldots, \langle b_{i-1}, x_j\rangle = \langle b_{i-1}, y_j\rangle$. In other words D_i is distance between u_i and v_i (recall, that if for some $i' < i$, $\langle b_{i'}, x_j\rangle \neq \langle b_{i'}, y_j\rangle$ then both x_j and y_j are replaced by 0^n and therefore the j-th coordinate of u_i and v_i must be the same).

The expected number of bits exchanged in an execution of the protocol given that $D_i = d$ is bounded by $c \cdot \sum_{i=1}^{k}\left(k + \log\binom{\ell}{d}\right)$ for some constant c. For any set of inputs, the expected value of D_{i+1} given that $D_i = d$ and that procedure exchange does not fail is bounded by $1/2d$. Therefore, conditioned on that exchange does not fail, $E[D_i] \leq \ell \cdot 2^{-i}$ and for all $0 \leq m \leq i$ we have $Prob[D_i > \ell 2^{m-i}] < 2^m$. If exchange does fail at some round, then at most $\ell \cdot k$ bits are exchanged as result. The expected total number of bits exchanged is therefore at most

$$E[c \cdot \sum_{i=1}^{k}\left(k + \log\binom{\ell}{D_i}\right)] + Prob[\text{ exchange fails}] \cdot \ell \cdot k \leq c \cdot k^2 + k \cdot \ell \log \ell 2^{-k} + c \cdot \sum_{i=1}^{k} E[\log\binom{\ell}{D_i}]$$
$$\leq \quad c \cdot k^2 + k \cdot \ell \log \ell 2^{-k} + c\sum_{i=1}^{k}\sum_{m=0}^{i-1} 2^{-m}\log\binom{\ell}{\ell 2^{m-i}} \leq c \cdot k^2 + k \cdot \ell \log \ell 2^{-k} + c\sum_{s=1}^{k}\log\binom{\ell}{\ell 2^{-s}}\sum_{t=0}^{k} 2^{-t}$$
$$\leq \quad c \cdot k^2 + k \cdot \ell \log \ell 2^{-k} + 2c\sum_{s=1}^{k}\log\binom{\ell}{\ell 2^{-s}}$$

which by Claim 5 is $O(k^2 + k \cdot \ell \log \ell 2^{-k} + \ell)$. If k is $\Theta(\sqrt{\ell})$, then the expected number of bits communicated is $O(\ell)$.

As for correctness, if $x_j \neq y_j$ then with probability at most 2^{-k} we have that for all $1 \leq i \leq k$, $\langle b_i, x_j\rangle = \langle b_i, y_j\rangle$. Therefore the probability that for some j, and for all $1 \leq i \leq k$ we have that $\langle b_i, x_j\rangle = \langle b_i, y_j\rangle$ is bounded by $\frac{\ell}{2^k}$. In addition there is the probability of failure each time we invoke $exchange(u, v)$. This probability is at most $\frac{\log \ell}{2^k}$. Thus the probability of error in our protocol is bounded by $\frac{\ell + k\log\ell}{2^k}$. Therefore, if $k = \sqrt{\ell}$, then the probability of error is at most $2^{-\Omega(\sqrt{\ell})}$. To summarize we have

Lemma 2: The protocol described above computes in the public coins model the identity function on ℓ instances while maintaining that the number of bits communicated is $O(\ell)$ and the probability of error on any instance is at most $2^{-\Omega(\sqrt{\ell})}$.

Newman [14] has considered the public-coins model vs. the private coins model. He showed that $C_R(f) = O(C_{pub}(f) + \log n)$, which in particular implies

$$C_R(f^{(\ell)}) = O(C_{pub}(f^{(\ell)}) + \log \ell n).$$

Clearly,

$$C_R(f^{(\ell)}) \geq C_{pub}(f^{(\ell)}) = O(\ell \cdot \log \ell \cdot C_{pub}(f)).$$

All together we have the following:

Theorem 9: Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a (partial) function. Then

1. $\overline{C}_R(f) = \Theta(\overline{C_{pub}}(f))$.
2. For every sufficiently large ℓ, $\frac{1}{\ell} \cdot C_R(f^{(\ell)}) = O(\log \ell \cdot C_{pub}(f))$.

In particular, this Theorem together with Lemma 2 give:

Theorem 10: $\overline{C}_R(ID) = O(1)$.

Note however that Newman's method is non-constructive in nature. In the rest of this section we turn to the question of constructively converting the protocols described above to run in the private coins model. We describe a way for the parties to agree on the random strings (i.e., the b_i's and c_i's) with not much additional cost in the communication.

We first describe how to agree on a single string b_i. A collection of vectors $B_m \subset \{0,1\}^m$ is called *ε-biased* if every $x \in \{0,1\}^m$ satisfies $Pr_{b \in B_m}(\langle b, x\rangle = 0) = \frac{1}{2} \pm \varepsilon$. In [13] and [3] the existence and construction of such sets which are of size polynomial in m (and thus each of them can be represented by $O(\log m)$ bits) is shown. For our purposes it is sufficient to take ε to be say 1/4. Fix B_n and B_ℓ, two ε-biased probability spaces. P_1 selects $b_i \in B_n$ by choosing $\log |B_n|$ random bits and sends those bits to P_2. They can both compute b_i. Clearly, if $x = y$ then $\langle b_i, x\rangle = \langle b_i, y\rangle$ while if $x \neq y$ then $\langle b_i, x\rangle \neq \langle b_i, y\rangle$ with probability at least 1/4. In order to pick k strings $b_1, b_2, \ldots, b_k$ the party P_1 samples k times B_n using $O(k \cdot \log n)$ bits altogether. He sends those bits to P_2. The probability that *multi_compare* errs is at most $\ell \cdot (\frac{3}{4})^k$ and the expectation of D_i is at most $\ell \cdot (\frac{3}{4})^{-i}$.

The strings $c_1, c_2, \ldots, c_k$ are selected similarly from B_ℓ using $O(k \log \ell)$ bits. Note however that step (1) in protocol *exchange*(u, v) should not be repeated, i.e. $c_1, c_2, \ldots c_k$ are chosen once and for all at the beginning of the protocol *multi_compare*. In the *public* coins model there is no reason for doing that; we can allow the parties to use new strings $c_1, \ldots, c_k$ each time that step (2b) of *exchange* is executed. However, the fixed choice of $c_1, \ldots, c_k$ makes the conversion to the *private* coins model easier. Choosing the c_i's once and for all, using ε-biased spaces, has the property that in protocol *exchange*(u, v), in case $u' \neq u$ the expected number of bits exchanged is $O(1)$. Also the probability of error is at most $(\frac{3}{4})^k$. Thus the analysis of Lemma 2 still applies and we get that the probability of error is at most $2^{-\Omega(\sqrt{\ell})}$ and the number of bits exchanged is $O(\ell + \sqrt{\ell} \log n)$.

For values of ℓ which are around $\log n$ we would like to replace the term $\sqrt{\ell} \log n$ with $\sqrt{\ell} + \log n$. This can be done by sampling the b_i's via a random walk in an expander à la Ajtai, Komlòs and Szemerèdi [2] (in such a case the b_i's are not independent): The elements of B_n are mapped to nodes of a constant degree expander G. Then, a random walk of length k in G is generated, and the vectors $b_1, b_2, \ldots, b_k$ are the vectors corresponding to the nodes of the walk. The number of bits required to specify the walk is $O(\log |B_n| + k)$ which is $O(\log n + k)$. (See e.g. [13] for details.) As before, P_1 selects the random bits and sends them to P_2, so that they both agree on the same sequence. If $x \neq y$ then the probability that $\langle b_i, x\rangle = \langle b_i, y\rangle$ for all $1 \leq i \leq k$ goes down exponentially in k. The strings $c_1, c_2, \ldots, c_k$ are selected similarly in B_ℓ using $O(k + \log \ell)$ bits.

To conclude, we have a randomized protocol, in the *private* coins model, for computing the identity function on ℓ instances with probability of error at most $2^{-\Omega(\sqrt{\ell})}$ and expected complexity of $O(\ell + \log n)$, which is $O(\ell)$, for ℓ sufficiently large. With a "small" additional error the protocol can be converted to a protocol that uses $O(\ell)$ bits in the *worst case*. This gives a constructive proof for Theorem 10.

7 Open Problems

We conclude this work by mentioning some open problems:

- In [7] it was conjectured that for any relation f, the communication complexity, $\overline{C}(f)$, can not be smaller than $C(f)$ by more than an additive factor of $O(\log n)$. The examples given in our paper do not contradict this conjecture. On the other hand, according to the best lower bound we are able to prove (Corollary 8), even for (non-partial) functions a quadratic gap between $C(f)$ and $\overline{C}(f)$ is possible (and the gap may be even bigger for general relations). Therefore, the main open problem is to try to close this gap by either improving the lower bound (in particular, trying to extend it to relations), or presenting relations with more than $O(\log n)$ difference between $C(f)$ and $\overline{C}(f)$. (Presenting other relations with $O(\log n)$ difference between $C(f)$ and $\overline{C}(f)$ may also be interesting).

- Another open problem is trying to achieve similar lower bounds for the *randomized* model. Namely, can one prove a lower bound on $\overline{C}_R(f)$ in terms of $C_R(f)$? In the randomized case, it is also not known whether $\overline{C}_R(f) \leq C_R(f)$, for every relation f.

- In the case of partial functions f, one can consider a weaker definition for the correctness of a protocol for computing $f^{(\ell)}$: The protocol is required to succeed in computing $f^{(\ell)}(\vec{x}, \vec{y})$ only if for all i $(1 \leq i \leq \ell)$ we have $|f(x_i, y_i)| = 1$ (otherwise, there is no requirement). In such a model we think of inputs such that $f(x_i, y_i) = \mathcal{D}$ as "illegal". Clearly, proving upper bounds under this definition is easier, while proving lower bounds is harder.

8 Acknowledgments

We thank Mauricio Karchmer and Avi Wigderson for raising the question and for helpful discussions, and Amos Beimel, Benny Chor, Alon Orlitsky and Steve Ponzio for many interesting comments on earlier versions of this paper. Finally, we thank an anonymous referee for his very helpful comments and criticism.

References

[1] Aho A., J. Ullman, and M. Yannakakis, "On Notions of Information Transfer in VLSI Circuits", *Proc. of 15th ACM Symposium on Theory of Computing,* 1983, pp. 133-139.

[2] Ajtai M., J. Komlòs, and E. Szemerèdi, "Deterministic Simulation in LOGSPACE", *Proc. of 19th ACM Symposium on Theory of Computing*, 1987, pp. 132-140.

[3] Alon N., O. Goldreich, J. Håstad and R. Peralta, "Simple construction of almost k-wise independent random variables", *Proc. of 31st IEEE Symposium on Foundations of Computer Science* 1990, pp. 544-553.

[4] Bshouty, N. H., "On The Extended Direct Sum Conjecture", *Proc. of 21th ACM Symposium on Theory of Computing*, 1989, pp. 177-185.

[5] Fredman M., J. Komlòs, and E. Szemerèdi, "Storing A Sparse Table with O(1) Access Time", *Journal of the Association for Computing Machinery*, Vol 31, 1984, pp. 538–544.

[6] Galibati G., and M. J. Fischer, "On The Complexity of 2-Output Boolean Networks", *Theoretical Computer Science*, Vol 16, 1981, pp. 177–185.

[7] Karchmer M., R. Raz, and A. Wigderson, "On Proving Super-Logarithmic Depth Lower Bounds via the Direct Sum in Communication Complexity", *Proc. of 6th IEEE Structure in Complexity Theory*, 1991, pp. 299–304.

[8] Karchmer M., and A. Wigderson, "Monotone Circuits for Connectivity Require Super-Logarithmic Depth", *Proc. of 20th ACM Symposium on Theory of Computing*, 1988, pp. 539-550.

[9] Karchmer M., and A. Wigderson, private communication.

[10] Linial N., and U. Vazirani, "Graph Products and Chromatic Numbers", *Proc. of 30th , IEEE Symposium on Foundations of Computer Science,* 1989, pp. 124-128.

[11] Lovász, L., "Communication Complexity: A Survey", in *Paths, Flows, and VLSI Layout,* edited by B. H. Korte, Springer Verlag, Berlin New York, 1990.

[12] Mehlhorn, K., and E. Schmidt, "Las-Vegas is better than Determinism in VLSI and Distributed Computing", *Proc. of 14th ACM Symposium on Theory of Computing,* pp. 330-337, 1982.

[13] Naor J., and M. Naor, "Small-Bias Probability Spaces: Efficient Constructions and Applications", *SIAM J. on Computing*, vol 22, 1993, pp. 838–856.

[14] Newman, I., "Private vs. Common Random Bits in Communication Complexity", *Information Processing Letters* 39, 1991, pp. 67-71.

[15] Orlitsky, A., "Communication Issues in Distributed Communication", PhD thesis, Stanford University, 1986.

[16] Orlitsky, A., "Two Messages are Almost Optimal for Conveying Information", *Proc. of 9th Symposium on Principles of Distributed Computing*, 1990, pp. 219-232.

[17] Paul W., "Realizing Boolean Function on Disjoint Sets of Variables", *Theoretical Computer Science* 2, 1976, pp. 383-396.

[18] Raz R., and A. Wigderson, "Monotone Circuits for Matching Require Linear Depth", *Proc. of 22nd ACM Symposium on Theory of Computing*, 1990, pp. 287-292.

[19] Razborov A., "Applications of Matrix Methods to the Theory of Lower Bounds in Communication Complexity", *Combinatorica*, 10(1), 1990, pp. 81-93.

[20] Q. F. Stout, "Meshes with multiple buses", *Proc. of 27th IEEE Symposium on Foundations of Computer Science,* 1986, pp. 264-273.

[21] Witsenhausen, H. S., "The Zero-Error Side Information Problem and Chromatic Numbers", *IEEE Transactions on Information Theory*, 1976, pp. 592-593.

[22] Witsenhausen, H. S. and A. D. Wyner, "Interframe Coder for Video Signals", *United States Patent number 4,191,970*, 1980.

[23] Yao, A. C., "Some Complexity Questions Related to Distributed Computing", *Proc. of 11th ACM Symposium on Theory of Computing*, 1979, pp. 209-213.