COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **19** (2) (1999) 241–254

THE LINEAR-ARRAY CONJECTURE IN COMMUNICATION COMPLEXITY IS FALSE*

EYAL KUSHILEVITZ, NATHAN LINIAL† and RAFAIL OSTROVSKY

Received March 1, 1996

A *linear array* network consists of $k+1$ processors $P_0, P_1, \ldots, P_k$ with links only between P_i and P_{i+1} $(0 \le i < k)$. It is required to compute some boolean function $f(x,y)$ in this network, where initially x is stored at P_0 and y is stored at P_k. Let $D_k(f)$ be the (total) number of bits that must be exchanged to compute f in worst case. Clearly, $D_k(f) \le k \cdot D(f)$, where $D(f)$ is the standard two-party communication complexity of f. Tiwari proved that for almost all functions $D_k(f) \ge k(D(f) - O(1))$ and conjectured that this is true for *all* functions.

In this paper we disprove Tiwari's conjecture, by exhibiting an infinite family of functions for which $D_k(f)$ is essentially at most $\frac{3}{4}k \cdot D(f)$. Our construction also leads to progress on another major problem in this area: It is easy to bound the two-party communication complexity of any function, given the least number of monochromatic rectangles in any partition of the input space. How tight are such bounds? We exhibit certain functions, for which the (two-party) communication complexity is *twice* as large as the best lower bound obtainable this way.

1. Introduction

The *linear array* network consists of $k+1$ processors $P_0, P_1, \ldots, P_k$ with links only between P_i and P_{i+1} $(0 \le i < k)$. The processors are to compute a boolean function $f(x,y)$ where initially x is stored in processor P_0 and y is stored in processor P_k. The complexity of a protocol is the *total* number of bits exchanged on all links at worst case. Let $D_k(f)$ be the (worst case) complexity of (the best protocol for) f. Obviously, $D_k(f) \le k \cdot D(f)$, where $D(f)$ is the standard two-party communication complexity of f (as defined in [14]; see also [6] for an extended background on communication complexity). This is because the processors can simulate an optimal two-party protocol for f (P_0 simulates one party, P_k simulates the other party, and the intermediate processors behave as relays and just propagate the messages they receive). The question is whether better protocols exist. This problem was extensively studied by Tiwari [13], who conjectured that the above naive protocol is essentially optimal. More specifically, he conjectured that for every boolean

Mathematics Subject Classification (1991): 68Q22, 68Q10, 94A05

* Part of this research was done while the authors were at ICSI, Berkeley. An early version of this paper appeared in the proceedings of the *28th ACM Symp. on Theory of Computing (STOC)*, pp. 1–10, May 1996.

† Supported in part by a grant from the Israeli Academy of Sciences.

function f,

$$D_k(f) \geq k \cdot (D(f) - O(1)).$$

Tiwari did establish his conjecture in every instance where a lower bound on $D(f)$ is provable either by fooling set arguments [14, 7] or by the rank method [8]. These results create an interesting state of affairs, where finding a counterexample to Tiwari's conjecture entails developing a new method for proving lower bounds, since the two standard methods of communication complexity cannot be employed. Furthermore, Tiwari's argument implies that his conjecture is valid for almost all functions.

In this paper we disprove this conjecture, by exhibiting an infinite family of functions for which

$$D_k(f) \leq \left(\frac{3}{4} + o(1)\right) \cdot k \cdot D(f)$$

thus, proving that the intermediate processors can take a role in the computation more active than just relaying messages.[1] As Tiwari's results imply, we do develop a novel technique for proving a lower bound on $D(f)$. The proof is based on a careful analysis of the protocol tree, and involves some graph theoretic arguments.

Ever since Yao's early study of two-party communication complexity [14], all lower bounds on communication complexity are derived from estimates on $C(f)$, the least number of monochromatic rectangles in any partition of the input space.[2] Obviously,

$$D(f) \geq \log_2 C(f),$$

and the determination of the exact relationship between $D(f)$ and $\log_2 C(f)$ is a fundamental problem in the field of communication complexity.[3] Results on nondeterministic communication complexity [1] imply that $D(f) = O((\log_2 C(f))^2)$. However, no cases are known where such a gap occurs.[4] As mentioned, the communication complexity of the function f, that disproves Tiwari's conjecture, is bounded via a nonstandard method, whence f is a candidate for establishing such a gap. Indeed, f does exhibit the presently largest known gap, i.e.,

$$D(f) \geq (2 - o(1)) \cdot \log_2 C(f).$$

[1] The above bound holds for even values of k; for odd k the bound is slightly larger; see Corollary 4.

[2] The only exception to this statement is the study of k-round protocols [11, 3, 9].

[3] Recently much effort was devoted to study the power of various lower bound techniques in communication complexity, e.g., the rank method [12, 10] and the rectangle size method for the nondeterministic case [5]. We also take this opportunity for the usual disclaimer that all logarithms are to base 2.

[4] This should not be confused with nondeterministic communication complexity where *covers* (and not necessarily partitions) of the input space are relevant. In that case, functions that exhibit a quadratic gap are known.

In other words, the actual communication complexity is twice as large as the best lower bound obtainable for this function by considering partitions of the space.

We believe that our technique may help in other problems about communication complexity.

Subsequent Work. After the conference version of our paper was published, Dietzfelbinger [2] was able to prove a weaker version of Tiwari's conjecture. Namely, he showed that for every function f, the linear-array complexity satisfies $D_k(f) \geq \gamma \cdot k \cdot D(f)$, for some constant $\gamma < 0.3$. Therefore, up to the question of determining the exact constant, our work together with Dietzfelbinger's work [2] completely solves the linear-array problem.

2. Preliminaries

The *linear array* network consists of $k+1$ processors $P_0, P_1, \ldots, P_k$ with links only between P_i and P_{i+1}, for $0 \leq i \leq k-1$. The processors are to compute a function $f(x,y)$ where the input x is initially stored in processor P_0 and the input y is initially stored in processor P_k. The complexity of a protocol is the *total* number of bits exchanged over all links on the worst input pair (x,y). Let $D_k(f)$ be the complexity of the best protocol for computing the function f on such a linear array. Obviously, $D_k(f) \leq k \cdot D(f)$, where $D(f) \stackrel{\triangle}{=} D_1(f)$ is the standard two-party communication complexity of f (as defined in [14]; the reader is referred to [6] for an extended background on communication complexity including more formal definitions). A simple general lower bound for $D_k(f)$ is given by the following lemma[5].

Lemma 1. *For every function f, $D_k(f) = k \cdot \left(\Omega(\sqrt{D(f)}) - \log k\right)$.*

Proof. We show that for every function f, $R_0(f) \leq \frac{D_k(f)}{k} + \log k$, where $R_0(f)$ is the randomized zero-error communication complexity of f (in the two-party model)[6]. Since it is known [1] that $R_0(f) = \Omega(\sqrt{D(f)})$, the lemma follows. Given a protocol for the linear array network that uses a total of $D_k(f)$ bits, we construct a randomized, zero-error, two-party protocol for f. The first player, Alice, chooses uniformly at random, one of the k links (P_i, P_{i+1}) and sends i to the second player, Bob (at a cost of $\log k$ bits). Alice and Bob simulate the linear array protocol, where Alice (who holds x) simulates $P_0, \ldots, P_i$ and Bob (who holds y) simulates $P_{i+1}, \ldots, P_k$. Note that the only bits that Alice and Bob actually need to exchange are those that go across the chosen link. The expected number of bits that are transmitted in the simulation is therefore at most $D_k(f)/k$. Hence, $R_0(f) \leq \frac{D_k(f)}{k} + \log k$. ∎

[5] As mentioned above, this bound was already improved by Dietzfelbinger [2].

[6] I.e., we measure the expected number of bits exchanged on the worst input (x,y).

3. The results

The definition of our function f depends on another function $g:\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. For our purposes, almost any function can play the role of g, but the specific properties that are required, will be discussed only in Section 4.1.

Definition 2. Let $f : \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}$. The value $f([x_1,x_2],[y_1,y_2])$ is defined by

$$f([x_1,x_2],[y_1,y_2]) \quad \triangleq \quad \begin{cases} g(x_1,y_2) & \text{if } g(x_1,y_1)=1 \\ g(x_2,y_1) & \text{if } g(x_1,y_1)=0 \end{cases}$$

where $x_1,x_2,y_1,y_2 \in \{0,1\}^n$.

Theorem 3. *For almost every choice of the function g,*

$$(2-o(1))\cdot n \le D(f) \le 2\cdot n+1.$$

The upper bound is trivially true, regardless of the choice of g. (In fact, clearly even $D(f) \le 2D(g)$, for every function f.) We defer the proof of the lower bound, which is the most technical part of this paper, to Section 4. Following are corollaries to Theorem 3, starting with a refutation of Tiwari's conjecture:

Corollary 4. *Assume that k is even. For almost every choice of the function g, the function f of Definition 2 satisfies:*

$$D_k(f) \le (\frac{3}{4}+o(1))\cdot k\cdot D(f).$$

(If k is odd, then the bound is $D_k(f) \le (\frac{3}{4}+\frac{1}{4k}+o(1))\cdot k\cdot D(f)$.)

Proof. Assume that k is even. Theorem 3 implies that $D(f) \ge (2-o(1))\cdot n$ for almost every choice of the function g. Here is a protocol to compute f on the linear array (for any g): Initially, P_0 holds $[x_1,x_2]$ and P_k holds $[y_1,y_2]$. Processor P_0 sends x_1 to $P_{k/2}$, and processor P_k sends y_1 to $P_{k/2}$. The total cost of these steps is $(k/2)\cdot n$ bits each. Now, $P_{k/2}$ computes $g(x_1,y_1)$ and it broadcasts this bit to all the processors (this costs total of k bits and it indicates to every processor how the protocol is going to proceed). If $g(x_1,y_1)=0$, processor $P_{k/2}$ sends y_1 to P_0 so that it can compute $g(x_2,y_1)$ (in this case the role of P_k in the protocol is over; this was indicated to it by broadcasting the value of $g(x_1,y_1)$). Otherwise, if $g(x_1,y_1)=1$, then $P_{k/2}$ sends x_1 to P_k for it to compute $g(x_1,y_2)$ (in this case the role of P_0 in the protocol is over which again was indicated by broadcasting the value of $g(x_1,y_1)$). The processor that computed the output bit sends this bit to all other processors (additional k bits of communication). The total cost of the protocol is $\frac{3k}{2}\cdot n+2k$ bits and the corollary follows. For odd k, D_k is only slightly larger – the term $\frac{3k}{2}$ is replaced by $\frac{3k+1}{2}$. ∎

We next turn to show a gap between the two-party communication complexity and the logarithm of the partition number.

Corollary 5. *For almost every choice of the function g, the function f of Definition 2 satisfies:*

$$D(f) \geq (2 - o(1)) \cdot \log_2 C(f).$$

Proof. Again we start from the inequality $D(f) \geq (2-o(1))n$ (that holds for almost every choice of the function g), and show that $C(f) \leq 4 \cdot 2^n$ (for every function g), whence $\log_2 C(f) \leq n+2$. For each string $w \in \{0,1\}^n$ and bit $b \in \{0,1\}$ define two rectangles (altogether $4 \cdot 2^n$ rectangles) as follows:

$$\begin{aligned} R_{w,b} = \{([x_1, x_2], [y_1, y_2]) | x_1 = w, \\ g(x_1, y_1) = 1, \\ g(x_1, y_2) = b \ \ \}, \end{aligned}$$

and

$$\begin{aligned} S_{w,b} = \{([x_1, x_2], [y_1, y_2]) | y_1 = w, \\ g(x_1, y_1) = 0, \\ g(x_2, y_1) = b \ \ \}. \end{aligned}$$

Note that these are indeed rectangles and that they are f-monochromatic. Moreover, these rectangles are disjoint and cover all the inputs. To see this, consider an input $([x_1,x_2],[y_1,y_2])$ and note how to find the unique rectangle to which it belongs. If $g(x_1,y_1)=1$ this rectangle is $R_{x_1,g(x_1,y_2)}$ while if $g(x_1,y_1)=0$ it belongs to $S_{y_1,g(x_2,y_1)}$ and only to it. ∎

4. Proof of Theorem 3

In this section we prove our main theorem (Theorem 3). We start by specifying the function g (subsection 4.1) and then, based on this choice, we provide the details of the proof.

4.1. Choosing the function g

While the upper bound on the two-party communication complexity of f does not depend on the choice of g, the lower bound does require a careful choice of g. The intuition is that a "complicated" function g would force the two players to first compute $g(x_1,y_1)$ and, furthermore, that this computation may turn out useless for the remainder of the computation (of either $g(x_1,y_2)$ or $g(x_2,y_1)$). The notion that a "complicated" g is sought, should be taken with a grain of salt, though. For example, the inner-product function, is "difficult" for many purposes in the theory of communication complexity (formally, the inner-product function is defined by $IP(x,y) \stackrel{\triangle}{=} \sum_{i=1}^{n} x_i y_i \bmod 2$). It is instructive to see that this choice will not do in our setting, and the bound $D(f) \leq 1.5n + O(1)$ holds in this case: Alice sends the $n/2$

most significant bits of x_1 to Bob who responds with the $n/2$ least significant bits of y_1. With the exchange of two more bits, both players will know the inner product of x_1 and y_1. According to the answer they need to compute either the inner product of x_1 and y_2 or the inner product of x_2 and y_1. In either case, one of the players already has $n/2$ bits of the other input so by receiving the complementary $n/2$ bits it can complete this computation. The total number of bits that are transmitted is $1.5n+O(1)$. Similar arguments hold for other natural choices of g as well.

Instead, we pick a random g. Denote $N \stackrel{\triangle}{=} 2^n$. Let $0 \leq \alpha \leq 1$ and $\beta \geq 1$ be constants; specifically, we choose $\alpha = 1/32$ and β sufficiently large as required by the proof of Lemma 6 below. Denote $L = \beta n$. Any function $g : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ is also represented through its table $\Gamma = \Gamma_g$, which is an $N \times N$ 0-1 matrix. Let R_1 consist of L disjoint pairs of rows and similarly, let R_2 be a set of L disjoint pairs of columns in Γ. Correspondingly, the rectangle (minor)[7] $R_1 \times R_2$ of Γ consists of L^2 disjoint 2×2 squares, each of which can have any one of the $2^4 = 16$ possible patterns. The $R_1 \times R_2$ minor is called α-balanced if each of these 16 patterns appears at least αL^2 times. Define the following property:

(P1) All minors of dimensions $2L \times 2L$ in Γ_g are α-balanced.

Lemma 6. *Almost every function g satisfies property (P1).*

Proof. Consider a particular rectangle. That is, choose L disjoint pairs of rows, R_1, and L disjoint pairs of columns, R_2. Fix one of the 16 possible patterns. If the function g is selected uniformly at random, the expected number of appearances of this pattern in the rectangle is exactly $L^2/16$. By Chernoff's inequality, the probability that this pattern appears fewer than $L^2/32 = \alpha L^2$ times is at most $2^{-\Theta(L^2)}$. The probability that *some* pattern appears less than $L^2/32$ times is at most 16 times larger, and still $2^{-\Theta(L^2)}$. In other words, any particular $2L \times 2L$ rectangle fails to be α-balanced with probability at most $2^{-\Theta(L^2)}$. The number of such rectangles can certainly be bounded by $[\binom{N}{2L} \cdot (2L)!]^2 = N^{O(L)} = 2^{O(nL)}$. Therefore, the probability that *some* rectangle is not α-balanced does not exceed $2^{O(nL)} \cdot 2^{-\Theta(L^2)}$. But $L = \beta n$, so for a sufficiently large constant β, this probability is at most $2^{-\Theta(L^2)}$. The lemma follows. ∎

The proof of the lower bound repeatedly uses property (P1), as well as the following immediate consequences of (P1):

(P2) The matrix Γ_g contains no $L \times L$ monochromatic rectangle.

[7] The term *rectangle* and the term *minor* have both the same meaning: a sub-matrix defined by some subset of the rows and some subset of the columns; the term rectangle comes from the theory of communication complexity while the term minor comes from matrix theory; we use these two terms interchangeably.

(P3) In every $2L \times L$ rectangle of Γ_g, each of the four 2×1 patterns appears at least αL^2 times. Similarly, in every $L \times 2L$ rectangle of Γ_g, each of the four 1×2 patterns appears at least αL^2 times.

A row x is said to be *balanced* with respect to a set of columns B, if the fraction of zeros in row x within this set is between 8α and $1-8\alpha$ (i.e., $8\alpha|B| \le |\{y \in B : g(x,y)=0\}| \le (1-8\alpha)|B|$). We define what it means for a column y to be balanced with respect to a set of rows A, in the natural way.

(P4) Let $A \times B$ be a minor of Γ_g. If $|B| \ge L$ then at most $2L$ rows in A are imbalanced with respect to B. Likewise, if $|A| \ge L$ then B has at most $2L$ columns that are imbalanced with respect to A.

The next property is never used in the actual proof, but may help in guiding the reader's intuition (it follows from property (P4) and the monochromatic rectangle-size bound).

(P5) Consider any $k \times k$ minor of Γ_g and the associated communication problem. The complexity of this problem is at least $\log k - O(\log\log N)$.

Remark. The correctness of our lower bound depends only on the leading term in the asymptotics of $D(f)$. At this level of resolution, it is not essential for L to be only $O(\log N)$, as stated and proved. Rather, the leading asymptotic term remains unchanged as long as $L = N^{o(1)}$. Unfortunately, no explicit construction is currently known for functions g that satisfy the above conditions with $L = O(\log N)$ (this is essentially the notorious question of explicit constructions for Ramsey graphs). However, if we settle for $L = N^{o(1)}$, then such constructions are known [4]. Consequently, the main statements of our article hold also with an explicitly constructed f.

4.2. The lower bound

Let g be any function that satisfies (P1). Consider an optimal protocol for computing f and the tree T of this protocol. We seek a node ζ in T such that many bits must be exchanged by the players in order to reach this node, but still many bits need to be transmitted to complete the computation of f. The next paragraph presents the recipe for finding such a node ζ, and subsequently all relevant arguments are proved.

Each node z in T, is naturally associated with two directed graphs G_1 and G_2 on vertex set $\{0,1\}^n$: If input $([x_1,x_2],[y_1,y_2])$ is consistent with the protocol reaching node z, then (x_1,x_2) and (y_1,y_2) are directed edges in $G_1 = (V_1,E_1)$ and in $G_2 = (V_2,E_2)$, respectively. To derive our bound, we traverse the tree T starting from the root on a path along which the protocol progresses "slowly": At each step in the protocol either E_1 or E_2 (but not both) is partitioned into two parts, and the two edges out of node z in the tree T correspond to these parts. Our traversal of the tree always follows the edge that corresponds to the *larger* of the two parts. As we traverse the tree we also "prune" the graphs G_1 and G_2: Certain edges in these graphs are declared "bad" along the traversal and are henceforth eliminated from the corresponding graph. The specifics of this pruning process will be explained later, but note that the elimination of edges can only decrease the depth of T, so it may only become harder to prove lower bounds on communication complexity. Isolated vertices are essentially irrelevant to our discussion, so we define, for each node of T, the set V_i of all non-isolated vertices in G_i. The quantity $\frac{|E_i|}{|V_i|}$ (i.e., the average out-degree of non-isolated vertices in G_i) is denoted ρ_i, and ρ is $\min\{\rho_1, \rho_2\}$. The desired node ζ in T is the first one we encounter where:

$$\frac{L^3}{2} < \rho \leq 2L^3 \tag{1}$$

The very existence of a node ζ in T that satisfies Condition (1) is not clear at this stage. The first step towards showing the existence of such a ζ, is to study the leaves of T (note that at the root of the tree $\rho = 2^n$), and prove:

Lemma 7. *At every leaf of T, $\rho \leq 2L$.*

Proof. We claim that in a leaf of T, it is impossible for both graphs, G_1 and G_2, to have a matching of size L. Otherwise, consider the $2L \times 2L$ minor of Γ_g, whose rows and columns correspond to the vertices in G_1 and G_2 that participate in these matchings. Since we are dealing with a leaf, it follows that the value of f is already uniquely defined, whence (by the definition of f) only 8 of the 16 patterns may appear in this minor, contrary to Property (P1).

Note that if M is an inclusion-maximal matching in a graph with e edges and v vertices, then M has at least $\frac{e}{2v}$ edges: Given $|M|$ and v, the number of edges, e, is maximized by making every pair of vertices adjacent, except if neither of them is covered by the matching. Therefore, we get that $e \leq 2|M|v$. In particular, the largest matching in a graph without isolated vertices has at least $\rho/2$ edges, and the conclusion follows. ∎

We also need to control the rate at which $\rho, |E_1|$ and $|E_2|$ decrease, as we traverse T. This will allow us to establish the existence of the desired node ζ, as well as to bound the time to reach this node in the protocol.

We now introduce *pruning*, a process that will be applied at every node during the traversal. Assume that at the current node the edge set E_1 is split. (Otherwise, interchange the roles of G_1 and G_2.) Consider the situation just after E_1 is split. Let $V_2^+ \subseteq V_2$ consist of all vertices in G_2 with a positive out-degree, and consider the $V_1 \times V_2^+$ minor of Γ_g. Since $|V_2^+| \geq \frac{|E_2|}{|V_2|} = \rho_2 \geq L^3/2$, by Property (P4) at most $2L$ vertices in V_1 are imbalanced with respect to V_2^+. All edges that are incident on these vertices are considered *bad* and are removed from G_1. When pruned and unpruned graphs need to be distinguished, we use *bars* to denote pruned graphs and their parameters. Pruning can isolate vertices, and consequently ρ_i may even increase as we traverse T. However, to show that Condition (1) is satisfied sometime, whence ζ exists, it suffices to show that ρ never *decreases* too rapidly. Indeed, pruning does not speed much the rate at which ρ decreases, since the number of pruned edges is never too big: Clearly, $|E_1| \geq L^3|V_1|/2$ and so $|\bar{E}_1| \geq |E_1| - 2L|V_1| \geq (1 - \frac{4}{L^2})|E_1|$, and in particular, $\bar{\rho}_1 \geq (1 - \frac{4}{L^2})\rho_1$. It follows that in passing from a node in T to its child, ρ may, at worst, get multiplied by a factor of $\frac{1}{2}(1 - \frac{4}{L^2})$. Since, by Lemma 7, $\rho \leq 2L$ at the leaves, and the bounds in Condition (1) differ by a factor of four, it follows that ζ exists.

We turn to bound the time for reaching ζ in the protocol, and claim that until this time, at least

$$t_1 \geq \log \frac{N^4}{|E_1^\zeta| \cdot |E_2^\zeta|} - o(1) \tag{2}$$

bits must be exchanged (where E_1^ζ and E_2^ζ are the edge sets corresponding to the graphs in the node ζ under consideration): At the protocol's outset $|E_1| \cdot |E_2| = N^4$ and in each step this quantity can be reduced by at most a factor of $\frac{1}{2}(1 - \frac{4}{L^2})$. But $\log(\frac{1}{2}(1 - \frac{4}{L^2})) = -(1 + O(L^{-2}))$, so Equation (2) follows where the $o(1)$ term is, in fact $O(L^{-1})$.

We now turn to the main part of the proof, in which we show that to complete the protocol, starting from ζ,

$$t_2 \geq \log \frac{|E_1^\zeta| \cdot |E_2^\zeta|}{N^2} - O(\log L) \tag{3}$$

additional bits must be transmitted. (From now on, whenever referring to the graphs G_1, G_2 or to the associated sets V_1, V_2, E_1 and E_2 we refer to the graphs corresponding to node ζ; henceforth, we omit ζ from the notation.) By summing the bounds in (2) and (3), it follows that the communication complexity is at least $2 \log N - O(\log \log N)$, as claimed. The intuition behind the proof of (3) is that since $|E_1| \geq L^3|V_1|/2$, there are still viable inputs, for which the bits communicated so far are of the "wrong type". To illustrate this idea, suppose that at ζ, the graph G_1 has a vertex u with a large out-degree. It follows that many inputs of the form (u, v) reach ζ. If it so happens that $g(u, w) = 0$ for many $w \in V_2^+$ (the set of vertices in G_2 of positive out-degree), then the evaluation of f, starting at ζ is still complex: It entails, at least, the evaluation of g on a large minor of Γ_g. The rows of this minor correspond to all those v with $(u, v) \in E_1$, the columns correspond to every $w \in V_2^+$ with $g(u, w) = 0$. The function g has a high communication complexity (Property (P5)), so the bound follows in this case.

Note that this is only a rough idea for a proof, and several difficulties arise in attempting to implement it as we see below. The reader is encouraged, however, to keep this intuition in mind.

To recap, we may assume that starting from ζ, the protocol computes the value of f on pairs of inputs from $E_1 \times E_2$, where:

(A1) $L^3/2 \leq \rho_1 \leq 2L^3 < \rho_2$.

(A2) All vertices in V_1 are balanced with respect to V_2^+.

We need the following graph-theoretic fact: In a graph with no isolated vertices there is a collection of vertex-disjoint proper stars that cover all vertices, where "proper" means stars with two vertices or more. (For the purpose of this definition we ignore the direction of the edges.) To prove this, construct a spanning forest in the graph. Any tree in this forest which is of diameter 1 or 2 is already a star and can be added to the list of stars. In a tree R of larger diameter, pick a diametrical pair of vertices x, y, and remove an edge h from the x, y path that is not incident on x nor on y. It is not hard to see that each of the two components of $R \setminus \{h\}$ has at least two vertices. Proceed with this process until only proper stars remain.

Apply this fact to the underlying graph of the digraph G_1, to obtain a collection of vertex-disjoint stars with a total of at least $|V_1|/2$ edges. We distinguish two cases: (I) at least half of the edges in the stars are oriented away from the stars' centers; and (II) at least half of the edges are oriented towards the centers. Denote $B \stackrel{\triangle}{=} \frac{|E_1||E_2|}{8N^2L^3}$. We now deal with each of these two cases.

Case (I). Let D be the set of edges oriented away from the stars' centers. Condition (A1), implies that

$$|D| \geq \frac{|V_1|}{4} \geq \frac{|E_1|}{8L^3} \geq \frac{|E_1||E_2|}{8L^3N^2} = B$$

and

$$|V_2^+| \geq \frac{|E_2|}{N} \geq \frac{|E_1||E_2|}{2N^2L^3} \geq B.$$

Consider the following communication problem, denoted Π, which is defined on $D \times V_2^+$. On inputs $(x_1, x_2) \in D$ and $y_1 \in V_2^+$, if $g(x_1, y_1) = 0$, then output $g(x_2, y_1)$, but if $g(x_1, y_1) = 1$, then any output is acceptable. This problem is represented again by a matrix with $0, 1$ and "*" entries, the latter corresponding to the "don't care" case. Since our protocol computes f, the sub-protocol starting from node ζ solves the problem Π. We conclude the desired Inequality (3) from a lower bound for this problem.

The proof is in two steps: we first compute the *effective area* of the $|D| \times |V_2^+|$ matrix, i.e, the number of non-"*" entries. Consider a specific row in the matrix, say row (x_1, x_2). By Property (A2), x_1 is balanced with respect to V_2^+. Therefore, a fraction of at least $\alpha' = 8\alpha$ of the y_1's in V_2^+, satisfy $g(x_1, y_1) = 0$, yielding a non-"*" entry. This holds for each individual row, whence also for the whole matrix $D \times V_2^+$. Consequently, the effective area of this matrix is at least $\alpha'|D||V_2^+|$.

As usual, any protocol for the communication problem Π partitions $D\times V_2^+$ into Π-monochromatic rectangles. In the ensuing discussion, a minor is "Π-monochromatic" if either 0 or 1 are missing from it. In other words, if the value of Π on all the entries of the minor is either b or "*", for some $b\in\{0,1\}$. It will be shown that every Π-monochromatic rectangle has a small effective area, whence their number is large, and the protocol must be long. Consider a Π-monochromatic rectangle $R\times S$ with some output value $b\in\{0,1\}$, where without loss of generality $|S|\geq L$ (otherwise, the effective area of this rectangle is clearly smaller than $D\cdot L$ which will make this an easy case). Recall that rows are indexed by ordered pairs (x_1,x_2), so we can speak of the *slice* $\Sigma_{x_1}\subseteq R$ of those rows in R with this x_1 (and we will bound the effective area of $R\times S$ by using this partition of R into slices). First, we show that the number of slices cannot be larger than L. For this, we pick from each slice $\Sigma_{\tilde{x}_1}$ one of its rows $(\tilde{x}_1,\tilde{x}_2)$. For every such pair $(\tilde{x}_1,\tilde{x}_2)$, consider the corresponding pair of rows in Γ_g. Note that by choosing the pairs $(\tilde{x}_1,\tilde{x}_2)$ as edges from vertex-disjoint starts (with the $\tilde{x}_1$ as the centers) it follows that all these pairs are disjoint. Hence, we can consider the minor of Γ_g with this set of rows and with the set of columns S, and make a census of its 2×1 minors. By assumption, none of these 2×1 minors satisfies $g(x_1,y_1)=0$ and $g(x_2,y_1)=\bar{b}$. Therefore, the number of slices must be at most L as otherwise we get a contradiction to property (P3). Next, we call a slice *wide* if it has at least L rows, and we call it *narrow* otherwise. Suppose that Σ_{x_1} is a wide slice. The only contribution to the effective area within this slice comes from columns in S_{x_1}, the set of those $y_1\in S$ with $g(x_1,y_1)=0$. Let R_{x_1} be the set of all x_2 such that $(x_1,x_2)\in\Sigma_{x_1}$. It follows that every entry in the $R_{x_1}\times S_{x_1}$ minor of Γ_g is b. Since $|R_{x_1}|=|\Sigma_{x_1}|\geq L$, it follows by property (P2) that $|S_{x_1}|\leq L$. Hence, this slice's contribution to the effective area of the rectangle does not exceed $|\Sigma_{x_1}|\cdot L$. By adding these inequalities together, the total contribution of wide slices to the effective area of the rectangle $R\times S$ is at most $|D|\cdot L$.

We turn to bound the contribution of narrow slices to the effective area of $R\times S$. Obviously, there are no more than L narrow slices (since we proved that even the total number of slices is at most L). A narrow slice has L rows or less, so the total area of narrow slices does not exceed $L^2|V_2^+|$. We conclude that the effective area of a Π-monochromatic rectangle is no more than $L^2(|D|+|V_2^+|)$. So, in partitioning the matrix to Π-monochromatic rectangles, the number of rectangles is at least

$$\frac{\alpha'|D||V_2^+|}{L^2(|D|+|V_2^+|)}\geq\frac{\alpha' B}{2L^2}=\frac{\alpha'|E_1||E_2|}{16N^2L^5}.$$

Finally, since the communication complexity is at least the logarithm of this expression it yields (3) as desired.

Case (II). This case is handled similarly; hence, we skip some of the details and mainly emphasize the differences between the two cases. Let C be the set of leaves in the vertex-disjoint stars that were extracted from G_1. Recall that in case (II), we concentrate on those stars that are oriented towards their centers, so vertices in C play the role of x_1. Also, by assumption, $|C|\geq|V_1|/4\geq B$. At this point we need to prune the graph G_2 (this is in addition to the pruning of G_1 that was made in this node ζ), and remove those vertices of V_2^+ that are imbalanced with respect to C. Since $|V_1| \geq \frac{|E_1|}{|V_1|} = \rho_1 \geq L^3/2$, it follows that $|C| \geq L^3/8$. Hence property (P4) may be used, to conclude that at most L vertices of G_2 are eliminated in this pruning. Also no more than $L|V_2|$ edges are thus lost, i.e., at most $\frac{|E_2|}{2L^2}$, since, by assumption $|E_2| \geq 2L^3|V_2|$. We henceforth assume, then that all vertices in V_2^+ are balanced with respect to C.

Consider the set V_2^- of those vertices in G_2 that have a positive in-degree. Clearly, $|V_2^-| \geq \frac{|E_2|}{N} \geq \frac{|E_1||E_2|}{2N^2L^3} > B$. Let F be a set of $|V_2^-|$ edges that is obtained by picking one edge (y_1, y_2) for each $y_2 \in V_2^-$. Consider the following communication problem, denoted Π', defined on $C \times F$. For $x_1 \in C$ and $(y_1, y_2) \in F$, if $g(x_1, y_1) = 1$ the output is $g(x_1, y_2)$, while if $g(x_1, y_1) = 0$, any output is acceptable. Again we use "*" to denote this "don't care" situation. As before, the original sub-protocol starting from node ζ solves this problem Π'. Again, the lower bound is proved by considering effective areas. Consider any column $(y_1, y_2) \in F$. Since y_1 is in V_2^+ it is balanced with respect to C, and so a fraction of at least $\alpha' = 8\alpha$ of the entries in the column are non-"*" (for those $x_1 \in C$ where $g(x_1, y_1) = 1$). By summing over all of F, the matrix has total effective area of at least $\alpha'|C||V_2^-|$. To bound the effective area of any Π'-monochromatic rectangle $R \times S$ (where, without loss of generality, $|R| \geq L$), we define the slice Σ_{y_1} as the set of columns with y_1 as the first element of the pair. As in Case (I), we start by bounding the number of slices. Here the argument is slightly different and we only show that this number is at most $3L$ (rather than L in Case (I)). First, we pick from each slice $\Sigma_{\tilde{y}_1}$ one of its columns $(\tilde{y}_1, \tilde{y}_2)$. Note that, by the choice of F, all the $\tilde{y}_2$ are distinct; moreover, since we picked one column from each slice all the $\tilde{y}_1$ are distinct. It may be though that some $\tilde{y}_1$ is equal to some $\tilde{y}_2$. However, it is always possible to pick (at least) $1/3$ of the pairs so that they are all distinct. We concentrate on these pairs and for each of them consider the corresponding pair of columns in Γ_g. Hence, we can consider the minor of Γ_g with this set of columns and with the set of rows R. The argument proceed, as in Case (I), to show that the number of columns is at most L (again, by using property (P3)) and hence the number of slices is at most $3L$.

Again, we say that a slice is *wide* if it contains at least L columns and otherwise we say that the slice is *narrow*. As in Case (I), the bound on the number of slices immediately yields a bound on the effective area of all narrow slices. It remains to estimate the effective area within *wide* slices (those with at least L columns). If row $x_1 \in C$ satisfies $g(x_1, y_1) = 0$, there are only "*" in this row and so it does not contribute to the effective area. If $g(x_1, y_1) = 1$, a g-monochromatic rectangle is obtained which by property (P2) has at most L rows. Altogether, wide slices contribute $L \cdot |V_2^-|$ to the effective area.

References

[1] A. Aho, J. Ullman, and M. Yannakakis: On notions of information transfer in VLSI circuits, *Proc. 15th STOC*, 133–139, 1983.

[2] M. Dietzfelbinger: The linear-array problem in communication complexity resolved, Proc. 29th STOC, 373–382, 1997.

[3] P. Duris, Z. Galil, and G. Schnitger: Lower bounds on communication complexity, *Information and Computation*, **73**(1) (1987), 1–22. (Early version in *Proc. 16th STOC*, 81–91, 1984.)

[4] P. Frankl, and R. M. Wilson: Intersection theorems with geometric consequences, *Combinatorica*, **1** (1981), 357–368.

[5] M. Karchmer, E. Kushilevitz, and N. Nisan: Fractional covers and communication complexity, *SIAM J. Discrete Math.*, **8**(1), (1995), 76–92. (Early version in *7th IEEE Structure in Complexity Theory*, (1992), 262–274.)

[6] E. Kushilevitz and N. Nisan: *Communication Complexity,* Cambridge University Press, 1997.

[7] R. J. Lipton and R. Sedgewick: Lower bounds for VLSI, *Proc. 13th STOC*, 300–307, 1981.

[8] K. Mehlhorn and E. Schmidt: Las-Vegas is better than determinism in VLSI and distributed computing, *Proc. 14th STOC,* 330–337, 1982.

[9] N. Nisan and A. Wigderson: Rounds in communication complexity revisited, *SIAM J. Computing*, **22**(1) (1993), 211–219. (Early version in *Proc. 23rd STOC*, 419–429, 1991.)

[10] N. Nisan and A. Wigderson: On rank vs. communication complexity, *Combinatorica*, **15**(4) (1995), 557–566. (Early version in *Proc. 35th FOCS*, 831–836, 1994.)

[11] C. Papadimitriou and M. Sipser: Communication complexity, *JCSS*, **28**(2) (1984), 260–269. (Early version in *Proc. 14th STOC*, 196–200, 1982.)

[12] R. Raz and B. Spieker: On the "log rank" conjecture in communication complexity, *Combinatorica*, **15**(4) (1995), 567–588. (Early version in *Proc. 34th FOCS*, 168–176, 1993.)

[13] P. Tiwari: Lower bounds on communication complexity in distributed computer networks, *JACM*, **34**(4) (1987), 921–938. (Early version in *Proc 25th FOCS*, 109–117, 1984.)

[14] A. C. Yao: Some complexity questions related to distributed computing, *Proc. 11th STOC*, 209–213, 1979.

Eyal Kushilevitz

Department of Computer Science,
Technion, Haifa 32000, Israel.
eyalk@cs.technion.ac.il
http://www.cs.technion.ac.il/∼eyalk

Nathan Linial

Institute of Computer Science,
Hebrew University, Israel
nati@cs.huji.ac.il

Rafail Ostrovsky

Bell Communications Research,
MCC-1C365B, 445 South Street,
Morristown, New Jersey 07960–6438,
rafail@bellcore.com