Probabilistic Communication Complexity*

RAMAMOHAN PATURI

Aiken Computation Laboratory, Harvard University, Cambridge, Massachusetts 02138

AND

JANOS SIMON

Department of Computer Science, University of Chicago, Chicago, Illinois 60637

Received February 15, 1985; revised February 26, 1986

Communication is a bottleneck in many distributed computations. In VLSI, communication constraints dictate lower bounds on the performance of chips. The two-processor information transfer model measures the communication requirements to compute functions. We study the unbounded error probabilistic version of this model. Because of its weak notion of correct output, we believe that this model measures the "intrinsic" communication complexity of functions. We present exact characterizations of the unbounded error communication complexity in terms of arrangements of hyperplanes and approximations of matrices. These characterizations establish the connection with certain classical problems in combinatorial geometry which are concerned with the configurations of points in d-dimensional real space. With the help of these characterizations, we obtain some upper and lower bounds on communication complexity. The upper bounds which we obtained for the functions—equality and verification of Hamming distance— are considerably better than their counterparts in the deterministic, the nondeterministic, and the bounded error probabilistic models. We also exhibit a function which has $\log n$ complexity. We present a counting argument to show that most functions have linear complexity. Further, we apply the logarithmic lower bound on communication complexity to obtain an $\Omega(n \log n)$ bound on the time of 1-tape unbounded error probabilistic Turing machines. We believe that this is the first nontrivial lower bound obtained for such machines. © 1986 Academic Press, Inc.

1. INTRODUCTION

It is well known that information transfer is a bottleneck in many parallel algorithms, VLSI implementations, and distributed systems. In fact, information transfer is a lower bound on the area-time square of VLSI chips [17]. Communication complexity measures the information transfer necessary for computing a function, when its two arguments are distributed over two processors which communicate according to some protocol [20, 16]. This measure, communication com-

* A preliminary version of this paper was presented on October 24, 1984, at the 25th Annual IEEE Symposium on Foundations of Computer Science, Singer Island, Florida. Research partially supported by USARO Research Contract DAAG 29–82–K–0110.

0022-0000/86 $3.00

plexity, is closely related to the time of 1-tape Turing machines, and to the size of branching programs, and of monotone circuits [19, 11]. Communication complexity allows us to study otherwise intractable questions (like the power of nondeterminism, the power of probabilistic choice, etc.) in a favorable environment, where it is possible to settle some of them [20, 16, 10, 1, 21].

In this paper, we study a model of unbounded error probabilistic communication complexity in which the processors are allowed to toss coins and the error in the computation is *not* bounded away from $\frac{1}{2}$. This unbounded error probabilistic model is not intended to serve as the basis for a theory of "reliable information transfer." Rather, we are interested in understanding the power of unrestricted probabilistic choice in distributed environments.

We show that this model is powerful. Consider the functions: $I(x, y) = (x = y)$; $\bar{I}(x, y) = (x \neq y)$; and $G(x, y) = (x \geqslant y)$, where x and y are interpreted as n-bit integers. We show that the functions I and $\bar{I}$ need just 2 bits of information transfer,[1] and the function G needs only 1 bit of information transfer in the unbounded error probabilistic communication model.

In contrast, we have the results that I, $\bar{I}$, and G require n bits of information transfer in the deterministic model, and $\bar{I}$ and G require n bits of information transfer in the nondeterministic model [20, 16]. Note that n bits of information transfer is equivalent to one processor sending its entire argument to the other processor. Even in the bounded error probabilistic model $\Omega(\log n)$ bits[2] must be exchanged to compute the functions I, $\bar{I}$, and G [19].

An immediate question is whether these facts mean that the model is trivial. After all, one processor could probabilistically guess the argument of the other processor, verify the guess using the equality protocol, and compute the function with just 2 bits of information transfer. Fortunately (?), the strategy does not work since, as one can verify, the computation is not reliable enough. This challenges us to try to prove lower bounds for the probabilistic information transfer. The results in this paper partially answer this challenge.

The problem of proving lower bounds for the probabilistic information transfer requires new techniques. In the case of the deterministic protocols, a counting argument immediately yields a (nonconstructive) proof of the existence of functions with asymptotically linear communication complexity. For example, there are $2^{2^{2n}}$ Boolean functions of $2n$ variables, but only $2^{2^{O(l)}}$ different deterministic protocols of length l. There are, on the other hand, nondenumerably many probabilistic protocols of length l, since the probabilities can be arbitrary. Although, by a continuity argument, we can restrict ourselves to rational probabilities with bounded denominators, the number of resulting protocols still makes the counting argument useless. In the case of the bounded error probabilistic model, both logarithmic and linear lower bound arguments make use of the fact that the error in the computation is bounded by a constant [19, 21].

[1] This fact was known to M. Rabin in the context of crossing sequences for Turing machines [7].

[2] All logarithms in this paper have base 2.

The following gives a summary of the techniques we developed to study unbounded error probabilistic communication complexity and the upper and lower bounds we obtained with these techniques.

We present two equivalent exact characterizations of the probabilistic communication complexity of a function: one in terms of the approximations of a Boolean matrix by rank 1 real matrices (Theorem 3), and the other, a geometric one, using arrangements of hyperplanes (Theorem 2). Because of these equivalences, questions about the probabilistic communication complexity can be formulated as combinatorial problems on arrangements of hyperplanes, oriented matroid, or lopsided sets [22, 6, 12].

A step towards obtaining these characterizations is the theorem (Theorem 1) which proves that the one-way probabilistic model is as powerful as the two-way one and thereby let us consider only one-way protocols. (In contrast, we have, in the deterministic model, exponential gaps between not only one-way and two-way protocols, but also k-turn and $k+1$-turn protocols [5].)

Using these characterizations, we construct a hierarchy of functions f_i, that require i bits of information transfer for $1 \leqslant i \leqslant \log n$ (Theorem 4). We obtain efficient protocols for computing equality and verification of Hamming distance (Section 7).

Using a recent result of Goodman and Pollack [9] on the number of equivalence classes of arrangements, Alon, Frankl, and Rödl [2] use a counting argument to show that most functions have linear unbounded error communication complexity. We give a brief sketch of this counting argument (Theorem 5).

We also give an $\Omega(n \log n)$ bound on the time of certain 1-tape probabilistic Turing machines with unbounded error (Theorem 6). This lower bound comes as an application of the logarithmic lower bound for the unbounded error probabilistic communication complexity. We also indicate a close relation between the lower bounds on the time of 1-tape Turing machines in the deterministic, the nondeterministic and the unbounded error probabilistic models and lower bounds on the communication complexity of a linear array of processors in the corresponding model [18] (Sect. 10). Finally, we discuss some open problems in Section 11.

2. Definition of the Model

The essentials of our model are the same as those of Yao [20], who introduced the notion of communication complexity (see also [16, 10] for variants of and extensions to the model).

Two processors P_0 and P_1 wish to compute a function of two arguments. (We assume in most of this paper that the function is Boolean.) The first argument, x_0, of the Boolean function f: $\{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ is known to P_0, and the second argument, x_1, is known to P_1. We treat an $x \in \{0, 1\}^n$ also as a number whose binary representation is the string x. With the function f, we associate a $2^n \times 2^n$ matrix F whose (x, y)th entry, $F[x, y]$, is $f(x, y)$.

In order to compute f, P_0, and P_1 communicate with each other in turns by sending messages (sequences of bits) according to some protocol. P_0 and P_1 have unlimited local computing power, and the ability to realize an arbitrary probability distribution over the set of messages they transmit in each turn. The complexity measure is the number of bits transmitted.

Given the input x_i to P_i for $i = 0, 1$, the computation, according to some protocol ϕ, will be as follows: P_0 is always the first one to send a message. The processors communicate in turns. The last message is always sent by P_1 and is a single bit. The last bit is the output produced. Each message will be sent with a certain probability, determined by the protocol.

A probabilistic computation can be viewed as a stochastic process. An *event* in this process is a sequence of messages $\beta_1, \beta_2, ..., \beta_{2k}$ (where message β_i is sent by processor $P_{(i+1) \bmod 2}$). The probability distribution, given by the protocol, assigns a probability to each event. The *result* of an event is the output produced by the associated sequence of messages. The protocol ϕ outputs the bit b ($b = 0$ or 1) if the probability of the events whose result is b is greater than $\frac{1}{2}$.

Formally, a protocol can be specified by a function ϕ: $\{0, 1\}^n \times \{0, 1\}^* \times \{0, 1\}^* \to [0, 1]$. ($[0, 1]$ is the closed interval on the real line with end points 0 and 1.) $\phi(x, \alpha, \beta)$ is the probability that the message β will be sent by a processor, where x is its input and α is the concatenation of the sequence of messages exchanged so far. ϕ has the property that the set $\{\beta \mid \exists x \phi(x, \alpha, \beta) \neq 0\}$ is finite and prefix free for each α. Note that $\sum_\beta \phi(x, \alpha, \beta) = 1$. Due to the prefix freeness property, a concatenated sequence of messages can again be decomposed into a sequence of messages which is unique for a given protocol.

Let ϕ be a protocol. Let x_i be the input at P_i for $i = 0, 1$. Let $(\beta_1, p_1), ..., (\beta_{2l}, p_{2l})$ be such that

$$
\begin{array}{rl}
\phi(x_0, \lambda, \beta_1) = p_1 & \text{where } \lambda \text{ is the null string;} \\
\phi(x_0, \beta_1 \cdots \beta_{2j}, \beta_{2j+1}) = p_{2j+1} & \text{for } j = 1, ..., l-1; \\
\phi(x_1, \beta_1 \cdots \beta_{2j-1}, \beta_{2j}) = p_{2j} & \text{for } j = 1, ..., l; \\
\phi(x_0, \beta_1 \cdots \beta_{2l}, \lambda) = 1 & (\beta_{2l} \text{ is the last message}); \\
|\beta_{2l}| = 1 & (\text{the last message transmitted is a single bit}).
\end{array}
$$

The set of all such sequences (β_1, p_1), (β_2, p_2), ..., (β_{2l}, p_{2l}) is the computation $T_\phi(x_0, x_1)$ using the protocol ϕ with inputs x_i at P_i.

Note that the probabilities $p_1, p_3, ..., p_{2l-1}$ do not depend on the input at the processor P_1. Similarly, $p_2, p_4, ..., p_{2l}$ do not depend on the input at P_0. We, therefore, define two functions ϕ_0, ϕ_1: $\{0, 1\}^n \times M_\phi \to [0, 1]^+$, where M_ϕ is the set of all concatenated sequences of messages that are transmitted between P_0 and P_1

with positive probability for some input. Let $\beta_1, ..., \beta_{2l}$ be the decomposition of $\alpha \in M_\phi$ according to the protocol ϕ. Define $\phi_i(x, \alpha) = (p_1, ..., p_l)$, where

$$p_j = \phi(x, \beta_1 \cdots \beta_{2j-2}, \beta_{2j-1}) \qquad \text{if} \quad i = 0,$$
$$p_j = \phi(x, \beta_1 \cdots \beta_{2j-1}, \beta_{2j}) \qquad \text{if} \quad i = 1,$$

for $j = 1, ..., l$.

The functions ϕ_0 and ϕ_1 together with the decomposition for each $\alpha \in M_\phi$ capture all the information contained in the protocol ϕ.

In the computation $T_\phi(x_0, x_1)$, the probability of outputting the bit b is $\sum_{\alpha b \in M_\phi} \Pi(\phi_0(x_0, \alpha b))\, \Pi(\phi_1(x_1, \alpha b))$. Here, Π is an operator which, when applied to a finite list of real numbers, yields their product.

The *communication complexity* $\tilde{C}_\phi$ of the protocol ϕ is $\max\{|\alpha| : \alpha \in M_\phi\}$. The protocol ϕ computes a function f if $f(x_0, x_1) = b$ iff the probability of outputting the bit b in the computation $T_\phi(x_0, x_1)$ is greater than $\frac{1}{2}$.

The *unbounded error probabilistic communication complexity* $\tilde{C}_f$ is $\min\{\tilde{C}_\phi \mid \phi$ computes $f\}$.

A restricted model in which only one processor, P_0, is allowed to send messages is also of interest because of its equivalence to the unrestricted two-way model. In this one-way model, P_0 sends the messages $\beta_1, ..., \beta_l$ with probabilities $p_1, ..., p_l$, respectively. P_1 on the receipt of β_i, outputs 1 with probability q_i and 0 with probability $1 - q_i$. The probability distribution on the set of messages sent by P_0 is entirely determined by the input at P_0 alone, and is not influenced by the input at P_1. Similarly, the probabilities q_i at P_1 depend only on its input and the message received. The one-way protocol ϕ can therefore be completely specified by two functions $\phi_0, \phi_1 : \{0, 1\}^n \times M_\phi \to [0, 1]$, where M_ϕ is the set of all messages that are sent by P_0 with positive probability for some input. $\phi_0(x, \alpha)$ is the probability that the message α is sent by P_0 with input x; $\phi_1(x, \alpha)$ is the probability that P_1 with input x outputs 1 upon receiving the message α. Since the particular set of messages is not relevant, ϕ_0 and ϕ_1 can be represented as functions from $\{0, 1\}^n$ to $[0, 1]^k$, where $|M_\phi| = k$. The communication complexity of the protocol ϕ is $\lceil \log_2 k \rceil$; k, the number of distinct messages used, is also called the *length* of the protocol ϕ. Other notions for one-way protocols are defined in an analogous manner.

3. Equivalence of One-way and Two-way Complexities

We exhibit a one-way protocol for each two-way protocol such that both compute the same function and their communication complexities differ by at most 1.

Theorem 1. *Let ϕ be a two-way protocol. Then, there exists a one-way protocol ϕ' such that*

(1) *ϕ and ϕ' compute the same function*

(2) $\tilde{C}_{\phi'} \leqslant \tilde{C}_\phi + 1$.

Proof. Let ϕ_0, ϕ_1, and M_ϕ be as defined earlier for the two-way protocol ϕ. Let $M_\phi = M^1_\phi \cup M^0_\phi$. $\alpha \in M^b_\phi$, if the last bit of α is b. Let $d^b_x = \sum_{\alpha \in M^b_\phi} \Pi(\phi_0(x, \alpha))$ and $d = \max_x d^1_x$.

We define a one-way protocol ϕ' such that the set $M_{\phi'}$ of messages transmitted in ϕ' is $M_\phi \cup \{\gamma\}$ for some message $\gamma \notin M_\phi$. The idea is that the one-way protocol ϕ' simulates the two-way protocol ϕ. In the protocol ϕ', P_0 assumes that P_1 sends its messages with "equal" probabilities, completes the communication according to two-way protocol ϕ without ever having to receive a message from P_1 and sends a record of this simulated communication to P_1 with an appropriate probability. P_1 then "corrects" the probability of the message it receives. The functions ϕ'_0 and ϕ'_1 corresponding to ϕ' are defined

$$\begin{array}{lll}
\phi'_0(x, \alpha) = \dfrac{1}{2d} \Pi(\phi_0(x, \alpha)) & \text{for} & \alpha \in M^1_\phi \\[2ex]
\phi'_0(x, \gamma) = \dfrac{1}{2}\left(1 - \dfrac{d^1_x}{d}\right) & & \\[2ex]
\phi'_0(x, \alpha) = \dfrac{1}{2d^0_x} \Pi(\phi_0(x, \alpha)) & \text{for} & \alpha \in M^0_\phi \\[2ex]
\phi'_1(x, \alpha) = \Pi(\phi_1(x, \alpha)) & \text{for} & \alpha \in M^1_\phi \\[1ex]
\phi'_1(x, \gamma) = 0 & & \\[1ex]
\phi'_1(x, \alpha) = 1 - \dfrac{1}{2d} & \text{for} & \alpha \in M^0_\phi.
\end{array}$$

ϕ'_1 are functions from $\{0, 1\}^n \times M_{\phi'}$ to $[0, 1]$. It can be easily verified that ϕ and ϕ' compute the same function. It is also clear that their complexities differ by at most 1. ∎

From now on, we consider only one-way protocols. All the upper bounds we derive in this paper are upper bounds for one-way protocols. If ϕ is a one-way protocol of length k, let ϕ_0, ϕ_1: $\{0, 1\}^n \to [0, 1]^k$ be the associated probability functions. Let $\hat{\phi}_1 = \phi_1 - (\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})$.

4. Arrangements of Hyperplanes

We present our first characterization of the probabilistic communication complexity in terms of arrangements of hyperplanes.

A hyperplane h in $\mathbf{R}^d$ is a set of points in $\mathbf{R}^d$ specified by some $\mathbf{a} = (a_1, a_2, ..., a_{d+1}) \in \mathbf{R}^{d+1}$, such that $\mathbf{b} \in h$ iff $\langle (a_1, a_2, ..., a_d), \mathbf{b} \rangle = a_{d+1}$ ($\langle \mathbf{s}, \mathbf{t} \rangle$ is the inner product of the vectors $\mathbf{s}$ and $\mathbf{t}$). A hyperplane in $\mathbf{R}^d$ which contains the origin is specified by some $(a_1, a_2, ..., a_d)$. An *arrangement* Arr(H) of hyperplanes is a finite set $H = \{h_1, h_2, ..., h_m\}$ of hyperplanes in $\mathbf{R}^d$ for some d [22]. The *regions* of an arrangement Arr(H) are the nonempty connected components of $\mathbf{R}^d$, when the hyperplanes in H are deleted. Each region r of an arrangement can be characterized by an m bit string whose ith bit (for $i = 1, ..., m$) is 1 iff the region r is in the positive half space of the hyperplane h_i. We call this bit string, the *signature* of the region r. We say that the arrangement Arr(H) *realizes* the set $S_H \subseteq \{0, 1\}^m$ if $S_H = \{w \in \{0, 1\}^m \mid w$ is a signature of some region r in Arr(H)$\}$.

We call each $w \in \{0, 1\}^m$ a *requirement.* A requirement $w \in \{0, 1\}^m$ is *satisfied* by an arrangement Arr(H) of m hyperplanes H in $\mathbf{R}^d$ for some d, if $w \in S_H$. Similarly, we say that a Boolean valued matrix M of order $k \times m$ is satisfied by an arrangement Arr(H) of m hyperplanes H in $\mathbf{R}^d$ if each row of M when viewed as a requirement belongs to S_H.

Let F denote the matrix associated with the Boolean function f, i.e., $F[x, y] = f(x, y)$.

THEOREM 2. *Let F be the matrix of a function f. Let d be the smallest dimension in which there is an arrangement* Arr(H) *of 2^n hyperplanes H that satisfies the matrix F. Then*

$$\lceil \log d \rceil \leqslant \tilde{C}_f \leqslant \lceil \log d \rceil + 1.$$

Proof. We exhibit, for each one-way protocol of length k that computes f, an arrangement of hyperplanes in $\mathbf{R}^k$ that satisfies the matrix F and, for each arrangement of hyperplanes in $\mathbf{R}^k$ that satisfies the matrix F, a one-way protocol of length $k + 2$ to compute the function f. Without loss of generality, let all the rows of F be distinct.

Let ϕ be a one-way protocol of length k that computes f. For each $x_0, x_1 \in \{0, 1\}^n$, $\phi_0(x_0)$ and $\hat{\phi}_1(x_1)$ (defined previously for one-way protocols) can be interpreted as a point, and a hyperplane containing the origin in $\mathbf{R}^k$, respectively. The point $\phi_0(x_0)$ lies in the positive (negative) half space of the hyperplane $\hat{\phi}_1(x_1)$ iff $\langle \phi_0(x_0), \hat{\phi}_1(x_1) \rangle$ is greater than (less than) zero. This means that the point $\phi_0(x_0)$ lies in the positive (negative) half space of the hyperplane $\hat{\phi}_1(x_1)$ iff $f(x_0, x_1) = 1$ (0). Let H be the set of 2^n hyperplanes and P be the set of 2^n points in $\mathbf{R}^k$ obtained from ϕ by the interpretation.

It can now easily be seen that, for each x, the signature of the region in Arr(H) in which the point $\phi_0(x)$ of P lies is the xth row of F. Therefore, the arrangement Arr(H) satisfies the matrix F.

In the other direction, assume that we have an arrangement of 2^n hyperplanes in $\mathbf{R}^k$ which satisfies the matrix F. Let H be the set of these hyperplanes. For each row in F, select a point in the region of the arrangement Arr(H) that corresponds to that row. Let P be the set of these 2^n points. Note that each point in P determines a row in F and each hyperplane in H a column in F. By interpreting these points and hyperplanes as the appropriate probability vectors, we can obtain a one-way protocol to compute f. This we do as follows.

For each hyperplane $\mathbf{a}=(a_1, a_2,..., a_k, a_{k+1}) \in H$, consider the hyperplane $\mathbf{a}'=(a_1,..., a_k, -a_{k+1}, a_{k+1}-\sum_{i=1}^{k} a_i)$ in $\mathbf{R}^{k+2}$ which contains the origin. Let $d=\max_i(abs(a_i'))$. Let $\mathbf{a}''=\frac{1}{2d}\mathbf{a}'$. Note that all the components of $\mathbf{a}''$ are in the interval $[-\frac{1}{2}, \frac{1}{2}]$. Let H'' be the set of all these hyperplanes $\mathbf{a}''$ in $\mathbf{R}^{k+2}$.

For each point $\mathbf{b} \in P$, let $d'=\max(abs(b_1), abs(b_2),..., abs(b_k), 1)$, and let $\mathbf{b}'=(d'+b_1, d'+b_2,..., d'+b_k, d'+1, d')$. Note that all components of $\mathbf{b}'$ are nonnegative and $\mathbf{b}'$ is nonzero. Let $\mathbf{b}''$ be obtained by normalizing $\mathbf{b}'$ to 1. Let P'' be the set of all these points $\mathbf{b}''$ in $\mathbf{R}^{k+2}$.

It is easy to verify that the signature of the region of $\text{Arr}(H'')$ in which a point $\mathbf{b}'' \in P''$ lies corresponds to the row of F determined by the point $\mathbf{b}$. Therefore, the arrangement of hyperplanes $\text{Arr}(H'')$ in $\mathbf{R}^{k+2}$ which contain the origin satisfies the matrix F. Now, the points of P'' and the hyperplanes of H'' can readily be interpreted as the probability vectors of a one-way protocol of length $k+2$ which computes f since, for each $(a_1, a_2,..., a_{k+2}) \in P''$ and $(b_1, b_2,..., b_{k+2}) \in H''$, we have that $0 \leqslant a_i \leqslant 1$, $-\frac{1}{2} \leqslant b_i \leqslant \frac{1}{2}$ and $\sum_{i=1}^{k+2} a_i = 1$ for $1 \leqslant i \leqslant k+2$. ∎

5. Approximations of Matrices

It is possible to give another equivalent characterization using rank 1 real matrices. We say that a real matrix $\hat{F}$ is an *approximation* of a Boolean matrix F of the same order if $\hat{F}[x, y] > 0$, when $F[x, y] = 1$ and $\hat{F}[x, y] < 0$ when $F[x, y] = 0$.

Theorem 3. *Let F be the matrix of a function f. Let d be the smallest number such that there are d rank 1 matrices $\hat{F}_i$ of order $2^n \times 2^n$, and $\hat{F}=\sum_{i=1}^{d} F_i$ is an approximation of F. Then*

$$\lceil \log d \rceil \leqslant \tilde{C}_f \leqslant \lceil \log d \rceil + 1.$$

Proof. Let $\hat{F}$ be an approximation of F such that $\hat{F}=\sum_{i=1}^{d} \hat{F}_i$, where each $\hat{F}_i$ is a rank 1 real matrix. Since $\hat{F}_i$ is a rank 1 matrix, $\hat{F}_i = \mathbf{a}^i \times \mathbf{b}^{iT}$ for some $\mathbf{a}^i$, and $\mathbf{b}^i \in \mathbf{R}^{2^n}$ ($\mathbf{b}^{iT}$ is the transpose of $\mathbf{b}^i$). Let $\mathbf{p}^x = (\mathbf{a}_1^x, \mathbf{a}_2^x,..., \mathbf{a}_d^x)$, and $\mathbf{q}^x = (\mathbf{b}_1^x, \mathbf{b}_2^x,..., \mathbf{b}_d^x)$. Note that $\hat{F}[x_0, x_1] = \langle \mathbf{p}^{x_0}, \mathbf{q}^{x_1} \rangle$. Consider the arrangement $\text{Arr}(H)$ in $\mathbf{R}^d$, where H consists of the hyperplanes $\mathbf{q}^x$ that contain the origin. The regions of $\text{Arr}(H)$ in which the points $\mathbf{p}^x$ lie correspond to the rows of the matrix F. We, therefore, have that the arrangement $\text{Arr}(H)$ satisfies the matrix F. From this arrangement, we can obtain, as in the proof of Theorem 2, a one-way protocol of length $d+2$, i.e., a protocol with $d+2$ messages to compute f.

Similarly, given a one-way protocol of length d to compute f, we can obtain an approximation of the matrix F which is a sum of d rank 1 matrices. ∎

This theorem generalizes a similar result already known for the deterministic communication complexity. Mehlhorn and Schmidt [14] showed that the logarithm of the rank of the matrix F is a lower bound on the two-way deterministic communication complexity. As observed by Kumar and Schnitger, if we define the approximation matrix appropriately for each of the models—deterministic, nondeterministic, bounded error probabilistic, and unbounded error probabilistic—it can be shown that the logarithm of the rank of the approximation matrix minimized over all approximation matrices is a lower bound on the two-way communication complexity.

6. A Logarithmic Lower Bound

Theorem 4. *There exists a function f such that* $\lceil \log_2 n \rceil \leqslant \tilde{C}_f \leqslant \lceil \log_2 n \rceil + 1$.

Proof. Consider the function f defined as

$$\begin{aligned} f(x, y) &= x\text{th bit of } y && \text{for} \quad 0 \leqslant x \leqslant n-1, \\ &= 0 && \text{otherwise.} \end{aligned}$$

Let H be a set of hyperplanes in $\mathbf{R}^d$ that satisfies the matrix F of the function f. Let F' be the submatrix containing the first n columns of F. There exists a subset $H' \subseteq H$ of n hyperplanes in R^d which satisfies the matrix F'. Since F' has 2^n distinct rows, the arrangement $\mathrm{Arr}(H')$ should at least have 2^n distinct regions. The number of distinct regions in any arrangement of n hyperplanes in $\mathbf{R}^d$ is bounded by $\sum_{i=0}^{d} \binom{n}{i}$ [4, 22]. Hence, $d \geqslant n$. This gives us the required lower bound.

Since any arrangement of d hyperplanes in general position in $\mathbf{R}^d$ contains 2^d regions, we also achieve our upper bound. ∎

The theorem can be easily extended to yield a complexity hierarchy for $0 \leqslant \tilde{C} \leqslant \lceil \log n \rceil$.

7. Protocols

In this section, we exhibit a protocol to compute the function equality which requires *two* bits of information transfer and a protocol to compute verification of Hamming distance which requires $2 \log n$ bits of information transfer.

Let the equality function I be $I(x, y) = (x = y)$. A protocol for I can be obtained, by finding a set P of 2^n points and a set H of 2^n hyperplanes in $\mathbf{R}^d$ for some d, such that the arrangement of these hyperplanes satisfies the matrix of I and, for each row in the matrix of I, there exists a point $p \in P$ which lies in a region of $\mathrm{Arr}(H)$ whose signature corresponds to the row. Since the matrix of I contains 1's in the diagonal and 0's elsewhere, any set P of points, and any set H of hyperplanes in R^d such that each $p \in P$ is separated from all the rest of the points in P by a hyperplane in H will yield a one-way protocol for equality. We show below that we can find such P and H in the 2-dimensional plane. From this, by Theorem 2, we can obtain a one-way protocol for equality which uses at most 2 bits of communication.

Protocol for Equality.

The idea is that the points of P are located in the first quadrant of $\mathbf{R}^2$ on the circumference of unit circle with origin as its center. To separate a point from the rest, we take a slightly displaced tangent to the circle at that point. The sets P and H are given below. Let

$$\beta = \pi/2^{n+1} \quad \text{and} \quad \delta = \beta/2,$$
$$P = \{(\cos i\beta, \sin i\beta) \mid 0 \leqslant i \leqslant 2^n - 1\}$$
$$H = \{(\cos i\beta, \sin i\beta, \cos \delta) \mid 0 \leqslant i \leqslant 2^n - 1\}.$$

It is easy to verify that P and H have the desired properties.

In fact, we can obtain a one-way protocol for equality which uses only three different messages and we cannot do with less than three messages. Similarly we can show that there exists a protocol for the function "greater than or equal to" which exchanges only one bit.

Protocol for Verification of Hamming Distance.

Let h_d, for some $d \in \{0, 1, ..., n\}$, be such that $h_d(x, y) = 1$ iff the Hamming distance between x and y is d. The following protocol computes h_d for $d = n/2$ with $2 \log n$ bits of information transfer. Protocols for other d can be devised similarly.

Processor P_0 sends two bits of its input x along with their addresses. Each pair of positions is equally likely to be selected. At P_1, after these two bits are received, one of the two following events, Event I or Event II, occurs such that the Event I happens with probability $1/n$.

Event I. Output 1 if the Hamming distance between the two bits received and the corresponding bits of y is 1; output 0 otherwise.

Event II. Output 1 with probability

$$\left(\frac{1}{2} - \frac{(n-2)(n+2)}{2n^2(n-1)} - \frac{1}{n^4}\right) \Big/ \left(1 - \frac{1}{n}\right).$$

It can be verified that this protocol indeed computes the function $h_{n/2}$.

8. A Linear Lower Bound

In this section, we show that almost all functions have linear probabilistic communication complexity (due to Alon, Frankl, and Rödl). This result follows from an upper bound of Goodman and Pollack [9] on the number of equivalence classes (under order equivalence relation) of simple configurations of points in R^d and our characterization of probabilistic communication complexity in terms of arrangements of hyperplanes. An improved counting argument can be found in [2].

THEOREM 5. *For almost all functions f: $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$,*

$$\tilde{C}_f \geqslant n/2 - \log n/2.$$

This proof is adapted from [2].

Proof. Let F be the matrix of a function f. F can be viewed as a family of 2^n subsets of the set $\{0,..., 2^n-1\}$. Note that if F (the family of subsets represented by F) is realizable in R^d by the points $\mathbf{a}_0,..., \mathbf{a}_{2^n-1}$ and hyperplanes $h_0,..., h_{2^n-1}$, then it is also realizable by $\mathbf{a}'_0,..., \mathbf{a}'_{2^n-1}$ and the same h_i's, whenever $\mathbf{a}'_j$ is sufficiently close to $\mathbf{a}_j$. Hence, we can assume that the points $\mathbf{a}_j$ of a realization are in *general position* in $\mathbf{R}^d$. A set of points in $\mathbf{R}^d$ is said to be in general position if no hyperplane in $\mathbf{R}^d$ contains more than d of these points. We now define two equivalence relations on configurations of points in general position in $\mathbf{R}^d$.

Two ordered sets $\mathbf{p}_1,..., \mathbf{p}_k$ and $\mathbf{q}_1,..., \mathbf{q}_k$ of points in general position in $\mathbf{R}^d$ are *equivalent*, if they can be partitioned by hyperplanes in precisely the same way, i.e., there exists a hyperplane h separating $\mathbf{p}_{j_1},..., \mathbf{p}_{j_r}$ from the rest of the $\mathbf{p}_j$'s if and only if there exists a hyperplane h' separating $\mathbf{q}_{j_1},..., \mathbf{q}_{j_r}$ from the rest of the $\mathbf{q}_j$'s.

A sequence $\mathbf{p}_0,..., \mathbf{p}_d$ of points in $\mathbf{R}^d$ with $\mathbf{p}_i = (p_{i1},..., p_{id})$ is said to have *positive orientation* if

$$\det(p_{ij}) > 0$$

where $p_{i0} = 1$ for each i. The *order type* of an ordered set of points $\mathbf{p}_1,..., \mathbf{p}_k$ in general position in $\mathbf{R}^d$ is the set of all $d+1$-tuples $j_1 < j_2 < \cdots < j_{d+1}$ such that the sequence $\mathbf{p}_{j_1}, \mathbf{p}_{j_2},..., \mathbf{p}_{j_d}$ has positive orientation.

It is easy and well known that $\mathbf{p}_1,..., \mathbf{p}_k$ and $\mathbf{q}_1,..., \mathbf{q}_k$ have the same order type, then they are equivalent. Recently, Goodman and Pollack [9] obtained the asymptotically best possible upper bound on the number of order types of k labeled points in $\mathbf{R}^d$ using a result of Milnor [15] on real algebraic varieties. They prove that the number of order types (and hence the number of equivalence classes) of k labeled points in $\mathbf{R}^d$ is at most $k^{d(d+1)k}$.

We now bound the number of families of m subsets that are realizable by a configuration of points in general position in $\mathbf{R}^d$. From [22], we know that the number of partitions of k points in $\mathbf{R}^d$ into two disjoint subsets separated by a hyperplane is at most $\sum_{i=0}^d \binom{k-1}{i}$. Therefore, the number of families of m subsets of $\{1, 2,..., k\}$ realizable in $\mathbf{R}^d$ is at most $(2\sum_{i=0}^d \binom{k-1}{i})^m$. Therefore, from the result of Goodman and Pollack, the total number of families of m of $\{1, 2,..., k\}$ that can be realized by k points in $\mathbf{R}^d$ is no more than

$$k^{d(d+1)k} \left(2 \sum_{i=0}^{d} \binom{k-1}{i}\right)^m \leqslant k^{d(d+1)k+(d+1)}.$$

Since the total number of families of m subsets of $\{1, 2,..., k\}$ is $\binom{2^k}{m}$, the theorem follows. ∎

9. A Lower Bound for 1-Tape Probabilistic Turing Machines

Consider the unbounded error 1-tape probabilistic Turing machines. We first prove a "bottleneck" lemma which states that an information transfer bottleneck exists in a "short" region of the tape. Using this and the logarithmic lower bound on the probabilistic communication complexity, we establish an $\Omega(n \log n)$ bound on the time of certain 1-tape probabilistic Turing machines.

In a probabilistic Turing machine, the next state is determined by not only the current state and the contents of the cells scanned by the heads but also the outcome of an unbiased coin toss. The concatenation of the outcomes of the coin tosses during a computation obtained by representing heads by 1 and tails by 0 is called a *guess string*. The probabilistic machines we consider have only finite computations. A probabilistic Turing machine M is said to accept (reject) a string x in time t if M stops after at most t steps in all computations with input x and the probability of the event, "M, started in its initial configuration with input x, will enter an accepting (rejecting) configuration" is greater the $\frac{1}{2}$ [7]. A 1-tape probabilistic Turing machine is a probabilistic Turing machine with a single one-way infinite tape.

Let $M(x, g)$ denote the deterministic computation of M with input x and guess string g. To obtain our results, we need to consider crossing sequences in $M(x, g)$ and a "cut and paste" lemma.

A crossing sequence is a maximal list of states that occur at a boundary in a computation. We require that the last state in this list is either an accepting or a rejecting state. This requirement can be satisfied if the Turing machine makes a sweep over the nonblank portion of its work tape, once it accepts or rejects. Let $CS(x, g)$ denote the set of all crossing sequences (along with the boundaries at which they occur) that occur in $M(x, g)$. More precisely, for $i \in N$, and $\bar{q}$, a sequence of internal states of M, $(i, \bar{q}) \in CS(x, g)$ iff $\bar{q}$ is the crossing sequence that occurs at the boundary of tape cells i and $i+1$ in $M(x, g)$. The length of a crossing sequence $(i, \bar{q})$ is the length of the sequence $\bar{q}$. It is clear that the time of the computation $M(x, g)$ is at least $\sum_{(i, \bar{q}) \in CS(x, g)} |\bar{q}|$.

Lemma 1 (cut and paste lemma). *Let $x = x_1 x_2 x_3$ be an input to M with guess string g. Let $CS(x, g)$ be the corresponding set of crossing sequences. Let the crossing sequences at the boundaries of x_1, x_2 and x_2, x_3 be identical. Then, there exists a guess string g' such that the set of crossing sequences $CS(x_1 x_3, g')$ corresponding to the computation $M(x_1 x_3, g')$ is given by*

$$\begin{aligned} (i, \bar{q}) \in CS(x_1 x_3, g') \quad \text{iff} \quad & (i, \bar{q}) \in CS(x, g) \quad \text{and} \quad i \leqslant |x_1| \\ \text{or} \quad & (i + |x_2|, \bar{q}) \in CS(x, g) \quad \text{and} \quad i > |x_1|. \end{aligned}$$

Proof. The guess string g' can be obtained by cutting and pasting the guess string g appropriately. ▌

We now formulate and prove the bottleneck lemma. Let M be an unbounded error probabilistic Turing machine which recognizes a certain language containing strings of the form $x\#^k y$, for some $k \in N^+$, $x, y \in \{0, 1\}^*$. Let $x\#^k y$ be an input to M, such that $|x| = n$ and $|y| = m$. Let the portion of the tape which contains the $\#$'s of the input be called the *bridge* for inputs of the form $x\#^k y$. The length of a bridge is the number of tape cells in it. Assume, for all guess strings g, the time spent by $M(x\#^k y, g)$ in the bridge is less than $\varepsilon k \log k$, for all sufficiently large k, m, and n; ε is a constant less than 1 and can be chosen to be sufficiently small.

Let $CS(x\#^k y, g)$ be the set of all crossing sequences in the computation $M(x\#^k y, g)$. A crossing sequence of $CS(x\#^k y, g)$ is long if its length is at least $c\varepsilon \log k$, for a suitable constant $c > 1$. Otherwise, a crossing sequence is *short*. It is clear that the number of long crossing sequences in the bridge is at most $1/c\, k$.

A *δ-bottleneck* is a region of k^δ consecutive tape cells in a bridge of length k such that a short crossing sequence exists in this region.

LEMMA 2 (bottleneck lemma). *For all δ and g, there exists a δ-bottleneck in the computation $M(x\#^k y, g)$, for all sufficiently large k, m, and n, provided ε and c are chosen appropriately.*

Proof. Consider the bridge. There are at least $(1 - 1/c)\, k$ short crossing sequences in the bridge. Note that the total number S of distinct short crossing sequences is at most $|Q|^{c\varepsilon \log k}$, where Q is the set of internal states of M. By applying the cut and paste lemma to the short crossing sequences of the bridge successively, we can obtain a computation of M with input $x\#^{k'} y$, and with some guess string g' in which the number s of short crossing sequences in the bridge is such that $S < s \leqslant 2S$ Let l be the number of long crossing sequences in the bridge of the resulting computation $M(x\#^{k'} y, g')$. We have that k' is at least $S + l + 1$ and at most $2S + l$.

Since a long crossing sequence is at least $c\varepsilon \log k$ long, the time spent in the bridge is at least $S + lc\varepsilon \log k$ in the computation $M(x\#^{k'} y, g')$. By hypothesis, the time spent in the bridge is at most $\varepsilon(2S + l) \log(2S + l)$ in the computation $M(x\#^{k'} y, g')$. Therefore, we have

$$S + lc\varepsilon \log k \leqslant \varepsilon(2S + l) \log(2S + l)$$

which means

$$l \leqslant k^\delta$$

where $\delta < 1$ can be made to be sufficiently small by selecting c and ε appropriately.

Consider the first $l + 1$ crossing sequences in the bridge in the computation $M(x\#^k y, g)$. One of them must be short. Otherwise, we would have $l + 1$ contiguous long crossing sequences in the bridge of the computation $M(x\#^{k'} y, g')$. This follows from the fact that these $l + 1$ crossing sequences are the leftmost crossing sequences in the bridge of $M(x\#^k y, g)$, and from the fact that we applied the cut and paste lemma only to short crossing sequences to obtain the computation $M(x\#^{k'} y, g')$. This proves the lemma. ▌

We now use the bottleneck lemma, and the logarithmic lower bound on the unbounded error probabilistic communication complexity to prove the following lower bound on the time of 1-tape probabilistic Turing machines.

THEOREM 6. *Let $L = \{x\#^k y \mid x, y \in \{0, 1\}^*, k \in N$ and bit y of x exists and is $1\}$. Then, any 1-tape probabilistic Turing machine M that accepts L takes $\Omega(n \log n)$ steps for an input of length n for infinitely many n.*

Yao [19] has obtained an $\Omega(n \log n)$ lower bound on the time required by certain 1-tape probabilistic Turing machines. However, the definition of acceptance used in [19] is more restrictive (the probability of acceptance is bounded away from $\frac{1}{2}$ by a constant) and the proofs use this restriction in an essential way.

Proof. Consider an input of the form $x\#^n y$, where $|x| = n$ and $|y| = \log n$. Assume that the theorem is not true. We can then find a sufficiently small $\varepsilon > 0$ such that M, with inputs of the form $x\#^n y$, always stops within time $\varepsilon n \log n$ for all sufficiently large n.

By the bottleneck lemma, we can find a δ-bottleneck starting from the leftmost "#" in the bridge of the computation $M(x\#^n y, g)$ for all x, y, and g. This δ-bottleneck contains a short crossing sequence of length at most $c\varepsilon \log n$. Note that the number of distinct short crossing sequences of length $c\varepsilon \log n$ is at most $n^{\delta'}$ for some δ'. $\delta, \delta' < 1$ can be selected to be sufficiently small by selecting ε and c appropriately.

We use these short crossing sequences along with their adresses in the bottleneck to produce a contradiction by obtaining a protocol to compute the function f of Theorem 4 which uses less than $\log n$ bits of communication. This is possible since a short crossing sequence in the bridge can be specified by less than $\log n$ bits.

More precisely, let $K = \{(i, \bar{q}) \mid 1 \leqslant i \leqslant n^{\delta}$, $\bar{q}$ is a sequence of states of length less than $c\varepsilon \log n$ and the last state in the sequence $\bar{q}$ is either an accepting or a rejecting state$\}$. For each $(i, \bar{q}) \in K$, let $p^{x\#^n y}(i, \bar{q})$ be the probability that the crossing sequence $\bar{q}$ occurs at the boundary of the tape cells $n + i$ and $n + i + 1$, when M is given the input $x\#^n y$. $p^{x\#^n y}(i, \bar{q})$ can be factored into two components: $p_0^x(i, \bar{q})$ which depends on x and $p_1^y(i, \bar{q})$ which depends on y. Let $\bar{q} = (q_{j_1}, q_{j_2}, ..., q_{j_r})$.

$p_0^x(i, \bar{q})$ is defined as the probability of the computation: "M is given the input $x\#^i$. $\bar{q}$ is the crossing sequence at the boundary of the cells $n + i$ and $n + i + 1$. Also $\bar{q}$ is the leftmost short crossing sequence in the bridge. Whenever M crosses the boundary of the tape cells $n + i$ and $n + i + 1$ to the right in state $q_{j_{r'}}$ for some odd $r' < r$, M crosses the boundary back to the left (for the first time since its entry to the right in state $q_{j_{r'}}$) in state $q_{j_{r'+1}}$ with probability 1. If M crosses the boundary to the right in state q_{j_r}, it never returns to the left and halts with probability 1."

$p_1^y(i, \bar{q})$ is defined as the probability of the computation: "M is given the input $\mathbf{b}^{n+1}\#^{n-i}y$, where $\mathbf{b}$ is the blank symbol, and M is started in the state q_{j_1} with its head scanning the first "#" symbol. $\bar{q}$ is the crossing sequence at boundary of the cells $n + i$ and $n + i + 1$. Whenever M crosses this boundary to the left in state $q_{j_{r'}}$ for some even $r' < r$, M crosses the boundary back to the right (for the first time since its entry to the left in state $q_{j_{r'}}$) in state $q_{j_{r'+1}}$ with probability 1. If M crosses the boundary to the right in state q_{j_r}, it never returns to the right and halts with probability 1."

Note that $p^{x\#^n y}(i, \bar{q}) = p_0^x(i, \bar{q})\, p_1^y(i, \bar{q})$.

Since the last state in each crossing sequence occurring in a computation is either an accepting or a rejecting one, the probability that M accepts the input $x\#^n y$ can be given by

$$\sum_{\substack{(i,\bar{q}) \in K \\ \bar{q} \text{ is accepting}}} p^{x\#^n y}(i, \bar{q}) = \sum_{\substack{(i,\bar{q}) \in K \\ \bar{q} \text{ is accepting}}} p_0^x(i, \bar{q})\, p_1^y(i, \bar{q}).$$

We now design a one-way probabilistic protocol to compute f in a way similar to Theorem 1. Processor P_0 has the input $x \in \{0, 1\}^n$ and processor P_1 has the input $y \in \{0, 1\}^{\log n}$. For some $\gamma \notin K$, $K \cup \{\gamma\}$ is the set of messages transmitted by P_0. Let K be partitioned into two sets K_a and K_r such that $K = K_a \cup K_r$. $(i, \bar{q}) \in K_a$ (K_r) if and only if $\bar{q}$ is accepting (rejecting) crossing sequence. Let $d_a^x = \sum_{(i,\bar{q}) \in K_a} p_0^x(i, \bar{q})$ and $d_r^x = \sum_{(i,\bar{q}) \in K_r} p_0^x(i, \bar{q})$. Let $d_a = \max_x d_a^x$. P_0, with input x, sends the message $(i, \bar{q}) \in K_a$ with probability $(1/2d_a)\, p_0^x(i, \bar{q})$, and the message $(i, \bar{q}) \in K_r$ with probability $(1/2d_r^x)\, p_0^x(i, \bar{q})$. The message γ will be sent with probability $\frac{1}{2}(1 - d_a^x/d_a)$.

P_1, after receiving the message $(i, \bar{q})$, outputs 1 with probability $p_1^y(i, \bar{q})$ if $\bar{q}$ is accepting, and with probability $1 - 1/2d_a$ if $\bar{q}$ is rejecting. (Note that d_a is independent of the input at the processor P_0.) If P_1 receives γ, it outputs 0 with probability 1.

Therefore, the protocol with inputs x and y outputs 1 with probability

$$\frac{1}{2} - \frac{1}{4d_a} + \frac{1}{2d_a} \sum_{\substack{(i,\bar{q}) \in K \\ \bar{q} \text{ is accepting}}} p_0^x(i, \bar{q})\, p_1^y(i, \bar{q}).$$

This means the protocol outputs 1 iff M accepts the input $x\#^n y$. Therefore, this protocol computes the function f, using less than $\log n$ bits of information transfer since $|K| < n^\alpha$ for some $\alpha < 1$. ▮

10. Linear Arrays of Processors

A linear array of processors is a sequence of processors $P_0, P_1, \ldots, P_k$ such that the adjacent processors are connected by a link or a communication channel. The end processors, P_0 and P_k, have the inputs of a function f which the processors cooperate to compute. We are interested in the total amount of communication (across all links) needed to compute f. The communication complexity C_f^k of a linear array of $k+1$ processors is defined as the total number of bits transmitted across all the links between the processors in the worst case minimized over all protocols that compute the function f. Tiwari [18] investigated this model and conjectured that C_f^k and C_f^1 are related by the equation: $C_f^k = kC_f^1$. (A complete description of this model can be found in [18].) He partially resolved this conjecture by showing that $C_{f_k} = kC_f^1 - k'$ (for some k', $0 \leqslant k' \leqslant k$) whenever the lower bound on C_f^1 is obtained by using one of the general techniques of Lipton and Sedgewick [13] or Mehlhorn and Schmidt [14].

We note the similarity between 1-tape Turing machines and linear array of processors. Boundaries between tape cells on the tape correspond to links between processors, crossing sequences to the sequences of messages exchanged and time to the number of bits exchanged over all links. This similarity indicates that bottleneck lemma is also applicable to linear arrays of processors. Therefore, techniques of Section 9 can be used to obtain the following corollary which says that an extension of Tiwari's conjecture (up to a multiplicative constant) is true for linear arrays of processors in the nondeterministic and the unbounded error probabilistic models.

COROLLARY. *$C_f^k = \Theta(kC_f^1)$ for the nondeterministic and the unbounded error probabilistic models.*

11. CONCLUSIONS AND OPEN PROBLEMS

Our results start a theory of probabilistic information transfer for unbounded error protocols. We provided interesting characterizations, some surprisingly efficient protocols, and a nontrivial lower bound.

It is pleasing that the basic questions about probabilistic information transfer are mathematically interesting. Approximations of matrices by matrices of rank 1 (in a different metric) play an important role in numerical analysis [8], and the decomposition of Euclidean space by hyperplanes is a classical geometric problem [4, 22]. Our lower bounds follow from the basic properties of these objects. Strengthening them would be equivalent to settling certain mathematical problems that are interesting on their own.

The main remaining open problem is to exhibit a function which has superlogarithmic lower bound. The lower bound proof of Alon, Frankl, and Rödl only says that a random function has linear communication complexity. We have done little to settle the problem. On the positive side: consider the problem of verifying whether x and y have the Hamming distance d for some $d \in \{0, ..., n\}$. We exhibited a protocol with $O(\log n)$ information transfer for this problem. Similar techniques yield $O(\log n)$ protocols for other problems. But, the technique fails for the function defined by a Hadamard matrix. We conjecture that this function has maximal (linear) probabilistic communication complexity. Proving this, however, seems to be difficult. Our logarithmic lower bound proof uses counting of regions in $\mathbf{R}^d$: A linear lower bound results from a choice of 2^n regions, that would require the existence of $2^{2^{\Omega(n)}}$ other regions in any arrangement of 2^n hyperplanes that contains these 2^n regions. The choice of orthogonal regions corresponding to the Hadamard matrix of order $2^n \times 2^n$ seems to be a suitable one, and hence the conjecture.

Acknowledgments

We have benefited from the work of P. Berman, G. Schnitger, and V. K. Prasanna Kumar, and from extensive discussions with the first two. In particular, the idea of using rank 1 matrices is due to Prasanna Kumar and Schnitger, who obtained an (essentially equivalent) lower bound. The efficient protocol for the verification of Hamming distance is suggested by P. Berman, who also helped in clarifying some of our ideas. We are grateful for their help. We also bothered a large number of mathematicians, believing that they must have solved our combinatorial problems long time ago. A partial list of these helpful mathematicians is P. Rejto, R. M. Hardt, B. Grunbaum (who suggested interesting ways to solve the open problem), J. Edmonds, S. Friedland, R. Bland, L. Vaserstein, and N. Alon. We appreciate their patience, and the time they spent on our questions. Finally, we thank the anonymous referees of our paper for their helpful comments.

References

1. A. V. Aho, J. D. Ullman, and M. Yannakakis, On notions of information transfer in VLSI circuits, *in* "Proceedings, 15th Ann. ACM Sympos. Theory of Computing," Boston, Massachusetts, 1983, pp. 133–139, Assoc. Comput. Mach., New York, 1983.
2. N. Alon, P. Frankl, and V. Rödl, Geometrical realization of set systems and probabilistic communication complexity, *in* "26th Ann. Sympos. Found. of Comput. Sci.," IEEE Computer Society, New York, 1985.
3. M. Ben-Or, private communication, May 1984.
4. R. C. Buck, Partition of space, *Amer. Math. Monthly* **50** (1943), 541–544.
5. P. Duris, Z. Galil, and G. Schnitger, Lower bounds on communication complexity, *in* "Proceedings, 16th Ann. ACM Sympos. Theory of Computing," Washington, D. C., 1984, pp. 81–91, New York, Assoc. Comput. Mach., 1984.
6. J. Folkman and J. Lawrence, Oriented matroids, *J. Combin. Theory Ser. B* **25** (1978), 199–236.
7. J. T. Gill, III, Computational complexity of probabilistic Turing machines, *SIAM J. Comput.* **6** (1977), 675–695.
8. G. H. Golub and C. F. Van Loan, "Matix Computations," Johns Hopkins Press, Baltimore, 1983.
9. J. E. Goodman and R. Pollack, Upper bounds for configurations and polytopes in R^d, to appear.
10. J. Ja' Ja', V. K. Prasanna Kumar, and J. Simon, Information transfer under different sets of protocols, *SIAM J. Comput.* **31**, No. 1 (1984), 150–162.
11. M. Klawe, W. J. Paul, N. Pippenger, and M. Yannakakis, On monotone formulae with restricted depth, *in* "Proceedings, 16th Annu. ACM Sympos. Theory of Computing," Washington, D. C., 1984, pp. 480–487, New York, Assoc. Comput. Mach., 1984.
12. J. Lawrence, Lopsided sets and orthant-intersection by convex sets, *Pacific J. Math.* **104**, No. 1 (1983), 155–173.
13. R. J. Lipton and R. Sedgewick, Lower bounds for VLSI, *in* "Proceedings, 13th Annu. ACM Sympos. Theory of Computing," 1981, pp. 300–307, Assoc. Comput. Mach., New York, 1981.
14. K. Mehlhorn and E. K. Schmidt, Las Vegas is better than determinism in VLSI and distributed computing (extended abstract), *in* "Proceedings, 14th Annu. ACM Sympos. Theory of Computing," Los Angeles, 1982, pp. 330–337, New York, Assoc. Comput. Mach., 1982.
15. J. Milnor, On the Betti numbers of real varieties, *Proc. Amer. Math. Soc.* **15** (1964), 275–280.

16. C. H. Papadimitriou and M. Sipser, Communication complexity, *in* "Proceedings, 14th Annu. ACM Sympos. Theory of Computing," Los Angeles, 1982, pp. 196–200, New York, Assoc. Comput. Mach., 1982.
17. C. D. Thompson, Area-time complexity for VLSI, *in* "Proceedings, 11th Annu. ACM Sympos. Theory of Computing," Atlanta, Georgia, 1979, pp. 81–88, New York, Assoc. Comput. Mach., 1979.
18. P. Tiwari, Lower bounds on communication complexity in distributed computer networks, *in* "25th Annu. Sympos. Found. of Comput. Sci.," pp. 109–117, IEEE Computer Society, New York, 1984.
19. A. C.-C. Yao, "A Lower Bound to Palindrome Recognition by Probabilistic Turing Machines," Technical Report, Computer Science Department, Stanford University, December 1977.
20. A. C.-C. Yao, Some complexity questions related to distributive computing, *in* "Proceedings, 11th Annu. ACM Sympos. Theory of Computing," Atlanta, Georgia, 1979, pp. 209–213, New York, Assoc. Comput. Mach., 1979.
21. A. C.-C. Yao, Lower bounds by probabilistic arguments, *in* "24th Annu. Sympos. Found. of Computer Sci.," pp. 420–428, IEEE Computer Society, New York, 1983.
22. T. Zaslavsky, Facing up to arrangements: Face-count formulas for partitions of space by hyperplanes," *Mem. Amer. Math. Soc.* **154** (1975).

Printed in Belgium