Reprinted from INFORMATION AND COMPUTATION

Vol. 116, No. 2, February 1, 1995
Printed in Belgium

A Communication-Randomness Tradeoff for Two-Processor Systems*

RUDOLF FLEISCHER

MPI für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany
E-mail: rudolf@mpi-sb.mpg.de

HERMANN JUNG

Fachbereich Informatik, Humboldt Universität Berlin, O-1086 Berlin, Germany

AND

KURT MEHLHORN

MPI für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany

We present a tight tradeoff between the expected communication complexity $\bar{C}$ (for a two-processor system) and the number R of random bits used by any Las Vegas protocol for the list-nondisjointness function of two lists of n numbers of n bits each. This function evaluates to 1 if and only if the two lists correspond in at least one position. We show a $\log(n^2/\bar{C})$ lower bound on the number of random bits used by any Las Vegas protocol, $\Omega(n) \leqslant \bar{C} \leqslant O(n^2)$. We also show that expected communication complexity $\bar{C}$, $\Omega(n \log n) \leqslant \bar{C} \leqslant O(n^2)$, can be achieved using no more than $\log(n^2/\bar{C}) + \lceil \log(2 + \log(n^2/\bar{C})) \rceil + 6$ random bits.

1. INTRODUCTION

The use of randomness has led, for a variety of problems, to algorithms which are more efficient than the best known deterministic algorithms. This suggests the following question: How much does a single random bit actually help?

One approach [KPU] to this question is to treat randomness as a resource and to analyze the amount of randomness needed to obtain a certain performance. If successful, this approach leads to tradeoffs between randomness and other complexity measures such as computation time and memory requirement. In this paper we prove such a tradeoff in the context of communication complexity. Previously a tradeoff between randomness and computation time was shown for oblivious routing in computer networks [KPU] and between randomness and memory for caching algorithms [RS89].

Recently, Canetti and Goldreich [CG] investigated tradeoffs between randomness and communication complexity for Monte Carlo protocols, i.e., protocols which are allowed to give wrong answers with some probability less than half. They proved tight tradeoffs between communication complexity and the number of random bits. Their results are related to ours in the following sense: Roughly speaking, if you truncate a Las Vegas protocol after $\bar{C}$ bits you get a Monte Carlo protocol with the same complexity. Hence lower bounds for Monte Carlo protocols also imply lower bounds for Las Vegas protocols (but not vice versa!). In fact, Theorem 2.b) in [CG] in the special setting of Las Vegas protocols implies our Theorem 4.2.

The communication complexity of Boolean functions is a complexity measure corresponding to the amount of information transfer necessary to compute the function (see [AUY, F87, F89, HR, LS, MS, PS, and Y79] for detailed definitions). Let $f: X_1 \times X_2 \to \{0, 1\}$ be a Boolean function. Assume that P_1 and P_2 are two processors such that P_i knows the argument $x_i \in X_i$, $i = 1, 2$. The processors now exchange messages according to some probabilistic protocol. The protocol terminates when both processors have correctly determined the function value $f(x_1, x_2)$. This type of protocol is called a Las Vegas protocol (uses randomness but always computes the correct answer). In this paper we only study Las Vegas protocols so we just call them protocols.

For a given protocol let $r_i(x_1, x_2)$ be the number of random bits used by P_i in the worst case on input (x_1, x_2) (maximized over all possible sequences of random bits). Let $m(x_1, x_2)$ be the number of bits exchanged between P_1 and P_2 on input (x_1, x_2) in the worst case (maximized over all possible sequences of random bits) and $\bar{m}(x_1, x_2)$ the expected number of bits exchanged. Let $p_t(x_1, x_2)$ be the probability that at least t bits are exchanged on input (x_1, x_2). The following quantities measure the performance of a protocol.

* This work was supported by the DFG, SFB 124, TP B2, VLSI-Entwurf und Parallelität and ESPRIT P3075 ALCOM.

0890-5401/95 $6.00

Randomness R, i.e., the maximal number of random bits used on any input (x_1, x_2). $R = \max_{(x_1, x_2)} (r_1(x_1, x_2) + r_2(x_1, x_2))$.

Communication complexity C, i.e., a bound on the number of bits exchanged by the processors in the worst case. $C = \max_{(x_1, x_2)} m(x_1, x_2)$.

The deterministic communication complexity (i.e., $R = 0$) is denoted by C_{det}.

Expected communication complexity $\bar{C}$, i.e., the maximal expected number of bits exchanged for any input (x_1, x_2). $\bar{C} = \max_{(x_1, x_2)} \bar{m}(x_1, x_2)$.

Failure probability Q_t, i.e., the maximum (over all inputs (x_1, x_2)) of the probability that the algorithm exchanges at least t bits. $Q_t = \max_{(x_1, x_2)} p_t(x_1, x_2)$.

Another measure for randomness is the entropy of a random source [KY]. We remark that, analogously to [KPU], our lower bounds in Section 4 (as well as the upper bounds) hold for this measure also instead of R.

We prove a tradeoff between randomness, expected communication complexity, and failure probability for the list-nondisjointness (LND) function. Let $X_1 = X_2 = X^n$ where $X = \{0, 1\}^n$. For $(x_1, ..., x_n)$, $(y_1, ..., y_n) \in X^n$ define $\text{LND}((x_1, ..., x_n), (y_1, ..., y_n)) := 1$ if there exists a j with $x_j = y_j$, and 0 otherwise. Note that the inputs to both processors consist of n^2 bits each. It is known [MS] that any deterministic algorithm (i.e., $R = 0$) for LND must exchange n^2 bits in the worst case (i.e., $C_{\text{det}} \geq n^2$, and if $t = o(n^2)$ then $Q_t > 0$) and that there is a probabilistic algorithm with $\bar{C} = O(n)$ [F87]. It is also known that this quadratic gap between deterministic and Las Vegas protocols is maximal [AUY].

We show in this paper the following

THEOREM A (Lower Bound). *Let* $0 \leq r \leq \log n$.[1]

(a) *Any protocol for LND that exchanges less than* $t = n^2/2^r$ *bits with probability at least* $1 - Q$, $0 < Q < 1$, *requires randomness* $R \geq r - \log Q = \log(n^2/(tQ))$.

(b) *Any protocol for LND that exchanges an expected number of* $\bar{C} = n^2/2^r$ *bits requires randomness* $R \geq r = \log(n^2/\bar{C})$.

In fact, we show a more general bound which applies to any function f (see Theorems 4.2 and 4.3).

THEOREM B (Nonconstructive Upper Bound). *For* $1 \leq r \leq \log n - \log\log n - 5$ *a protocol exists which uses* $R = r + \lceil \log r \rceil + 4$ *random bits and exchanges an expected number of* $\bar{C} \leq n^2/2^{r-2}$ *bits. In other words,* $R \leq \log(n^2/\bar{C}) + \lceil \log(2 + \log(n^2/\bar{C})) \rceil + 6$ *for* $2^7 \cdot n \log n \leq \bar{C} \leq n^2$.

[1] log denotes a logarithm of base 2, whereas ln denotes the natural logarithm of base e.

Theorem B implies that one additional random bit reduces the expected communication complexity of LND by nearly a factor of $\frac{1}{2}$ which is optimal by Theorem A. The proof of Theorem B involves a counting argument and is in this sense non-constructive. We also have an explicit construction which does not, however, give quite as good bounds.

THEOREM C (Explicit Upper Bound). *For* $1 \leq r \leq \log n - \log\log n - 1$ *a simple protocol with randomness* $R = 2r$ *and expected communication complexity* $\bar{C} \leq n^2/2^{r-2}$ *exists. In other words,* $R \leq 2\log(n^2/\bar{C}) + 4$ *for* $8n \log n \leq \bar{C} \leq n^2$.

This paper is organized as follows. In Section 2 we restate Fürer's protocol for LND [F87]. In Section 3 we improve it with respect to the number of random bits used (thus proving Theorem C) and prove the existence of an even more efficient protocol (Theorem B). In Section 4 we finally prove the lower bounds (Theorem A), and we conclude with some open problems in Section 5.

2. THE OLD ALGORITHM

Let $x_1, ..., x_n, y_1, ..., y_n$ be n-bit integers (i.e., $0 \leq x_i, y_i < 2^n$). Then the list-nondisjointness function LND is defined as

$$\text{LND}((x_1, ..., x_n), (y_1, ..., y_n)) := \begin{cases} 1 & \text{if } \exists j: x_j = y_j \\ 0 & \text{if } \forall j: x_j \neq y_j. \end{cases}$$

Fürer's $\bar{C}$-optimal Las Vegas algorithm for LND [F87] uses $\Omega(n)$ random bits, which is far away from the lower bound stated in Section 4. However, we will use the main idea of his algorithm to prove our much more randomness-efficient results, so we first present Fürer's algorithm. We remark that Fürer was only interested in an efficient protocol and did not care about randomness.

Let $2^{k-1} \leq n < 2^k$ for a $k \in \mathbb{N}$. Let $\mathcal{P}_i := \{p \text{ prime} \mid 2^{2^{i-1}} \leq p < 2^{2^i}\}$, i.e., all $p \in \mathcal{P}_i$ have binary length between $2^{i-1} + 1$ and 2^i.

ALGORITHM A ([F87]).

(0) If $k \leq 4$ then P_1 sends $x_1, ..., x_n$ to P_2 which can now compute the result.

Otherwise:

(1) P_1 chooses for $i = 4, ..., k-1$ a prime p_i from $\mathcal{P}_i$ at random with uniform distribution and sends it to P_2. Then P_1 computes for all j, $1 \leq j \leq n$, the sequence $x_j^k := x_j$, $x_j^i := x_j^{i+1} \bmod p_i$, $i = k-1, ..., 4$. P_2 does the same with the y_j.

(2) P_1 continues as follows:

```
j := 0;
repeat
  j := j + 1;
  i := 3;
  repeat      (* inner loop checks x_j ?= y_j *)
    i := i + 1;
    P_1 sends x_j^i to P_2;
    P_2 answers 1 if x_j^i = y_j^i, 0 otherwise;
  until x_j^i ≠ y_j^i or i = k;
until x_j^i = y_j^i or j = n;      (* x_j^i = y_j^i only if i = k, i.e., x_j = y_j *)
if x_j^i = y_j^i then f((x_1, ..., x_n), (y_1, ..., y_n)) := 1
else f((x_1, ..., x_n), (y_1, ..., y_n)) := 0;
```

Observe that $x_j^i \neq y_j^i$ implies $x_j \neq y_j$. Hence the inner loop checks $x_j \stackrel{?}{=} y_j$ correctly.

For the analysis of the expected communication complexity of Algorithm A and the other algorithms we need the following two lemmas which generalize [F87].

LEMMA 2.1. *$|\mathscr{P}_i| > 2^{2^i - i - 2}$ for $i \geq 4$.*

Proof. We know by the prime number theorem for $x \geq 12$ [G72, HW],

$$|\{\text{primes less than or equal to } x\}| \begin{cases} \geq \frac{1}{2} \cdot \ln 2 \cdot \frac{x}{\ln x} = \frac{1}{2} \cdot \frac{x}{\log x} \\ \leq 12 \cdot \ln 4 \cdot \frac{x}{\ln x} = 24 \cdot \frac{x}{\log x}. \end{cases}$$

Therefore

$$\begin{aligned} |\mathscr{P}_i| &\geq \frac{1}{2} \cdot \frac{2^{2^i}}{2^i} - 24 \cdot \frac{2^{2^{i-1}}}{2^{i-1}} \\ &= \frac{2^{2^i} - 3 \cdot 2^5 \cdot 2^{2^{i-1}}}{2^{i+1}} \\ &> \frac{2^{2^i} - 2^{2^{i-1}+7}}{2^{i+1}} \\ &\geq 2^{2^i - i - 2} \qquad \text{for} \quad i \geq 4. \quad \blacksquare \end{aligned}$$

LEMMA 2.2. *If $z_1 \neq z_2$ are m-bit integers and $\mathscr{P}$ is a set of primes, each with binary length at least l, then there are at most $\lfloor (m-1)/(l-1) \rfloor$ different primes $p \in \mathscr{P}$ with $z_1 \equiv z_2$ modulo each of them.*

Proof. Otherwise we would have $z_1 \equiv z_2$ modulo the product of these primes; but since this product has at least $(l-1) \cdot (\lfloor (m-1)/(l-1) \rfloor + 1) + 1 \geq m + 1$ bits, this would imply $z_1 = z_2$.

Analysis of Algorithm A. If $k \leq 4$, then P_1 uses no random bits and the communication complexity is $n^2 + 1 \leq 16n$. If $k \geq 5$, then P_1 uses in step (1) at least $\sum_{i=4}^{k-1} \log |\mathscr{P}_i| \geq \sum_{i=4}^{k-1} (2^i - i - 2) \geq n/2 - 6$ random bits and also sends them to P_2. On the other hand, it is sufficient for P_1 to use $2^4 + 2^5 + \cdots + 2^{k-1} \leq 2^k \leq 2n$ random bits (of course, a slightly better bound could be obtained by using upper bounds on $\mathscr{P}_i$ similar to the proof of Lemma 2.1).

In step (2) first assume $x_j \neq y_j$. Given $x_j^{i+1} \neq y_j^{i+1}$, the relative probability for $x_j^i = y_j^i$ is at most $3/2^{2^i - i - 2}$ by Lemmas 2.1 and 2.2 (with $m = 2^{i+1}$ and $l = 2^{i-1} + 1$). In this case at most $\sum_{l=4}^{i+1} 2^l \leq 2^{i+2}$ bits are sent by P_1 to P_2 until $x_j \neq y_j$ is recognized; P_2 sends only $i - 2$ one-bit answers. x_1^4 and P_2's first answers are always sent. Therefore the expected communication complexity of the inner loop is less than 18 if $x_j \neq y_j$:

$$\begin{aligned} \bar{C}_{x_j \neq y_j} &\leq \sum_{i=4}^{k-1} \frac{3}{2^{2^i - i - 2}} \cdot (2^{i+2} + (i-2) \cdot 1) + 2^4 + 1 \\ &< \sum_{i=4}^{k-1} \frac{3 \cdot 2^{i+3}}{2^{2^i - i - 2}} + 17 \\ &< \sum_{i=4}^{k-1} \frac{1}{2^{2^i - 2i - 7}} + 17 \\ &< 18. \end{aligned}$$

If $x_j = y_j$, then P_1 sends $\sum_{i=4}^{k} 2^i \leq 4n$ bits at most (and P_2 sends $k - 3$ one-bit answers), but this can only happen once because the algorithm stops afterwards. So we have proven

THEOREM 2.3. *If $n \geq 16$ (i.e., $k \geq 5$) then Algorithm A has expected communication complexity $\bar{C} \leq 23n$ and uses between $n/2 - 6$ and $2n$ random bits.*

3. IMPROVED ALGORITHMS

It is easy to reduce the number of random bits in Algorithm A as follows:

ALGORITHM B. Run Algorithm A, but do not use primes with length more than $2\lceil \log n \rceil$, i.e., restrict i to the range $4 \leq i \leq 1 + \lceil \log \log n \rceil$.

THEOREM 3.1. *Algorithm B has expected communication complexity $\bar{C} = O(n)$ and randomness R with $2 \log n - 6 \leq R \leq 8 \log n$.*

Proof. Substitute $k' = \lceil \log \log n \rceil + 2$ for $k = \lceil \log n \rceil$ in the analysis of Algorithm A. Further observe that the relative probability for $x_j \equiv y_j \bmod p_{k'-1}$ given $x_j \neq y_j$ is less than $(n/2^{\lceil \log \log n \rceil})/2^{2 \log n - \lceil \log \log n \rceil - 3} = 2^3/n$ by Lemmas 2.1 and 2.2 (because $p_{k'-1}$ has length $2^{\lceil \log \log n \rceil} + 1$ at least). Therefore the expected number of bits exchanged for checking $x_j \neq y_j$ is still $O(1)$. $\blacksquare$

However, our main goal is a tradeoff between the number of random bits and the expected communication com-

plexity. We will first present an easy algorithm which is optimal up to a factor of two for the number r of random bits used (if this number is not too big). It is a direct generalization of an algorithm proposed by [RY].

ALGORITHM C. Let S be any set of 4^r primes of length $\lceil n/2^r \rceil + 1$ each. S is known to both P_1 and P_2.

(1) P_1 chooses a prime p from S at random and tells P_2 which one was chosen.

(2) P_1 sends $x_j \bmod p$ to P_2 for all j, and if $x_j \equiv y_j \bmod p$, then it also sends x_j. If $x_j = y_j$, then the algorithm halts.

THEOREM 3.2 (Theorem C). *Algorithm C can be applied if $n \geqslant 16$ and $1 \leqslant r \leqslant \log n - \log\log n - 1$. It uses $2r$ random bits and has expected communication complexity $\bar{C} \leqslant n^2/2^{r-2}$.*

Proof. From $2^r \leqslant n/2 \log n$ and $n \geqslant 16$ follows $x := 2^{\lceil n/2^r \rceil} \geqslant 2^{2\log n} \geqslant 2^8 > 21$. Hence we can apply formula (3.8) of [RS62] and get

$$\begin{aligned} &|\{\text{primes of length } \lceil n/2^r \rceil + 1\}| \\ &\quad = \pi(2x) - \pi(x) > \frac{3}{5 \ln 2} \cdot \frac{x}{\log x} \geqslant \frac{4}{5} \cdot 2^{n/2^r} \cdot \frac{2^r}{n} \\ &\quad \geqslant \frac{4}{5} \cdot 2^{2\log n + r - \log n} \geqslant \frac{4}{5} \cdot 2^{2r + \log\log n + 1} \geqslant 4^r. \end{aligned}$$

Hence the set S of primes exists as required. Observe next that for $p \in S$

$$\text{prob}(x_j \equiv y_j \bmod p \text{ if } x_j \neq y_j) \leqslant \frac{n}{n/2^r} \cdot \frac{1}{4^r} = \frac{1}{2^r}$$

by Lemma 2.2. Therefore an expected number of $\lceil n/2^r \rceil + 2 + (1/2^r) \cdot (n+1)$ bits have to be exchanged to check $x_j \neq y_j$. If $x_j = y_j$, then $\lceil n/2^r \rceil + 2 + n + 1$ bits are exchanged. ∎

Algorithm C uses twice as many random bits as the lower bound requires (Theorem 4.2). We will now show that $\bar{C} = O(n^2/2^r)$ can already be obtained with $r + o(r)$ random bits; this is optimal up to lower order terms. Unfortunately, we have no explicit construction of such an algorithm. We can only prove its existence by a counting argument.

The algorithm is a refinement of Algorithm A. The main idea is to restrict the choice of random primes to small subsets of $\mathscr{P}_i$. Also, because we aim at communication complexity $O(n^2/2^r)$, we only need the modulo cascade from $\mathscr{P}_{k-1}$ to $\mathscr{P}_{k-r}$ (any $p \in \mathscr{P}_{k-r}$ has $2^{k-r} \leqslant n/2^{r-1}$ bits at most, so we can afford to send x_j^{k-r}).

DEFINITION 3.3. Let $\mathscr{P}$ be the set of all sequences $s = (p_{k-1}, \ldots, p_{k-r})$ with primes $p_j \in \mathscr{P}_j$, and let (x, y) be a pair of numbers with $x \neq y$.

(1) A sequence $s \in \mathscr{P}$ is called *i-bad* for (x, y) if

$$\begin{aligned} &(\cdots(x \bmod p_{k-1})\cdots) \bmod p_i \\ &\quad = (\cdots(y \bmod p_{k-1})\cdots) \bmod p_i. \end{aligned}$$

Let $b_i(x, y)$ be the number of i-bad sequences for (x, y).

(2) A subset $\mathscr{L} \subseteq \mathscr{P}$ is called *bad* for (x, y) if an i exists such that at least $Q_i \cdot L$ sequences of $\mathscr{L}$ are i-bad for (x, y); here $L := |\mathscr{L}|$ and $Q_i := n/(r \cdot 2^{r+i+1}) \leqslant \frac{1}{2}$.

THEOREM 3.4. *If we restrict Algorithm A to sequences s of an $\mathscr{L} \subseteq \mathscr{P}$ which is good for all (x, y), $x \neq y$, and $0 \leqslant x, y < 2^n$, we get expected communication complexity $\bar{C} \leqslant 3n^2/2^r + (r+2) \cdot n + \lceil \log L \rceil$ and use only $\lceil \log L \rceil$ random bits.*

Proof. $\lceil \log L \rceil$ bits are required to transmit the choice of the sequence s. For fixed $x_j \neq y_j$ and all i there are at most $\lfloor Q_i \cdot L \rfloor$ sequences of $\mathscr{L}$ which are i-bad for (x_j, y_j), i.e., $x_j^i = y_j^i$. Hence P_1 sends an expected number of

$$\frac{n}{2^{r-1}} + \sum_{i=k-r}^{k-1} \frac{\lfloor Q_i \cdot L \rfloor}{L} \cdot 2^{i+1} \leqslant \frac{n}{2^{r-1}} + \sum_{i=k-r}^{k-1} \frac{n \cdot 2^{i+1}}{r \cdot 2^{r+i+1}} = \frac{3n}{2^r}$$

bits to check $x_j \neq y_j$ (and P_2 sends at most r one-bit answers).

If $x_j = y_j$, then at most $2n + r$ bits are exchanged. ∎

THEOREM 3.5. *If $r \leqslant \log n - \log\log n - 5$, then an $\mathscr{L} \subseteq \mathscr{P}$, $|\mathscr{L}| = r \cdot 2^{r+4}$, exists which is good for all (x, y), $x \neq y$ and $0 \leqslant x, y < 2^n$.*

COROLLARY 3.6 (Theorem B). *If $r \leqslant \log n - \log\log n - 5$, then there exists a protocol for LND with expected communication complexity $\bar{C} \leqslant n^2/2^{r-2}$ which uses only $r + \lceil \log r \rceil + 4$ random bits.*

To prove Theorem 3.5 we first need two technical lemmas.

LEMMA 3.7. *$b_i(x, y) \leqslant 6 \cdot (|\mathscr{P}|/|\mathscr{P}_i|)$ for $i \geqslant 4$.*

Proof. We have by Lemma 2.2

$$b_{k-1}(x, y) \leqslant 3 \cdot \frac{|\mathscr{P}|}{|\mathscr{P}_{k-1}|}$$

and

$$b_l(x, y) \leqslant b_{l+1}(x, y) + |\mathscr{P}_{k-1}| \cdots |\mathscr{P}_{l+1}| \cdot 3 \cdot |\mathscr{P}_{l-1}| \cdots |\mathscr{P}_{k-r}|.$$

Because of $|\mathscr{P}_{l+1}| \geqslant 2 \cdot |\mathscr{P}_l|$ (Lemma 2.1 together with $|P_l| \leqslant 2^{2^l}$) we conclude

$$b_i(x, y) \leqslant 3 \cdot \left(\frac{|\mathscr{P}|}{|\mathscr{P}_{k-1}|} + \cdots + \frac{|\mathscr{P}|}{|\mathscr{P}_i|} \right) \leqslant 6 \cdot \frac{|\mathscr{P}|}{|\mathscr{P}_i|}. \quad ∎$$

LEMMA 3.8. *If* $\log n \geqslant i \geqslant \log n - r \geqslant 5$, $r \leqslant \log n - \log\log n - 5$, $L = r \cdot 2^{r+4}$ *and* $1 \leqslant c \leqslant \min\{2^7, n^2\}$, *then*

$$2^{2n} \cdot \left(2^4 \cdot \left(\frac{c}{Q_i \cdot |\mathscr{P}_i|}\right)^{Q_i}\right)^L < \frac{1}{2r}.$$

Proof. The proof consists of a long but easy calculation which is divided into two preliminary parts (1) and (2) followed by the main proof (3).

(1) Since $i \leqslant \log n$ and $\log c \leqslant 2 \log n$ we have

$$\begin{aligned} i &\geqslant \log n - r \geqslant \log\log n + 5 \\ &= \log(2\log n + 2\log n) + 3 \\ &\geqslant \log[\log n + (\log n - \log\log n - 5) + 3 \\ &\quad + \log(\log n - \log\log n - 5) + \log c] + 3 \\ &\geqslant \log(\log n + r + \log r + \log c + 3) + 3 \\ &\geqslant \log(\underbrace{2i - \log n + r + \log r + \log c + 3}_{=: d}) + 3. \end{aligned}$$

Hence $2^i - d \geqslant 2^i - 2^{i-3}$.

(2) Let

$$a := \frac{1}{2^i \cdot Q_i} = \frac{r \cdot 2^{r+1}}{n} \leqslant \frac{\log n \cdot 2^{\log n - \log\log n - 4}}{n} = \frac{1}{16}.$$

Then for $c \leqslant 2^7$ and $i \geqslant 5$:

$$\begin{aligned} |\mathscr{P}_i| &\geqslant 2^{2^{i-i-2}} > 2^{2^{i-1} + i + \log c + \log a} \\ &\geqslant 2^{8a2^i} \cdot c \cdot a \cdot 2^i \\ &= c \cdot 2^{8/Q_i} \cdot \frac{1}{Q_i}. \end{aligned}$$

Hence $Q_i \cdot \log(Q_i \cdot |\mathscr{P}_i|/c) > 8$ and $Q_i \cdot \log(Q_i \cdot |\mathscr{P}_i|/c) - 4$ is positive.

(3) It follows that

$$\begin{aligned} &\frac{2n + \log 2r}{Q_i \cdot \log(Q_i \cdot |\mathscr{P}_i|/c) - 4} \\ &\leqslant \frac{2n + 1 + \log\log n}{(n/r \cdot 2^{r+i+1}) \cdot \log((n/r \cdot 2^{r+i+1}) \cdot (2^{2^{i-i-2}}/c)) - 4} \\ &\leqslant \frac{4 \cdot n \cdot r \cdot 2^{r+i+1}}{n \cdot (2^i - d) - 4 \cdot r \cdot 2^{r+i+1}} \\ &\overset{(1)}{\leqslant} \frac{n \cdot r \cdot 2^{r+i+3}}{n \cdot (2^i - 2^{i-3}) - a \cdot n \cdot 2^{i+2}} \\ &\overset{(2)}{<} \frac{r \cdot 2^{r+i+3}}{2^i - 2^{i-2} - 2^{i-2}} \\ &= r \cdot 2^{r+4} \\ &= L \\ &\Rightarrow \quad L \cdot \left(Q_i \cdot \log \frac{Q_i \cdot |\mathscr{P}_i|}{c} - 4\right) > 2n + \log 2r \\ &\Rightarrow \quad 2n + 4L + L \cdot Q_i \cdot \log \frac{c}{Q_i \cdot |\mathscr{P}_i|} < \log \frac{1}{2r}. \quad \blacksquare \end{aligned}$$

Proof of Theorem 3.5. We show that the fraction of $\mathscr{L}$'s which are bad for some pair (x, y), $x \neq y$ and $0 \leqslant x, y < 2^n$, is less than $\frac{1}{2}$.

For fixed (x, y) an $\mathscr{L} \subseteq \mathscr{P}$ is bad for (x, y) if there is an i, $k - r \leqslant i \leqslant k - 1$, such that at least $\lceil Q_i \cdot L \rceil$ sequences of $\mathscr{L}$ belong to the at most $6 \cdot |\mathscr{P}|/|\mathscr{P}_i|$ sequences which are i-bad for (x, y) (Lemma 3.7). Hence

$$\begin{aligned} \frac{\#\text{ bad } \mathscr{L}}{\#\mathscr{L}} &\leqslant \left(2^{2n} \cdot \sum_{i=k-r}^{k-1} \binom{6 \cdot |\mathscr{P}|/|\mathscr{P}_i|}{\lceil Q_i \cdot L \rceil} \right. \\ &\qquad \left. \cdot \binom{|\mathscr{P}|}{L - \lceil Q_i \cdot L \rceil}\right) \Big/ \binom{|\mathscr{P}|}{L} \\ &\leqslant 2^{2n} \cdot \sum_{i=k-r}^{k-1} \left(\frac{6 \cdot |\mathscr{P}| \cdot e}{|\mathscr{P}_i| \cdot \lceil Q_i \cdot L \rceil}\right)^{\lceil Q_i \cdot L \rceil} \\ &\qquad \cdot \left(\frac{|\mathscr{P}| \cdot e}{L - \lceil Q_i \cdot L \rceil}\right)^{L - \lceil Q_i \cdot L \rceil} \cdot \left(\frac{L}{|\mathscr{P}| - L}\right)^L \end{aligned}$$

(because of Sterling's Approximation $\binom{n}{m} \leqslant (n \cdot e/m)^m$)

$$\begin{aligned} &= 2^{2n} \cdot \sum_{i=k-r}^{k-1} \left(\frac{6 \cdot (L - \lceil Q_i \cdot L \rceil)}{|\mathscr{P}_i| \cdot \lceil Q_i \cdot L \rceil}\right)^{\lceil Q_i \cdot L \rceil} \\ &\qquad \cdot \left(\frac{|\mathscr{P}| \cdot e}{L - \lceil Q_i \cdot L \rceil} \cdot \frac{L}{|\mathscr{P}| - L}\right)^L \\ &\leqslant 2^{2n} \cdot \sum_{i=k-r}^{k-1} \left(\frac{6}{|\mathscr{P}_i| \cdot Q_i}\right)^{\lceil Q_i \cdot L \rceil} \\ &\qquad \cdot \left(\frac{2 \cdot e \cdot L}{L - Q_i \cdot L - 1}\right)^L \qquad (|\mathscr{P}| \geqslant 2L) \\ &\leqslant \sum_{i=k-r}^{k-1} 2^{2n} \cdot \left(\frac{6}{|\mathscr{P}_i| \cdot Q_i}\right)^{\lceil Q_i \cdot L \rceil} \cdot 2^{4L} \qquad (Q_i \leqslant \tfrac{1}{2}) \\ &< \tfrac{1}{2} \qquad \text{(by Lemma 3.8)} \end{aligned}$$

i.e., at least half of all $\mathscr{L} \subseteq \mathscr{P}$, $|\mathscr{L}| = r \cdot 2^{r+4}$, are good for all (x, y), $x \neq y$ and $0 \leqslant x, y < 2^n$. $\blacksquare$

4. THE LOWER BOUND

Our proof of the lower bound is based on the lower bound proof for deterministic communication complexity as given in [MS] and generalized in [F89]. From the latter we take the following definition of rank-functions.

Let F be the functional matrix of some function $f: X \times Y \to \{0, 1\}$, i.e., $F[x, y] := f(x, y)$. A function rank: 0/1-matrices $\to \mathbb{N}$ is called a *rank-function* if

(1) For F constant, rank(F) is equal to one.

(2) For $i = 1, 2$, all $a \in \mathbb{N}$, and all partitions $X = X^1 \cup \cdots \cup X^{2^a}$ (some of the X^j might be empty) there exists at least one j with rank$(F^j) \geq$ (rank(F))$/2^a$, where F^j is the submatrix of F induced by X^j.

(3) The same as (2) with X replaced by Y.

Examples of rank functions are the rank of F over the field of two elements (or any other field) or the function $2^{C_{\det}(f)}$, where $C_{\det}(f)$ is the deterministic communication complexity of the function f corresponding to the matrix F. In [F89] (generalizing [MS]) we have shown that log rank(F) is a lower bound for $C_{\det}(f)$.

It is well known that any probabilistic algorithm $\mathscr{A}$ can be transformed into an equivalent algorithm $\mathscr{A}'$ which first does some random step and afterwards runs deterministically. If we have r random bits, this means that we first choose a deterministic algorithm out of a set of 2^r algorithms, which then solves the problem.

LEMMA 4.1. *For any set of 2^r deterministic algorithms for f there exists an input (x, y) such that all of these 2^r algorithms need at least communication complexity* (log rank(F))$/2^r$ *on that input. Here* rank *is any rank function.*

Proof. Let $\{A_1, \ldots, A_{2^r}\}$ be an arbitrary set of deterministic algorithms for f. Assume w.l.o.g. that algorithms $A_1, \ldots, A_a$ start with processor P_1 sending the first bit (and $A_{a+1}, \ldots, A_{2^r}$ with P_2). Then the set of all functions $h: \{1, \ldots, a\} \to \{0, 1\}$ induces a disjunct partition of the input values X (of processor P_1) as follows:

$$X_h := \{x \in X \mid \text{Algorithm } A_i \text{ sends } h(i) \text{ as its first bit, } 1 \leq i \leq a\}$$

i.e., X_h is a maximal subset of X with the property that each algorithm does the same for all $x \in X_h$. Because there are only 2^a different functions h there must be one function h_1 such that F restricted to X_{h_1} has rank at least rank$(F)/2^a$.

In the same way we can find a subset Y_{h_2} of the input values Y of processor P_2 (considering the algorithms $A_{a+1}, \ldots, A_{2^r}$ which have P_2 sending the first bit) such that the resulting functional matrix has rank at least rank$(F)/(2^a \cdot 2^{2^r - a}) =$ rank$(F)/2^{2^r}$.

As long as the matrix has rank > 1 the computation cannot stop (because the matrix is not constant). Hence we can iterate this procedure until $(2^{2^r})^t \geq$ rank(F), but then all algorithms have to communicate t bits. ∎

From this follows immediately

THEOREM 4.2. *Any probabilistic algorithm for f using r random bits has expected communication complexity* $\bar{C} \geq$ (log rank(F))$/2^r$.

Choosing rank$(F) = 2^{C_{\det}(f)}$ we obtain the well-known lower bound $\bar{C} \geq C_{\det}(f)/2^r$ which can also be obtained by simulating all sequences of random bits by a deterministic protocol (see [CG]). However, we think that Lemma 4.1 is of independent interest because we can use it in the following theorem to derive a relationship between communication complexity, failure probability, and randomness (as suggested by [KPU]).

THEOREM 4.3. *Any protocol for f that exchanges less than $t =$ (log rank(F))$/2^r$ bits with probability $1 - Q$, $0 < Q < 1$, requires randomness*

$$R \geq r - \log Q = \log \frac{\log \operatorname{rank}(F)}{tQ}.$$

Proof. If the probabilistic algorithm exchanges less than t bits with probability at least $1 - Q$ on every input, then the sum of the probabilities given to any set of 2^r deterministic algorithms must be bounded by Q (this follows directly from Lemma 4.1). However, then there must be at least $2^r/Q$ deterministic algorithms that are selected with positive probability. ∎

COROLLARY 4.4 (Theorem A). *Let $0 \leq r \leq \log n$.*

(a) *Any protocol for LND that exchanges less than $t = n^2/2^r$ bits with probability at least $1 - Q$, $0 < Q < 1$, requires randomness $R \geq r - \log Q = \log(n^2/(tQ))$.*

(b) *Any protocol for LND that exchanges an expected number of $\bar{C} = n^2/2^r$ bits requires randomness $R \geq r = \log(n^2/\bar{C})$.*

Proof. Mehlhorn and Schmidt [MS] have shown $C_{\det}(\text{LND}) \geq n^2$.

5. FURTHER RESEARCH

(1) We should try to close the gap between the upper and lower bound (Corollaries 3.6 and 4.4).

(2) The upper bound (Theorem 3.5) applies only for $r \leq \log n - \log \log n - 5$. We should also find efficient algorithms for $\log n - \log \log n - 5 < r \leq \log n$.

(3) Halstenberg and Reischuk [HR] exhaustively studied k-round protocols (i.e., the processors are only allowed to send a restricted number of messages of arbitrary length). It should be possible to prove similar results for this kind of communication complexity (note that all algorithms presented in this paper may need more than a constant number of rounds, whereas the lower bound holds for any number of rounds).

Received February 8, 1991

REFERENCES

[AUY] Aho, A. V., Ullman, J. D., and Yannakakis, M. (1983), On notions of information transfer in VLSI circuits, *in* "Proc., 15th ACM STOC 1983," pp. 133–139.

[CG] Canetti, R., and Goldreich, O. (1990), Bounds on tradeoffs between randomness and communication complexity, *in* "Proc. 31th IEEE FOCS 1990," pp. 766–775.

[F87] Fürer, M. (1987), The power of randomness for communication complexity, *in* "Proc. 19th ACM STOC 1987," pp. 178–181.

[F89] Fleischer, R. (1989), Communication complexity of multiprocessor systems, *Inform. Process. Lett.* **30**, 57–65.

[G72] Gundlach, K.-B. (1972), Einführung in die Zahlentheorie, Bibliographisches Institut BI Bd. 772.

[H86] Halstenberg, B. (1986), Zweiprozessorkommunikationskomplexität, Diplomarbeit, Universität Bielefeld.

[HR] Halstenberg, B., and Reischuk, R. On different modes of communication, *in* "Proc. 20th ACM STOC 1988," pp. 162–172.

[HW] Hardy and Wright (1979), An Introduction to the Theory of Numbers, 5th ed., Oxford Univ. Press, Oxford.

[KPU] Krizanc, D., Peleg, D., and Upfal, E. (1988), A time-randomness tradeoff for oblivious routing, *in* "Proc. 20th ACM STOC 1988," 93–102.

[KY] Knuth, D. E., and Yao, A. C. (1976), The complexity of nonuniform random number generation, *in* "Algorithms and Complexity" (J. E. Traub, Ed.), pp. 357–428, Academic Press, New York.

[LS] Lovász, L., and Saks, M. (1988), Lattices, Möbius functions and communication complexity, *in* "Proc. 29th IEEE FOCS 1988," pp. 81–90.

[MS] Mehlhorn, K., and Schmidt, E. M. (1982), Las Vegas is better than determinism in VLSI and distributed computing, *in* "Proc. 14th ACM STOC 1982," pp. 330–337.

[PS] Papadimitriou, C. H., and Sipser, M. (1982, 1984), Communication complexity, *in* "Proc. 14th ACM STOC 1982," pp. 196–200; *J. Comput. System Sci.* **28**, 260–269.

[RS62] Rosser, J. B., and Schoenfeld, L. (1962), Approximate formulas for some functions of prime numbers, *Illinois J. Math.* **6**, 64–94.

[RS89] Raghavan, P., and Snir, M. (1989), Memory versus randomization in on-line algorithms, *in* "Proc. ICALP '89," Lectures Notes in Computer Science, Vol. 372, pp. 687–703, Springer, Berlin.

[RY] Rabin, M., and Yao, A. C. (1979), Unpublished work; but see [CG].

[Y77] Yao, A. C. (1977), Probabilistic computations: Toward a unified measure of complexity, *in* "Proc. 18th IEEE FOCS 1977," pp. 222–227.

[Y79] Yao, A. C. (1979), Some complexity questions related to distributive computing, *in* "Proc. 11th ACM STOC 1979," pp. 209–213.

[Y83] Yao, A. C. (1983), Lower bounds by probabilistic arguments, "Proc. 24th IEEE FOCS 1983," pp. 420–428.

Printed by Catherine Press, Ltd., Tempelhof 41, B-8000 Brugge, Belgium