Algorithmica (1996) 16: 298–301

Algorithmica
© 1996 Springer-Verlag New York Inc.

Communication Complexity in a 3-Computer Model[1]

A. Ambainis[2]

Abstract. It is proved that the probabilistic communication complexity of the identity function in a 3-computer model is $O(\sqrt{n})$.

Key Words. Communication complexity, Probabilistic algorithms, Error-correcting codes.

1. Introduction. One of the problems in distributed computing is performing some computations by several processors if part of the input data is known to only one processor and another part to another processor. One trivial solution to this problem is sending all data to one processor and performing all computations by this processor. If the size of the input data is large, sending all data may be difficult and time-consuming. Very often better solutions exist.

For example, we consider the following problem: one processor has one array, another processor another array. We wish to know whether these arrays are equal. If we consider deterministic algorithms, it is known that for each algorithm there are always bad cases when it is necessary to send the whole array from one processor to another. In fact, for any algorithm, there are always cases when two arrays are equal for which it takes $\Omega(n)$ bits of exchange between the two processors.

If we allow probabilistic algorithms with an arbitrary small probability of error, the situation changes. It becomes possible to compute whether arrays are equal just with one message of length $O(\log n)$ from one processor to another [3]. This result cannot be improved, $O(\log n)$ is also the lower bound.

We can consider another model, rather similar to the one mentioned above. We have three processors, one array is given to the first processor, another to the second. These two processors can send messages to the third. (Exchange of information between the first and second processor is impossible. Also, the third processor cannot send any messages to the first and second.) This is the 3-computer model introduced by Yao [3] together with other models of communication complexity.

In this model it appears to be more difficult to compute whether two arrays are equal. We prove that it is possible to compute it by sending $O(\sqrt{n})$ bits only. Thus we give a partial answer to an open problem posed by Yao in 1979 [3]. The best-known lower bound is $\Omega(\log n)$.

[1] This research was supported by Latvian Science Council Grant No. 93.599.

[2] Institute of Mathematics and Computer Science, University of Latvia, Raina bulv. 29, Riga, LV-1459, Latvia. ambainis@mii.lu.lv.

Received December 4, 1993; revised February 24, 1994. Communicated by A. C.-C. Yao.

More formally speaking, we consider the Boolean function

$$f(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = (x_1 = y_1)\ \&\ (x_2 = y_2)\ \&\cdots\&\ (x_n = y_n).$$

We call it the *identity function.*

We consider the following 3-computer model: There are three computers A, B, and C. A has variables $x_1, x_2, \dots, x_n$, and B has variables $y_1, y_2, \dots, y_n$. A analyzes his variables, sends a message to C, B analyzes his variables and sends a message to C, too. Then C analyzes the two messages received from A and B and announces the result of the computation.

The *communication complexity* of function f is a worst-case number of bits sent from A and B to C when f is computed. Communication complexity was introduced by Yao in [3]. For a survey on communication complexity, see [1].

We prove that the identity function in a 3-computer model has communication complexity $O(\sqrt{n})$.

2. Combinatorics.

In this section we present a result from the theory of error-correcting codes which will be used further.

DEFINITION 1. If $x, y \in \{0, 1\}^n$, $x = (x_1, x_2, \dots, x_n)$, and $y = (y_1, y_2, \dots, y_n)$, then the Hamming distance between x and y is the number of i such that $x_i \neq y_i$. It is denoted by $d(x, y)$.

DEFINITION 2. $M \subset \{0, 1\}^n$ is called the $[n, k, d]$-code if it contains 2^k elements and $d(x, y) \geq d$ for every two distinct $x, y \in M$.

We denote $H_2(x) = -x \cdot \log_2 x - (1 - x) \cdot \log_2(1 - x)$.

LEMMA 1 [2, Theorem 17.30]. *If $0 < \delta < \frac{1}{2}$, then for each n there is a $[n, k, d]$-code such that $d/n \geq \delta$ and $k/n \geq 1 - H_2(d/n)$.*

We use following particular case of this lemma.

LEMMA 2. *For each m there is a $[3m, m, m/2]$-code.*

PROOF. In Lemma 1 replace n by $3m$ and δ by $\frac{1}{6}$. □

3. Complexity of the Identity Function.

We prove:

THEOREM 1. *It is possible to compute the identity function in the 3-processor model so that A and B transmit $\sqrt{3n} + o(\sqrt{n})$ bits each to C and the probability of the correct answer is at least* $6/11$.

PROOF. We denote by m the smallest integer satisfying $(6m)^2 \geq 3n$. From Lemma 2 we know that there is a $[(6m)^2, (6m)^2/3, (6m)^2/6]$-code. We choose 2^n elements of this code to obtain a $[(6m)^2, n, (6m)^2/6]$-code. We fix some such code and establish a one-to-one correspondence between the elements of the code and words $x \in \{0, 1\}^n$.

Our algorithm is as follows:

For A. Find the codeword $s = (s_1, \ldots, s_{(6m)^2})$ corresponding to input data $x = (x_1, \ldots, x_n)$. Take a $6m \times 6m$ table with $(6m)^2$ *squares* (i.e., positions) and write the numbers $s_1, \ldots, s_{(6m)^2}$ in the squares of the table. Choose a random row i, where i is uniformly distributed over $\{1, \ldots, 6m\}$, and transmit $(i, a_1, a_2, \ldots, a_{6m})$, where $(a_1, a_2, \ldots, a_{6m})$ is the content of row i, to C.

For B. Find the codeword $s = (s_1, \ldots, s_{(6m)^2})$ corresponding to the input data $y = (y_1, \ldots, y_n)$ and write $s_1, \ldots, s_{(6m)^2}$ in the squares in a $6m \times 6m$ table as in the case for A. Choose some column of the table equiprobably and transmit $(j, b_1, b_2, \ldots, b_{6m})$, where j is the column number and $(b_1, b_2, \ldots, b_{6m})$ is the content of column j, to C.

For C. C compares a_j and b_i. If they are different, C announces that $g = 0$ $(x \neq y)$. If they are equal, C announces that $g = 1$ $(x = y)$ with probability $6/11$ and that $g = 0$ with probability $5/11$.

The number of bits transmitted from A (or B) to C is $6m + \lceil \log_2(6m) \rceil = \sqrt{3n} + o(\sqrt{3n})$.

Now, we prove that the algorithm really computes g with the probability of a correct answer being at least $6/11$. Note that a_j is the (i, j)th entry of A's table and b_i is the (i, j)th entry of B's table. If $g = 1$ $(x = y)$, then the tables constructed by A and B are equal. Hence $a_j = b_i$. So, with probability $6/11$ C will give the answer $g = 1$. If $g = 0$ $(x \neq y)$, then A and B construct two different tables. As we have chosen, for writing into these tables, the codewords from a $[(6m)^2, n, (6m)^2/6]$-code, these tables are different in at least $(6m)^2/6$ squares (one-sixth of all the squares).

Each possible value for the pair (i, j) is chosen by A and B with equal probability. So, each square becomes the square contents of which C receives from both A and B with equal probability. With probability $p_0 \geq \frac{1}{6}$ the square in which the numbers in two tables are different is chosen. It means that with probability p_0 C receives two different values and with probability $1 - p_0$ two equal values. So, C will announce the correct answer $g = 0$ with probability $p_0 + \frac{5}{11}(1 - p_0) = \frac{5}{11} + \frac{6}{11}p_0 \geq \frac{5}{11} + \frac{6}{11} \cdot \frac{1}{6} = \frac{6}{11}$.

This proves the theorem. □

By repeating this algorithm several times and taking the majority of the outcomes as the final result by C, the probability of error can be made arbitrarily small. The amount of transmitted bits will still remain $O(\sqrt{n})$.

Acknowledgments. I thank Rusins Freivalds who suggested working on this problem and an anonymous referee whose comments helped to improve this paper.

References

[1] L. Lovasz. Communication complexity: a survey. In *Paths, Flows and VLSI Layout* (Korte, Lovasz, Promel, Schrijver, eds.). Springer-Verlag, New York, 1990, pp. 235–266.

[2] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1977.

[3] Andrew C. Yao. Some complexity questions related to distributed computing. *Proceedings of the* 11*th ACM Symposium on the Theory of Computing*, 1979, pp. 209–213.