Algorithmica (1999) 23: 341–353

Algorithmica


# Harmonic Analysis, Real Approximation, and the Communication Complexity of Boolean Functions[1]

V. Grolmusz[2]

**Abstract.** The two-party communication complexity of Boolean function $f$ is known to be at least $\log \operatorname{rank}(M_f)$, i.e., the logarithm of the rank of the communication matrix of $f$ [19]. Lovász and Saks [17] asked whether the communication complexity of $f$ can be bounded from above by $(\log \operatorname{rank}(M_f))^c$, for some constant $c$. The question was answered affirmatively for a special class of functions $f$ in [17], and Nisan and Wigderson proved nice results related to this problem [20], but, for *arbitrary* $f$, it remained a difficult open problem.

We prove here an analogous polylogarithmic upper bound in the stronger multiparty communication model of Chandra et al. [6], which, instead of the rank of the communication matrix, depends on the $L_1$ norm of function $f$, for *arbitrary* Boolean function $f$.

**Key Words.** Complexity of Boolean functions, Communication complexity, Fourier coefficients.

## 1. Introduction

1.1. *Communication Complexity.* In the *two-party communication game*, introduced by Yao [23], two players, $P_1$ and $P_2$, attempt to compute a Boolean function $f(x_1, x_2)$: $\{0,1\}^n \to \{0,1\}$, where $x_1, x_2 \in \{0,1\}^{n'}$, $2n' = n$. Player $P_1$ knows the value of $x_2$, $P_2$ knows the value of $x_1$, but $P_i$ does not know the value of $x_i$, for $i = 1, 2$. The minimum number of bits that must be communicated by the players to compute $f$ is the *communication complexity* of $f$, denoted by $\kappa(f)$.

This model has been widely studied and was applied to prove time–area tradeoffs for VLSI circuits, and has other numerous applications and remarkable properties (e.g., [1], [10], [11], [17], [19], or see [16] for a survey).

An important problem in complexity theory is giving lower and upper estimations for the communication complexity of function $f$. The following general lower bound to $\kappa(f)$ was introduced in [19]:

$$\kappa(f) \geq \log \operatorname{rank}(M_f),$$

where $M_f$ is a binary $2^{n'} \times 2^{n'}$ matrix, containing the value of $f(x_1, x_2)$ in the intersection of the row of $x_1$ and the column of $x_2$.

Lovász and Saks asked in [17] whether there existed an integer $c$ such that, for all

[1] This research was supported by Grants OTKA T017580 and F014919.
[2] Department of Computer Science, Eötvös University, Múzeum krt.6-8, H-1088 Budapest, Hungary. grolmusz@cs.elte.hu.

Received August 24, 1996; revised October 15, 1997. Communicated by J.-Y. Cai and C. K. Wong.

Boolean functions $f$,

$$\kappa(f) \leq (\log \operatorname{rank}(M_f))^c. \tag{1}$$

In [17], (1) was proved for a special class of functions. Nisan and Wigderson [20] also have nice results concerning this inequality. However, for general $f$, (1) is open, and seems to be a difficult problem.

The main contribution of this paper is an analogous polylogarithmic upper bound for *arbitrary* $f$, in the stronger $k$-party communication model of [6]:

$$C^{(k)}(f) = O((\log (n\mathrm{L}_1(f)))^3),$$

for $k = c\log(n\mathrm{L}_1(f))$ players, where $C^{(k)}(f)$ is the *$k$-party communication complexity* of $f$, and $\mathrm{L}_1(f)$ is the $\mathrm{L}_1$ *spectral norm* of Boolean function $f$ (both are defined below).

REMARK. Recently, Lu [18] observed that a slight modification in our ODDCOUNT protocol (Lemma 11) yields an $O(\left(\log(n\mathrm{L}_1(f))\right)^2)$ upper bound to $C^{(k)}(f)$.

1.2. *Multiparty Games*. The *multiparty communication game*, defined by Chandra et al. [6], is a generalization of the two-party case. In this game, $k$ players, $P_1, P_2, \ldots, P_k$, intend to compute a Boolean function $f(x_1, x_2, \ldots, x_n)\colon \{0,1\}^n \to \{0,1\}$. On set $S = \{x_1, x_2, \ldots, x_n\}$ of variables there is a fixed partition $A$ of $k$ classes $A_1, A_2, \ldots, A_k$, and player $P_i$ knows every variable, *except* those in $A_i$, for $i = 1, 2, \ldots, k$. The players have unlimited computational power, and they communicate with the help of a blackboard, viewed by all players. The goal is to compute $f(x_1, x_2, \ldots, x_n)$, such that at the end of the computation, every player knows this value. The cost of the computation is the number of bits written on the blackboard for the given $x = (x_1, x_2, \ldots, x_n)$ and $A = (A_1, A_2, \ldots, A_k)$. The cost of a multiparty protocol is the maximum number of bits communicated for any $x$ from $\{0,1\}^n$ and the given $A$. The $k$-party communication complexity, $C_A^{(k)}(f)$, of a function $f$, with respect to partition $A$, is the minimum of costs of those $k$-party protocols which compute $f$. The $k$-party symmetric communication complexity of $f$ is defined as

$$C^{(k)}(f) = \max_A C_A^{(k)}(f),$$

where the maximum is taken over all $k$-partitions of set $\{x_1, x_2, \ldots, x_n\}$.

This model was used by Babai et al. [3] for constructing pseudorandom generators. Håstad and Goldmann [13] and the author [7], [12] have used it for proving lower bounds to the size of hard-to-handle circuit classes.

For a general upper bound for both two and more players, we suppose that $A_1$ is one of the smallest classes of $A_1, A_2, \ldots, A_k$. Then $P_1$ can compute any Boolean function of $S$ with $|A_1|+1$ bits of communication: $P_2$ writes down the $|A_1|$ bits of $A_1$ on the blackboard, $P_1$ reads it, and computes and announces the value $f(x_1, x_2, \ldots, x_n) \in \{0,1\}$. So

$$C^{(k)}(f) \leq \left\lfloor \frac{n}{k} \right\rfloor + 1. \tag{2}$$

For certain functions, much better upper bounds were proven in [6], [9], and [7]. However, to the author's knowledge, before the present paper, no general upper bounds were known, other than (2).

1.3. *Spectral Norms*. There is a vast literature on representing the Boolean functions by polynomials over some field or ring (see, e.g., [2], [5], [22], [15], [14], or [4] for a survey). One reason for this may be that the polynomials offer a more developed machinery than the "pure" Boolean functions. One tool in this machinery is the Fourier expansion of Boolean functions [15], [5]:

We represent the Boolean function $f$ as a function $f\colon \{-1,1\}^n \to \{-1,1\}$ where $-1$ stands for "true." The set of all real-valued functions over $\{-1,1\}^n$ forms a $2^n$-dimensional vector space over the reals with an inner product:

$$\langle g, h\rangle = 2^{-n} \sum_{x\in\{-1,1\}^n} g(x)h(x).$$

We define, for $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \{0,1\}^n$,

$$x^\alpha = \prod_{i=1}^{n} x_i^{\alpha_i}.$$

The monomials $x^\alpha$ for $\alpha \in \{0,1\}^n$ form an *orthonormal basis* in this $2^n$-dimensional vector space; consequently, any function $h\colon \{-1,1\}^n \to \mathbf{R}$ can be uniquely expressed as

$$h(x_1, x_2, \ldots, x_n) = \sum_{\alpha\in\{0,1\}^n} a_\alpha x^\alpha. \tag{3}$$

The right-hand side of (3) is called the *Fourier expansion* of $h$, and numbers $a_\alpha$ for $\alpha \in \{0,1\}^n$ are called *the spectral* (*or Fourier*) *coefficients* of $h$. The $\mathrm{L}_1$ norm of $h$ is

$$\mathrm{L}_1(h) = \sum_{\alpha\in\{0,1\}^n} |a_\alpha|.$$

The $\mathrm{L}_2$ norm is

$$\mathrm{L}_2(h) = \left(\sum_{\alpha\in\{0,1\}^n} a_\alpha^2\right)^{1/2} = \langle h, h\rangle^{1/2}.$$

1.3.1. *Examples*

- The PARITY function in this setting is $x_1x_2\cdots x_n$, its $\mathrm{L}_1$ norm is 1, while its degree is $n$.
- It is easy to verify that

$$\begin{aligned}\bigvee_{i=1}^{n} x_i &= -\frac{1}{2^{n-1}}\left(2^{n-1} - \prod_{i=1}^{n}(x_i+1)\right)\\ &= -\frac{1}{2^{n-1}}(2^{n-1} - (1 + x_1 + x_2 + \cdots + x_n + x_1x_2 + \cdots + x_1x_2\cdots x_n))\end{aligned}$$

and

$$\begin{aligned}\bigwedge_{i=1}^{n} x_i &= \frac{1}{2^{n-1}}\left(2^{n-1} - \prod_{i=1}^{n}(1-x_i)\right)\\ &= \frac{1}{2^{n-1}}(2^{n-1} - (1 - x_1 - x_2 - \cdots - x_n + x_1x_2 + \cdots + (-1)^n x_1x_2\ldots x_n)).\end{aligned}$$

Observe that both the $n$-fan-in OR and AND have exponentially many nonzero Fourier coefficients, their degree is $n$, while their $L_1$ norms are less than three.

- The inner product mod 2 function (IP) is defined as follows:

$$\mathrm{IP}(x_1, x_2, ..., x_{2n}) = \prod_{i=1}^{n}(x_{2i-1} \wedge x_{2i}).$$

It is easy to verify that $L_1(\mathrm{IP})$ is the highest possible for any $2n$ variable Boolean functions: $2^n$.

Bruck and Smolensky [5] established a relation between the $L_1$ norm and the computability of $f$ by polynomial threshold function. A generalization of one of their results plays a main role (Lemma 8) in the present work.

**2. Main Results.** First we present a general theorem, which implies several corollaries in a more natural setting. Theorem 1 shows that if a Boolean function can be approximated by a *real* function with small error, then there exists a $k$-party protocol which computes the Boolean function, and the number of communicated bits in this protocol depends only on the $L_1$ norm of the *approximating real function*.

THEOREM 1. *Let $f$ be a Boolean function $f\colon \{-1, 1\}^n \to \{-1, 1\}$, and let $g$ be a real function $g\colon \{-1, 1\}^n \to \mathbf{R}$. Suppose that, for all $x \in \{-1, 1\}^n$,*

$$|g(x) - f(x)| < \tfrac{1}{5}.$$

*Then the $k$-party symmetric communication complexity of $f$ is*

$$O\left(k^2 \log(nL_1(g)) \left\lceil \frac{nL_1^2(g)}{2^k} \right\rceil\right).$$

In particular, choosing $g = f$ in Theorem 1:

COROLLARY 2. *Let $f$ be a Boolean function $f\colon \{-1, 1\}^n \to \{-1, 1\}$. Then the $k$-party symmetric communication complexity of $f$ is*

$$O\left(k^2 \log(nL_1(f)) \left\lceil \frac{nL_1^2(f)}{2^k} \right\rceil\right).$$

Or, setting $k$ large enough:

COROLLARY 3. *Let $f$ be an arbitrary Boolean function of $n$ variables. Let $k = c\log(nL_1(f))$ with $c > 0$. Then*

$$C^{(k)}(f) = O\left(\log^3\left(nL_1(f)\right)\right).$$

In other words, if the $L_1$ spectral norm of $f$ is bounded by a polynomial in $n$, then the *symmetric* $k$-party communication complexity of $f$ is at most $O(\log^3 n)$, with $k = c\log n$.

Let $f$ and $g$ be two functions, such that $|f - g| < \frac{1}{5}$. Then their $L_1$ norms may differ even exponentially, e.g., $f \equiv 0$, $g'$ is a Boolean function of exponential $L_1$ norm, then $g = \frac{1}{6}g'$ also has exponential $L_1$ norm, while $|f - g| \le \frac{1}{6}$. So the following corollary of Theorem 1 may yield a much better bound than Corollary 3:

COROLLARY 4. *Let*

$$\gamma = \inf\{L_1(g) \mid g : \{-1,1\}^n \to \mathbf{R},\ \textit{and}\ \forall x \in \{-1,1\}^n :\ |g(x) - f(x)| < \tfrac{1}{5}\}.$$

*Then*

$$C^{(k)}(f) = O\left(k^2 \log(n\gamma)\left\lceil \frac{n\gamma^2}{2^k} \right\rceil\right).$$

Suppose that $f$ is a Boolean function of large (say, exponential) $L_1$ norm in $n$. Our Corollary 3 can guarantee only a communication protocol with too many communicated bits: the trivial $\lfloor n/k \rfloor + 1$ protocol may be better. However, if the Fourier coefficients of $f$ are distributed "unevenly enough," i.e., they can be divided into two parts, one with small $L_1$, the other with small $L_2$ norms, then we can do much better:

THEOREM 5. *Let*

$$f(x) = \sum_{\alpha \in \{0,1\}^n} a_\alpha x^\alpha,$$

*and let* $S \subset \{0,1\}^n$ *such that*

$$\sum_{\alpha \in S} a_\alpha^2 \le \varepsilon,$$

*for some* $\varepsilon < \frac{1}{2500}$. *Let*

$$g(x) = \sum_{\alpha \in \{0,1\}^n - S} a_\alpha x^\alpha.$$

*Then, for all* $k \ge 2$ *and for all* $k$*-partitions of the inputs, there exists a* $k$*-party protocol with*

$$O\left(k^2 \log(nL_1(g))\left\lceil \frac{nL_1^2(g)}{2^k} \right\rceil\right)$$

*bits of communication, and this protocol computes* $f$ *correctly on at least* $(1 - 25\varepsilon) > \frac{99}{100}$ *fraction of the inputs.*

The following results of [8] show the power of our upper bounds in Theorems 1 and 5, proving that almost all Boolean functions have very high multiparty communication complexity:

THEOREM 6 [8]. *Let* $f$ *be a uniformly chosen random member of set*

$$\{f \mid f : \{-1,1\}^n \to \{-1,1\}\}.$$

*Then the probability that, for some A k-equipartition of* $x = \{x_1, x_2, \ldots, x_n\}$, *there exists a k-party protocol which computes* $f$ *with communication of at most* $\lfloor n/k \rfloor - \log n$ *bits, is less than*

$$2^{-2^{\Omega(n)}}.$$

The communication complexity remains high even if we compute $f$ on *most* of the inputs:

THEOREM 7. *Let* $f$ *be a uniformly chosen random member of set*

$$\{f \mid f : \{-1, 1\}^n \to \{-1, 1\}\}.$$

*Then the probability that, for some A k-equipartition of* $x = \{x_1, x_2, \ldots, x_n\}$, *there exists a k-party protocol which correctly computes* $f$ *on a fraction of at least* $\frac{1}{2} + \varepsilon$ *of inputs, with communication of at most* $\lfloor n/k \rfloor - \log(n/\varepsilon)$ *bits, is less than*

$$2^{-2^{\Omega(n)}}.$$

Comparing Theorem 1 with Theorem 6, and Theorem 5 with Theorem 7, we have that for almost all Boolean functions $f$:

- $f$ has exponential $L_1$ norm,
- if $f$ is approximated by a real function $g$ with error less than $\frac{1}{5}$, then the $L_1$ norm of $g$ is exponential in $n$,
- the Fourier coefficients of $f$ are "evenly distributed": they cannot be divided into two sets, one with subexponential $L_1$ norm, the other with a small $L_2$ norm.

## 3. The Proof of Theorem 1.

The following lemma is a generalization of a lemma of Bruck and Smolensky [5].

LEMMA 8. *Let* $U \subset \{-1, 1\}^n$ *such that* $|U| \geq (1 - \frac{1}{100})2^n$. *Let* $g\colon \{-1, 1\}^n \to \mathbf{R}$. *Suppose that, for all* $x \in U$, $\frac{4}{5} < |g(x)| < \frac{6}{5}$ *is satisfied. Then there exists polynomial* $G_0(x)$ *with integer coefficients and with* $L_1$ *norm*

$$L_1(G_0) \leq 400n L_1^2(g)$$

*such that*

$$\mathrm{sgn}(G_0(x)) = \mathrm{sgn}(g(x))$$

*for all* $x \in U$.

PROOF. The Fourier expansion of $g$:

$$g(x) = \sum_{\alpha \in \{0,1\}^n} a_\alpha x^\alpha,$$

where $a_\alpha$, for $\alpha \in \{0,1\}^n$, are the Fourier coefficients of $g$. Then by definition

$$\mathrm{L}_1(g) = \sum_{\alpha \in \{0,1\}^n} |a_\alpha|$$

and

$$\mathrm{L}_2^2(g) = \langle g, g \rangle = 2^{-n} \sum_{x \in \{-1,1\}^n} g^2(x) = \sum_{\alpha \in \{0,1\}^n} a_\alpha^2,$$

using the Parseval identity.

Since $|g(x)| \geq \frac{4}{5}$ for $x \in U$, and $|U| \geq (1 - \frac{1}{100})2^n$,

$$\mathrm{L}_2(g) \geq (1 - \tfrac{1}{100})\tfrac{16}{25}.$$

Our next step is giving a lower bound to the $\mathrm{L}_1$ norm of $g$.

(i) Suppose that there exists an $\alpha$ such that $|a_\alpha| > \frac{1}{2}$. If $\mathrm{sgn}(x^\alpha) = \mathrm{sgn}(g(x))$ for all $x \in U$, then we are done, $G_0(x) = x^\alpha$ suffices. Otherwise, for some $x \in U$, $\mathrm{sgn}(x^\alpha) \neq \mathrm{sgn}(g(x))$. Then the other terms of $g$ must compensate for $x^\alpha$, so the sum of the absolute values of their coefficients should be greater than $\frac{4}{5}$. So

$$\mathrm{L}_1(g) \geq \tfrac{4}{5} + |a_\alpha| \geq \tfrac{13}{10}.$$

(ii) Otherwise, if all $|a_\alpha| \leq \frac{1}{2}$, then

$$(1 - \tfrac{1}{100})\tfrac{16}{25} \leq \sum_{\alpha \in \{0,1\}^n} a_\alpha^2 \leq \tfrac{1}{2} \sum_{\alpha \in \{0,1\}^n} |a_\alpha|,$$

so

$$(1 - \tfrac{1}{100})\tfrac{32}{25} \leq \sum_{\alpha \in \{0,1\}^n} |a_\alpha| = \mathrm{L}_1(g).$$

Consequently, either we have found a suitable $G_0(x)$, or we have concluded that

$$\mathrm{L}_1(g) \geq (1 - \tfrac{1}{100})\tfrac{32}{25} \geq \tfrac{127}{100}. \tag{4}$$

We define random monomials $Z_i(x)$ as follows:

$$Z_i(x) = \mathrm{sgn}(a_\alpha)x^\alpha \qquad \text{with probability } \frac{|a_\alpha|}{\mathrm{L}_1(g)}.$$

Let random polynomial $G(x)$ be defined as the sum of $N = \lfloor 400n\mathrm{L}_1^2(g) \rfloor$ monomials $Z_i(x)$:

$$G(x) = \sum_{i=1}^{N} Z_i(x).$$

Computing the expectation of $Z_i(x)$:

$$\mathrm{E}(Z_i(x)) = \sum_{\alpha \in \{0,1\}^n} \frac{|a_\alpha|}{\mathrm{L}_1(g)} \mathrm{sgn}(a_\alpha)x^\alpha = \frac{g(x)}{\mathrm{L}_1(g)},$$

where we used the fact that $\mathrm{sgn}(v)|v| = v$. The expectation of $G(x)$ is

$$\mathrm{E}(G(x)) = \frac{Ng(x)}{\mathrm{L}_1(g)}. \tag{5}$$

The variance of $Z_i(x)$ is

$$\mathrm{Var}(Z_i(x)) = \mathrm{E}(Z_i^2(x)) - \mathrm{E}^2(Z_i(x)) = 1 - \frac{g^2(x)}{\mathrm{L}_1^2(g)}.$$

The variance of $G(x)$ is

$$\mathrm{Var}(G(x)) = N\left(1 - \frac{g^2(x)}{\mathrm{L}_1^2(g)}\right).$$

Since $|g(x)| \leq \frac{6}{5}$, and because of (4),

$$\frac{g^2(x)}{\mathrm{L}_1^2(g)} \leq (\tfrac{120}{127})^2 \leq \tfrac{9}{10},$$

so

$$\frac{N}{10} \leq \mathrm{Var}(G(x)) \leq N$$

or

$$\sqrt{\frac{N}{10}} \leq D(G(x)) \leq \sqrt{N}, \tag{6}$$

where $D(G(x)) = \sqrt{\mathrm{Var}(G(x))}$, the standard deviation of $G(x)$.

From (5), the sign of $E(G(x))$ is the same as the sign of $g(x)$. Consequently,

$$\begin{aligned}\Pr(\mathrm{sgn}(G(x)) \neq \mathrm{sgn}(g(x)) &= \Pr(\mathrm{sgn}(G(x)) \neq \mathrm{sgn}\left(\mathrm{E}(G(x))\right))\\ &\leq \Pr\left(\left|G(x) - \mathrm{E}(G(x))\right| \geq \frac{N|g(x)|}{\mathrm{L}_1(g)}\right)\\ &\leq \Pr\left(\left|G(x) - \mathrm{E}(G(x))\right| \geq \frac{4N}{5\mathrm{L}_1(g)}\right).\end{aligned}$$

From the Bernstein inequality (see [21]) (or from the Central Limit Theorem), with $D = D(G(x))$, we get

$$\Pr(|G(x) - \mathrm{E}(G(x))| \geq \mu D) \leq 2\exp\left(-\frac{\mu^2}{2\left(1 + \mu/D\right)^2}\right), \tag{7}$$

where $0 < \mu < D/2$.

For $\mu = 3\sqrt{n}$, $N = \lfloor 400n\mathrm{L}_1^2(g)\rfloor$ we get that the probability in (7) is less than $e^{-n}$. On the other hand,

$$\mu D \leq \frac{4N}{5\mathrm{L}_1(g)},$$

so

$$\Pr(\text{sgn}(G(x)) \neq \text{sgn}(g(x))) < e^{-n}.$$

Consequently,

$$\Pr(\exists x \in U : \text{sgn}(G(x)) \neq \text{sgn}(g(x))) \\ \leq \sum_{x \in U} \Pr(\text{sgn}(G(x)) \neq \text{sgn}(g(x))) \leq |U| e^{-n} \leq 2^n e^{-n} < 1,$$

and since this probability is less than one, there exists a polynomial $G_0(x)$ for which $\text{sgn}(G_0(x)) = \text{sgn}(g(x))$ for all $x \in U$. The coefficients of this $G_0$ are integers, and its $L_1$ norm is at most $N$. □

PROOF OF THEOREM 1. Function $g$ satisfies the requirements of Lemma 8, for $U = \{-1, 1\}^n$. Then there exists a polynomial $G_0(x)$ with integer coefficients and an $L_1$ norm of at most $400n L_1^2(g)$, such that

$$\text{sgn}(g(x)) = \text{sgn}(G_0(x))$$

for all $x \in \{-1, 1\}^n$. Since $\text{sgn}(g(x)) = f(x)$, we have that $\text{sgn}(G_0(x)) = f(x)$, for all $x \in \{-1, 1\}^n$. By Theorem 9, $G_0(x)$ has the required symmetric $k$-party communication complexity. □

THEOREM 9. *Let*

$$G(x) = \sum_{i=1}^{N} Z_i(x),$$

*where* $Z_i(x) = x^\alpha$ *or* $Z_i(x) = -x^\alpha$, *for some* $\alpha \in \{0, 1\}^n$, *and for* $x \in \{-1, 1\}^n$. *Then the symmetric k–party communication complexity of G is*

$$O\left(k^2 \log(nN) \left\lceil \frac{nN}{2^k} \right\rceil \right).$$

PROOF OF THEOREM 9. Let $G_1(x)$ be the sum of the $Z_i$'s with positive sign, and let $G_2(x)$ be the sum of the $(-Z_i)$'s, where $Z_i$ has a negative sign. So

$$G(x) = G_1(x) - G_2(x),$$

and $G_1$ has $N_1$ terms, $G_2$ has $N_2$ terms, $N_1 + N_2 = N$.

Observe that $G_j(x)$ is the sum of $N_j$ terms of the form

$$x^\alpha = \prod_{i=1}^{n} x_i^{\alpha_i} = \prod_{i:\alpha_i=1} x_i$$

for $j = 1, 2$. Clearly,

$$x^\alpha = \begin{cases} -1 & \text{if } \quad |\{i : x_i = -1, \alpha_i = 1\}| \text{ is odd,} \\ 1 & \text{otherwise.} \end{cases}$$

For $j = 1, 2$, let $b_j$ denote the number (counting the possible multiplicities) of those terms $x^\alpha$ in $G_j(x)$, for which $|\{i : x_i = -1, \alpha_i = 1\}|$ is odd. Then $G_j(x) = (N_j - b_j) - b_j = N_j - 2b_j$, so

$$G(x) = G_1(x) - G_2(x) = N_1 - N_2 + 2b_2 - 2b_1. \tag{8}$$

We denote

$$y_i = \begin{cases} 1 & \text{if} \quad x_i = -1, \\ 0 & \text{if} \quad x_i = 1, \end{cases}$$

then

$$x^\alpha = -1 \quad \Longleftrightarrow \quad \sum_{i=1}^{n} y_i \alpha_i = 1 \bmod 2.$$

We form a matrix $M^{(j)}$ with $N_j$ rows and $n$ columns, for $j = 1, 2$. Each row corresponds to a term $x^\alpha$ in $G_j(x)$, and the $i$th entry of that row is $y_i \alpha_i$. Obviously, the number of those rows of $M^{(j)}$ which have an odd sum is equal to $b_j$. Suppose now that we are given polynomial $G(x)$, players $P_1, P_2, \ldots, P_k$, and a $k$-partition $A = (A_1, A_2, \ldots, A_k)$ of the set $\{x_1, x_2, \ldots, x_n\}$. We assume that player $P_\ell$ knows function $G(x)$, partition $A$, functions $G_1(x)$, $G_2(x)$, and the values of all variables, except those in $A_\ell$, for $\ell = 1, 2, \ldots, k$. Then the players, without any communication, can privately compute matrices $M^{(1)}$ and $M^{(2)}$, and exactly those entries of these matrices will not be known to player $P_\ell$, which correspond to variables in class $A_\ell$. The set of these entries is called $B_\ell$, for $\ell = 1, 2, \ldots, k$. The following lemma shows a protocol by which the players can first compute $b_1$ and then $b_2$, and, consequently, $G(x)$, by (8).

LEMMA 10. *Let $M \in \{0,1\}^{m \times n}$, $M = \{m_{ij}\}$, and let $B = \{B_1, B_2, \ldots, B_k\}$ be a partition of the set $\{m_{ij}\colon 1 \le i \le m, 1 \le j \le n\}$, such that player $P_\ell$ knows every $m_{ij}$ except those in $B_\ell$, for $\ell = 1, 2, \ldots, k$. Then there exists a $k$-party protocol which computes the number of the rows with an odd sum in $M$ by communicating*

$$O\left(k^2 \log m \left\lceil \frac{m}{2^k} \right\rceil\right)$$

*bits.*

PROOF. First, the players compute a matrix $Q \in \{0,1\}^{m \times k}$ from $M$, with no communication: a row of $Q$ is associated with each row of $M$; the first element of row $j$ of $Q$ is the mod 2 sum of those entries of the $j$th row of $M$ which are the elements of $B_1$ at the same time. Analogously, the $i$th element of row $j$ of $Q$ is the mod 2 sum of those entries of the $j$th row of $M$ which are the elements of $B_i$ at the same time. Clearly, the number of rows with an odd sum in $M$ and in $Q$ is the same. Moreover, player $P_\ell$ knows every column of matrix $Q$, except column $\ell$, for $\ell = 1, 2, \ldots, k$.

With an additional assumption, Lemma 11 gives a protocol with $O(k^2 \log m)$ communication. This protocol is implicit in [2], in [9], and is used in a more general form in [7].

LEMMA 11. *Let $\beta \in \{0, 1\}^k$. Suppose it is known to each player that $\beta$ does not occur as a row of $Q$. Then there exists a $k$-party protocol which computes the number of rows with an odd sum with a communication of $O(k^2 \log m)$ bits.*

PROOF OF LEMMA 11. Without restricting the generality we may suppose that $\beta$ is the all-1 vector of length $k$. Let $\mathrm{ODD}(\gamma_1\gamma_2\cdots\gamma_\ell)$ and $\mathrm{EVEN}(\gamma_1\gamma_2\cdots\gamma_\ell)$ denote the number of those rows of $Q$ which have odd (resp. even) sums, and they begin with $\gamma_1\gamma_2\cdots\gamma_\ell$, $\ell \leq k$, $\gamma_i \in \{0, 1\}$. Clearly, $\mathrm{ODD}(\gamma_1\gamma_2\cdots\gamma_{\ell-1}, 0) + \mathrm{ODD}(\gamma_1\gamma_2\cdots\gamma_{\ell-1}, 1) = \mathrm{ODD}(\gamma_1\gamma_2\cdots\gamma_{\ell-1})$. For example, $P_1$ does not know the first column of $Q$, but he can communicate $\mathrm{ODD}(0) + \mathrm{EVEN}(1)$ if $P_1$ counts those rows which have an odd sum in their second through $k$th position. Similarly $P_2$ can communicate $\mathrm{ODD}(10) + \mathrm{EVEN}(11)$ if he counts those rows which begin with 1, and the sum of their first, third, fourth,..., $k$th elements is odd.

This observation motivates the following protocol:

**Protocol ODDCOUNT**

The goal: to compute $b$, the number of rows with an odd sum in $Q$. Observe that $b = \mathrm{ODD}(1) + \mathrm{ODD}(0)$. Number $b$ will be computed as the sum of values $u_i$ announced by player $P_i$, $i = 1, 2, \ldots, k$.

$P_1$ announces $u_1 = \mathrm{ODD}(0) + \mathrm{EVEN}(1)$.

Note: $b = u_1 - \mathrm{EVEN}(1) + \mathrm{ODD}(1)$.

$P_2$ announces $u_2 = \mathrm{ODD}(10) + \mathrm{EVEN}(11) - \mathrm{EVEN}(10) - \mathrm{ODD}(11)$.

Note: $b = u_1 + u_2 - 2\mathrm{EVEN}(11) + 2\mathrm{ODD}(11)$

$P_3$ announces $u_3 = 2\mathrm{ODD}(110) + 2\mathrm{EVEN}(111) - 2\mathrm{EVEN}(110) - 2\mathrm{ODD}(111)$.

Note: $b = u_1 + u_2 + u_3 - 4\mathrm{EVEN}(111) + 4\mathrm{ODD}(111)$

$$\vdots$$

$$P_i \text{ announces } u_i = 2^{i-2}\mathrm{ODD}(1\cdots 10) + 2^{i-2}\mathrm{EVEN}(1\cdots 11) - 2^{i-2}\mathrm{EVEN}(1\cdots 10) - 2^{i-2}\mathrm{ODD}(1\cdots 11)$$

Note: $b = \sum_{j=1}^{i} u_j - 2^{i-1}\mathrm{EVEN}(\overbrace{11\cdots 1}^{i \text{ times}}) + 2^{i-1}\mathrm{ODD}(\overbrace{11\cdots 1}^{i \text{ times}})$.

After $P_k$ announces $u_k$, the players privately add up the $u_i$'s from $i = 1$ through $k$. Note that

$$b = \sum_{j=1}^{k} u_j - 2^{k-1}\mathrm{EVEN}(\overbrace{11\cdots 1}^{k \text{ times}}) + 2^{k-1}\mathrm{ODD}(\overbrace{11\cdots 1}^{k \text{ times}}).$$

However, as we assumed at the beginning, there are no all-1 rows in $Q$, so

$$b = \sum_{j=1}^{k} u_j$$

and we are done. Each $u_i$ can be communicated using $O(k \log m)$ bits, so the total communication is $O(k^2 \log m)$. □

Now we return to the proof of Lemma 10. We divide the rows of matrix $Q$ into blocks of $2^{k-1} - 1$ contiguous rows plus a leftover of at most $2^{k-1} - 1$ rows. The players cooperatively determine the number of the odd rows in each block, and then privately add up the results.

Next we show how to obtain the number of odd rows for a single block at the cost of $O(k^2 \log m)$ bits of communication. $P_1$ knows all the columns, except the first, so he knows at most $2^{k-1} - 1$ rows of length $k - 1$ in a block, so he can find a $\beta' \in \{0, 1\}^{k-1}$, $\beta' = (\beta_2, \beta_3, \ldots, \beta_k)$, which is not a row of the $k - 1$ column wide part of the block seen by $P_1$. Let $\beta = (1, \beta_2, \beta_3, \ldots, \beta_k)$. Then $\beta$ does not occur as a row in this block. So if $P_1$ communicates $\beta$, and they play protocol ODDCOUNT of Lemma 11 in a given block, they use $k^2 \log m$ bits for the block, and, since there are at most $\lceil m/(2^{k-1} - 1)\rceil$ blocks, the total communication is

$$O\left(k^2 \log m \left\lceil \frac{m}{2^k} \right\rceil\right). \qquad \square$$

## 4. Proof of Theorem 5

LEMMA 12. *Let $f$ be a Boolean function and let $h$: $\{-1, 1\}^n \to \mathbf{R}$ such that*

$$\mathrm{L}_2^2(f - h) = \langle f - h,\ f - h\rangle \le \varepsilon.$$

*Then*

$$\Pr_x(|f(x) - h(x)| > \tfrac{1}{5}) \le 25\varepsilon,$$

*where* $\Pr_x$ *is the probability measure associated with the uniform distribution over* $\{-1, 1\}^n$.

PROOF.

$$\begin{aligned}\varepsilon &\ge \langle f(x) - h(x),\ f(x) - h(x)\rangle \\ &= \mathrm{E}_x(f(x) - h(x))^2 \ge \tfrac{1}{25}\Pr_x(|f(x) - h(x)| > \tfrac{1}{5}).\end{aligned} \qquad \square$$

Now we prove Theorem 5. Let $U$ be defined as

$$U = \{x \in \{-1, 1\}^n : |f(x) - g(x)| \le \tfrac{1}{5}\}.$$

From Lemma 12, $|U| \ge (1 - 25\varepsilon)2^n$. If $\varepsilon \le \frac{1}{2500}$, then we can apply Lemma 8 for $g$. The proof then proceeds exactly as the proof of Theorem 1. □

**Acknowledgment.** The author is grateful to Chi-Jen Lu for discussions on this topic.

## References


[1] A. Aho, J. D. Ullman, and M. Yannakakis, On the notions of information transfer in VLSI circuits, in *Proc*. 15*th Ann*. *ACM Symp*. *Theory Comput*., 1983, pp. 151–158.

[2] J. Aspnes, R. Beigel, M. L. Furst, and S. Rudich, The expressive power of voting polynomials, in *Proc*. 23*rd Ann*. *ACM Symp*. *Theory Comput*., 1991, pp. 402–409.

[3] L. Babai, N. Nisan, and M. Szegedy, Multiparty protocols, pseudorandom generators for logspace, and time–space trade-offs, *J. Comput*. *System Sci*., **45** (1992), 204–232.

[4] R. Beigel, The polynomial method in circuit complexity, in *Proc*. *Eighth Ann*. *Conf*. *Structure in Complexity Theory* (*SCT*), IEEE Computer Society Press, Los Alamitos, CA, 1993, pp. 82–95.

[5] J. Bruck and R. Smolensky, Polynomial threshold functions, $AC^0$ functions and spectral norms, in *Proc*. 32*nd Ann*. *IEEE Symp*. *Found*. *Comput*. *Sci*., 1991, pp. 632–641.

[6] A. K. Chandra, M. L. Furst, and R. J. Lipton, Multiparty protocols, in *Proc*. 15*th Ann*. *ACM Symp*. *Theory Comput*., 1983, pp. 94–99.

[7] V. Grolmusz, Circuits and multi-party protocols, *Comput*. *Complexity*, **7** (1998), 1–18.

[8] V. Grolmusz, On multi-party communication complexity of random functions, Tech. Report MPII-1993-162, Max Planck Institut für Informatik, December 1993.

[9] V. Grolmusz, The BNS lower bound for multi-party protocols is nearly optimal, *Inform*. *and Comput*., **112** (1994), 51–54.

[10] V. Grolmusz, A weight–size trade-off for circuits with mod $m$ gates, in *Proc*. 26*th Ann*. *ACM Symp*. *Theory Comput*., 1994, pp. 68–74.

[11] V. Grolmusz, On the weak mod $m$ representation of Boolean functions, *Chicago J*. *Theoret*. *Comput*. *Sci*., Vol. 1995, Article 2.

[12] V. Grolmusz, Separating the communication complexities of *MOD m* and *MOD p* circuits, *J*. *Comput*. *System Sci*., **51** (1995), 307–313. Also in *Proc*. 33*rd Ann*. *IEEE Symp*. *Found*. *Comput*. *Sci*., 1992, pp. 278–287.

[13] J. Håstad and M. Goldmann, On the power of the small-depth threshold circuits, *Comput*. *Complexity*, **1** (1991), 113–129.

[14] J. Kahn, G. Kalai, and N. Linial, The influence of variables on Boolean functions, in *Proc*. 29*th Ann*. *IEEE Symp*. *Found*. *Comput*. *Sci*., 1988, pp. 68–80.

[15] N. Linial, Y. Mansour, and N. Nisan, Constant depth circuits, Fourier transform and learnability, *J*. *Assoc*. *Comput*. *Mach*., **40** (1993), 607–620.

[16] L. Lovász, Communication complexity: a survey, in *Paths*, *Flows*, *and VLSI Layout*, B. Korte, L. Lovász, H. Prömel, and A. Schrijver, eds., Springer-Verlag, New York, 1989, pp. 235–265.

[17] L. Lovász and M. Saks, Lattices, Möbius functions, and communication complexity, in *Proc*. 29*th Ann*. *IEEE Symp*. *Found*. *Comput*. *Sci*., 1988, pp. 81–90.

[18] C.-J. Lu, Private correspondence, 1995.

[19] K. Mehlhorn and E. Schmidt, Las Vegas is better than determinism in VLSI and distributive computing, in *Proc*. 14*th Ann*. *ACM Symp*. *Theory Comput*., 1982, pp. 330–337.

[20] N. Nisan and A. Wigderson, On rank vs. communication complexity, in *Proc*. 35*th Ann*. *IEEE Symp*. *Found*. *Comput*. *Sci*., 1994, pp. 831–836.

[21] A. Rényi, *Probability Theory*, North-Holland, Amsterdam, 1970.

[22] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, in *Proc*. 19*th Ann*. *ACM Symp*. *Theor*. *Comput*., 1987, pp. 77–82.

[23] A. C. Yao, Some complexity questions related to distributed computing, in *Proc*. 11*th Ann*. *ACM Symp*. *Theory Comput*., 1979, pp. 209–213.