Separating the Communication Complexities of MOD m and MOD p Circuits

Vince Grolmusz*

Max Planck Institute for Computer Science and Eötvös University

ABSTRACT:

We prove in this paper that it is much harder to evaluate depth–2, size–N circuits with MOD m gates than with MOD p gates by k–party communication protocols: we show a k–party protocol which communicates $O(1)$ bits to evaluate circuits with MOD p gates, while evaluating circuits with MOD m gates needs $\Omega(N)$ bits, where p denotes a prime, and m a composite, non-prime power number. As a corollary, for all m, we show a function, computable with a depth–2 circuit with MOD m gates, but not with any depth–2 circuit with MOD p gates.

Obviously, the k–party protocols are not weaker than the k'–party protocols, for $k' > k$. Our results imply that if there is a prime p between k and k': $k < p \leq k'$, then there exists a function which can be computed by a k'–party protocol with a constant number of communicated bits, while any k–party protocol needs linearly many bits of communication. This result gives a hierarchy theorem for multi–party protocols.

1. INTRODUCTION

The connection between the circuit complexity and the communication complexity plays an important role in the recent literature of the circuit lower bound theory.

The notion of the (2–party) communication complexity was introduced by *Yao* [11]. Due to the algebraic characterization of the communication complexity, several strong lower bounds were proved for this model (see [6] for a survey). Many nice results appeared in the literature concerning the connection of the (2–party) communication complexity and the circuit complexity: [5], [7], [8], [9], [12].

The *multi–party communication game*, defined by *Chandra, Furst* and *Lipton* [3], is an interesting generalization of the 2–party communication game. In this game, k players: $P_1, P_2..., P_k$ intend to compute the value of $g(A_1, A_2, ..., A_k)$, where $g : \{0, 1, 2, ..., m - 1\}^{kn} \to \mathbf{N}$, where $\mathbf{N}$ denotes the set of natural numbers, $m \in \mathbf{N}$ and $A_i \in \{0, 1, 2, ..., m - 1\}^n$, for $i = 1, 2, ..., k$. Player P_i knows every variable, *except* A_i, for $i = 1, 2, ..., k$. The players have unlimited computational power, and they communicate with the help of a blackboard, viewed by all players. Only one player may write on the blackboard at a time. The goal is to compute $g(A_1, A_2, ..., A_k)$, 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 $A = (A_1, A_2, ..., A_k)$. The cost of a multi–party protocol is the maximum number of bits communicated for any A from $\{0, 1, 2, ..., m - 1\}^{nk}$. The k-party communication complexity, $C^{(k)}(g)$, of a function g, is the minimum of costs of those k-party protocols which compute g.

The theory of the 2–party communication games is well developed [6], but much less is known about the multi–party communication complexity of functions. As a general upper bound, P_1 can compute any function of A with n bits of communication: P_2 writes down the n bits of A_1 on the blackboard, P_1 reads it, and computes the value $g(A)$ at no cost. The additional cost of diffusing the result $g(A)$ to other players is the binary length of $g(A)$.

An important progress was made by *Babai, Nisan* and *Szegedy*, [2], proving an $\Omega(\frac{n}{4^k})$ lower bound for the k–party communication complexity of the GIP function. *Goldmann* and *Håstad* [13] found a surprising application of the BNS–lower bound to circuit–complexity.

In this paper we use multi–party techniques to characterize some hard–to–handle circuit classes.

Smolensky [10] showed an exponential lower bound for the sizes of circuits with MOD p, AND and OR gates, using algebraic methods in finite fields. Deriving superpolynomial lower bounds — without using uniformity conditions — for the size of circuits with MOD m gates remained unsuccessful, despite the widespread opinion that the powers of MOD m gates and MOD p gates do not differ considerably, where (and throughout this paper) m is a non-prime power composite number and p is a prime. Recently, for *uniform* circuits with MOD m, AND and OR gates, *Allender* and *Gore* [1] showed a subexponential lower bound for the permanent function.
On the other hand, *Kahn* and *Meshulam* [4] showed that OR_n can be computed by a depth–2 circuit with MOD $(2p)$ gates, while it can not be computed by any constant-depth circuits with MOD p gates.

We show a large gap between multi–party complexities of evaluating circuits with MOD p and MOD m gates, where a MOD r gate outputs 1 if the sum of its input bits is divisible by r, otherwise it outputs 0.

Definition 1. *Let $\mathcal{C}$ be a circuit, and let $k \geq 2$ be an integer. Let X denote the set of the input–variables of $\mathcal{C}$, i.e. $X = \{x_1, x_2, ..., x_\ell\}$. We say that circuit $\mathcal{C}$ is k-evaluated with b bits of communication, if for all partitions of X into k classes $X_1, X_2, ...X_k$, there exists a k–party protocol with players $P_1, P_2, ..., P_k$, such that all the players know circuit $\mathcal{C}$ and partition $X_1, X_2, ...X_k$, and player P_i knows the values of all the variables, except those in X_i, for $i = 1, 2, ..., k$; and the k–party protocol computes the output of the circuit,* *communicating at most b bits.*

Heuristically, we can consider a circuit to be "hard" if it needs a large number of communicated bits for evaluation, otherwise it can be said "easy". The statement of the main lemma of [2] (whose generalization is our Lemma 12.), implies that the circuit, with a PARITY gate at the top and fan–in k AND gates at level one is hard for k–party protocols. The lower bound of [13] uses the fact that any circuit, with a SYMMETRIC gate at the top, and arbitrary gates of fan–in at most $k-1$ at level 1 are easy for k–party protocols.

Szegedy has considered the (2–party) communication complexity of evaluating Boolean functions in [9], using the 2–party version of Definition 1. He proved that circuits with gates of bounded symmetric communication–complexity, can be simulated by circuits with MOD m, AND and OR gates of similar depth and size.

Obviously, if m and p are constants, then there is no difference between the evaluations of one MOD m or one MOD p gate. However, we shall show here, that if we consider two layers of MOD p gates versus two layers of MOD m gates, the difference is dramatic (Theorem 2 *vs.* Theorem 5), and the k–party technique becomes very important (Theorem 2 *vs.* Theorem 3).

Theorem 2. *Let p be a prime, $k \geq p$ an integer, and let $\mathcal{C}$ be a circuit of depth 2 and size N with a MOD p^ℓ gate on the top, for $1 \leq \ell \leq \lfloor k/p \rfloor$ and $N-1$ MOD p gates on level 1. Then $\mathcal{C}$ is k–evaluated with $O(k\ell)$ bits of communication.*

Note. *When p and k are constants, then the circuit is k–evaluated by a* constant *number of communicated bits.*

Remark. *As* Richard Beigel *pointed out to us [14], one may allow negated MOD p^ℓ and negated MOD p gates in circuit $\mathcal{C}$ in Theorem 2, since a negated MOD p gate on level 1 can be simulated with $p^\ell - 1$ copies of MOD p gates plus one constant-gate 1. If the circuit has a negated MOD p^ℓ gate at the top, then it can also be evaluated by the same protocol as the original circuit $\mathcal{C}$, as we shall see in the proof of Theorem 2.*

Theorem 3. *Let $q > k$, and $N \in \mathbf{N}$. Then there exists a depth–2, size–N circuit with MOD q gates, which needs $\Omega(\frac{N}{4^k})$ bits of communication, if evaluated by any k–party protocol.*

Let us note that the k–party protocols separate the powers of the circuits with MOD p gates and with MOD q gates, where $q > k \geq p$.

The next is an immediate corollary of Theorem 2:

Corollary 4. *Let $k \geq 2$, integer, and let $f : \{0, 1, 2, ..., m-1\}^{kn} \to \mathbf{N}$ be a function, and suppose that the k–party communication complexity of f is non–constant. Then f cannot be computed by a depth–2 circuit of MOD p gates, for $p \leq k$.* ∎

Theorem 5. *Let m be a positive integer with at least two different prime divisors, p_1 and p_2, and let N and k be positive integers. Then there exists an explicitly constructible depth–2, size–N circuit $\mathcal{C}$ with MOD m gates on the first and on the second level, such that the k-evaluation of $\mathcal{C}$ needs $\Omega(\frac{N}{c_m^k})$ bits of communication, where constant $c_m > 1$ depends only on m.*

Obviously, the k–party communication complexity of the function, computed by $\mathcal{C}$, is $\Omega(\frac{N}{c_m^k})$, so, by Corollary 4, for any $p \leq k$, this function cannot be computed by any depth–2 circuits with MOD p gates. For any m and p, choosing a $k \geq p$, this result separates the powers of depth–2 circuits with MOD m and with MOD p gates.

It is easy to see that the k'–party protocols are not weaker than the k–party protocols, for $k' > k$. Theorem 2, and, on the other hand, Theorem 3 directly imply the following hierarchy–theorem:

Theorem 6. *Let $k < k'$ two positive integers, and suppose that there is a prime p between k and k': $k < p \leq k'$. Then for all $N \in \mathbf{N}$, there exists a function of kN variable which can be computed by a k'–party protocol with a constant number of communicated bits, while any k–party protocol needs $\Omega(N)$ bits of communication to compute the function.*
∎

2. SEPARATING CIRCUIT–CLASSES

Proof of Theorem 2. By Definition 1, we must show a k–party protocol for any k–partition $\{X_1, X_2, ..., X_k\}$ of set X which evaluates $\mathcal{C}$ with $O(k\ell)$ bits of communication. Let the partition $\{X_1, X_2, ..., X_k\}$ be fixed.

The players first compose a matrix $B \in \{0, 1, 2, ..., p-1\}^{(N-1)\times k}$, then play a k–party protocol, using data only from this matrix. Let B_i denote column i, B^j row j of B, and B_i^j the entry in the intersection of B_i and B^j. Let $G_1, G_2, ..., G_{N-1}$ denote the MOD p gates on level 1 of $\mathcal{C}$. Gate G_j will be corresponded to row B^j as follows:

B_i^j is the sum, modulo p, of the values of those inputs of G_j which are in class X_i. The value of x_ℓ (or $\bar{x}_\ell$) should be added with multiplier c_ℓ if G_j is connected to x_ℓ (or to $\bar{x}_\ell$) with c_ℓ wires.

Let us observe that players can compose matrix B without any communication, and P_j knows every column of B, except B^j, $j = 1, 2, ..., k$.

It is easy to see that circuit $\mathcal{C}$ outputs 1 if and only if the number of those rows of B, whose sums are divisible by p, is 0 mod $p^{\lfloor k/p \rfloor}$.

Lemma 7. *Let $B \in \{0, 1, 2, ..., p-1\}^{n\times k}$, where p is a prime and $k \geq p$ an integer. Then there exists an explicitly constructible protocol, which computes the number, modulo p^ℓ, of those rows of B, whose sums are divisible by p. Moreover, this protocol uses $O(k\ell)$ bits of communication for $1 \leq \ell \leq \lfloor k/p \rfloor$.*

Proof. The following protocol "**MOD m**" was first described in [15] and was only used to matrices with 0-1 entries. The present version is applied to matrices with entries $\{0, 1, ..., m-1\}$, and its analysis is much more intricate than that of [15].

We state that the following protocol will satisfy the requirements, with $m = p$:

The strategy of the players in protocol **MOD m** is the following: Player P_i ($1 \leq i \leq k$) assumes that column i of B, B_i is the all–1 vector. P_1 – using his assumption –

communicates the number of rows in each congruency– classes mod m:

$$\alpha = (\alpha_0, \alpha_1, ..., \alpha_{m-1}),$$

where α_i denotes the number of those rows, whose sums are believed to be i mod m. Next P_2 corrects P_1 in case of those rows which begin with 0 or 2, or 3, or ..., $m-1$, instead of the assumed 1: P_2 communicates the corrections, to be added to vector α. P_2 computes this correction, assuming that he knows the entire input. Then P_3 corrects P_1 and P_2, in case of those rows, which begins with two non–ones, and so on, until P_k comes. Then P_k corrects $P_1, P_2, ..., P_{k-1}$ in case of those rows which begins with $k-1$ non–ones. The protocol makes errors only in the case of those rows, for which *neither of the assumptions* were satisfied: the rows without $1'$s. Every other row will be counted correctly: since at least one player's assumption was right, he saw the row correctly, and counted it to the proper congruency–class, corrected the errors of the players with lower indices. Player P_i will not count those rows, which contain a 1 in a position lower than i.

Example. Let $m = p = 3, k = 3$, and consider row 022.

P_1 assumes this row to be 122, so he counts this row to vector α as $(0, 0, 1)$.

P_2 assumes this row to be 012, so he counts it as $(1, 0, 0)$, and P_2 assumes that P_1 saw the row to be 112, and because of this, P_1 communicated $(0, 1, 0)$ for this row, which should be corrected by P_2, subtracting it. In total, P_2 adds $(1, -1, 0)$ to the α of P_1.

P_3 assumes the row to be 021, he adds $(1, 0, 0)$, and he corrects first P_1, next P_2. P_3 assumes that P_1 saw the row to be 121, and corrects him adding $(0, -1, 0)$ to α. P_3 assumes that P_2 saw the row to be 011, and corrects him by adding $(0, 0, -1)$. However, P_3 assumes that P_2 erroneously corrected P_1, P_3 thinks that P_2 thinks that P_1 saw the row to be 111, so P_2 is thought to correct P_1 adding $(-1, 0, 0)$, so P_3 corrects P_2 by adding $(1, 0, 0)$. So P_3 adds in total $(2, -1, -1)$.

The sum of the corrections here is $(3, -2, 0)$ instead of the correct value $(0, 1, 0)$.

Let us observe that $(3, -2, 0) \equiv (0, 1, 0) \pmod 3$, i.e. the value computed is correct if seen modulo 3. The following lemma gives a formula for the number, computed by our protocol for rows without entry "1". We shall see that the error is $0 \pmod{p^{\lfloor k/p \rfloor}}$.

Notation 8. *Let* $\mathbf{N}$ *denote the set of natural numbers. We denote the elements of set* $\mathbf{N}^m$ *by small–case greek letters, and we index their coordinates from* 0 *through* $m-1$*. Let* $S^{n\times k}$ *denote the set of all* $n \times k$ *matrices with entries from set* S*. Let* $B \in \{0,1,...,m-1\}^{n\times k}$*. Let*

$$\delta^{(m)}(B) = (\delta_0, \delta_1, ..., \delta_{m-1})$$

denote a vector where δ_i *is the number of those rows of* B*, which are congruent to* $i \pmod{m}$*. Let* $v \in \{0,1,...,m-1\}^k$*, then* $CT(v, B)$ *denotes the number of those rows of* B*, which are equal to* v*. Let* $\mathbf{0} = (0,0,...,0) \in \{0,1,...,m-1\}^k$*.*

Lemma 9. *Protocol* **MOD m** *computes the number*

$$\delta^{(m)}(B) - \sum_{v\in\{0,2,3,...,m-1\}^k} CT(v,B)w_v,$$

where $w_v \in \mathbf{N^m}$*, and when* v *contains* d_2 *2 coordinates,* d_3 *3's, ...,* d_{m-1} $m-1$*'s, and* d_0 *0's, then*

$$w_v = \nu\Pi^{d(v)}(I-\Pi^{m-1})^{d_2}(I-\Pi^{m-2})^{d_3}...(I-\Pi^2)^{d_{m-1}}(I-\Pi)^{d_0},$$

where $\nu = (1,0,0,...,0) \in \mathbf{N}^m$*,* $d(v) = 2d_2 + 3d_3 + ... + (m-1)d_{m-1}$*, and* Π *is the* $m \times m$ *cyclic right–shift permutation matrix:*

$$\Pi = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Let us note that a row vector multiplied by Π is the vector with coordinates shifted with one position to right. Similarly, if a row-vector is multiplied by Π^{-1} the result is the vector, with coordinates shifted with one position to left.

Before proving Lemma 9, let us see how it implies Lemma 7. Let $m = p$. Since matrix Π commutes with its own powers, one can write w_v into the form:

$$w_v = \nu \Pi^{d(v)} (I - \Pi)^k P(\Pi),$$

where $P(\Pi)$ is a polynomial of matrix Π, since $k = d_2 + d_3 + ... + d_{m-1} + d_0$, and one can write $(I - \Pi^s) = (I - \Pi)Q(\Pi)$, where Q is also a polynomial.

By the binomial theorem:

$$(I - \Pi)^p = \binom{p}{0} I - \binom{p}{1} \Pi + ... + (-1)^p \binom{p}{p} \Pi^p \equiv$$

$$\equiv I + (-1)^p \Pi^p \equiv I + (-1)^p I \equiv 0 \pmod{p},$$

so

$$((I - \Pi)^p)^{\lfloor \frac{k}{p} \rfloor} \equiv 0 \pmod{p^{\lfloor \frac{k}{p} \rfloor}}, \text{ and}$$

$$(I - \Pi)^k \equiv 0 \pmod{p^{\lfloor \frac{k}{p} \rfloor}}.$$

Hence

$$w_v \equiv 0 \pmod{p^{\lfloor \frac{k}{p} \rfloor}},$$

for all $v \in \{0, 2, 3, ..., p-1\}^k$. This means that protocol **MOD p** computes $\delta^{(m)}(B)$ mod $p^{\lfloor \frac{k}{p} \rfloor}$.

However, the players are enough to communicate their α vectors only mod p^ℓ. Hence each player communicates p numbers of size $O(\ell \log p)$, and protocol **MOD p** uses $O(k\ell p \log p) = O(k\ell)$ bits of communication, which is constant if k is constant.

Proof of Lemma 9. First we prove

Sublemma 10. *The vector, computed by protocol* **MOD m** *for a row* $v \in \{0, 2, 3, ..., p - 1\}^k$ *is the same for any permutation of the coordinates of* v.

Proof. It is enough to prove that our protocol computes the same vector for

$$v = (v_1, v_2, ..., v_i, v_{i+1}, ..., v_k)$$

and

$$v' = (v_1, v_2, ..., v_{i+1}, v_i, ..., v_k).$$

Obviously, P_s communicates the same vector for v and v' if $s \neq i$ or $s \neq i+1$. P_i assumes v to be v_{P_i} and v' to be v'_{P_i}:

$$v_{P_i} = (v_1, v_2, ..., 1, v_{i+1}, ..., v_k)$$

$$v'_{P_i} = (v_1, v_2, ..., 1, v_i, ..., v_k),$$

while P_{i+1} assumes v to be $v_{P_{i+1}}$ and v' to be $v'_{P_{i+1}}$:

$$v_{P_{i+1}} = (v_1, v_2, ..., v_i, 1, ..., v_k)$$

$$v'_{P_{i+1}} = (v_1, v_2, ..., v_{i+1}, 1, ..., v_k).$$

P_i sees v in the same congruency–class as P_{i+1} sees v', and P_i sees v' in the same congruency–class as P_{i+1} sees v. Moreover, P_i corrects players $P_1, P_2, ..., P_{i-1}$ for row v exactly as P_{i+1} corrects them in row v', and P_i corrects players $P_1, P_2, ..., P_{i-1}$ for row v' exactly as P_{i+1} corrects them in row v. P_{i+1} ,both in v and in v', corrects P_i assuming

$$(v_1, v_2, ..., 1, 1, ..., v_k).$$

So the sum of the vectors, communicated by P_i and P_{i+1} is the same for v and for v'. ∎

By Sublemma 10, we may assume that the first d_2 coordinates are $2's$, then d_3 3's,..., d_{m-1} $m-1$'s, and, at the end, d_0 0's. Let us note that the correct vector, to be added up for v to get $\delta^m(B)$, is $\nu\Pi^{d(v)}$. However:

P_1 assumes the first coordinate to be 1 instead of 2, so he communicates

$$\nu\Pi^{d(v)}\Pi^{-1}.$$

P_2 assumes the second coordinate to be 1, so he adds up $\nu\Pi^{d(v)}\Pi^{-1}$, too, but corrects P_1 by subtracting $\nu\Pi^{d(v)}\Pi^{-2}$, since the sum, supposed to be seen by P_1, is less by one. So P_2 communicates:

$$\nu\Pi^{d(v)}\Pi^{-1}(I - \Pi^{-1}).$$

P_i $(i \le d_2)$ communicates the same vector as P_{i-1} communicated plus the correction for P_{i-1}. This correction is $(-\Pi^{-1})$ times the vector, communicated by P_{i-1}, so P_i communicates:

$$\nu\Pi^{d(v)}\Pi^{-1}(I-\Pi^{-1})^{i-1}.$$

The sum of the vectors communicated by $P_1, P_2, ..., P_{d_2}$ is:

$$\beta^{(2)} = \nu\Pi^{d(v)}\Pi^{-1}\left[I + (I-\Pi^{-1}) + (I-\Pi^{-1})^2 + ... + (I-\Pi^{-1})^{d_2-1}\right] =$$

$$= \nu\Pi^{d(v)}\big(I - (I-\Pi^{-1})^{d_2}\big).$$

Remark: $d_2 = 0$ implies that $\beta^{(2)} = 0$.

P_{d_2+1} assumes v_{d_2+1} to be 1, instead of the correct 3. So P_{d_2+1} sees the sum of v one less than P_{d_2} has seen, this also applies to the corrections for $P_1, P_2, ..., P_{d_2-1}$. So P_{d_2} communicates $\nu\Pi^{d(v)}\Pi^{-1}(I-\Pi^{-1})^{d_2-1}\Pi^{-1}$ plus the correction for P_{d_2}: what is the $(-\Pi^{-2})$ times that P_{d_2} has communicated. P_{d_2} communicates:

$$\nu\Pi^{d(v)}\Pi^{-1}(I-\Pi^{-1})^{d_2-1}(\Pi^{-1}-\Pi^{-2}) =$$

$$= \nu\Pi^{d(v)}\Pi^{-2}(I-\Pi^{-1})^{d_2}.$$

P_{d_2+2} tells the same for the sum of v and the corrections for $P_1, P_2, ..., P_{d_2}$ as P_{d_2+1}, but he also corrects P_{d_2+1}, by subtracting Π^{-2} times the vector that P_{d_2+1} has communicated, so in total, P_{d_2+2} communicates:

$$\nu\Pi^{d(v)}\Pi^{-2}(I-\Pi^{-1})^{d_2}(I-\Pi^{-2}).$$

P_{d_2+i} $(i \le d_3)$ communicates

$$\nu\Pi^{d(v)}\Pi^{-2}(I-\Pi^{-1})^{d_2}(I-\Pi^{-2})^{i-1}.$$

$\beta^{(3)}$, the sum of the vectors, communicated by $P_{d_2+1}, P_{d_2+2}, ..., P_{d_2+d_3}$ is

$$\beta^{(3)} = \nu\Pi^{d(v)}(I - \Pi^{-1})^{d_2}\big(I - (I - \Pi^{-2})^{d_3}\big).$$

Similarly, $\beta^{(j)}$, the sum of the vectors, communicated by $P_{d_2+...+d_{j-1}+1}$, $P_{d_2+...+d_{j-1}+2}, ...,$ $P_{d_2+...+d_{j-1}+d_j}$, is

$$\nu\Pi^{d(v)}(I - \Pi^{-1})^{d_2}...(I - \Pi^{-j+2})^{d_{j-1}}\big(I - (I - \Pi^{-j+1})^{d_j}\big)$$

The result of the telescopic sum $\beta^{(2)} + \beta^{(3)} + ... + \beta^{(m)} + \beta^{(0)}$ is:

$$\nu\Pi^{d(v)} - \nu\Pi^{d(v)}(I - \Pi^{-1})^{d_2}...(I - \Pi^{-m+1})^{d_m}.$$

So the vector w_v is equal to

$$w_v = \nu\Pi^{d(v)}(I - \Pi^{-1})^{d_2}(I - \Pi^{-2})^{d_3}...(I - \Pi^{-m+1})^{d_m}.$$

Noticing that $\Pi^m = I$, our result follows. ∎

Proof of Theorem 5. By Definition 1, we must give a circuit $\mathcal{C}$ and a k partition $X_1, X_2, ...X_k$ of X, for which every k–party protocol needs $\Omega(\frac{N}{c_m^k})$ bits for evaluation. In fact we shall prove the statement only for k's of the form $k = p_1^c$, since if for a k–partition $X_1, X_2, ..., X_k$ the k-evaluation of circuit $\mathcal{C}$ needs a bits of communication, then for $k' < k$, and for the partition $X'_1 = X_1, ..., X'_{k'-1} = X_{k'-1}, X'_{k'} = \bigcup_{i=k'}^{k} X_i$, the k'–evaluation needs also at least a bits of communication. If we prove a lower bound of $\Omega(\frac{N}{4^k})$ for the least $k \geq k'$ of the form $k = p_1^c$, then it implies a lower bound $\Omega(\frac{N}{c_m^{k'}})$ with $c_m = 4^{p_1}$ for the original k', and that is stated in the theorem.

Let $X = \{y_1, y_2, ..., y_m; x_{11}, x_{12}, ..., x_{1k}, x_{21}, ..., x_{2k}\ , ..., x_{(N-1)1}, ..., x_{(N-1)k}\}$,

The partition on X is defined as follows: $X_1 = \{y_1, y_2, ..., y_m; x_{11}, x_{21}, ..., x_{(N-1)1}\}$, and $X_j = \{x_{1j}, x_{2j}, ..., x_{(N-1)j}\}$, for $j = 2, 3, ..., k$.

Let $q_1 = m/p_1$, and $q_2 = m/p_2$.

Circuit $\mathcal{C}$ is defined as follows: there is a MOD m gate G on the top, and MOD m gates $G_1, G_2, ..., G_{N-1}$ on the first level; the variables of X are situated on the bottom. G is connected to variables $y_1, y_2, ..., y_m$ with one–one input wire, while to each gates $G_1, G_2, ..., G_{N-1}$ with q_1 input wires. The fan–in of G is $(N-1)q_1 + m$. Gate G_i is connected to each variable from $\{x_{i1}, x_{i2}, ..., x_{ik}\}$ with q_2 input–wires, the fan–in of the MOD m gates is kq_2.

Let us remark that G_i is 1 iff $x_{i1}+x_{i2}+...+x_{ik} \equiv 0 \pmod{p_2}$. Suppose that $\sum_{i=1}^m y_i \equiv q_1 s \pmod{m}$. Then G is 1 iff $q_1 s + q_1(G_1 + G_2 + ... + G_{N-1}) \equiv 0 \pmod{m}$. Or, in other words, G is 1 iff $s + (G_1 + G_2 + ... + G_{N-1}) \equiv 0 \pmod{p_1}$.

Let A denote matrix $\{x_{ij}\}$, $i = 1, 2, ..., N-1$; $j = 1, 2, ..., k$. Because of the definition of our partition, player j knows all the columns of this matrix, except column j. Gate G_i is 1 iff the sum of row i is divisible by p_2, and gate G is 1 iff the number of those rows of A, whose sums are divisible by p_2, is congruent to $-s \pmod{p_1}$.

Suppose now, that players $P_1, P_2, ..., P_k$ evaluates circuit $\mathcal{C}$ with communicating b bits. Then for any s and for any $A \in \{0, 1, \}^{(N-1)\times k}$, they can decide, communicating b bits, whether the number of those rows of A, whose sums are divisible by p_2, is congruent to $-s \pmod{p}_1$, or not. So the players can *compute* the number, mod p_1, of those rows of A, whose sums are divisible by p_2 with bp_1 bits of communication.

The following lemma gives a lower bound to bp_1:

Lemma 11. *Let p_1 and p_2 be different primes, $k = p_1^c$, and $A \in \{0,1\}^{n\times k}$. Then any k–party protocol computing mod p_1 the number of those rows of A which are divisible by p_2, needs $\Omega(\frac{n}{4^k})$ bits of communication.*

Proof. By Lemma 9, players can compute vector

$$\delta^{(p_2)}(A) - \nu(I - \Pi)^k CT(\mathbf{0}, A) \tag{1}$$

using $O(k \log n)$ bits of communication, where $\delta_i^{(p_2)}(A)$ is the number of rows of A, whose sum is $\equiv i \pmod{p_2}$, and Π is the $p_2 \times p_2$ cyclic right-shift permutation matrix.

$$(I-\Pi)^k = \sum_{i=0}^{k}(-1)\binom{k}{i}\Pi^i \equiv I - \Pi^{p_1^c} \pmod{p_1},$$

since $k = p_1^c$ and p_1 divides $\binom{k}{i}$ if $0 < i < k$.

Since $\nu = (1,0,0,...,0)$, $\nu(I-\Pi)^{p_1^c} = \nu(I-\Pi^{p_1^c})$ is the first row of $(I-\Pi^{p_1^c})$. The first entry in the first row of $\Pi^{p_1^c}$ is 0, since $\Pi^{p_1^c} \neq I$, because p_2 does not divide p_1^c. So the first entry of vector $\nu(I-\Pi^{p_1^c})$ is 1, thus the first coordinate of vector $\delta^{(p_2)}(A) - \nu(I-\Pi)^k CT(\mathbf{0}, A)$ is

$$\delta_0^{(p_2)}(A) - CT(\mathbf{0}, A) \pmod{p_1}. \tag{2}$$

From the assumption, $\delta_0^{(p_2)}(A) \bmod p_1$ is computed by the protocol, say, with z bits of communication. Then, because of (2), $CT(\mathbf{0}, A) \bmod p_1$ can also be computed using $z + O(k \log n)$ bits of communication. The following generalization of ([2], Theorem 1) yields that $z + O(k\log n) = \Omega(\frac{n}{4^k})$.

Lemma 12. *Let p be a prime, and $A \in \{0,1,\}^{n\times k}$. Then any k–party protocol, which computes $CT(\mathbf{0}, A)$ mod p, needs $\Omega(\frac{n}{4^k})$ bits of communication.*

Proof. We adopt the notation and some of the definitions of [2]. Let $S \subset \{0,1\}^{n\times k}$. S is called a *cylinder* if the membership of S does not depend on column i, for some $i \in \{1,2,...,k\}$. S is called a *cylinder–intersection* if it can be represented as the intersection of some cylinders.

It is easy to verify that for any k–party protocol, the subset $S \subset \{0,1\}^{n\times k}$, whose elements, if they are taken as inputs, lead to the same string s of communicated bits, is a cylinder intersection. Any cylinder intersection in $\{0,1\}^{n\times k}$ can be represented as the intersection of at most k cylinders.

Definition 13. *Let $g : \{0,1\}^{n\times k} \to \{0,1,2,...,p-1\}$ be a function. The discrepancy of g is*

$$\Gamma(g) = \max_S \Big|\sum_{i=0}^{p-1} \varepsilon^i Pr(g(A) = i, A \in S)\Big|,$$

where ε is a p-th complex root of unity, which minimizes $|1+\varepsilon|$, and A is chosen uniformly from $\{0,1\}^{n\times k}$, and S runs over all the cylinder intersections of $\{0,1\}^{n\times k}$.

Lemma 14. *([2], Lemma 2.2.) For any function g:*

$$C(g) \geq \log\left(\frac{1}{\Gamma(g)}\right).$$

Proof. Let S_0 be the cylinder–intersection of the largest probability, on which g is constant. Then $\Pr(S_0) \leq \Gamma(g)$, and, on the other hand, $C(g) \geq \log\left(\frac{1}{\Pr(S_0)}\right)$. ∎

Let $g(A) = g_{n,k,p}(A) = CT(\mathbf{0}, A) \bmod p$, and let

$$f(A) = \varepsilon^{g(A)} = \varepsilon^{CT(\mathbf{0},A)}.$$

Let

$$\Delta^{(k)}(n) = \max_{\phi_1,\phi_2,\ldots,\phi_k} |\mathop{\mathrm{E}}_{A}(f(A)\phi_1\phi_2...\phi_k)|,$$

where ϕ_i is a shorthand for $\phi_i(A) = \phi_i(A_1, A_2, ..., A_k)$, where A_j denotes column j of matrix A, and where the maximum is taken over all functions $\phi_i : \{0,1\}^{n\times k} \to \{0,1\}$ such that ϕ_i does not depend on A_i. E denotes the expected value on the uniformly distributed $A = (A_1, A_2, ..., A_k) \in \{0,1\}^{n\times k}$.

Let us note that $\Delta^{(k)}(n) = \Gamma(g_{n,k,p})$. Because of Lemma 14, an upper bound to $\Delta^{(k)}(n)$ yields a lower bound to $C(g)$.

Lemma 15.

$$\Delta^{(k)}(n) \leq \mu_k^n,$$

where $\mu_1 = \frac{1}{2}$, and $\mu_i = \sqrt{\frac{1+\mu_{i-1}}{2}}$.

Note: It is easy to show by induction that $\mu_k \leq 1 - 4^{-k}$, which is about $e^{-4^{-k}}$.

Proof. The proof is by induction. For $k = 1$,

$$\Delta^{(1)}(n) \leq 2^{-n}\left|\binom{n}{0}\varepsilon^0 + \binom{n}{1}\varepsilon^1 + ... + \binom{n}{n}\varepsilon^n\right| =$$

$$= 2^{-n}|(1+\varepsilon)^n| \le 2^{-n} = \mu_1^n,$$

since $|(1+\varepsilon)| \le 1$. Let $k \ge 2$. Since ϕ_k does not depend on A_k:

$$\Delta^{(k)}(n) \le \mathop{\mathrm{E}}_{A_1,A_2,...,A_{k-1}} \Big|\Big(\mathop{\mathrm{E}}_{A_k} (f(A)\phi_1\phi_2...\phi_{k-1})\Big)\Big|.$$

We will use the following version of the Cauchy-Schwarz inequality:

Cauchy–Schwarz inequality. *For any random variable x:*

$$(E(x))^2 \le E(x^2).$$

Using the Cauchy–Schwarz inequality with

$$x = \Big| \mathop{\mathrm{E}}_{A_k} (f(A_1, A_2, ..., A_k)\phi_1\phi_2...\phi_{k-1})\Big|,$$

and noticing that

$$x^2 = \Big| \mathop{\mathrm{E}}_{A_k} (f(A)\phi_1\phi_2...\phi_{k-1})\Big|^2 =$$

$$= \Big(\mathop{\mathrm{E}}_{A_k} (f(A)\phi_1\phi_2...\phi_{k-1})\Big)\Big(\mathop{\mathrm{E}}_{A_k} (\bar{f}(A)\phi_1\phi_2...\phi_{k-1})\Big),$$

where $\bar{f}$ denotes the complex conjugate of f.

We can estimate

$$\text{(3)} \qquad \Delta^{(k)}(n) \le \left[\mathop{\mathrm{E}}_{A_1,A_2,...,A_{k-1}} \Big(\mathop{\mathrm{E}}_{A_k} (f(A)\phi_1\phi_2...\phi_{k-1})\Big)\Big(\mathop{\mathrm{E}}_{A_k} (\bar{f}(A)\phi_1\phi_2...\phi_{k-1})\Big)\right]^{\frac{1}{2}} =$$

$$= \left[\mathop{\mathrm{E}}_{U,V,A_1,A_2,...,A_{k-1}} \Big(f^U \bar{f}^V \phi_1^U \phi_1^V \phi_2^U \phi_2^V ... \phi_{k-1}^U \phi_{k-1}^V \Big)\right]^{\frac{1}{2}}$$

where $U, V \in \{0,1\}^n$, and f^U stands for $f(A_1, A_2, ..., A_{k-1}, U)$, $\bar{f}^V$ stands for $\bar{f}(A_1, A_2, ..., A_{k-1}, V)$, and ϕ_i^U stands for $\phi_i(A_1, A_2, ..., A_{k-1}, U)$, ϕ_i^V stands for $\phi_i(A_1, A_2, ..., A_{k-1}, V)$.

Note: The domain of f^U, f^V and ϕ_i^U, ϕ_i^V is $\{0,1\}^{n\times(k-1)}$.

Let us partition the rows of matrix $A' = (A_1, A_2, ..., A_{k-1})$ into four classes: A_{00}, A_{11}, A_{01} and A_{10}, where A_{xy} contains row i of A' iff $U_i = x, V_i = y$, $1 \le i \le n$,

$x, y \in \{0,1\}$. Let f^U_{xy} denote the restriction of f^U to A_{xy}: $f^U_{xy} = \varepsilon^{CT(\mathbf{0},A_{xy})}$, for $x, y \in \{0,1\}$. f^V is defined similarly.

From the definition of f:

$$f^U = f^U_{00} f^U_{01} f^U_{10} f^U_{11}, \text{ and } f^V = f^V_{00} f^V_{01} f^V_{10} f^V_{11}.$$

So

$$f^U \bar{f}^V = f^U_{00} \bar{f}^V_{00} f^U_{01} \bar{f}^V_{01} f^U_{10} \bar{f}^V_{10} f^U_{11} \bar{f}^V_{11}.$$

Let us observe that $f^U_{11} \bar{f}^V_{11} = 1$, since among those rows there are no all–0 ones, because their last coordinates are 1. $f^U_{00} = f^V_{00} = \varepsilon^{CT(\mathbf{0},A_{00})}$, so $f^U_{00} \bar{f}^V_{00} = 1$. Moreover, $f^U_{10} = \varepsilon^0 = 1$, $\bar{f}^V_{01} = \varepsilon^0 = 1$, so we have got:

$$f^U \bar{f}^V = f^U_{01} \bar{f}^V_{10}.$$

For $i = 1, 2, ..., k-1$, let A_i be composed of two parts: B_i and C_i, where C_i corresponds to the coordinates of A_i in the rows of A_{10}, and B_i to the remaining coordinates.

Let$\xi_i^{U,V,B_1,B_2,...,B_{k-1}}(C_1, C_2, ..., C_{k-1}) = \phi_i^U(A_1, A_2, ..., A_{k-1}) \phi_i^V(A_1, A_2, ..., A_{k-1})$. Then we can estimate (3):

$$\Delta^{(k)}(n) \leq \left[\mathop{\mathrm{E}}_{U,V} \left| \mathop{\mathrm{E}}_{B_1,B_2,...,B_{k-1}} f^U_{01} \left(\mathop{\mathrm{E}}_{C_1,C_2,...,C_{k-1}} \left(\bar{f}^V_{10} \xi_1 \xi_2 ... \xi_{k-1} \right) \right) \right| \right]^{\frac{1}{2}},$$

since f^U_{01} does not depend on the C_i's.

From the induction hypothesis:

$$\left| \mathop{\mathrm{E}}_{C_1,C_2,...,C_{k-1}} \left(\bar{f}^V_{10} \xi_1 \xi_2 ... \xi_{k-1} \right) \right| \leq \mu_{k-1}^{m_{10}},$$

where m_{10} is the number of rows in A_{10}.

For $i = 1, 2, ..., k-1$ let B_i be composed of two parts: D_i and F_i, where F_i corresponds to the coordinates of B_i in the rows of A_{01}, and D_i corresponds to the remaining coordinates.

Then

$$\Delta^{(k)}(n) \le \left[\mathop{\mathrm{E}}_{U,V,D_1,D_2,...,D_{k-1}} \left(\mu_{k-1}^{m_{10}} \left| \mathop{\mathrm{E}}_{F_1,F_2,...,F_{k-1}} \left(f_{01}^U \right) \right| \right) \right]^{\frac{1}{2}}.$$

Again, from the induction hypothesis, choosing $\phi_1 = \phi_2 = ... = \phi_{k-1} = 1$:

$$\left| \mathop{\mathrm{E}}_{F_1,F_2,...,F_{k-1}} \left(f_{01}^U \right) \right| \le \mu_{k-1}^{m_{01}},$$

where m_{01} is the number of the rows of A_{01}. So we have got

$$\Delta^{(k)}(n) \le \left[\mathop{\mathrm{E}}_{U,V,D_1,D_2,...,D_{k-1}} \left(\mu_{k-1}^{m_{10}+m_{01}} \right) \right]^{\frac{1}{2}}.$$

$m_{10} + m_{01}$ is equal to the number of those coordinates i: $U_i \ne V_i$. Since U and V is distributed uniformly, the probability that $m_{10} + m_{01} = m$ is $\binom{n}{m} 2^{-n}$, so:

$$\Delta^{(k)}(n) \le \left(\sum_{m=0}^{n} \binom{n}{m} 2^{-n} \mu_{k-1}^m \right)^{\frac{1}{2}} = \left(2^{-n} (1 + \mu_{k-1})^n \right)^{\frac{1}{2}} = \mu_k^n,$$

and this completes the proof of Lemma 15. ∎

Lemma 15 yields that $\Delta^{(k)}(n) \le \mu_k^n \le e^{-n4^{-k}}$, and from Lemma 14:

$$C(g) \ge \log \left(e^{n4^{-k}} \right) = \Omega\left(\frac{n}{4^k} \right)$$

which completes the proof of Lemma 12. ∎

We have got that $z + O(k \log n) = \Omega(\frac{n}{4^k})$, that is, $z = \Omega(\frac{n}{4^k})$, so any protocol computing $\delta_0^{(p_2)}(A) \bmod p_1$ needs $\Omega(\frac{n}{4^k})$ bits of communication, and this is the statement of Lemma 11. ∎

Since $bp_2 = \Omega(\frac{n}{4^k})$, then $b = \Omega(\frac{n}{4^k})$ also holds, thus evaluating circuit $\mathcal{C}$ needs also $\Omega(\frac{n}{4^k})$ bits of communication for $k = p_1^c$, and $\Omega(\frac{n}{c_m^k})$ bits for general k. This completes the proof of Theorem 5. ∎

Proof of Theorem 3. Let p be a prime–divisor of q. Let $X = \{y_1, y_2, ..., y_q; x_{11}, x_{12}, ..., x_{1k},\ x_{21}, ..., x_{2k}, ..., x_{(N-1)1}, ..., x_{(N-1)k}\}$, The partition on X

is defined as follows: $X_1 = \{y_1, y_2, ..., y_q; x_{11}, x_{21}, ..., x_{(N-1)1}\}$, and $X_j = \{x_{1j}, x_{2j}, ..., x_{(N-1)j}\}$, for $j = 2, 3, ..., k$. Let $q_1 = q/p$.

Circuit $\mathcal{C}'$ is defined as follows: there is a MOD q gate G on the top, and MOD q gates $G_1, G_2, ..., G_{N-1}$ on the first level; the variables of X are situated on the bottom. G is connected to variables $y_1, y_2, ..., y_q$ with one–one input wire, while to each gates $G_1, G_2, ..., G_{N-1}$ with q_1 input wires. The fan–in of G is $(N-1)q_1 + q$. Gate G_i is connected to each variable from $\{x_{i1}, x_{i2}, ..., x_{ik}\}$ with 1 input–wire. The fan–in of the MOD q gate G_i is k, for $i = 1, 2, ..., N-1$.

Let us remark that G_i is 1 iff $x_{i1} = x_{i2} = ... = x_{ik} = 0$. Suppose that $\sum_{i=1}^{q} y_i \equiv q_1 s \pmod q$. Let A denote matrix $\{x_{ij}\}$, $i = 1, 2, ..., N-1$; $j = 1, 2, ..., k$. Then G is 1 iff $q_1 s + q_1 CT(\mathbf{0}, A) \pmod q$. Or, in other words, G is 1 iff $s + CT(\mathbf{0}, A) \equiv 0 \pmod p$. Because of the definition of our partition, player j knows all the columns of matrix A, except column j. Gate G_i is 1 iff row i is the all–0 row, and gate G is 1 iff the number of the all–0 rows of A is congruent to $-s \pmod p$.

Suppose now, that players $P_1, P_2, ..., P_k$ evaluates circuit $\mathcal{C}'$ with communicating b bits. Then for any s and for any $A \in \{0, 1, \}^{(N-1)\times k}$, they can decide, communicating b bits, whether the number of the all–0 rows of A, is congruent to $-s \pmod p)_1$, or not. So the players can *compute* the number of the all–0 rows of A, mod p. From Lemma 12 our statement follows. ∎

REFERENCES

[1] E. Allender, V. Gore: A Uniform Circuit Lower Bound for the Permanent, *preprint.*

[2] L. Babai, N. Nisan, M. Szegedy: Multiparty Protocols and Pseudorandom Sequences, *Proc. 21st ACM STOC,* 1989, pp. 1-11.

[3] A. K. Chandra, M. L. Furst, R. J. Lipton: Multi-party Protocols, *Proc. 15th ACM STOC,* 1983, pp. 94–99.

[4] J. Kahn, R. Meshulam: On mod p Transversals, *Combinatorica*, 1991, (11) No. 1. pp. 17–22.

[5] M. Karchmer, A. Wigderson: Monotone Circuits for Connectivity Require Super–Logarithmic Depth, *Proc. 20th ACM STOC,* 1988, pp.

[6] L. Lovász: Communication Complexity: A Survey, *Technical Report, CS–TR–204–89*, Princeton University, 1989.

[7] R. Raz, A. Wigderson: Probabilistic Communication Complexity of Boolean Relations. *Proc. 30th IEEE FOCS,* 1989, pp. 562-567

[8] R. Raz, A. Wigderson: Monotone Circuits for Matching Require Linear Depth. *Proc. 22nd ACM STOC*, pp. 287–292.

[9] M. Szegedy: Functions with Bounded Symmetric Communication Complexity and Circuits with MOD m Gates, *Proc. 22nd ACM STOC,* pp. 278–286.

[10] R. Smolensky, Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity, *Proc. 19th IEEE FOCS*, pp. 77-82, (1987).

[11] A.C. Yao: Some Complexity Questions Related to Distributive Computing, *Proc. 11th ACM STOC*, 1979, pp. 209–213.

[12] A.C. Yao: Circuits and Local Computation, *Proc. 21st ACM STOC,* 1989, pp. 186–196

[13] M. Goldmann, J. Håstad: On the Power of Small–Depth Threshold Circuits, *Proc. 31st IEEE FOCS*, 1990, pp. 610–618.

[14] R. Beigel, private correspondence.

[15] V. Grolmusz: Circuits and Multi–Party Protocols, Technical Report No. MPII-1992-104, Max Planck Institute for Computer Science, Saarbruecken, Germany, 1992,