Multiparty Communication Complexity of Vector–Valued and Sum–Type Functions

Ulrich Tamm

German Language Department of Business Informatics,
Marmara University, Istanbul, Turkey
tamm@ieee.org

Dedicated to the memory of Rudolf Ahlswede

Abstract. Rudolf Ahlswede's work on communication complexity dealt with functions defined on direct sums: vector–valued functions and sum–type functions. He was interested in single–letter characterizations and provided several lower bound techniques to this aim. In this paper we shall review these lower bounds and extend them to the "number in hand" multiparty model of communication complexity.

Keywords: communication complexity, direct sum functions, tensor product.

1 Introduction

Sum-type functions f_n and vector-valued functions f^n are defined on the powers $\mathcal{X}^n, \mathcal{Y}^n$ of the sets from the domain of some basic function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Elements of $\mathcal{X}^n$ and $\mathcal{Y}^n$ are denoted as x^n and y^n, respectively. Hence, e. g., $x^n = (x_1, \ldots, x_n)$ for some $x_1, \ldots, x_n \in \mathcal{X}$. With this notation

$$f^n(x^n, y^n) = \big(f(x_1, y_1), \ldots, f(x_n, y_n)\big), \qquad f_n(x^n, y^n) = \sum_{i=1}^{n} f(x_i, y_i),$$

where it is required that the range $\mathcal{Z}$ is a subset of an additive group G.

Motivated by the communication complexity of the Hamming distance [8], in a series of papers Rudolf Ahlswede ([1] - [7]) and his group in Bielefeld ([17] - [20]) studied the communication complexity of sum–type and vector–valued functions. The results are summarized in [21]. Rudolf Ahlswede was mainly interested in a single–letter characterization basing the communication complexity of f_n and f^n on the communication complexity of the function f. To this aim he and his coauthors demonstrated that several lower bounds behave multiplicatively. These results and also their applications yielding the exact communication complexity for special functions as Hamming distance and set intersection are presented in Section 2.

H. Aydinian, F. Cicalese, and C. Deppe (Eds.): Ahlswede Festschrift, LNCS 7777, pp. 451–462, 2013.

A further line of research leading to direct sum methods in communication complexity goes back to the question if it is easier to solve communication problems simultaneously than separately, cf. [15], pp. 42 - 48. Recall the definition of a vector-valued function $f^n((x_1,\ldots,x_n),(y_1,\ldots,y_n)) = (f(x_1,y_1),\ldots,f(x_n,y_n))$. An obvious upper bound on the communication complexity $C(f^n)$ is obtained by evaluating each component $f(x_i,y_i)$ separately and communicating the result for component i using the optimal protocol for f. Can we do better by considering all components simultaneously? Ahlswede et al. [3], [5] using data compression could show that for set intersection it is $C(f) = 2$ but $C(f^n) = \lceil n \cdot \log_2 3 \rceil$.

The measure $\limsup_{n\to\infty} \frac{1}{n} C(f^n)$ is also called amortized communication complexity (see [13]). One of the main open problems in communication complexity is the question if there can exist a significant gap between the communication complexity and the amortized communication complexity of a function. Direct sum methods in communication complexity are also useful in the comparison of lower bound techniques and the study of their power. The famous log-rank conjecture states that the gap between the rank lower bound and the communication complexity cannot be too big.

The last problem was recently extended to the "number in hand" model of multiparty communication complexity [12]. Yao's model of communication complexity can be generalized to several multiparty models depending on the information accessible to each person. Most well studied is the "number on the forehead" model in which each person knows all inputs but her own, for instance [10]. The "number in hand" model, in which each person knows just her own input, was not so popular in the beginning but later found an important application in streaming [9]. The problem with "number in hand" is that a generalization of the lower bound techniques is rather difficult. The most powerful lower bound in two-party communication complexity is the rank lower bound. But the rank of a matrix is generalized by a tensor rank (3 and higher dimensional matrices), which is not so easy to determine. Besides, the matrix rank is multiplicative under the tensor product (very important for functions on direct sums). This is no longer the case for higher dimensional tensors, cf. [11].

Vector-valued and sum-type functions can straightforwardly be extended to functions in 3 and more arguments. In Section 3 we shall study the multiparty communication complexity of several generalizations of the Hamming distance and set intersection, the two functions mainly discussed by Ahlswede and his coauthors. Fortunately, a bound introduced by Ahlswede and Cai [3] via the independence number can replace the rank lower bound in this case, such that sharp lower bounds are still possible. This will be demonstrated with four boolean functions in more than two arguments.

For sum–type functions the independence number is not an appropriate lower bound. Since also the rank lower bound is not easily applicable, it hence remains to study largest monochromatic rectangles. In Section 4 a generalization of Ahlswede's 4-word property is presented. This yields tight bounds on the size of the largest monochromatic rectangles for some functions only if there is an even

number of persons involved in the multiparty communication. For odd numbers one dimension can not be included in the formula.

As an example that it may occur that three-party communication behaves much like two-party communication the pairwise comparison of the inputs is analyzed in Section 5.

2 Bounds on Communication Complexity

The notion of communication complexity was introduced by Yao in 1979 [22]. Since then it found many applications in Computer Science, for which we refer to the books by Kushilevitz and Nisan [15] or by Hromkovic [14]. The communication complexity of a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ (where $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ are finite sets), denoted as $C(f)$, is the number of bits that two persons, P_1 and P_2, have to exchange in order to compute the function value $f(x, y)$, when initially P_1 only knows $x \in \mathcal{X}$ and P_2 only knows $y \in \mathcal{Y}$. To this aim they follow a predetermined interactive protocol in which the set of messages a person is allowed to send at each instance of time form a prefix code.

Upper bounds are usually obtained by special protocols. Often, the trivial protocol, in which one person sends all the bits of his input and the other person returns the result, is at least asymptotically optimal.

Lower bounds are expressed via the function matrix $M(f) = \big(f(x, y)\big)_{x\in\mathcal{X}, y\in\mathcal{Y}}$. and the function value matrices $M_z(f) = (a_{xy})_{x\in\mathcal{X}, y\in\mathcal{Y}}$ for all $z \in \mathcal{Z}$ defined by $a_{xy} = \begin{cases} 1 & \text{if } f(x, y) = z \\ 0 & \text{if } f(x, y) \neq z. \end{cases}$

Yao [22] already showed that $C(f) \geq \log D(f)$, where the decomposition number $D(f)$ denotes the minimum size of a partition of $\mathcal{X} \times \mathcal{Y}$ into monochromatic rectangles, i. e., products $A \times B$ of pairs $A \subset \mathcal{X}, B \subset \mathcal{Y}$ on which the function is constant. The decomposition number usually is hard to determine, however, further lower bounds can be derived from it. Immediately, we have

$$C(f) \geq \left\lceil \log \frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{Lmr\big(M(f)\big)} \right\rceil, \tag{1}$$

where $Lmr\big(M(f)\big)$ denotes the size of the largest monochromatic rectangle in the function matrix $M(f)$.

In order to make induction proofs possible Ahlswede weakened the conditions on the rectangles. He no longer required that the function is constant on the rectangle $A \times B$ but that the so called 4-word- property has to be fulfilled, i. e., for all $a, a' \in A$, $b, b' \in B$

$$f(a, b) - f(a', b) - f(a, b') + f(a', b') = 0$$

Denoting by $Lfw(f)$ the size of the largest rectangle, on which the 4-word-property holds, we obtain

$$C(f) \geq \left\lceil \log \frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{Lfw(f)} \right\rceil. \tag{2}$$

A z–independent set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ for the function value z in $M(f)$ is a set of pairs with $f(x^{(i)}, y^{(i)}) = z$ for all $i = 1, \ldots, N$ such that no two members of the set are in the same monochromatic rectangle. Denoting the size of a z–independent set by $ind(M_z(f))$ and $Ind(f) = \sum_{z \in \mathcal{Z}} ind(M_z(f))$ we obtain [3]

$$C(f) \geq \lceil \log Ind(f) \rceil. \tag{3}$$

$C(f)$ can also be lower bounded by the rank of the corresponding function matrices

$$C(f) \geq \lceil \log r(f) \rceil, \text{ where } r(f) = \sum_{z \in \mathcal{Z}} \text{rank} M_z(f) \tag{4}$$

It can be shown that the function f has the same communication complexity as the function g defined by $g(x, y) = c^{f(x,y)}$ for all x, y, when the number c is chosen appropriately ($c \neq 0, |c| \neq 1$). So it is also possible to lower bound $C(f)$ by the rank of $M(g) = \exp(M(f), c) = (c^{f(x,y)})_{x \in \mathcal{X}, y \in \mathcal{Y}}$, the exponential transform of the matrix $M(f)$. This yields

$$C(f) \geq \lceil \log \text{rank } \exp(M(f), c) \rceil. \tag{5}$$

Central in the following arguments is the observation that the function matrices of the vector-valued and sum-type functions can be expressed in terms of the Kronecker product or tensor product, defined for two matrices $A = (a_{ij})_{i,j}$ and $B = (b_{kl})_{k,l}$ as $A \otimes B = (a_{ij} \cdot b_{kl})_{i,j,k,l}$. The n-fold Kronecker product of a matrix is denoted as $A^{\otimes n}$. We have (cf. [3], [4], [17], [19])

$$M_{(z_1, \ldots, z_n)}(f^n) = M_{z_1}(f) \otimes M_{z_2}(f) \otimes \cdots \otimes M_{z_n}(f) \tag{6}$$

$$M_z(f_n) = \sum_{\substack{(z_1, \ldots, z_n) \\ z_1 + \cdots + z_n = z}} M_{z_1}(f) \otimes \cdots \otimes M_{z_n}(f) \tag{7}$$

$$\exp\big(M(f_n), c\big) = \big[\exp\big(M(f), c\big)\big]^{\otimes n} \tag{8}$$

It can be shown that the parameters in the bounds (2) - (5) behave multiplicatively, since the rank and hence also $r(f) = \sum_{z \in Z} \text{rank} M_z(f)$ are multiplicative under the Kronecker product.

Theorem 1. *([3], [4]):*

$$Lfw(f_n) = n \cdot Lfw(f) \tag{9}$$

$$rank \exp\big(M(f_n), c\big) = (rank\big[\exp\big(M(f), c\big)\big])^n \tag{10}$$

$$r(f^n) = r(f)^n \tag{11}$$

$$Ind(f^n) \geq Ind(f)^n \tag{12}$$

Using these bounds Ahlswede et al. analyzed several sum – type functions especially the Hamming distance and set intersection defined by the basic function matrices $M(h) = \begin{pmatrix} 0\,1 \\ 1\,0 \end{pmatrix}$ and $M(si) = \begin{pmatrix} 0\,0 \\ 0\,1 \end{pmatrix}$

For sum-type and vector–valued functions defined on more than two arguments the corresponding function tensors (higher dimensional function matrices) can again be described in terms of the tensor product, where $A \otimes B$ is the tensor obtained by multiplying each entry of A with each entry of B. This way, the descriptions (6), (7), and (8) generalize. However, the tensor product now is no longer multicplicative (cf. [11]), that means, $\mathrm{rank}(A \otimes B)$ can be smaller than $\mathrm{rank}(A) \cdot \mathrm{rank}(B)$. So, the rank lower bounds (10) and (11) can no longer be applied. However, for vector–valued functions still (12) and for sum–type functions (9) can be used.

3 Multiparty Communication Complexity of Vector-Valued Functions

In this section we try to extend Rudolf Ahlswede's methods to determine the multi–party communication complexity of some functions defined on direct sums. First is repeated the result for symmetric difference and the set intersection, since we need it for the multiparty protocols below.

Theorem 2. *([3], [5]):*

$$C(h^n) = 2n, \qquad C(si^n) = \lceil \log 3 \rceil \tag{13}$$

Proof: For the symmetric difference h^n, the trivial protocol requires $\lceil \log 2^n \rceil + \lceil \log 2^n \rceil = 2n$ bits of communication. With the rank lower bound (4) and (11) this can be shown to be optimal, since

$$C(h^n) \geq \log r(h^n) = n \cdot \log r(h) = n \cdot \log(\mathrm{rank}M_0(h) + \mathrm{rank}M_1(h))$$

$$= n \cdot \log(\mathrm{rank} \begin{pmatrix} 1\,0 \\ 0\,1 \end{pmatrix} + \mathrm{rank} \begin{pmatrix} 0\,1 \\ 1\,0 \end{pmatrix}) = n \cdot \log 4 = 2n$$

For set-intersection the rank lower bound yields

$$C(si^n) \geq \lceil \log r(si^n) \rceil = \lceil n \cdot \log r(si) \rceil = \lceil n \cdot \log(\mathrm{rank}M_0(si) + \mathrm{rank}M_1(si)) \rceil$$

$$= \lceil n \cdot \log(\mathrm{rank} \begin{pmatrix} 1\,1 \\ 1\,0 \end{pmatrix} + \mathrm{rank} \begin{pmatrix} 0\,0 \\ 0\,1 \end{pmatrix}) \rceil = \lceil n \cdot \log 3 \rceil$$

In order to obtain the same upper bound, we shall modify the trivial protocol, which would require $2n$ bits of transmission. Again, in the first round person P_1 encodes his input $x^n \in \{0,1\}^n$. P_2 then knows both values and hence is able to

compute the result $si^n(x^n, y^n)$, which is returned to P_1. However, in knowledge of x^n the set of possible function values is reduced to the set $S(x^n) = \{y^n : y^n \subset x^n\}$. Hence, only $\lceil \log S(x^n) \rceil$ bits have to be reserved for the transmission of $si^n(x^n, y^n)$ such that P_1 can assign longer messages to elements with few subsets. So, in contrast to the trivial protocol, the messages $\{\phi_1(x^n) : x^n \in \{0,1\}^n\}$ are now of variable length. Since the prefix property has to be guaranteed, Kraft's inequality for prefix codes yields a condition, from which the upper bound can be derived. Specifically, we require that to each x^n there corresponds a message $\phi_1(x^n)$ of (variable) length $l(x^n)$ such that for all $x^n \in \{0,1\}^n$ the sum $l(x^n) + \lceil \log S(x^n) \rceil$ takes a fixed value, L say. Kraft's inequality states that a prefix code exists, if $\sum_{x^n} 2^{-l(x^n)} \leq 1$. This is equivalent to $\sum_{x^n} 2^{\lceil \log S(x^n) \rceil} \leq 2^L$. With the choice $L = \lceil \log 3^n \rceil$ Kraft's inequality holds.

The functions in more than 2 arguments below are canonical extensions of the symmetric difference (r and s below) and the set intersection function (basic functions t and u). Namely, the basic functions defined on the product $\{0,1\} \times \{0,1\} \times \cdots \times \{0,1\}$ are the following boolean functions:

1) $r(x_1, x_2, \ldots, x_k) = x_1 + x_2 + \cdots + x_k \bmod 2$

2) $s(x_1, x_2, \ldots, x_k) = \begin{cases} 1\ , x_1 = x_2 = \cdots = x_k \\ 0\ ,\ \text{else} \end{cases}$

3) $t(x_1, x_2, \ldots, x_k) = \begin{cases} 1\ , x_1 = x_2 = \cdots = x_k = 1 \\ 0\ ,\ \text{else} \end{cases}$

4) $u(x_1, x_2 \ldots, x_k) = \begin{cases} 1\ \text{ if at least half of the inputs } x_i = 1 \\ 0\ \ \text{else} \end{cases}$

The big problem is that for more than 2 parties communicating the rank lower bound loses much of its power. The function matrices are now replaced by tensors (i.e. , higher dimensional matrices). The rank of a matrix can be extended to tensors, but it is not so easy to determine any more. The most efficient methods to determine the rank of a matrix - eigenvalues and diagonalization of matrices - cannot be applied any more. As for matrices, the rank of a tensor can be combinatorially expressed as the minimal number of rank 1 tensors whose sum is the tensor, the rank of which has to be determined. Unfortunately, this tensor rank does not behave multiplicatively under the tensor product. This means that the rank lower bound for sum–type and vector–valued functions can not be easily applied any more.

As an alternative the independence number may be considered for vector–valued functions. Following the argumentation by Ahlswede and Cai in [3], as for functions in two arguments $Ind(f^n) \geq Ind(f)^n$ also holds for vector–valued functions in $k > 2$ arguments, such that we have the lower bound $C(f^n) \geq n \cdot \log Ind(f)$. In general, the independence number is, of course, very difficult to determine, but for basic functions over small alphabets it may yield sharp bounds, as in the following theorem.

Theorem 3.

$$C(r^n) = k \cdot n$$

$$\lceil n \cdot \log(k+2) \rceil \leq C(s^n) \leq \lceil n \cdot (2 \log k) + k - 3 \rceil$$

$$\lceil n \cdot \log(k+1) \rceil \leq C(t^n) \leq \lceil n \cdot \log(k+1) \rceil + k - 2$$

$$C(u^n) = \lceil n \cdot \log 6 \rceil \text{ for } k = 3.$$

Proof: Obviously, no two of the 2^k entries of the function tensor of r can be contained in a monochromatic rectangle such that the independence number $Ind(r) = 2^k$. Hence $C(r^n) \geq \log Ind(r)^n = \log 2^{kn} = k \cdot n$, which is also the complexity of the trivial protocol, in which all k persons transmit all their inputs.

Next, let us consider the vector–valued function $t^n(x_1^n, x_2^n, \ldots x_k^n)$, which gives the intersection of the k sets represented by the binary strings $x_1^n, \ldots, x_k^n$. The function tensor of the basic function t contains exactly one entry 1 namely for $x_1 = x_2 = \cdots = x_k = 1$, i.e., the all–1 vector of length k . All other entries are 0. The k neighbours of the all–1 vector, i.e. all $(x_1, \ldots, x_k)$ with exactly one $x_i = 0$ and all other $x_j = 1$ obviously must be contained in different monochromatic rectangles. Since also the all-1 vector must be contained in a separate monochromatic rectangle, the independence number $Ind(t) = k + 1$ and hence $C(t^n) \geq \lceil n \log Ind(t) \rceil = \lceil n \log(k+1) \rceil$.

A protocol that almost achieves this lower bound is again obtained by assigning an appropriate prefix code to the messages in the trivial protocol. As for the set intersection function si^n in two arguments, again Person 1 can assign longer messages to inputs with few 1s. The other persons then can determine the exact value following an optimal protocol for set intersection of $k-1$ sets. For $k = 2$ we already know that $\lceil n \cdot \log 3 \rceil$ bits are optimal. So, for $k = 3$, Person 1 transmits $l(x)$ bits, say for an input x. Since the total number of bits transmitted should be a fixed value L, say, $L = l(x) + f(x)$, where $f(x)$ is the number of bits the other persons should still transmit to agree on the result. In order to guarantee the existence of a prefix code, Kraft's inequality $\sum_x 2^{-l(x)} \leq 1$ must hold. This is equivalent to $\sum_x 2^{-(L-f(x))} \leq 1$ or $\sum_x 2^{f(x)} \leq 2^L$. Now if Person 1 has an input $x = x_1$ with exactly i many 1's then by the protocol for si we know already that $f(x) = \lceil i \cdot \log 3 \rceil$ bits are enough to determine the set intersection of the remaining two sets by persons 2 and 3. So, Kraft's inequality reduces to $\sum_i \binom{n}{i} 2^{\lceil i \log 3 \rceil} \leq 2^L$. This can be assured by the choice $L = \lceil n \log 4 \rceil + 1$. Analogously, for $k > 3$ we inductively obtain from Kraft's inequality $\sum_i \binom{n}{i} 2^{\lceil i \cdot \log(k) \rceil + k - 3} \leq 2^L$, which is fulfilled for $L = \lceil n \cdot \log(k+1) \rceil + k - 2$.

For the function s^n the first person can send all the n bits of her input x_1. In knowledge of this the other $k-1$ persons have to determine for each component either the function t or $1-t$. Hence, their task is to evaluate a function equivalent to set intersection t^n on $k-1$ arguments, which can be done with $\lceil n \cdot \log k \rceil + k - 3$ bits of communication by the previous considerations. However, there is a gap to the lower bound, since a maximal independent set only has size $k + 2$ - the

two 1's (for $x_1 = x_2 = \cdots = x_k = 0$ or 1, respectively) plus the k 0's adjacent (at Hamming distance 1 to the all-one or all-zero vector) to one of these 1's.

The same protocol as for s^n can be used for u^n in the case of $k = 3$ inputs. Again after Person 1 has transmitted all the bits of its input in each component the function t or $1 - t$ must be computed, which can be done with $\lceil n \cdot \log 3 \rceil$ bits of communication. Here, the situation is better than for the function s, since we can find an independent set of size 6 in the function tensor of the basic function u: the three 1's $u(1,1,0) = u(1,0,1) = u(0,1,1) = 1$ and the three 0's $u(0,0,1) = u(0,1,0) = u(1,0,0) = 0$ must be contained in different monochromatic rectangles, such that $C(u^n) \geq \lceil n \log Ind(u) \rceil = \lceil n \cdot \log 6 \rceil$, which is exactly the complexity of the protocol described above.

Remarks

1) Unfortunately, for $k > 3$, the function u^n is not so nicely analyzable.

2) The lower bound for the function s^n is not so easy to improve as already the case $k = 3$ demonstrates. Here $\lceil n \log 5 \rceil \leq C(f) \leq \lceil n \log 6 \rceil$. However, there is a decomposition of the function tensor of s into just 5 monochromatic rectangles: $\{0\} \times \{0\} \times \{0\}$ and $\{1\} \times \{1\} \times \{1\}$ for the two 1s and $\{0\} \times \{1\} \times \{0,1\}$, $\{1\} \times \{0,1\} \times \{0\}$ as well as $\{0,1\} \times \{0\} \times \{1\}$ for the 0s.

4 Largest Monochromatic Rectangles for Multiparty Sum–Type Functions and a Generalization of Ahlswede's 4-Word Property

In order to be able to inductively determine the size of monochromatic rectangles Ahlswede and his coauthors[2], [6] introduced the weaker 4-word property It is no longer required that the function is constant on the rectangle $A \times B$ but that for all $a, a' \in A$, $b, b' \in B$

$$f(a,b) - f(a',b) - f(a,b') + f(a',b') = 0$$

This 4-word property behaves multiplicatively for sum-type functions in the sense that if the 4-word property holds on a rectangle $A \times B$ for the basic function f then it also holds on the rectangle $A^n \times B^n$ for the sum–type function f_n. Indeed, if $M(f,R,n)$ is the size of the largest rectangle with the 4-word property in f_n it can be shown that $M(f,R,n) = M(f,R,1)^n$. This often allows to determine exactly the size of the largest monochromatic rectangle for sum–type functions and hence bound the communication complexity from below.

In my PhD thesis [17] following a question posed by Rudolf Ahlswede an extension of the 4-word property to functions in more than 2 arguments was derived. Actually, this is just the 4-word property applied to the two-dimensional projections of higher dimensional rectangles. For instance, for a basic function $f : X_1 \times X_2 \times X_3 \times X_4 \to R$ in 4 arguments this yields an 8-word property, namely:

A rectangle (A, B, C, D) with $A \subset X_1$, $B \subset X_2$, $C \subset X_3$, $D \subset X_4$ fulfills the 8-word property if for all $a, a' \in A$, $b, b' \in B$, $c, c' \in C$ and $d, d' \in D$ it holds

$$f(a, b, c_*, d_*) - f(a', b, c_*, d_*) - f(a, b', c_*, d_*) + f(a', b', c_*, d_*) = 0$$

for $c_* \in \{c, c'\}$ and $d_* \in \{d, d'\}$ and

$$f(a_*, b_*, c, d) - f(a_*, b_*, c', d) - f(a_*, b_*, c, d') + f(a_*, b_*, c', d') = 0$$

for $a_* \in \{a, a'\}$ and $b_* \in \{b, b'\}$.

This can be straightforwardly generalized to a 2t-word property for functions in an even number t of arguments. The proof follows the lines of the one in [6]. Again, the 2t-word property is multiplicative in the above sense that $M(f, R, n) = M(f, R, 1)^n$ for the sum-type function f_n.

Unfortunately, these rectangles usually become too large in order to prove asymptotically tight lower bounds for the communication complexity of sum–type functions in more than two arguments. However, for some very natural functions the size of the largest monochromatic rectangles can be determined. Let discuss sum-type functions with the basic boolean functions $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ from the previous section.

1) $r(x, y, z, w) = x + y + z + w \bmod 2$. Then for $n = 2m$ r_n takes the constant value m on the rectangle $A \times B \times C \times D$ with $A = \{00, 11\}^m$, $B = C = D = \{01, 10\}^m$. Hence, the size of the largest monochromatic rectangle of the sum-type function r_n is at least 2^{2n}. On the other hand, obviously $\{01, \} \times \{0\} \times \{0\} \times \{0\}$ is a maximal 8-word set (of size 2) for the basic function r, such that the maximal 8-word set for r_n can have size at most 2^{2n}. Hence the above configuration yields the largest monochromatic rectangle.

2) $s(x, y, z, w) = \begin{cases} 1 \text{ , } x = y = z = w \\ 0 \text{ ,else} \end{cases}$. Here s_n takes the constant value 0 on the rectangle $\{0\}^n \times \{1\}^n \times \{0,1\}^n \times \{0,1\}^n$. This yields a monochromatic rectangle of size 4^n. On the other hand, with the largest 8-word set $\{01, \} \times \{0\} \times \{0,1\} \times \{0\}$ for the basic function s it can be shown that there is no larger monochromatic rectangle.

It would be interesting to find an analogue for the 4-word property also for sum–type functions with an odd number of arguments. For instance we conjecture the monochromatic rectangles in the function matrices of the sum–type function f_n in the following three examples for basic functions $f : \{0,1\} \times \{0,1\} \times \{0,1\} \to \{0,1\}$ in three arguments to be optimal, but there is no suitable lower bound, so far.

1') $r(x, y, z) = x + y + z \bmod 2$. For $n = 2m$ the sum–type function r_n takes the constant value m on the rectangle $\{01, 10\}^m \times \{01, 10\}^m \times \{01, 10\}^m$, hence the largest monochromatic rectangle has size at least $2^{\frac{3}{2}n}$.

2') $s(x, y, z) = \begin{cases} 1 \text{ , } x = y = z \\ 0 \text{ ,else} \end{cases}$.

On $\{0\}^n \times \{1\}^n \times \{0,1\}^n$ s_n takes the constant value 0. Hence, the size of the largest monochromatic rectangle in the function matrix of s_n is at least 2^n. Another configuration achieving this bound with constant value m for $n = 4m$ is $\{0000, 1111\}^m \times \{0011, 1100\}^m \times \{0101, 0110, 1001, 1010\}^m$.

3') $u(x, y, z) = 1$ iff at least two of the arguments are 1 (and 0 else). Again for $n = 2m$ u_n takes the constant value m on the rectangle $\{01, 10\}^m \times \{01, 10\}^m \times \{01, 10\}^m$, which means that the size of the largest monochromatic rectangle is at least $2^{\frac{3}{2}n}$.

5 Communication Complexity of Pairwise Comparison

Let there be k persons $P_1, P_2, \ldots, P_k$ each holding a binary string $x_i \in \{0,1\}^n$ $(i = 1, \ldots, k)$. Their task is to pairwisely compare their strings in the "number in hand" model with a minimum amount of data exchange. So we have to determine the communication complexity $C(f)$ of the function $f : \{0,1\}^n \times \ldots \{0,1\}^n \to \{0,1\}^{\binom{k}{2}}$ where $f(x_1, \ldots x_k) = (f_{(i,j)}(x_i, x_j))_{i<j\in\{1,\ldots,k\}}$ with $f_{(i,j)}(x_i, x_j) = 1$ iff $x_i = x_j$ (and 0 iff $x_i \neq x_j$).

A lower bound is obviously $C(f) \geq \lfloor \frac{k}{2} \rfloor n + 1$, since f automatically compares the two strings obtained by concatenating the first $\lfloor \frac{k}{2} \rfloor$ and the next $\lfloor \frac{k}{2} \rfloor$ inputs in the two - party comunication model. Here the trivial protocol is optimal for the equality function.

Theorem 4. $\lim_{n\to\infty} \frac{1}{n} C(f) = \lfloor \frac{k}{2} \rfloor$

Proof: With the following "divide and conquer" protocol it can be shown that this lower bound is asympotically optimal. This is somehow surprising, since for odd k one might expect some additional communication. This is, however, negligible - only a $\sqrt{n}$ term:

For $k = 2$ Person 1 transmits her complete string and Person 2 returns the result.

For $k \geq 3$ Person 1 transmits the first $\lceil \sqrt{n} \rceil$ bits of her input. The other persons then send a 0 if their inputs coincide on these $\lceil \sqrt{n} \rceil$ bits or a 1, respectively, if this is not the case. If all other persons have sent a 0, then Person 1 transmits the next $\lceil \sqrt{n} \rceil$ bits of her input and the other persons respond in the same way. After Person 1 has transmitted, say, $t\lceil \sqrt{n} \rceil$ bits for the first time some of the other persons, say $k - i$ of them, will answer with a 1. Their $k - i$ inputs then have to be compared on $n - (t-1)\lceil \sqrt{n} \rceil$ bits, the other i inputs have to be compared on $n - t\lceil \sqrt{n} \rceil$ bits.

Let $M(k, n)$ be the number of bits transmitted during this protocol in the worst case. Then

$$M(k,n) \leq \begin{cases} \frac{k}{2}n + a_k & , k \text{ even} \\ \frac{k-1}{2}n + b_k\sqrt{n} + c_k & , k \text{ odd} \end{cases}$$

for certain numbers a_k, b_k, c_k only depending on the number of partys k and not on n. With this, the asymptotic statement of the theorem is immediate.

The above formula for $M(k,n)$ can be proven by induction. Obviously $M(2,n) = n+1$. Further, $M(3,n) \leq n+3\lceil\sqrt{n}\rceil+1$. To see this, assume that after Person 1 has sent $t\lceil\sqrt{n}\rceil$ bits for the first time the other Persons do not reply with 0 both. So at least one of their inputs does not coincide with x_1 on the last $\lceil\sqrt{n}\rceil$ bits transmitted. If only one person, Person 3 say, sent a 1, it is clear that x_3 is different from x_1 and x_2, which then have to be compared on the remaining $t - \lceil\sqrt{n}\rceil$ bits.

If both persons replied with 1, then x_1 is different from x_2 and x_3, which then still have to be compared on $n-(t-1)\lceil\sqrt{n}\rceil$ bits. This is obviously the worst case and here still $M(2, n-(t-1)\lceil\sqrt{n}\rceil+1)$ further bits must be exchanged to obtain the result. Hence

$M(3,n) \leq t\lceil\sqrt{n}\rceil+2t+n-(t-1)\lceil\sqrt{n}\rceil+1 = n+\lceil\sqrt{n}\rceil+2t+1 \leq n+3\lceil\sqrt{n}\rceil+1$.

For $k \geq 4$ the above protocol yields the recursion

$$M(k,n) \leq \max_t \max_{i=1,\ldots,k} t\lceil\sqrt{n}\rceil + (k-1)t + M(i, n-t\lceil\sqrt{n}\rceil) + M(k-i, n-(t-1)\lceil\sqrt{n}\rceil).$$

from which the numbers a_k, b_k, and c_k can be recursively calculated with several case investigations (k, i even or odd).

References

1. Ahlswede, R.: On code pairs with specified Hamming distances. Colloquia Math. Soc. J. Bolyai 52, 9–47 (1988)
2. Ahlswede, R., Mörs, M.: Inequalities for code pairs. European J. Combinatorics 9, 175–188 (1988)
3. Ahlswede, R., Cai, N.: On communication complexity of vector-valued functions. IEEE Trans. Inf. Theory 40(6), 2062–2067 (1994)
4. Ahlswede, R., Cai, N.: 2-Way communication complexity of sum-type functions for one processor to be informed. Probl. Inf. Transmission 30(1), 1–10 (1994)
5. Ahlswede, R., Cai, N., Tamm, U.: Communication complexity in lattices. Appl. Math. Letters 6(6), 53–58 (1993)
6. Ahlswede, R., Cai, N., Zhang, Z.: A general 4–word–inequality with consequences for 2–Way communication complexity. Advances in Applied Mathematics 10, 75–94 (1989)
7. Ahlswede, R., Zhang, Z.: Code pairs with specified parity of the Hamming distances. Discr. Math. 188, 1–11 (1998)
8. Ahlswede, R., El Gamal, A., Pang, K.F.: A two–family extremal problem in Hamming space. Discr. Math. 49, 1–5 (1984)
9. Alon, N., Matias, M., Szegedy, M.: The space complexity of approximating the frequency moments. J. Comput. Syst. Sci. 58(1), 137–147 (1999)
10. Babai, L., Frankl, P., Simon, J.: Complexity classes in communication complexity theory. In: Proc. IEEE FOCS, pp. 337–347 (1986)
11. Chen, L., Chitambar, E., Duan, R., Ji, Z., Winter, A.: Tensor rank and stochastic entaglement catalysis for multipartite pure states. Physical Review Letters 105 (2010)
12. Draisma, J., Kushilevitz, E., Weinreb, E.: Partition arguments in multiparty communication complexity. Theoretical Computer Science 412, 2611–2622 (2011)
13. Feder, T., Kushilevitz, E., Naor, M., Nisan, N.: Amortized communication complexity. SIAM J. Comp. 24(4), 736–750 (1995)

14. Hromkovic, J.: Communication Complexity and Parallel Computing. Springer (1997)
15. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press (1997)
16. Nisan, N., Wigderson, A.: On rank vs communication complexity. Combinatorica 15(4), 557–566 (1995)
17. Tamm, U.: Communication complexity of sum-Type functions. PhD thesis, Bielefeld (1991)
18. Tamm, U.: Still another rank determination of set intersection matrices with an application in communication complexity. Appl. Math. Letters 7, 39–44 (1994)
19. Tamm, U.: Communication complexity of sum - type functions invariant under translation. Inform. and Computation 116(2), 162–173 (1995)
20. Tamm, U.: Deterministic communication complexity of set intersection. Discr. Appl. Math. 61, 271–283 (1995)
21. Tamm, U.: Communication complexity of functions on direct sums. In: Althöfer, I., Cai, N., Dueck, G., Khachatrian, L., Pinsker, M., Sárközy, A., Wegener, I., Zhang, Z. (eds.) Numbers, Information and Complexity, pp. 589–602. Kluwer (2000)
22. Yao, A.C.: Some complexity questions related to distributive computing. In: Proc. ACM STOC, pp. 209–213 (1979)