Non-deterministic Communication Complexity with Few Witnesses

Mauricio Karchmer*
Dept. of Mathematics, M.I.T.
Cambridge, MA 02139 USA

Ilan Newman†
Dept. of Mathematics and Computer Science, Haifa University
Haifa, 31905, Israel

Mike Saks‡
Dept. of Mathematics, Rutgers University
New Brunswick, NJ 08903 USA

Avi Wigderson
Dept. of Computer Science, Hebrew University
Givat-Ram, Jerusalem, Israel

*Supported by NSF grant # CCR90-10533

†This work done while the author was at The Royal Institute of Technology, Stockholm

‡On leave from Dept. of Comp. Sci and Eng, UCSD. Supported in part by NSF Grant # CCR89-11388 and AFOSR Grants 89-0512 and 90-0008.

Proposed running title: Nondet communication comp.

Mailing address:

Ilan Newman

Dept. of Mathematics and Computer Science

University of Haifa

Haifa, 31905, Israel

Abstract

We study non-deterministic communication protocols in which no input has too many witnesses. Define $n_k(f)$ to be the minimum complexity of a non-deterministic protocol for the function f in which each input has at most k witnesses. We present two different lower bounds for $n_k(f)$. Our first result shows that $n_k(f)$ is bounded below by $\Omega(\sqrt{c(f)}/k)$ where $c(f)$ is the deterministic complexity. Our second results bounds $n_k(f)$ by $\log(rk(M_f))/k - 1$ where $rk(M_f)$ is the rank of the representing matrix of f. As a consequence, it follows that the communication complexity analogue of the Turing-complexity class $FewP$ is equal to the analogue of the class P.

1 Introduction

In the two-party communication complexity model, two parties compute a function that depends on both of their (initially private) inputs. Roughly speaking, the deterministic communication complexity of a function $f(x, y)$ is the minimum number of bits that must be exchanged in order for one of the parties to be able to deduce the value of f. Many variants and applications of the communication complexity model were studied (see e.g. the survey paper of Lovasz [11]) since it was introduced by Yao [20] in 1979.

In 1986 Babai, Frankl and Simon [4] began a systematic study of complexity classes associated with this model, by defining a communication protocol to be "tractable" if the number of bits that must be exchanged is at most polylogarithmic in the input length. For example, the classes $P^{cc}, NP^{cc}, BPP^{cc}$ correspond respectively to the languages recognized by tractable deterministic, nondeterministic and bounded-error probabilistic protocols. (The reader is referred to [4] for the exact definitions.) The natural complexity theoretic problems, such as $P \stackrel{?}{=} NP$ and $PH \stackrel{?}{=} PSPACE$ immediately pose the analogous questions in the communication model. They showed that existing results can be concisely phrased in this language, e.g $P^{cc} \neq NP^{cc}$, $P^{cc} = NP^{cc} \cap coNP^{cc}$. They also proved some new results, e.g. $BPP^{cc} \subset PH^{cc}$ but BPP^{cc} and NP^{cc} are unrelated.

In this paper we investigate the communication complexity analogues of the Turing-complexity classes UP and $FewP$. The classes UP and $FewP$ are subclasses of NP consisting of those languages recognizable by non-deterministic Turing Machines that satisfy a restriction: for every string in the language, the number of accepting computations (witness strings) is very small (exactly one for UP and polynomially bounded for FewP). The class UP was introduced by Valiant [16] in 1976 and was intensively studied. It is not known whether UP is strictly larger than P, although this is believed to be the case, (for related results see [18]). In contrast, Yannakakis [19] proved that in communication complexity this restriction is as severe as can be, namely, that $P^{cc} = UP^{cc}$. The somewhat less restrictive class $FewP$ was introduced by Allender [1] and studied in [5, 6]. As it contains UP, it is at least as hard. A corollary to our main result (Corollary 1) is a strengthening of the result of Yannakakis above, namely we prove $P^{cc} = FewP^{cc}$ (where in $FewP^{cc}$ we allow each input to have polylogarithmic number of witnesses, as well as total polylog number of communication bits). Stated differently, we show that $FewP^{cc}$ is as easy as UP^{cc}. A similar statement could be quite plausible for the Turing machine analogous classes, however, the relative power of $FewP$ and UP is not known, (for related results see [5]).

For each integer k, we define $n_k(f)$ to be the minimum complexity of a non-deterministic communication protocol that computes f, subject to the condition that every input has at most k witnesses. Our two main results are (nearly tight) lower bounds on $n_k(f)$. The first (Theorem 1) says that $n_k(f)$ is at least of the order of the square root of deterministic communication complexity divided by k, which implies the above-mentioned corollary. The second result (Theorem 2) says that $n_k(f)$ is at least of the order of the log of the rank of the representing matrix of f, divided by k. As this rank is a well-known lower bound [10] on the deterministic communication complexity and is, in fact, equal for "most" functions, the second bound is "usually" better than the first, but in general, we do not know if it always implies the first.

Our result also yields a generalization of a well known graph theoretic result concerning the covering of the edges of the complete graph K_n on n vertices by complete bipartite graphs. It is easy to see that $\lceil \log n \rceil$ bipartite graphs are necessary and sufficient to cover K_n. On the other hand, if it is required that each edge belong to exactly one of the bipartite graphs (i.e. the cover is a partition of the edges), then it has been proven that $n-1$ graphs are necessary (and trivially sufficient). The lower bound was first proved by Graham and Pollak [7, 8] (see also [2],K), using a linear-algebra (rank) argument. As a consequence of our results we obtain a tradeoff between these two extremes: any cover of the complete graph by bipartite graphs in which each edge is covered at most k times requires at least $n^{1/k}$ graphs, and there is such a cover that uses $kn^{1/k}$ graphs.

This paper is organized as follows. In section 2, we provide some background and preliminary results. The reader familiar with past work on communication complexity can skip much of this section. In section 3, we state our main results, which are then proved in section 4.

2 Preliminaries

2.1 Matrices

All matrices we consider have complex entries. We define an equivalence relation $\equiv_0$ on the set of matrices of a given size with $A \equiv_0 B$ if A and B have the same set of 0 entries. The rank (over the complex numbers) of a matrix A is denoted $rk(A)$. The *triangular rank* of A, $trk(A)$, is the size of the largest nonsingular lower triangular submatrix of A. The following properties are easily verified:

Proposition 1

1. For every matrix A, $trk(A) \leq rk(A)$.
2. If $A \equiv_0 B$ then $trk(A) = trk(B)$.

A Boolean matrix of rank 1 is called a *rectangle.* A *rectangle cover* of a Boolean matrix M is a set $\mathcal{R} = \{R_i\}_{i\in[t]}$ of rectangles satisfying $M \equiv_0 R$, where $R = \sum_{i=1}^{t} R_i$, i.e., every '1' entry of M is covered by at least one of the R_i's, while every '0' entry of M is '0' in all the R_i's. The *degree* of an entry (x, y) of M with respect to $\mathcal{R}$ is the number of $R_i \in \mathcal{R}$ such that $R_i(x,y) = 1$. A rectangle cover is a *k-cover* if all of the degrees are at most k, i.e., $R \leq kJ$ (where J denotes the all 1's matrix). Define $\kappa(M)$ (resp., $\kappa_k(M)$) as the minimum cardinality of a cover (resp., k-cover) of M.

Proposition 2 For any Boolean matrix M:

1. $\kappa_1(M) \geq rk(M)$,
2. $\kappa(M) \geq trk(M)$,
3. $\kappa_1(M) \geq \kappa_2(M) \geq \ldots \geq \kappa_i(M) \geq \ldots \geq \kappa(M)$.

Proof : For the first part: if $\{R_i\}_{i\in[t]}$ is a 1-cover of M then $M = \sum_{i=1}^{t} R_i$ and so by the subadditivity of matrix rank, $rk(M) \leq t$. For the second part: in any rectangle cover of M, each diagonal entry of a nonsingular lower triangular matrix must belong to a different rectangle. The third part is immediate from the definition. ∎

Let X and Y be finite sets. For $S \subseteq X \times Y$, the *characteristic matrix* A of S is the Boolean matrix with rows indexed by X and columns by Y with $A(x,y) = 1$ if and only if $(x,y) \in S$. For a function f with domain $X \times Y$, its *representing matrix* M_f has rows indexed by X and columns indexed by Y and $M_f(x,y) = f(x,y)$.

2.2 Communication Protocols

A deterministic two-party Boolean-output communication protocol P on $X \times Y$ is described by a rooted binary tree as follows:(i) the two children of each interior node v are distinguished as $c_0(v)$ and $c_1(v)$, (ii) each node is classified as either type X or type Y, (iii) each node v of type $Z \in \{X, Y\}$ is labeled by a function $b_v : Z \mapsto \{0,1\}$. (When we do not need to specify the type of the node we write $b_v(x, y)$ although it is understood that b_v depends only on one of its arguments). The set of leaves of the tree is denoted L_P.

Such a protocol corresponds to an interactive computation between two parties in which party X has input $x \in X$ and party Y has input $y \in Y$. Each node of the tree is a computation state. Starting from the root, the parties exchange bits, where at a node of type Z, party Z sends $b_v(x, y)$ thereby specifying one of the children of v as the new computation state. In this way, the parties arrive at a leaf, $l = l_P(x, y)$, and the output of the protocol is the value $b_l(x, y)$. The function $f_P : X \times Y \mapsto \{0, 1\}$ specified in this way is the *function computed by* P.

A non-deterministic protocol P is defined similarly, except that the domain of each interior node function b_v is $\{0, 1, *\}$ instead of $\{0, 1\}$. At node v, if b_v evaluates to $*$ on input (x, y) then the computation moves non-deterministically to either child of v. Let $L_P(x, y)$ denote the set of leaves which can be reached from the root on input (x, y). The function computed for (x, y) is 1 if and only if $f_l(x, y) = 1$ for one of the leaves $l \in L_P(x, y)$.

The complexity of a (deterministic or non-deterministic) protocol is the maximum depth of a leaf in the tree. We use $c(f)$ (resp., $n(f)$), respectively, to denote the the minimum complexity of any deterministic (resp. non-deterministic) protocol that computes f.

We define the *witness multiplicity* of input (x, y) in the non-deterministic protocol P to be the number of accepting paths for the input, i.e., the number of leaves $l \in L_P(x, y)$ such that $b_l(x, y) = 1$. For each positive integer k, we define $n_k(f)$ to be the minimum complexity of a non-deterministic protocol for f for which the witness multiplicity of every input is at most k.

2.3 Communication Complexity Classes

As suggested in [14] and described in more detail in [4], these complexity measures for functions extend naturally to a complexity measure for languages. Let L be a language in $\{0,1\}^*$ such that each string is of even length. We associate to L a family of functions $\{L_n | n \geq 1\}$ where $L_n : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ is defined by $L_n(x,y) = 1$ if and only if $xy \in L$. A communication protocol for L is then a (non-uniform) sequence of protocols, one for each L_n. A protocol is said to have polynomial time communication complexity if there is a polynomial $p()$ such that, for each $n \geq 1$, the complexity of the protocol for L_n is at most $p(\log n)$. The complexity classes P^{cc} and NP^{cc} are defined, respectively to be the set of languages that have polynomial time deterministic and non-deterministic protocols. In a similar way, [4] define analogues for many other standard complexity classes.

As described in the introduction, we are concerned with analogues of the Turing classes UP and $FewP$. The class UP^{cc} can be formally defined as the class of languages L such that there is a polynomial $p()$ such that for all $n \geq 1$, $n_1(L_n) \leq p(\log n)$. Similarly, a language L is in $FewP^{cc}$ if there are polynomials $q(), p()$, such that for each $n \geq 1$, $n_{q(\log n)}(L_n) \leq p(\log n)$.

We now review some needed background results about communication complexity. For any protocol P, and each leaf v we define the set $S_v \subseteq X \times Y$ to be the set of inputs (x,y) such that $v \in L_P(x,y)$ (v is a possible leaf for the input (x,y)) and $b_v(x,y) = 1$ (input (x,y) is accepted at v). The characteristic matrix of S_v is denoted R_v.

An elementary but important observation due to [20] and [3] is that S_v is always a product set; $S_v = X' \times Y'$ for some $X' \subseteq X$, $Y' \subseteq Y$ and that each of the matrices R_v is a rectangle. Moreover, in any protocol for f, the set $\{R_v : v \in L_P\}$ is a rectangle cover of the matrix M_f. It is also easy to see that the set is a k-cover if and only if each input (x,y) has witness multiplicity at most k with respect to the protocol P. This implies that any non-deterministic protocol for f must have at least $\kappa(M_f)$ leaves, and any such protocol with witness multiplicity at most k must have at least $\kappa_k(M_f)$ leaves. Since the complexity of the protocol is bounded below by the base two logarithm of the number of leaves, we have $n(f) \geq \log \kappa(M_f)$ and for each $k \geq 1$, $n_k(f) \geq \log \kappa_k(M_f)$. In fact, up to round-off these bounds are tight.

Proposition 3 For any function f, $n(f) = \lceil \log \kappa(M_f) \rceil$. Similarly, for $k \geq 1$, $n_k(f) = \lceil \log \kappa_k(M_f) \rceil$.

Proof : The first part of this proposition is from [3]. It is enough to prove that $n(f)$ and $n_k(f)$ are bounded above by the associated quantities. For this it suffices to show that for every rectangle cover (resp., k-cover) by t rectangles there is a non-deterministic protocol (resp., protocol with witness multiplicity at most k) of complexity $\lceil \log t \rceil$. Given any cover $\mathcal{R}$ of M_f of size t, we may define a non-deterministic protocol as follows: encode each of the rectangles by a binary string of length $\lceil \log t \rceil$. The first party (non-deterministically) sends the name of some rectangle in which row x is nonzero. The second party then evaluates the function to 1 if column y in this rectangle is nonzero. Clearly, there is some computation path which evaluates to 1 if and only if there is some rectangle belonging to $\mathcal{R}$ which contains entry (x, y), i.e., $f(x, y) = 1$. The complexity of the computation is $\lceil \log t \rceil$ and the witness multiplicity for a given input (x, y) is equal to the degree of (x, y) in the cover. ■

For deterministic complexity, no such exact characterization is known. It is easy to see that $2^{n(f)} \geq c(f) \geq n(f)$, and simple known examples show that each inequality can be made an equality: the lower bound is tight for the function $ID(x, y) = 1$ iff $x = y$ and the upper bound is tight for the complementary function $\overline{ID}$. If one has simultaneous upper bounds on the nondeterministic complexity of a function and its complement, one gets much better upper bounds on the deterministic complexity. This was shown by Aho, Ullman and Yannakakis [3]. Let $\bar{f}$ denote the complement of f.

Proposition 4 [3] $c(f) = O(n(f)n(\bar{f}))$. In particular, $P^{cc} = NP^{cc} \cap coNP^{cc}$.

This result was improved by Lovasz and Saks [12], who showed that $n(f)$ in the upper bound above can be replaced by the quantity $\log(trk(M_f))$, which is smaller than n_f by Propositions 2 and 3:

Proposition 5 [12] For every function f, $c(f) = O(n(\bar{f})\log(trk(M_f)))$.

The lower bound $c(f) \geq n(f)$ can be improved by $c(f) \geq n_1(f)$, since every deterministic protocol is a non-deterministic protocol of witness multiplicity 1. Together with Proposition 3 and the first part of Proposition 2, this yields the following bounds:

Proposition 6 [10] For any function f,

$$c(f) \geq n_1(f) = \lceil \log \kappa_1(f) \rceil \geq \log rk(M_f).$$

The inequality $c(f) \geq rk(M_f)$ has been a major tool for proving explicit lower bounds for specific functions, but it is not known how good it is in general. The largest known gap between these two quantities is a constant factor [15], but it may well be exponential (see [11]).

On the other hand, Yannakakis [19] showed that the gap between $c(f)$ and $n_1(f)$ (which he called the unambiguous non-deterministic complexity) can not be more than polynomial.

Proposition 7 [19] $c_f \leq n_1(f)^2$.
In particular, $P^{cc} = UP^{cc}$.

It is not known if this bound is best possible.

3 Main Results

The main results of this paper are two lower bounds on $n_k(f)$. The first generalizes Yannakakis' bound (Proposition 7) for $n_1(f)$.

Theorem 1 For any function $f : X \times Y \mapsto \{0,1\}$ and integer $k \geq 1$:

$$n_k(f) = \Omega(\frac{\sqrt{c(f)}}{k}),$$

or equivalently

$$c(f) = O((kn_k(f))^2)$$

.

Applying the definition of the complexity class $FewP^{cc}$ we obtain:

Corollary 1 $P^{cc} = FewP^{cc}$.

The second result generalizes the bound $n_1(f) \geq \log rk(M_f)$.

Theorem 2 For any non-zero function $f : X \times Y \mapsto \{0,1\}$ and integer $k \geq 1$,

$$n_k(f) \geq \frac{\log rk(M_f)}{k} - 1$$

.

Theorem 2 is an arithmetic consequence of the fact (Proposition 3) that $n_k(f) = \lceil \log \kappa_k(M_f) \rceil$ and the following lemma.

Lemma 1 For any Boolean matrix A and positive integer k,

$$(1 + \kappa_k(A))^k \geq rk(A).$$

We defer the proof of lemma 1 to the next section.

For all presently known examples, the second result is stronger than the first, but as remarked after Proposition 6, it is possible that the second result is much weaker for some functions. Finally, we show that the last lower bound on $n_k(f)$ is nearly best possible. For this, define the identity function $ID : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ by $ID(x,y) = 1$ iff $x = y$, and $\overline{ID}$ its complement.

Theorem 3 $n(\overline{ID}) = \log rk(\overline{ID}) = n$, and for every k, $n_k(\overline{ID}) \leq (n/k) + \log k$.

Finally, we note the following graph theoretic corollary of Lemma 1 and Theorem 3 , generalizing a theorem of Graham and Pollak([7, 8]):

Corollary 2 Let G be a graph and k a positive integer. Let A_G be the adjacency matrix of G. Any set of bipartite subgraphs of G whose union is G and which cover each edge of G at most k times must have at least $rk(A)^{1/k} - 1$ members. In particular, for the complete graph K_n, any such set must have at least $n^{1/k} - 1$ members and there exists such a set consisting of $kn^{1/k}$ subgraphs.

4 Proofs

4.1 Proof of Theorem 1:

This proof is very similar to the one in [13], which is a special case of this one. Let f be given, and an optimal k-cover $\{R_i\}_{i\in[t]}$ for M_f with $R = \sum R_i$. Note that $\log t \leq n_k(f)$ (proposition 3). The proof is by induction on k, with the base case $k = 1$ given by Proposition 7. Assume $k > 1$.

Define a function $g : X \times Y \mapsto \{0, 1\}$ by $g(x, y) = 0 \Leftrightarrow R_{xy} = k$. It is easy to see that $n(\bar{g}) \leq k \cdot n_k(f)$, as the (unique) cover is given by the intersections of exactly k rectangles from the original cover. Also observe that M_g and $kJ - R$ (where J denotes the all 1's matrix) have the same set of zero entries. Thus by fact 1 we have $trk(M_g) = trk(kJ - R) \leq rk(kJ - R) \leq 1 + t$. From Proposition 5 we deduce $c(g) = O(kn_k(f)^2)$.

This motivates the following deterministic protocol for $f(x, y)$. First evaluate g using $O(kn_k(f)^2)$ bits as above. The leaves of this protocol partition the input into product sets $\{S_\ell = X_\ell \times Y_\ell\}$, one for each leaf ℓ. In a leaf ℓ in which the answer is $g = 0$, we halt and output $f = 1$. Let ℓ be leaf in which the answer is $g = 1$. By the definition of g, the entries of R indexed by $X_\ell \times Y_\ell$ are bounded by $k-1$. Let $R(\ell)$ be the set of matrices $\{R_i[X_\ell \times Y_\ell]\}_{i\in[t]}$, where $R_i[X_\ell \times Y_\ell]$ is the minor of R_i defined by the row set X_ℓ and column set Y_ℓ. It is easy to see that $R(\ell)$ is a $k-1$ cover of the function f_ℓ, which is the restriction of f to the set $S_\ell = X_\ell \times Y_\ell$. By the inductive assumption it can be solved using $O((k-1)^2 n_k(f)^2)$ bits. $\square$

4.2 Proof of Theorem 2:

As noted above, it is enough to prove Lemma 1. Let M be a matrix , $t = \kappa_k(M)$ and $\{R_i\}_{i\in[t]}$ be a minimum k-cover of M. Let $R = \sum_{i=1}^{t} R_i$. As $rk(R) \leq t$, it suffices to prove $(1 + rk(R))^k \geq rk(M_f)$.

By definition of k-cover, $R \equiv_0 M$. Let $\mathcal{M}_k$ be the set of all matrices A, whose nonzero entries take on at most k distinct values. Note that R belongs to $\mathcal{M}_k$. Thus it suffices to prove the following stronger result, which may be of independent interest.

Theorem 4 Let A, B be matrices (over $\mathcal{C}$) with $A \equiv_0 B$, $A \in \mathcal{M}_k$ and B Boolean. Then $(1 + rk(A))^k \geq rk(B)$.

Remarks:

1. The bound above is nearly best possible, i.e, the example in the proof of Theorem 3 exhibits a Boolean matrix B with $rk(B) = 2^n$ which has a k-cover by $k2^{n/k}$ rectangles. Let A be the sum of these rectangles, A, B meet the conditions of the above Theorem and $rk(B) \geq (rk(A)/k)^k$.

2. We note here that this theorem has a flavour of a rigidity type statement, (see [17]). i.e. One can start with a Boolean matrix B and get a matrix A by changing as many

non zero entries as long as the entries take no more than k values, then the rank of A cannot drop too much.

Proof : The theorem follows from the two lemmata below, with the following definition. For two matrices X, Y of the same dimensions $(r \times s)$, let $Z = X \circ Y$ be the $(r \times s)$ matrix for which $Z_{ij} = X_{ij} Y_{ij}$.

Lemma 2 Rank is sub-multiplicative under $\circ$, i.e.

$$rk(X_1 \circ X_2 \circ \cdots \circ X_s) \leq \prod_{i=1}^{s} rk(X_i).$$

Lemma 3 Under the conditions of the theorem, there exist matrices $A_1, A_2, \ldots, A_k$ such that $rk(A_i) \leq 1 + rk(A)$ for all $i \in [k]$, and $\lambda B = A_1 \circ A_2 \circ \cdots \circ A_k$ for some constant $\lambda \neq 0$.

Using $A_1, \ldots, A_k$ and λ as in lemma 3 we have $rk(B) = rk(\lambda B) = rk(A_1 \circ A_2 \circ \cdots \circ A_k) \leq (1 + rk(A))^k$. ■

Proof of Lemma 2: It clearly suffices to prove it for $s = 2$, as for larger s it follows by induction. So we want to prove that for any X, Y, $rk(X \circ Y) \leq rk(X) rk(Y)$. Note that if $rk(X) = rk(Y) = 1$, then $rk(X \circ Y) \leq 1$. Write $X = \sum_{i=1}^{rk(X)} X_i$, $Y = \sum_{j=1}^{rk(Y)} Y_j$ such that for all i, j $rk(X_i) = rk(Y_j) = 1$. Now using subadditivity of the rank function under matrix addition, we have

$$rk(X \circ Y) = rk\left(\left(\sum_{i=1}^{rk(X)} X_i\right) \circ \left(\sum_{j=1}^{rk(Y)} Y_i\right)\right) = rk\left(\sum_{i,j} X_i \circ Y_j\right) \leq$$
$$\leq \sum_{i,j} rk(X_i \circ Y_j) \leq rk(X) \times rk(Y)$$

■

Proof of Lemma 3: Let $\alpha_1, \alpha_2, \cdots, \alpha_k$ be the distinct non-zero elements of $\mathcal{C}$ appearing in A. We shall prove that there exist $x_1, x_2, \cdots, x_k \in \mathcal{C}$, with $x_1 = 0$, and $\lambda \in \mathcal{C}$ with $\lambda \neq 0$, such that

$$\text{for all } j \in [k] \ , \ \prod_{i=1}^{k}(x_i + \alpha_j) = \lambda \tag{1}$$

Note that this implies the lemma, as taking $A_i = x_i J + A$ satisfy $rk(A_i) \leq 1 + rk(A)$ and $\lambda B = A_1 \circ A_2 \circ \cdots \circ A_k$.

Let $\overline{x} = (x_1, x_2, \cdots, x_k)$, and for $\ell \in [k]$, let $S_\ell(\overline{x})$ denote the ℓ^{th} elementary symmetric function of $\overline{x}$, namely $S_\ell(\overline{x}) = \sum_{\substack{T \subseteq [k] \\ |T| = \ell}} \prod_{i \in T} x_i$. Then the system of equations (1) can be written in matrix form as

$$\begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{k-1} \\ \vdots & & & & \\ 1 & \alpha_k & \alpha_k^2 & \cdots & \alpha_k^{k-1} \end{pmatrix} \begin{pmatrix} S_k(\overline{x}) \\ S_{k-1}(\overline{x}) \\ \vdots \\ S_1(\overline{x}) \end{pmatrix} = \begin{pmatrix} \lambda - \alpha_1^k \\ \lambda - \alpha_2^k \\ \vdots \\ \lambda - \alpha_k^k \end{pmatrix}$$

Let V be the Vandermonde matrix generated by the α's, i.e. $V_{ij} = \alpha_i^{j-1}$; it is well known that V is invertible. Denote by $\overline{\alpha}^{(j)} = (\alpha_1^j, \alpha_2^j, \cdots, \alpha_k^j)$ for every $0 \leq j \leq \ell$.

Finally let $\overline{r}(\lambda)$ denote the right hand side of the above equation, and $\overline{w}(\lambda)$ denote the vector $V^{-1}\overline{r}(\lambda)$. Then for a particular choice of λ, we seek to find $\overline{x}$ so that for each $\ell \in [k]$, $S_\ell(\overline{x}) = w_{k+1-\ell}(\lambda)$. This is done by choosing the x_i to be the roots of the polynomial $p(z) = z^k + \sum_{\ell=0}^{k-1}(-1)^{k-\ell} w_{\ell+1}(\lambda) z^\ell$. To guarantee that one of these roots is zero, it suffices to select $\lambda \neq 0$ for which $w_1(\lambda) = 0$. Letting $\overline{u}$ denote the first row of V^{-1}, we have $w_1(\lambda) = \lambda \overline{u} \cdot \overline{1} - \overline{u} \cdot \overline{\alpha}^{(k)} = \lambda - \overline{u} \cdot \overline{\alpha}^{(k)}$ and so we set $\lambda = \overline{u} \cdot \overline{\alpha}^{(k)}$. We need only check that this is non-zero, which follows from the fact that $\overline{u}$ is non-zero and is orthogonal to the last

$k-1$ columns of V, and thus cannot be orthogonal to $\overline{\alpha}^{(k)}$, since these k vectors are linearly independent. ■

4.3 Proof of Theorem 3:

Assume for simplicity that k divides n (otherwise add dummy bits). We give a family of $k2^{n/k}$ rectangles which constitute a k-cover of $\overline{ID}$. Let $I_1, I_2, \cdots I_k$ be a partition of $[n]$ into blocks of n/k bits each. For every $j \in [k]$ and every string $\sigma \in \{0,1\}^{n/k}$ define the rectangle $R_{j,\sigma}$ as follows. An input (x, y) is in $R_{j,\sigma}$ iff x agrees with σ on I_j and y doesn't. It is easy to see that every input x, y with $x \neq y$ belongs to at most k rectangles, one for each choice of j. ■

References

[1] E.W. Allender, The complexity of sparse sets in P, *Proc. of the 1th Symp. on Structures in Complexity theory,* Springer Verlag Lecture notes in computer science, Vol 223(1986), 1–11.

[2] N. Alon, R.A. Brualdi, B.L. Shader, *Multicolored Forests in bipartite decompositions of graphs*, manuscript.

[3] A. Aho, J. Ullman, M. Yannakakis, On Notions of Information Transfer in VLSI Circuits, *Proc. of the 15th STOC,* (1983), 133–139.

[4] L. Babai, P. Frankel and J. Simon, Complexity Classes in Communication Complexity Theory, *Proc. of the 27th FOCS,* (1986), 337–347.

[5] R. Beigel, J. Gill, U. Hertrampf, Counting classes: thresholds, parity, mods, and fewness, *Proc. 7th STACS,* Springer Verlag Lecture Notes in Computer science, Vol 415(1990), 49–57.

[6] J. Cai, L.A. Hemachandra, On the power of parity, *Proc. of the 6th STACS,* Lecture Notes in computer science, Vol 349 (1989), 229–240, Springer Verlag.

[7] R.L. Graham, H.O. Pollak, On the addressing problem for loop switching, *Bell Syst. Tech. J. 50(1971),* 2495-2519.

[8] R.L. Graham, H.O. Pollak, On embedding graphs in squashed cubes, *Lecture Notes in Mathematics 303,* pp 99-110, Springer Verlag, 1973.

[9] D.J. Kleitman, A new proof of a theorem of Graham and Pollak, *Discrete Mathematics,* 49 (1984), 327-8.

[10] K. Mehlhorn and E. Schmidt, Las Vegas is Better than Determinism in VLSI and Distributed Computing, *Proc. of the 14th STOC (1982),* 330–337.

[11] L. Lovász, Communication Complexity A Survey, *Paths, Flows and VLSI Layout*, Korte, Lovász, Promel, Schrijver Eds., Springer-Verlag (1990), 235–266.

[12] L. Lovász and M. Saks, Lattices, Mobius Functions and Communication Complexity, *29th FOCS (1988)*, 81–90.

[13] I. Newman, On Read-Once Boolean functions, to appear in *Boolean function complexity selected papers from LMS Symp. Durham, July 1990*, Edited by M.S. Paterson, Cambridge U. Press.

[14] C.H. Papadimitriou and M. Sipser, Communication Complexity, *Proc. 14th ACM STOC (1982),* 330-337.

[15] A. A. Razborov, *The Gap between the Chromatic Number of a Graph and the Rank of its Adjacency Matrix is Superlinear*, Manuscript.

[16] L. G. Valiant, Relative Complexity of Checking and Evaluating, *Information Processing Letters 5,* (1976), 20–23.

[17] L. G. Valiant, Graph theoretic arguments in low-level complexity, *Technical Report CS 13-77,* University of Edinburgh (1977).

[18] L. G. Valiant and V. V. Vazirani, NP is as Easy as Detecting Unique Solutions, *Proc. of the 17th STOC,* (1985), 458–463.

[19] M. Yannakakis, Expressing Combinatorial Optimization Problems by Linear Programs, *Proc. 29th FOCS,* (1988), 223–228.

[20] A. C. -C. Yao, Some Complexity Questions Related to Distributive Computing, *Proc. of the 11th STOC,* (1979) 209–213.