The communication complexity of threshold gates

Noam Nisan*

November 2, 1994

Abstract

We prove upper bounds on the randomized communication complexity of evaluating a threshold gate (with arbitrary weights). For linear threshold gates this is done in the usual 2 party communication model, and for degree-d threshold gates this is done in the multiparty model. We then use these upper bounds together with known lower bounds for communication complexity in order to give very easy proofs for lower bounds in various models of computation involving threshold gates. This generalizes several known bounds and answers several open problems.

1 Introduction

1.1 Threshold gates

A (linear) threshold gate is a boolean function on m boolean inputs $x_1 \ldots x_m$, giving the value true if $\sum_i w_i x_i > \theta$, where w_i, the weights, and θ, the threshold value, are real constants defining the gate. (For the input variables x_i we identify false with the real number 0, and true with the real number 1.) These types of functions have attracted much interest as computing elements ([All89, GHR92, HHK91, HMP+87, HG91, MK61, SBKH91, RW92, SB91,

*Dept. of CS, Hebrew University, Jerusalem. Supported by BSF 89-00126 and by a Wolfson research award.

Smo90, MP88, Bei92], and many more). This interest has been sparked, among other reasons, because their natural interpretation as counting elements and because of some similarity they have to real neurons.

In some papers a generalization of these gates is considered: instead of taking a weighted sum of the input variables, we are allowed weighted sums of arbitrary functions of at most d variables each [MP88, Bru90, BS92, GHR92, HG91, Bei92, BRS91]. These generalized thresholds are sometimes called threshold gates of degree d or perceptrons of order d. It is not difficult to see that the following definition captures this generalization.

Definition 1 *A boolean function is called a threshold gate of degree d (or in short d-threshold gate) if it can be expressed as the sign of a real polynomial of degree at most d.*

When we say that a boolean function is represented by a sign of a polynomial we mean that a positive value is interpreted as true, a negative value as false, and zero is never allowed. Notice that now linear threshold gates are just threshold gates of degree 1.

There have been several lower bounds in the literature on the number of threshold elements needed (in various models of computation) in order to compute certain functions [HMP+87, GT91, ROS93, Vat92, HG91, Smo90]. In this paper we obtain these lower bounds in a unified and simple manner and in fact generalize each of these known lower bounds in at least one of three ways:

1. We show a lower bound for general d-threshold gates while the known lower bound held only for linear threshold gates.

2. We show a lower bound for threshold gates whose weights may be arbitrary real numbers while the known bounds only applied to threshold gates with small integer weights.

3. We show a lower bound for computing functions for which the known lower bounds do not apply.

1.2 Communication Complexity

In this paper we will obtain all our lower bounds as corollaries to known lower bounds in communication complexity models. We will use both the well studied two-party model of Yao [Yao79], and the somewhat less studied multiparty model of Chandra, Furst and Lipton [CFL83]. Definitions of these models appear in section 2.

The basic argument is the same in all cases: If a function with high communication complexity is computed by elements each having small communication complexity, then many of these elements are required in order to compute the function. Arguments along these lines were used in varying levels of explicitness and in different scenarios in [HMP+87, GT91, ROS93, Vat92, HG91, CG85, Vaz87].

We will prove our lower bounds on the number of threshold gates required to compute functions for which there are known "large" communication complexity lower bounds. In order to derive our bounds we will thus require "small" *upper bounds* on the communication complexity of evaluating a threshold gate. There are two difficulties which need to be overcome:

1. The communication complexity even of a linear threshold gate may be large. (Only if the weights are all polynomially bounded integers then the communication complexity is guarentied to be small.) In order to overcome this difficulty we consider randomized communication complexity.

2. Even the randomized complexity of 2-threshold gates may be large. In order to overcome this difficulty, for d-threshold gates, we consider the $(d+1)$-party communication model.

The main new technical result of this paper is thus the following upper bounds for the communication complexity of threshold gates.

Theorem 1. The randomized communication complexity of a linear threshold gate with m inputs is $O(\log m)$. The $(d+1)$-party randomized communication complexity of a d-threshold gate on m variables is $O(d \log^2 m)$.

The result regarding linear threshold gate was obtained jointly with Muli Safra. The result regarding multiparty complexity generalizes a result of [HG91] proving a stronger deterministic bound for the case of polynomially bounded weights.

Remark: These bounds are for two-sided error randomized complexity. It is not difficult to see that a linear threshold gate may have linear zero-error (or even one-sided error) randomized complexity.

Our lower bounds will be proven for the following functions for which strong (linear) randomized communication complexity lower bounds are known.
$IP_n(x_1 \dots x_n, y_1 \dots y_n) = \sum_i x_i y_i \; (mod\, 2)$
$DISJ_n(x_1 \dots x_n, y_1 \dots y_n) = \bigwedge_i (x_i = 0 \; or \; y_i = 0)$
$GIP_{k,n}(x_{11} ... x_{1n}, ..., x_{k1} ... x_{kn}) = \sum_i \prod_j x_{ji} \; (mod\, 2)$

1.3 New lower bounds

Circuits

The first model of computation we consider is simply a circuit composed of threshold gates. The size of the circuit is the number of gates, and since each gate may have unbounded fanin, no super-constant lower bound is trivial. This model was considered in [Smo90] where an $\Omega(n/\log n)$ lower bound is proved for several functions (including IP) for the case of linear threshold gates with small integer weights. In [ROS93] this was improved to a linear lower bound (for IP) for linear threshold gates with arbitrary weights. We derive a somewhat weaker bound, but that also applies to $DISJ$ (even for the case of arbitrary weights as opposed to the techniques of [ROS93],) and that also applies to d-threshold gates.

Theorem 2. Any circuit of linear threshold gates computing IP_n or $DISJ_n$ requires $\Omega(n/\log n)$ gates. Any circuit of d-threshold gates computing $GIP_{d+1,n}$ requires $\Omega(c_d n/log^2 n)$ gates, where $c_d = d^{-3} 4^{-d}$.

Decision trees

The second model we consider is that of decision trees, where each query may be a threshold function. (The decision tree computes a boolean function of the boolean inputs.) This model was considered in [GT91] who prove a linear lower bound (for IP) for linear threshold gates and in [Vat92] who prove a near-linear lower bound (for GIP) for d-threshold gates with small integer weights. We prove a slightly weaker bound but which applies also to $DISJ$ (for which the techniques of [GT91] do not apply,) and which applies also to d-threshold gates with arbitrary weights. This answers an open problem posed in [GT91] and in [Vat92].

Theorem 3. Any decision tree of linear threshold gates computing IP_n or $DISJ_n$ requires depth $\Omega(n/logn)$. Any decision tree of d-threshold gates computing $GIP_{d+1,n}$ requires depth $\Omega(c_d n/log^2 n)$, where $c_d = d^{-3} 4^{-d}$.

Remark: Theorem 1 may in fact be obtained as a corollary of this theorem.

Majority vote

The third model we consider is that of a depth two circuit, the top gate being a simple majority gate, and the bottom gates being threshold gates. In other words, the model is a majority vote of threshold gates. This model was considered in [HMP$^+$87] who prove an exponential lower bound for linear threshold gates with small integer weights, and in [HG91] who prove exponential lower bounds for d-threshold gates with small integer weights. We prove this bound also for threshold gates with arbitrary weights, answering a question of [ROS93].

Theorem 4. Any majority vote of linear threshold gates computing IP_n requires size $2^{\Omega(n)}$. Any majority vote of d-threshold gates computing $GIP_{d+1,n}$ requires size $2^{\Omega(c_d n/\log n)}$, where $c_d = d^{-3}4^{-d}$.

Remark: Proving super-polynomial lower bounds for the power of a linear threshold gate of linear threshold gates, both layers with arbitrary weights, is still an open problem.

2 Preliminaries

2.1 Communication complexity

Yao [Yao79] considered the following model where two players, one holding $x_1 \ldots x_n$, and the other holding $y_1 \ldots y_n$, wish to evaluate a boolean function $f(x_1 \ldots x_n; y_1 \ldots y_n)$. The two players exchange messages with each other according to some pre-defined protocol, and at the end of the protocol both of them must know the value of f. The communication complexity of f is defined to be the number of bits that need to be exchanged in the worst case by the best protocol for computing f. This model was extensively studied in the literature ([Yao79, BFS86, CG85, DGS84, FKN91, HR88, MS82, NW91, Raz90, AUY83], and many more).

In this paper we also deal with boolean functions $f(x_1 \ldots x_m)$, where the partition of the input bits to the two players is not predefined. In this case we define the communication complexity of f as the *maximum* over all possible partitions of the communication complexity under that partition.

2.2 Randomized complexity

In this paper we will be interested in randomized complexity. We will only consider in the 2-sided error case, where a small probability of error is allowed on any input. A randomized protocol for computing f is a distribution over deterministic protocols, and we define the cost of the randomized protocol as the worst case number of bits sent over all possible inputs and over all choices of the deterministic protocol.

Definition 2 *The randomized ϵ-error complexity of a boolean function f, $R_\epsilon(f)$, is defined to be the cost of the best randomized protocol for f that computes the correct answer with probability $1-\epsilon$. (Probability taken over the random choices of the protocol, but for the worst case input.)*

We will also use two abbreviations:

Definition 3 $R(f) = R_{1/3}(f)$. $R_{[\epsilon]}(f) = R_{1/2-\epsilon}(f)$.

The notation $R_{[\epsilon]}$ is useful in cases where the protocol is correct with probability only slightly greater than $1/2$.

Remark: Changing the definition of the cost to be the expected number of bits sent (expectation taken over the random choices) changes the complexity by only a constant factor.

Remark: We can view this type of randomized protocol as a protocol where the two parties share a common random string viewed by both (public coins model). This string tells them which deterministic protocol to run. A different definition often used is to allow each party separately to flip coins (private coins model). This obviously weakens the power of the players, but [New91] shows that the complexity increases by at most an additive $O(\log n)$ factor.

Remark: As stated previously, for functions $f(x_1 \dots x_m)$, where the partition of the variables to the two players is not pre-defined, $R(f)$ is defined to be the randomized complexity under the worst case partition.

Finally, we will introduce the following notation for the communication complexity of *families* of functions.

Definition 4 *Let G be a set of functions, then we define $R_\epsilon(G) = \sup_{g \in G} R_\epsilon(g)$.*

2.3 Known lower bounds

A lower bound of $\Omega(\sqrt{n})$ for the randomized complexity of $DISJ$ is given in [BFS86], this was improved to a tight $\Theta(n)$ in [KS87], and the proof was simplified in [Raz90].

Theorem A.

$$R(DISJ_n) = \Theta(n)$$

□

A lower bound of $\Omega(\log^2 n)$ for the randomized complexity of IP was proven in [Yao83]. The bound was improved to $\Omega(n/\log n)$ in [Vaz87] and then to a tight $\Theta(n)$ in [CG85]. These lower bounds show in fact that $\Theta(n)$ bits are needed even to be correct with probability slightly better than 1/2.

Theorem B.

$$R_{[\epsilon]}(IP_n) \geq n/2 - \log \epsilon^{-1}$$

□

2.4 Multiparty protocols

We will also consider the multiparty model of Chandra, Furst and Lipton [CFL83] which was also studied in [BNS89, HG91, NW91]. In this model a function of k n-bit strings, $f(\vec{x_1}...\vec{x_k})$, is to be evaluated by k players, where each player i knows the values of all x_j *except* x_i. The players communicate according to a fixed protocol by taking turns writing on a blackboard viewed by all. The cost of the protocol is the number of bits that are written in the worst case in order to evaluate f. Note that for $k = 2$ this model becomes the communication complexity model of Yao.

As in the two party case, for a function $f(x_1 \ldots x_m)$, where the partition of the bits into k subsets is not predefined, we define the k-way communication complexity of f as the maximum over all partitions into k subsets of the complexity relative to the partition. As in the two party case we consider randomized protocols, in the public coins model, and which are allowed two-sided error.

Definition 5 *The randomized ϵ-error k-way complexity of f, $R^k_\epsilon(f)$, is the cost of the best randomized k-party protocol which computes f correctly with probability $1 - \epsilon$ on every input. As before we use the abbreviations $R^k(f) = R^k_{1/3}(f)$ and $R^k_{[\epsilon]}(f) = R^k_{1/2-\epsilon}(f)$.*

A lower bound for GIP is proven in [BNS89].

Theorem C.

$$R^k_{[\epsilon]}(GIP_{k,n}) \geq n/4^k - \log \epsilon^{-1}.$$

3 Communication complexity of threshold gates

In this section we give upper bounds to the communication complexity of threshold gates: for linear threshold gates in the 2-party model, and for higher degree thresholds in the multiparty model. The first step we must take is to limit somehow the range of the weights (i.e. of the coefficients of the polynomial). By definition, the weights of the inputs to the threshold gate may be arbitrary reals. However it is known that for linear threshold gates, without loss of generality, we may assume that all the weights are $O(m \log m)$ bit integers, where m is the number of input variables. This fact seems to be folklore, and a proof can be found in [GHR92]. From this we easily also deduce:

Lemma 1 *Given an arbitrary d-threshold gate on m variables, there exists an equivalent d-threshold gate in which all weights are $O(m^d d \log m)$-bit integers.*

Proof: A d-threshold gate on m variables may also be viewed as a linear threshold gate whose inputs are all the possible monomials of degree at most d out of m variables. There are at most $O(m^d)$ such monomials, and the lemma thus follows from the statement regarding linear threshold gates. $\square$

3.1 The two-party case

In this section we provide a randomized protocol for evaluating a linear threshold gate.

Our protocol uses as a black box an equality testing protocol. While it is known that in the private coins model, randomized equality testing of n-bit strings requires $\Theta(\log n)$ bits of communication, it is also known that using public coins $O(1)$ bits suffice (this result is attributed in [RW89] to M. Karchmer). For completeness we give a proof. Denote by EQ_n the problem of testing whether two n-bit strings are equal (where each party recieves one of the two strings).

Lemma 2 $R_\epsilon(EQ_n) = O(\log \epsilon^{-1})$.

Proof: The two parties share k random n-bit strings, $\vec{r_1}...\vec{r_k}$, where $k = \log \epsilon^{-1}$. Each party computes the inner product of its string with each of the random strings, and the parties compare inner products (mod 2). (This requires k bits of communication as one party must send the k bits denoting the inner products to the other party.) It is clear that if the two strings were equal then all inner products will be equal. Elementary linear algebra in $GF(2)$ will also reveal that if the two strings were not equal then the probability that all inner products are equal is exactly 2^{-k}. $\Box$

We are now ready to give a protocol for evaluating a linear threshold gate.
Theorem 1a. Let $g(x_1 \ldots x_m)$ be a linear threshold function, then $R_\epsilon(g) \leq O(\log m + \log \epsilon^{-1})$.
Proof: Assume, without loss of generality, that g is given by $\sum_{i=1}^m w_i x_i > \theta$, where θ and all w_i are $O(m \log m)$-bit integers. The m inputs to the threshold gate are partitioned between the two parties, say the first party holds $x_1...x_k$, and the second $x_{k+1}...x_m$. The first party privately computes $y = \sum_{i=1}^k w_i x_i$, and the second privately computes $z = \theta - \sum_{i=k+1}^m w_i x_i$. The problem is now reduced to testing whether $y > z$, where y and z are n-bit integers and $n = O(m \log m)$. By adding 2^n to y and to z, we can assume without loss of generality that both y and z are positive.

The players will solve this question by finding the most significant bit, i_{sig}, in which y and z differ (or "none"). (This of course suffices in order to decide which is the larger integer.) Finding i_{sig} will be done by binary search on the n bits. Let us first describe a very simple but not quite optimal protocol, and then describe how to improve it.

The simple protocol performs a binary search on $1...n$ by first deciding whether $i_{sig} > n/2$ and then recursively searching in the correct half of the bits. Each query needed in this binary search is of the form "$i_{sig} > i$?", and is implemented by testing whether the first i bits of y are equal to the first i bits of z using the equality protocol described above. The probability of error of the equality tests must be kept small enough so that when we sum the error probabilities of all equality checks made during the binary search process ($\log n$ of them), the combined error is still ϵ.

In order to obtain the more efficient protocol we will always use equality tests which may err with constant probability (1/4) and thus require only $O(1)$ bits of communication. We are now in the situation of searching for a number, i_{sig}, in the range $1..n$ using queries of the form "$i_{sig} > i$?", where each answer may be incorrect with a small constant probability. This situation has been studied in the literature, and [FPRU90] show how it can be done, with probability of error ϵ, using $O(\log n + \log \epsilon^{-1})$ queries. $\square$

Remark: The improved protocol is not really needed for any of the applications, without it we would simply loose another $\log n$ factor in all our lower bounds.

Remark: The upper bound is in fact tight.

3.2 The multiparty case

In this section we give a randomized protocol for $k = d + 1$ parties in the multiparty model for evaluating a d-threshold gate.

We will first need to consider the complexity of two problems in the following simpler scenario: k parties which to evaluate a function of k n-bit integers $x_1 ... x_k$, where the i'th party *knowns only* x_i.

Equality of the sum

The players need to determine whether $\sum_i x_i = \theta$, where θ is some fixed integer known to all parties.

Equality of the sum protocol

The parties jointly choose a random prime p uniformly from the first $n^{O(1)}$ primes. Each party broadcasts the value $(x_i \, mod \, p)$, and then each party tests whether $\sum_i (x_i \, mod \, p) = \theta \, (mod \, p)$. It is clear that if inequality is found then $\sum_i x_i \neq \theta$. It is also easy to see that if indeed $\sum_i x_i \neq \theta$ then the protoocl may find equality for at most $O(n)$ different primes p, i.e. with probability $O(n^{-\Omega(1)})$. We have thus obtained an $O(k \log n)$-bit randomized protocol which has probability of error $O(n^{-\Omega(1)})$.

Remark: As opposed to the two-party case we do not know whether an $O(1)$ bit protocol is possible.

Threshold of the sum

The players need to determine whether $\sum_i x_i > \theta$, again where θ is some fixed integer known to all parties.

Threshold of the sum protocol

The parties will, as in the 2-party case, try to perform a binary search to find the most significant bit in which $\sum_i x_i$ and θ differ. It is simpler to describe the protocol recursively: Let x_i^M and x_i^L denote, respectively, the most significant and least significant halfs of x_i, and similarly for θ^M and θ^L. The players first perform the following set of tests:

$$\sum_i x_i^M = \theta^M?,$$

$$\sum_i x_i^M = \theta^M - 1?,$$

$$\ldots,$$

$$\sum_i x_i^M = \theta^M - k?$$

If equality is detected anywhere, say, $\sum_i x_i^M = \theta^M - h$, then the players need only recursively solve $\sum_i x_i^L > (\theta^L + h \cdot 2^{n/2})?$. If no equality was detected then the players need only recursively solve $\sum_i x_i^M > \theta^M?$, as the carry from the least significant half of the bits cannot be more than k.

For $k \leq n^{O(1)}$, the parties can implement all k equality tests simultaneously, by one run of the "equality of the sum" protocol, and thus using $O(k \log n)$ bits with total error probability $n^{-\Omega(1)}$. The recursion depth is $O(\log n)$, so the total communication is $O(k \log^2 n)$, and the total error is $n^{-\Omega(1)}$.

We now return to the usual multiparty model, and to d-threshold gates.

Theorem 1b. Let g be a d-threshold gate on m variables then $R_\epsilon^{d+1}(g) = O(d^3 \log m \log(m/\epsilon))$.

Proof: Assume g is defined by the degree d polynomial $p(x_1 \ldots x_m) > 0$. We assume without loss of generality that all coefficients of p are n-bit integers where $n = O(m^d d \log m)$. The bits $x_1 \ldots x_m$ are partitioned somehow into $k = d + 1$ sets, where each of the k players knows the values of all the bits except those in one of the sets.

Notice that the value of each monomial of p depends on at most d variables, and thus on variables in at most d sets, and thus some player can compute by itself the value of the monomial. We fix some assignment of monomials to players such that each monomial is assigned to some player that can compute its value. At this point each player i computes s_i, the sum of all monomials that were assigned to it. We have now reduced the problem of evaluating g to determining whether $\sum_i s_i > 0$. This can be done by the "threshold of the sum" protocol and requires $O(k \log^2 n)$ bits with error probability $n^{-\Omega(1)}$. To reduce the error probability to ϵ (for $\epsilon = n^{-\omega(1)}$) we repeat the protocol $O(\log(\epsilon^{-1})/\log m)$ times and take a majority vote. $\square$

4 Applications

4.1 Circuits

Lemma 3 *Let G be a family of functions. If a function f can be computed by a circuit consisting of s gates from G then, for all k, $R^k(f) \leq sR^k_{1/(3s)}(G)$.*

Proof: In order to evaluate f the k parties will simulate the circuit. The gates will be simulated in a topological order, and each gate with error probability of $1/(3s)$. The inputs to each gate are either input variables, distributed somehow between the k parties, or outputs of previous gates known to all. Thus each gate can be simulated using at most $R^k_{1/(3s)}(G)$ bits of communication. $\square$

Remark: Clearly a similar lemma holds for deterministic communication complexity.

Combining this lemma, the upper bounds from theorem 1 and the lower bounds from theorems A, B and C, we immediately get:

Theorem 2. Any circuit of linear threshold gates computing IP_n or $DISJ_n$ requires $\Omega(n/\log n)$ gates. Any circuit of d-threshold gates computing $GIP_{d+1,n}$ requires $\Omega(c_d n/log^2 n)$ gates, where $c_d = d^{-3}4^{-d}$.

4.2 Decision trees

Lemma 4 *Let G be a family of functions. If a function f can be computed by a decision tree of height h over queries from G then $R^k(f) \leq hR^k_{1/(3h)}(G)$.*

Proof: In order to evaluate f the parties will simulate the decision tree. Each query along the path taken can be simulated, with error $1/(3h)$, using at most $R^k_{1/(3h)}(G)$ bits of communication. $\square$

Combining this lemma, the upper bounds from theorem 1 and the lower bounds from theorems A,B and C, we get:

Theorem 3. Any decision tree of linear threshold gates computing IP_n or $DISJ_n$ requires depth $\Omega(n/\log n)$. Any decision tree of d-threshold gates computing $GIP_{d+1,n}$ requires depth $\Omega(c_d n/log^2 n)$, where $c_d = d^{-3}4^{-d}$.

4.3 Majority vote

As opposed to previous two cases where we can get simulations of f with small probability of error, in this case we can only get a small bias on computing f, i.e. can only compute f correctly with probability slightly better than 1/2. This, however, still suffices for proving lower bounds since the lower bounds we have for IP and GIP hold even if we only want a small bias.

Lemma 5 *Let G be a family of boolean functions. If f can be computed as the majority vote of s functions from G then $R^k_{[1/(4s)]}(f) \leq R^k_{1/(4s)}(G)$.*

Proof: The k parties will choose uniformly at random one of the s functions from G and will evaluate it (with probability of error of $1/(4s)$). The parties will predict that this is the value of f. Let us now compute a lower bound on the probability that the prediction is correct. Since f is the majority of s functions, the values of at least $(s+1)/2$ of these functions are equal to f. The probability that one of these functions was chosen is thus at least $(s+1)/(2s) = 1/2 + 1/(2s)$. The probability that the prediction is correct is thus this value minus the probability that the randomly chosen function was computed incorrectly. $\square$

From this lemma, the upper bounds from theorem 1 and the lower bounds from theorems A and C, we get:

Theorem 4. Any majority vote of linear threshold gates computing IP_n requires size $2^{\Omega(n)}$. Any majority vote of d-threshold gates computing $GIP_{d+1,n}$ requires size $2^{\Omega(c_d n/\log n)}$, where $c_d = d^{-3}4^{-d}$.

5 Acknowledgements

I thank Muli Safra for his collaboration on theorem 1a. I thank Avi Wigderson for his comments on an earlier version of this paper.

References

[All89] E. Allender. A note on the power of threshold circuts. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 580–584, 1989.

[AUY83] A.V. Aho, J.D. Ullman, and M. Yannakakis. On notions of information transfer in VLSI circuits. In *Proceedings of the* 15*st ACM STOC*, pages 133–139, 1983.

[Bei92] R. Beigel. Perceptrons, PP, and the polynomial time heirarchy. In *Proceedings of the 7th structures*, pages 14–19, 1992.

[BFS86] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory. In *Proceedings of the* 27*th IEEE FOCS*, pages 337–347, 1986.

[BNS89] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols and logspace-hard pseudorandom sequences. In *Proceedings of the* 21*st ACM STOC*, pages 1–11, 1989.

[BRS91] R. Beigel, N. Reingold, and D. Spielman. The perceptron strikes back. In *Proceedings of the* 6*th structures*, pages 286–291, 1991.

[Bru90] J. Bruck. Harmonic analysis of polynomial threshold functions. *SIAM Journal on Discrete Mathematics*, 3(2):168–177, May 1990.

[BS92] J. Bruck and R. Smolensky. Polynomial threshold functions, AC^0 functions and spectral norms. *SIAM Journal on Computing*, 21(1):33–42, February 1992.

[CFL83] A. Chandra, M. Furst, and R. Lipton. Multiparty protocols. In *Proceedings of the* 15*th ACM STOC*, pages 94–99, 1983.

[CG85] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. In *FOCS*, pages 429–442, 1985.

[DGS84] P. Duris, Z. Galil, and G. Schnitger. Lower bounds of communication complexity. In *Proceedings of the 16th STOC*, pages 81–91, 1984.

[FKN91] T. Feder, E. Kushilevitz, and M. Naor. Amortized communication complexity. In *Proceedings of the 32nd FOCS*, pages 239–248, 1991.

[FPRU90] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Computing with unreliable information. In *Proceedings of the 22nd STOC*, pages 128–137, 1990.

[GHR92] M. Goldmann, J. Håstad, and A. Razborov. Majority gates vs. general weighted threshold gates. In *Proceedings of the 7th Structure in Complexity Theory Annual Conference*, 1992. To appear in *Computational Complexity.*

[GT91] H.D. Groger and G. Turan. On linear decison trees computing boolean functions. In *Proceedings of 18th ICALP, Lecture notes in computer science 510, Springer*, pages 707–718, 1991.

[HG91] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1(2):113–129, 1991.

[HHK91] T. Hofmeister, W. Honberg, and S. Köling. Some notes on threshold circuits and multiplication in depth 4. *Information Processing Letters*, 39:219–225, 1991.

[HMP+87] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. In *Proceedings of* 28*th IEEE FOCS*, pages 99–110, 1987.

[HR88] B. Halstenberg and R. Reischuk. On different modes of communication. In *Proceedings of the 20th STOC*, pages 162–172, 1988.

[KS87] B. Kalyanasundaram and G. Schnitger. The probabilistic communication complexity of set intersection. In 2*nd Structure in Complexity Theory Conference*, pages 41–49, 1987.

[MK61] J. Myhill and W.H. Kautz. On the size of weights required for linear-input switching functions. *IRE Trans. on Electronic Computers*, EC10(2):288–290, June 1961.

[MP88] M.L. Minsky and S.A. Papert. *Perceptrons.* MIT press, 1988. Expanded edition. First edition 1968.

[MS82] K. Mehlhorn and E. M. Schmidt. Las Vegas is better than determinism in VLSI and distributive computing. In *Proceedings of the* 14*th ACM STOC*, pages 330–337, 1982.

[New91] I. Newman. Private vs. common random bits in communication complexity. *Information processing letters*, 39:67–71, 1991.

[NW91] N. Nisan and A. Wigderson. Rounds in communication complexity revisited. In *Proceedings of the 23rd STOC*, pages 419–429, 1991.

[Raz90] A. Razborov. On the distributional complexity of disjointness. In *Proceedings of the ICALP*, pages 249–253, 1990. (to appear in *Theoretical Computer Science*).

[ROS93] V.P. Roychowdhury, A. Orlitsky, and K.Y. Siu. Lower bounds on threshold and related circuits via communication complexity. Manuscript, 1993.

[RW89] R. Raz and A. Wigderson. Probabilistic communication complexity of boolean relations. In *Proceedings of the 30th FOCS*, 1989.

[RW92] A. Razborov and A. Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth 3 threshold circuits with and gates at the bottom, 1992. To appear.

[SB91] K.-I. Siu and J. Bruck. On the power of threshold circuits with small weights. *SIAM Journal on Discrete Mathematics*, 4(3):423–435, 1991.

[SBKH91] K.-I. Siu, J. Bruck, T. Kailath, and T. Hofmeister. Depth-efficient neural networks for division and related problems. Technical Report RJ 7946, IBM Research, January 1991. To appear in IEEE Trans. Information Theory.

[Smo90] R. Smolensky. On interpolation by analytical functions with special properties and some weak lower bounds on the size of circuits with symmetric gates. In *Proceedings of the 31st FOCS*, pages 628–631, 1990.

[Vat92] F. Vatan. Some lower and upper bounds for algebraic decision trees and the separation problem. In *Proceedings of 7th structures*, pages 295–304, 1992.

[Vaz87] U.V. Vazirani. Strong communication complexity or generating quasirandom sequences from two communicating semirandom sources. *Combinatorica*, 7:375–392, 1987.

[Yao79] A. Yao. Some complexity questions related to distributive computing. In *Proceedings of the* 11*th ACM STOC*, pages 209–213, 1979.

[Yao83] A. Yao. Lower bounds by probabilistic arguments. In *Proceedings of the* 24*th IEEE FOCS*, pages 420–428, 1983.