Size–Depth Trade–offs for Threshold Circuits

Russell Impagliazzo*, Ramamohan Paturi*, Michael E. Saks†

Abstract

The following size–depth trade–off for threshold circuits is obtained: any threshold circuit of depth d that computes the parity function on n variables must have at least $n^{1+c\theta^{-d}}$ edges, where constants $c > 0$ and $\theta \leq 3$ are constants independent of n and d. Previously known constructions show that up to the choice of c and θ this bound is best possible. In particular, the lower bound implies an affirmative answer to the conjecture of Paturi and Saks that a bounded depth threshold circuit that computes parity requires a super–linear number of edges. This is the first super–linear lower bound for an explicit function that holds for any fixed depth, and the first that applies to threshold circuits with unrestricted weights.

The trade-off is obtained as a consequence of a general restriction theorem for threshold circuits with a small number of edges: For any threshold circuit with n inputs, depth d and at most kn edges, there exists a partial assignment to the inputs that fixes the output of the circuit to a constant, while leaving $\lfloor n/(c_1k)^{c_2\theta^d} \rfloor$ variables unfixed, where $c_1, c_2 > 0$ and $\theta \leq 3$ are constants independent of n,k, and d.

A trade-off between the number of gates and depth is also proved: any threshold circuit of depth d that computes the parity of n variables has at least $(n/2)^{1/2(d-1)}$ gates. This trade-off, which is essentially the best possible, was proved previously (with a better constant) for the case of threshold circuits with polynomially bounded weights in [22]; the result in the present paper holds for unrestricted weights.

*Department of Computer Science and Engineering, University of California, San Diego, La Jolla, Ca 92093

†Dept. of Mathematics, Rutgers University, New Brunswick, NJ 08903 and Dept. of Computer Science and Engineering, UCSD, La Jolla, Ca. 92093. Research supported in part by NSF grant CCR-8911388 and AFOSR grants 89-0512 and 90-0008.

25th ACM STOC '93-5/93/CA,USA

1 Introduction

A fundamental problem in complexity theory is to prove lower bounds on the size and the depth of general boolean circuits for specific problems of interest such as arithmetic operations, graph reachability, linear programming, satisfiability [10, 4]. Unfortunately, current research has not begun to provide lower bounds for such computationally significant problems in general models. For example,the best known lower bound on the size of boolean circuits over the standard basis { AND, OR, NOT } for any problem in NP is a $4n-4$ bound on the parity function [19]; over the basis of all 2 input functions, the best known lower bound is $3n-3$ [3].

Since proving bounds for general circuits seems very difficult, it is interesting to look at restricted families of circuits, for example, small depth circuits over various bases. Some of these classes of circuits are interesting on their own. For example, the size and the depth required for unbounded fan–in circuits over the basis {AND, OR, NOT} to compute a function f are the same as the number of processors (up to a polynomial factor) and the parallel time (up to a constant factor) required to compute f on a CREW PRAM model.

Another basis of interest is the family of linear threshold gates. Circuits over this basis, threshold circuits, have attracted interest as a model for neural networks [13, 11], and because of the potential that hardware implementations of threshold circuits might become a reality [14]. Bounded depth threshold circuits are also appealing theoretically since they provide a surprisingly strong bounded depth computational model. Indeed, it has been shown that basic operations like addition, multiplication, division and sorting can be performed by bounded depth polynomial size threshold circuits [6, 16, 21, 4, 2, 5, 23, 26, 12]. On the other hand, unbounded fan–in bounded depth polynomial size circuits over the standard basis (even when supplemented with mod p gates for prime p) cannot compute majority [4, 20, 24]. Therefore, separating the class of functions computable by bounded depth polynomial size threshold circuits, TC^0, from those computable by polynomial time Turing machines would be an extremely interesting result in complexity theory.

In this paper, we give the first superlinear separation between bounded depth threshold circuits and **P**. More precisely, our main result (Theorem 1 and its refinement Theorem 3) says that for any threshold circuit with n inputs, depth d and kn edges, there exists a partial assignment to the inputs that fixes the output of the circuit to a constant, while leaving at least $\lfloor n/(c_1 k)^{c_2 \theta^d} \rfloor$ variables unfixed, where $c_1, c_2 > 0$ and $\theta \leq 3$ are constants. In particular, this implies (Corollary 1 and its refinement Corollary 4) that any depth d circuit that computes the parity function on n variables must have at least $n^{1+c\theta^{-d}}$ edges for the same θ and some constant $c > 0$, proving the conjecture of Paturi and Saks [17]. (The value of θ obtained in this paper is $1+\sqrt{2} = 2.414\ldots$ as compared to the value $(1+\sqrt{5})/2 = 1.618\ldots$ in the upper bound given in [17].) In particular any linear size threshold circuit for parity requires depth $\Omega(\log\log n)$ matching the upper bound given in [17].

The only lower bounds known previously for the number of edges needed to compute the parity function were for depth 2 and 3 circuits with polynomial size weights. In [17], it is proved that $\Omega(n^2/\log^2 n)$ edges are required for depth 2 threshold circuits and $\Omega(n^{1.2}/\log^{5/3} n)$ edges for depth 3 circuits. These results are obtained by showing that small size depth 2 and 3 threshold circuits can be approximated by low degree rational functions. The results in this paper are more general in that they hold for threshold circuits with arbitrary weights and all depths. However, for the special cases mentioned above, our techniques yield weaker bounds.

Our proof uses a random restriction method as in [1, 8, 25, 9, 15]. However, unlike previous random restrictions proofs, our proof uses a distribution on the partial assignments that depends on the structure of the circuit. The main restriction lemma (Lemma 1) shows that for any family of linear threshold gates on a common set of n variables with a total of δn edges, there is a partial assignment that leaves $n/(4\delta^2+2)$ variables free and makes every gate in the family dependent on at most one variable. Given a threshold circuit, this lemma can be applied to the set of gates at the first level in order to reduce the depth of the circuit by 1. A straightforward induction argument then yields the main result with $\theta = 3$. A more careful induction argument improves this to $\theta = 1+\sqrt{2}$.

In fact, the restriction lemma applies to a more general class of functions than threshold functions, called *generalized monotone* functions. A boolean function $f(\vec{x})$ is *generalized monotone* if $f(\vec{x}) = g(\vec{x} \oplus \vec{b})$ for some monotone boolean function g and boolean vector $\vec{b}$ where $\oplus$ represents the component-wise addition mod 2. (These functions are sometimes referred to as "units".)

We also prove analogous results for the number of gates in a small depth threshold circuit. We prove a lemma (Lemma 2) that is analogous to Lemma 1 and says that for any family of N generalized monotone function gates on a common set of n variables there is a partial assignment that leaves $n/(N^2+1)$ variables free and fixes all of the functions. This result, together with a simple induction argument, proves Theorem 2, that for any threshold circuit with n inputs, depth d and N gates, there exists a partial assignment to the inputs that fixes the output of the circuit to a constant, while leaving at least $\lfloor n/2N^{2(d-1)} \rfloor$ variables unfixed. This theorem easily implies a $(n/2)^{1/2(d-1)}$ bound on the number of threshold gates required to compute parity by a depth d threshold circuit (Corollary 2). A similar bound was obtained previously in [22] in the special case of circuits with polynomial size weights.

Section 2 contains definitions and some preliminary observations. In section 3, we state the main restriction theorem with $\theta = 3$ and show how it follows from Lemma 1. We also formalize the statement of Theorem 2 and show how it follows from 2. These two lemmas are proved in the succeeding two sections. In Section 6, a more careful argument is used to improve the value of θ in the main restriction theorem to $1+\sqrt{2}$. In the last section, we present some related combinatorial results and discuss some possible strengthenings.

2 Definitions

A *threshold gate* with fan-in n is an $n+1$ tuple $g = (\vec{w}; b)$ where $\vec{w} \in \mathbf{R}^n$ and $b \in \mathbf{R}$. w_i is called the *weight* of variable i and b is called the *threshold value* for the gate. The boolean function $f_g : \{0,1\}^n \to \{0,1\}$ computed by g is defined on input $(x_1, \ldots, x_n) = \vec{x} \in \{0,1\}^n$ by $f_g(\vec{x}) = \mathrm{sgn}(g(\vec{x}))$ where the weighted sum $g(\vec{x})$ is given by $g(\vec{x}) = \langle \vec{w}, \vec{x} \rangle - b = \sum_{i+1}^{n} w_i x_i - b$ and $\mathrm{sgn} : \mathbf{R} \to \{0,1\}$ is defined as

$$\mathrm{sgn}(\alpha) = \begin{cases} 1 & \text{if } \alpha > 0 \\ 0 & \text{otherwise} \end{cases}$$

A boolean function f which is representable as f_g for some threshold gate is called a *threshold function.*

A threshold circuit T on n inputs is a directed acyclic graph with a designated node (output) and exactly n *source* nodes, one for each input. Each non-source node is labelled by a threshold gate with its fan-in equal to the in-degree of the node. The function $f_v(x_1, \ldots, x_n)$ computed by the node v is obtained by functional composition in the obvious way. The function $f_T : \{0,1\}^n \to \{0,1\}$ computed by T is the function computed by the designated output node.

The *gate complexity* of T is defined as the number of non-source nodes of T. The *edge complexity* of T is defined as the number of edges in T.

The *level* of a node in a circuit T is defined inductively. The level of each source node is 0. The level of any other node i is 1 more than the maximum level of its immediate predecessors. The *depth* of T is the level of the output node. The circuit T is *layered* if the inputs to each gate are from gates of level one less.

It will be convenient to fix a variable set X of cardinality n and define an *assignment* of X to be a function $\alpha : X \to \{0,1\}$. Letting $\mathcal{A}(X)$ denote the set of assignments, we then view an n-variable boolean function f as a function from $\mathcal{A}(X)$ to $\{0,1\}$. We say that f *depends on* variable $x \in X$ if there are two assignments α and β that differ only in their values at x such that $f(\alpha) \neq f(\beta)$. The set of variables that f depends on is denoted by $S(f)$, and $s(f) = |S(f)|$.

As usual, we write $\alpha \leq \beta$ if $\alpha(x) \leq \beta(x)$ for all $x \in X$, and denote the complement of α by $\bar{\alpha}$. A monotone boolean function h is one that satisfies $h(\alpha) \leq h(\beta)$ whenever $\alpha \leq \beta$. The sum $\alpha \oplus \beta$ of two assignments is defined by $(\alpha \oplus \beta)(x) = (\alpha(x) + \alpha(y)) \bmod 2$. A boolean function f is a *generalized monotone* function if there exists an assignment β and a monotone function h such that $f(\alpha) = h(\alpha \oplus \beta)$. The assignment β is called an *orientation* of f. It is easy to see that any threshold function is a generalized monotone function, with its orientation determined by the signs of the weights of the variables.

A *partial assignment* α of X is a function from a subset Y of X to $\{0,1\}$. The domain Y of α is denoted $\Delta(\alpha)$, and elements $x \in Y$ are said to be *assigned* or *fixed* by α. The variables in the set $\Phi(\alpha) = X - \Delta(\alpha)$ are said to be *unassigned* or *free*. We denote by $\mathcal{P}(X)$ the set of all partial assignments of X. This set contains $\mathcal{A}(X)$; if we wish to emphasize that an assignment α is in $\mathcal{A}(X)$ we say that it is a *total assignment*. If Y is a subset of variables and α is a total assignment, then α_Y denotes the partial assignment with domain Y and $\alpha_Y(x) = \alpha(x)$ for $x \in Y$.

If α and β are partial assignments such that $\Delta(\beta) \subseteq \Delta(\alpha)$ and $\beta(x) = \alpha(x)$ for $x \in \Delta(\beta)$ then we say that α *extends* or is an *extension* of β. If α and β are partial assignments that fix disjoint sets of variables then the partial assignment $\alpha\beta$ is the unique minimal extension of both α and β. For a boolean function f and a partial assignment α, the *restriction of f induced by α*, written as $f(\alpha)$, is the boolean function with variable set $\Phi(\alpha)$ obtained by assigning the variables in $\Delta(\alpha)$ according to α.

An *ordering* of a set Y is a bijection $\Gamma : [|Y|] \to Y$ where $[k]$ denotes the set $\{1, 2 \ldots, k\}$. Given Γ, we refer to $\Gamma(i)$ as the i^{th} element of Y. Also, $\Gamma(\leq i)$ denotes the set $\{\Gamma(j) : j \leq i \text{ and } j \in [|Y|]\}$ and $\Gamma(\geq i)$ denotes the set $\{\Gamma(j) : j \geq i \text{ and } j \in [|Y|]\}$.

3 Results

Our main result concerns the computational power of depth d threshold circuits with a small number of edges.

Theorem 1 *Let C be an n input threshold circuit with depth d and nk edges, where $k \geq 1$. Let f denote the function computed by C. Then there exists a partial assignment α that leaves at least $\lfloor n/2(3k)^{3^{d-1}-1} \rfloor$ variables free such that $f(\alpha)$ is a constant function.*

If f is the parity function, then $f(\alpha)$ is constant only if α is a total assignment. Thus it follows from the above theorem that if C is a depth d circuit with nk edges that computes the parity function on n variables, then $n < 2(3k)^{3^{d-1}-1}$. This yields:

Corollary 1 *Any threshold circuit of depth d that computes parity of n variables has at least $n^{1+1/(3^{d-1}-1)}/(3\sqrt{2})$ edges.*

The key to proving Theorem 1 is the following:

Lemma 1 (Main Lemma) *Let F be a collection of generalized monotone functions on n variables and let $\delta = \frac{1}{n}\sum_{f \in F} s(f)$. (So the total support of the functions is $n\delta$.) Then there exists a partial assignment α that leaves at least $n/(4\delta^2+2)$ variables free such that for every $f \in F$, $f(\alpha)$ depends on at most one variable.*

Proof of Theorem 1 from Main Lemma: We proceed by induction on the depth d of the circuit. If $d = 1$, the circuit consists of a single threshold gate and the conclusion follows from Corollary 3. For $d > 1$, let F be the family of functions corresponding to the gates at depth 1. By hypothesis, the sum of the fan-ins of these gates is at most nk. Lemma 1 implies that there is a partial assignment that leaves at least $n' = n/(4k^2+2) \geq n/(6k^2)$ variables free such that the induced restriction of each function in F depends on at most one variable. We may then collapse the first level of the circuit, i.e., if

g is a gate at depth 2, then each input to g is either an input to the circuit or the output of a gate at level 1, which after the restriction is equal to a variable or its complement. Thus, each gate g at depth 2 can now be reexpressed as a threshold gate that depends only on the original inputs. (Note that g may have several edges entering which depend on the same variable but these can be combined into one edge by adjusting the weights of g). Hence, we obtain a depth $d-1$ circuit C' on at least n' variables with at most $n'k'$ edges, where $k' = 6k^3$. By induction hypothesis, there exists a partial assignment of the variables of C' such that the number of free variables is at least

$$\begin{aligned} \lfloor \frac{n'}{2(3k')^{3^{d-2}-1}} \rfloor &\geq \lfloor \frac{n/(6k^2)}{2(3(6k^3))^{3^{d-2}-1}} \rfloor \\ &\geq \lfloor \frac{n}{2(3k)^{3^{d-1}-1}} \rfloor, \end{aligned}$$

as required to prove the theorem. ♣

Remark: Since the main lemma applies to generalized monotone functions, it might appear that Theorem 1 could be generalized to apply to circuits whose gates compute arbitrary generalized monotone functions. However, the proof fails to generalize because when the circuit is collapsed in the induction step, a level 2 gate may have more than one input corresponding to the same variable. In that case, it is not true that the gate computes a generalized monotone function of the original variables; indeed it is easy to see that *every* n-variable boolean function can be represented as a single generalized monotone function on $2n$ variables by identifying variables in pairs.

To bound the number of gates in a small depth circuit instead of the number of edges, we use the following (simpler) relative of Lemma 1:

Lemma 2 *Let F be a collection generalized monotone functions on n variables. Then there exists a partial assignment α that leaves at least $\lfloor n/(|F|^2 + 1)\rfloor$ variables free such that for each $f \in F$, $f(\alpha)$ is a constant function.*

This leads to the following result for threshold circuits with a small number of gates. In this case, the result holds for generalized monotone functions

Theorem 2 *Let C be a circuit consisting of generalized monotone function gates of depth d on n inputs with at most N gates. Then there exists a partial assignment α leaving $\lfloor \frac{n}{2N^{2(d-1)}} \rfloor$ variables free such that $f_C(\alpha)$ is constant.*

Proof of Theorem 2 from Lemma 2: We proceed by induction on the depth d of the circuit, as in the proof of Theorem 1. If $d = 1$, the circuit consists of a single threshold gate and the conclusion follows from Corollary 3. For $d > 1$, consider the family F of threshold functions corresponding to the depth 1 gates. Note that $|F| \leq N - 1$. We apply Lemma 2 to F to obtain a partial assignment that leaves at least $n' = \lfloor n/(|F|^2 + 1)\rfloor \geq \lfloor n/N^2 \rfloor$ variables free such that the induced restriction of each function in F is constant. After the restriction, the only non-constant inputs to the second level gates are the inputs to the circuit. Thus, the resulting circuit C' has depth at most $d-1$, at most N gates and at least n' variables. By the induction hypothesis, there exists a partial assignment of the variables of C' which leaves at least $\lfloor n'/2(N)^{2(d-2)} \rfloor \geq \lfloor n/2(N)^{2(d-1)} \rfloor$ variables free (where the inequality follows from the fact that for positive integers n, A, B, $\lfloor \lfloor n/A \rfloor /B \rfloor = \lfloor n/AB \rfloor$). ♣

Again using the fact that the only partial assignments that make the parity function constant are the total assignments, we deduce that the number N of gates of a depth d parity circuit satisfies $2N^{2(d-1)} \geq n$ and thus

Corollary 2 *Any circuit of depth d consisting of generalized monotone function gates that computes the parity of n inputs has at least $(n/2)^{1/2(d-1)}$ gates.*

Slightly stronger bounds than those obtained in Theorem 2 and Corollary 2 were previously proved in [16, 22] for the case of threshold circuits with polynomially bounded weights.

It remains to prove Lemmas 1 and 2, and these proofs constitute the main part of the paper. The proofs of these lemmas are similar; both use a probabilistic method to demonstrate the existence of the required partial assignment.

The proof of Lemma 2 is somewhat simpler, so we present the proof in the next section. The proof of Lemma 1 will be presented in section 5.

4 Proof of Lemma 2

Before proving the main result for this section, we establish the following easy fact:

Lemma 3 *Let f be a non-constant generalized monotone function on X with an orientation β, and let Γ be an ordering of X. Then there exists a $j \in \{0, 1, \ldots, n\}$ such that $f(\beta_{\Gamma(\leq j)})$ is identically 0 and $f(\bar{\beta}_{\Gamma(\geq j)})$ is identically 1.*

Proof: Consider first the case that β is identically 0, i.e., f is monotone. Since f is not a constant function, we have $f(\beta) \equiv f(\beta_{\Gamma(\leq n)}) \equiv 0$. Let j be the least index such that $f(\beta_{\Gamma(\leq j)})$ is identically 0. $j \geq 1$ since f is not constant function. Then $f(\beta_{\Gamma(\leq j-1)})$ is not identically 0 which implies that there is a total assignment α that extends $\beta_{\Gamma(\leq j-1)}$ such that $f(\alpha) = 1$. Then $f(\bar{\beta}_{\Gamma(\geq j)})$ is identically 1 by monotonicity since every total assignment that extends $\bar{\beta}_{\Gamma(\geq j)}$ is greater than or equal to α.

In the case that f is not monotone, the desired result follows immediately by applying the previous argument to the monotone function $h(\alpha) = f(\alpha \oplus \beta)$. ♣

One useful consequence of this lemma is:

Corollary 3 *Any generalized monotone function f on n variables has a partial assignment α that leaves at least $\lfloor n/2 \rfloor$ variables free, such that $f(\alpha)$ is constant.*

We now prove Lemma 2. In the probabilistic arguments in this section and the next we adopt the following notational convention. Random variables are denoted by placing a ˜ over the identifier. When we refer to a specific value that a random variable may assume, we denote that value by an identifier without a ˜ .

We have a family F of boolean generalized monotone functions on n variables and seek a partial assignment that makes all of the functions constant. It will be convenient to fix an indexing $f^1, f^2, \ldots, f^m$ of the functions in F. Let β^i be an orientation for f^i.

Fix an ordered partition $Y_1, Y_2, \ldots, Y_q$ of the variable set X into $q = m^2 + 1$ blocks of nearly equal size (each having $\lfloor n/q \rfloor$ or $\lfloor n/q \rfloor + 1$ variables). The desired partial assignment will be obtained by fixing the variables in all but one of the blocks. We describe a randomized procedure P which produces such a partial assignment $\tilde{\alpha}$ and show that with positive probability $f^i(\tilde{\alpha})$ is constant for all $i \in [m]$.

The procedure P is as follows. Let $\mathcal{U}$ be a symbol (meaning "unallocated"). Choose uniformly at random a 1–1 function $\tilde{M}$ from $[m] \times [m] \cup \{\mathcal{U}\}$ to $[q]$. Intuitively, we think of $\tilde{M}$ as "allocating" sets $Y_{\tilde{M}(i,1)}, \ldots Y_{\tilde{M}(i,m)}$ to function f^i, while leaving set $Y_{\tilde{M}(\mathcal{U})}$ unallocated. In addition choose a vector $(\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_m)$ uniformly from the set $\{0, 1, 2, \ldots, m\}^m$. For each $1 \leq i, j \leq m$, if $j \leq \tilde{t}_i$ then fix the variables in $Y_{\tilde{M}(i,j)}$ according to β^i, and if $j > t_i$ then fix the variables in $Y_{\tilde{M}(i,j)}$ according to $\bar{\beta}^i$. Thus, all of the variables except those in the unique block $Y_{\tilde{M}(\mathcal{U})}$ are fixed. Call the resulting partial assignment $\tilde{\alpha}$.

The key property of this distribution is given by:

Lemma 4 *For each $h \in [m]$, the probability that $f^h(\tilde{\alpha})$ is not constant is at most $1/(m+1)$.*

It follows from this lemma that the probability that there exists $i \in [m]$ with $f^i(\tilde{\alpha})$ not constant is at most $m/(m+1)$. Thus there exists a particular α such that $f^i(\alpha)$ is constant for all $i \in [m]$ and this α satisfies the conclusion of Lemma 2.

Proof of Lemma 4 Fix $h \in [m]$. We define a modification P^h of the procedure P. It will be easy to see that this modified construction produces the same distribution; we then use the modified construction to verify the conclusion of the lemma.

The modified construction is as follows. Choose $\tilde{t'}_i \in \{0, \ldots, m\}$ uniformly at random for $i \neq h$, and pick $\tilde{t'}_h \in \{1, \ldots, m+1\}$ uniformly at random. Pick $\tilde{M}'$, a random 1–1 function from $[m] \times [m] \cup \{(h, m+1)\}$ to $[q]$. For $i \neq h$, assign variables in $Y_{\tilde{M}'(i,j)}$ as before, according to β^i if $j \leq \tilde{t'}_i$, and according to $\bar{\beta}^i$ otherwise. For $j < \tilde{t'}_h$, assign the variables in $Y_{\tilde{M}'(h,j)}$ according to β^h and for $j > \tilde{t'}_h$ according to $\bar{\beta}^h$. We leave the variables in $Y_{\tilde{M}'(h,\tilde{t'}_h)}$ unassigned.

As in the original procedure, each gate is allocated m random sets of variables, with one random set of variables being unallocated. For each gate, the number of these sets fixed according to the orientation of the gate is randomly chosen between 0 and m, and the rest are set according to the negation of the orientation. Thus, the two distributions are identical. More formally, we could define $\tilde{M}(i,j) = \tilde{M}'(i,j)$ for $i \neq h$, $\tilde{M}(h,j) = \tilde{M}'(h,j)$ for $j < \tilde{t_h}'$, $\tilde{M}(h,j) = \tilde{M}'(h,j+1)$ for $\tilde{t'}_h < j \leq m$, and $\tilde{M}(\mathcal{U}) = \tilde{M}'(h, \tilde{t_h}')$, and define $\tilde{t}_i = \tilde{t'}_i$ for $i \neq h$, $\tilde{t}_h = \tilde{t'}_h - 1$. Then the distributions on $\tilde{M}$ and $\tilde{t}$ are identical to those in the original process, and all values M and $t_1, \ldots, t_m$ of these random variables, if chosen by the original process, would determine the same value of α as M' and the t'_i's do in the modified process.

Thus, it will suffice to upper bound the probability that $f^h(\tilde{\alpha})$ is not constant when $\tilde{\alpha}$ is constructed according to P^h. For this, fix any value M' for $\tilde{M}'$, and fix values t'_i for $\tilde{t'}_i$, $i \neq h$. This determines the setting of $\tilde{\alpha}$ for all the variables in $Y_{M'(i,j)}$ for $i \neq h$. We will show that, given the above information, the probability that f^h is non-constant when restricted by $\tilde{\alpha}$ is at most $1/m+1$. Let g be f^h restricted to the variables in the blocks $Y_{h,j}$, $1 \leq j \leq m+1$, with the other variables set according to $\tilde{\alpha}$. (As we noted before, the value of $\tilde{\alpha}$ at all other variables has been fixed by the information we are conditioning on.) g

is a generalized monotone function with the same orientation β^h as f^h.

For each block $Y_{M'(h,j)}$, fix an arbitrary order on variables of the block; extend these orders to an ordering Γ on all the variables for g by ordering the blocks according to j. Then we can apply Lemma 3 to obtain an index l such that the functions $g(\beta^h_{\Gamma(\leq l)})$ and $g(\bar{\beta}^h_{\Gamma(\geq l)})$ are both constant. Let $0 \leq r \leq m+1$ be such that $\Gamma(l) \in Y_{M'(h,r)}$, i.e., the l–th variable is in the r–th block allocated to f^h. We claim that $g(\tilde{\alpha})$ and hence $f^h(\tilde{\alpha})$ is constant unless $\tilde{t'}_h = r$, an event which happens with probability $1/(m+1)$ (since $\tilde{t'}_h \in [m+1]$ is chosen independently from $\tilde{M}'$ and the $\tilde{t'}_i$'s for $i \neq h$.). If $\tilde{t'}_h > r$, all variables in blocks labelled r or less are fixed by $\tilde{\alpha}$ to β^h, so $\tilde{\alpha}$ extends $\beta^h_{\Gamma(\leq l)}$, so $g(\tilde{\alpha})$ is constant. Similarly, if $\tilde{t'}_h < r$, $\tilde{\alpha}$ extends $\bar{\beta}^h{}_{\Gamma(\geq l)}$ and $g(\tilde{\alpha})$ is constant. Thus, with probability $1 - 1/(m+1)$, $f^h(\tilde{\alpha}) = g(\tilde{\alpha})$ is constant, as required to complete the proof of Lemma 4 and hence of Lemma 2. ♣

5 Proof of the Main Lemma

Again, index the functions of F as $f^1, \ldots, f^m$ and let β^i denote an orientation for f^i. For each variable x, let D_x be the subfamily of F consisting of those functions that depend on variable x and let $\delta_x = |D_x|$. Thus the quantity $\delta = \frac{1}{n}\sum_{f\in F} s(f)$ in the lemma is just the average of the δ_x. We seek a partial assignment α that leaves at least $n/(4\delta^2+2)$ variables free and so that for every $f \in F$, $f(\alpha)$ depends on at most one variable.

We will describe a randomized algorithm $A(L)$, where L is a positive real parameter, for constructing a partial assignment $\tilde{\alpha}$ and show that, for an appropriate choice of L, $\tilde{\alpha}$ has the desired properties with positive probability. The random procedure in the previous proof can be viewed as associating a fraction $m/(m^2+1)$ of the variables to each function and then fixing the variables associated with a function in a way that is determined by the orientation of the function. We will do something similar here; however here we will require that the set of variables assigned to f^i is a subset of $S(f^i)$, the set of variables on which f^i depends.

Procedure $A(L)$.

1. *Partition the variables.* (Intuitively, this step assigns each function a set of variables in proportion to its support size, leaving a few variables unassigned.) Construct a random partition of the variable set X into $m+1$ parts $\tilde{R}, \tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_m$. For each variable x, the block of the partition containing x is determined independently according to the following rule. With probability $1/(1+L\delta_x)$, $x \in \tilde{R}$. Otherwise x is assigned to block $\tilde{C}_{\tilde{a}(x)}$ where $\tilde{a}(x)$ is the index of a uniformly chosen element of D_x. In other words, for each $f^i \in D_x$, the probability that $x \in \tilde{C}_i$ is $L/(1+L\delta_x)$. Let $\tilde{r} = |\tilde{R}|$ and for each $i \in [m]$, let $\tilde{c}_i = |\tilde{C}_i|$.
2. *For each $i \in [m]$, fix all of the variables in $\tilde{C}_i$* . (Intuitively, this step fixes the variables assigned to each f^i so that any particular function f^i becomes constant with a good probability.) For each $i \in [m]$, choose $\tilde{b}_i$ uniformly at random from $\{0, 1, \ldots, \tilde{c}_i\}$. Choose a subset $\tilde{B}_i$ of $\tilde{C}_i$ uniformly from all $\tilde{b}_i$ element subsets of $\tilde{C}_i$. Let $\tilde{\gamma}^i$ denote the partial assignment which fixes the variables of $\tilde{B}_i$ according to β^i and fixes the variables of $\tilde{C}_i - \tilde{B}_i$ according to $\bar{\beta}^i$. Let $\tilde{\gamma}$ be the union of the partial assignments $\tilde{\gamma}^i, i \in [m]$.
3. *Fix some of the variables in $\tilde{R}$.* (Intuitively, this step cleans up the few remaining functions that are still non-constant, so that they depend on at most one variable.) For each $i \in [m]$: Let $\tilde{T}_i$ denote the set of variables on which $f^i(\tilde{\gamma})$ depends, and if $\tilde{T}_i \neq \emptyset$, let $\tilde{T'}_i$ be an arbitrary subset containing all but one element of $\tilde{T}_i$; otherwise, $\tilde{T'}_i = \emptyset$. Let $\tilde{\alpha}$ be the restriction obtained from $\tilde{\gamma}$ by setting all the elements of each $\tilde{T'}_i$ to 1.

The third step above ensures that the partial assignment $\tilde{\alpha}$ has the required property that $f^i(\tilde{\alpha})$ depends on at most one variable for each i. Thus it remains to show that with positive probability the number $\tilde{\phi}$ of variables left free is sufficiently large.

The set of free variables of $\tilde{\alpha}$ consists of those variables in $\tilde{R}$ that are not fixed during Step 3. Thus $\tilde{\phi} \geq \tilde{r} - \sum_{i=1}^m \max\{0, |\tilde{T}_i| - 1\}$. Our goal is to obtain a lower bound on the expectation of $\tilde{\phi}$. Note that $\mathbf{E}[\tilde{r}] = \sum_{x\in X} 1/(L\delta_x + 1)$.

The harder part is to upper bound the expectation of $a(|\tilde{T}_i|)$ where $a(m) = \max\{0, m-1\}$. The key lemma of this section is:

Lemma 5 *For each $h \in [m]$,*

$$\mathbf{E}[a(|\tilde{T}_h|)] \leq \frac{1}{L} \sum_{x\in S(f^h)} \frac{1}{L\delta_x + 1}$$

Assuming the lemma for the moment, we have:

$$\mathbf{E}[\tilde{\phi}] \;\; \geq \;\; \mathbf{E}[\tilde{r}] - \sum_{i=1}^{m} \mathbf{E}[a(|\tilde{T}_i|)]$$

$$\begin{aligned}
&\geq \sum_{x\in X} 1/(Ld_x+1) - \sum_{i=1}^{m} \frac{1}{L} \sum_{x\in S(f^i)} \frac{1}{L\delta_x+1} \\
&\geq \sum_{x\in X} 1/(Ld_x+1) - \frac{1}{L}\sum_{x\in X}\sum_{f_i\in D_x} \frac{1}{L\delta_x+1} \\
&\geq \sum_{x\in X} 1/(Ld_x+1) - \sum_{x\in X} \frac{\delta_x/L}{L\delta_x+1} \\
&= \sum_{x\in X} \frac{1-\delta_x/L}{L\delta_x+1} \\
&\geq n(\frac{1-\delta/L}{L\delta+1}),
\end{aligned}$$

where the last inequality follows from the convexity of the function $\lambda(x) = (1-x/L)/(Lx+1)$ for positive x. Choosing the parameter $L = 2\delta$ to (approximately) maximize this quantity, we will have that the expectation of $\tilde{\phi}$ is at least $n/(4\delta^2+2)$. Thus, among the partial assignments that could be produced by the procedure $A(2\delta)$ there must exist a partial assignment that leaves at least $n/(4\delta^2+2)$ variables free, as required to prove the Main Lemma.

It remains to prove Lemma 5. Fix $h \in [m]$. Let $\tilde{\chi}_h$ denote the random variable which is 1 if the h^{th} function is not fixed after Step 2, i.e., $f^h(\tilde{\gamma})$ is not constant, and is 0 if $f^h(\tilde{\gamma})$ is constant. Clearly $\tilde{T}_h$ is empty if $\chi_h = 0$ and otherwise $\tilde{T}_h$ is a subset of the set $\tilde{U}_h = \tilde{R} \cap S(f^h)$. Letting $\tilde{u}_h$ denote the cardinality of $|\tilde{U}_h|$, we have $a(|\tilde{T}_h|) \leq \tilde{\chi}_h a(\tilde{u}_h)$. Lemma 5 is an immediate consequence of:

Lemma 6 *Let $h \in [m]$ and let q be an arbitrary nonnegative valued function defined on the natural numbers. Then*

$$\mathbf{E}[\tilde{\chi}_h q(\tilde{u}_h)] \leq \frac{1}{L}\mathbf{E}[q(\tilde{u}_h+1)]$$

Applying this lemma with $q = a$, the right hand side of the inequality is just $\mathbf{E}[\tilde{u}_h]/L$, which, by linearity of expectation, is $\frac{1}{L}\sum_{x\in S(f^h)} \mathbf{P}[x \in \tilde{R}] = \frac{1}{L}\sum_{x\in S(f^h)} \frac{1}{L\delta_x+1}$ as required to prove Lemma 5.

Proof of Lemma 6 Let $\tilde{K} = \tilde{C}_h \cup \tilde{U}_h$, i.e., the set of variables on which f^h depends that are assigned to either $\tilde{C}_h$ or $\tilde{R}$. Let $\tilde{k} = |\tilde{K}|$.

Fix a particular instantiation C_i and B_i for all $i \neq h$ and let Ξ denote the event that $\tilde{C}_i = C_i$ and $\tilde{B}_i = B_i$ for all $i \neq h$. Note that Ξ determines the value K of $\tilde{K}$ and also determines $\tilde{\gamma}$ on all variables in $S(f^h) - \tilde{K}$. Thus, let g be the function of the variables in K determined by restricting f^h according to $\tilde{\gamma}_i$ for each $i \neq h$.

We will show that for any such event Ξ:

$$\mathbf{E}[\tilde{\chi}_h q(\tilde{u}_h)|\Xi] \leq \frac{1}{L}\mathbf{E}[q(\tilde{u}_h+1)|\Xi].$$

The lemma then follows by deconditioning the expectation.

Given Ξ, the variables in K are partitioned into the two sets $\tilde{C}_h$ and $\tilde{U}_h$ as follows: for $x \in K$, the conditional probability given Ξ that x is in $\tilde{R}$ (and hence in $\tilde{U}_h$) is $p = 1/(L+1)$ and otherwise (with probability $L/(L+1)$), x is in $\tilde{C}_h$. Furthermore, these events are independently determined for each $x \in K$. Thus the conditional distribution given Ξ of $\tilde{u}_h$ is a binomial distribution $B(k,p)$, i.e., $\mathbf{P}[\tilde{u}_h = i|\Xi] = \binom{k}{i}p^i(1-p)^{k-i}$. Let P_i denote the probability $\mathbf{P}[g(\tilde{\gamma})\text{is not constant}|\Xi \wedge (\tilde{u}_h = i)]$. We have:

$$\begin{aligned}
\mathbf{E}[\tilde{\chi}_h q(\tilde{u}_h)|\Xi] &\leq \sum_{i=0}^{k} P_i q(i) \mathbf{P}[\tilde{u}_h = i|\Xi] \\
&= \sum_{i=1}^{k} P_i q(i) \binom{k}{i} p^i (1-p)^{k-i}
\end{aligned}$$

We next determine an upper bound for $P_i = \mathbf{P}[g(\tilde{\gamma})\text{is not constant}|\Xi \wedge (\tilde{u}_h = i)]$.

The conditional distribution of $\tilde{C}_h, \tilde{B}_h$ given $\Xi \wedge (\tilde{u}_h = i)$ can be described as follows. $\tilde{C}_h$ is a uniformly chosen $k-i$ element subset of K, $\tilde{b}_h$ is chosen uniformly at random from $\{0,1,\ldots,k-i\}$ and $\tilde{B}_h$ is a uniformly chosen $\tilde{b}_h$ element subset of $\tilde{C}_h$.

An alternative way to generate this same distribution on $\tilde{B}_h, \tilde{C}_h$ is as follows: Choose an order $\tilde{\Gamma}$ of the elements of K uniformly at random. Choose $\tilde{b}_h$ uniformly from $\{0,1,\ldots,k-i\}$. Let $\tilde{B}_h$ be the first $\tilde{b}_h$ elements of K and let $\tilde{C}_h$ consist of $\tilde{B}_h$ together with the last $k-i-\tilde{b}_h$ elements of K. It is clear that this distribution is equivalent to the one described in the previous paragraph.

We want to determine the conditional probability that g is not constant given Ξ and $\tilde{u}_h = i$. Lemma 3 applied to the function g and the ordering $\tilde{\Gamma}$ of K implies that there is an index $\tilde{j} = \tilde{j}(\tilde{\Gamma})$ in $\{0,1,\ldots,k\}$ such that $f(\beta^h_{\tilde{\Gamma}(\leq\tilde{j})})$ is identically 0 and $f(\bar{\beta}^h_{\tilde{\Gamma}(\geq\tilde{j})})$ is identically 1. Now, observe that if $\tilde{b}_h$ is chosen to be greater than or equal to $\tilde{j}$ then the partial assignment $\tilde{\gamma}^h$ is an extension of $\beta^h_{\tilde{\Gamma}(\leq\tilde{j})}$ and is thus identically 0. Similarly, if $\tilde{b}_h$ is chosen to be less than $\tilde{j}-(k-i)$ then the partial assignment $\tilde{\gamma}^h$ is an extension of $\bar{\beta}^h_{\tilde{\Gamma}(\geq\tilde{j})}$ and is thus identically 1. Thus, the only way that $\tilde{w}_h$ can be non-zero is if $\tilde{b}_h$ satisfies $\tilde{j}-(k-i) \leq \tilde{b}_h \leq \tilde{j}-1$, and since $\tilde{b}_h$ is chosen uniformly in the range $\{0,1,\ldots,k-i\}$, this happens with probability at most $i/(k-i+1)$. We conclude that the conditional probability given $\Xi \wedge (\tilde{u}_h = i)$ that g is not constant is at most $i/(k-i+1)$. Using this probability, we can rewrite the expression for the conditional expectation of $\tilde{\chi}_h q(\tilde{u}_h)$ as:

$$
\begin{aligned}
&\mathbf{E}[\tilde{\chi}_h q(\tilde{u}_h)|\Xi] \\
&\leq \sum_{i=1}^{k} q(i)\binom{k}{i}p^i(1-p)^{k-i}\frac{i}{k-i+1} \\
&= \frac{p}{1-p}\sum_{i=1}^{k} q(i)\binom{k}{i-1}p^{i-1}(1-p)^{k-(i-1)} \\
&= \frac{p}{1-p}\sum_{i'=0}^{k-1} q(i'+1)\binom{k}{i'}p^{i'}(1-p)^{k-i'} \\
&= \frac{p}{1-p}\sum_{i'=0}^{k-1} q(i'+1)\mathbf{P}[\tilde{u}_h = i'|\Xi] \\
&\leq \frac{p}{1-p}\mathbf{E}[q(\tilde{u}_h+1)|\Xi] \\
&= \frac{1}{L}\mathbf{E}[q(\tilde{u}_h+1)|\Xi],
\end{aligned}
$$

as required to complete the proof of Lemma 6, which in turn completes the proofs of 5 and the main lemma. ♣

6 An Improved Lower Bound

In this section, we present refined versions of Theorem 1 and Corollary 1 for which the parameter θ is reduced from 3 to $1+\sqrt{2}$. In the following, our results are stated for layered threshold circuits. This is sufficient for our purposes since an arbitrary threshold circuit can be converted to a layered one that computes the same function by increasing the number of edges by a factor at most d.

To state the improvement of Theorem 1, define ν_i for $i \geq 1$, to be the solution to the recurrence equation $\nu_{i+2} = 2\nu_{i+1} + \nu_i$ with the initial conditions $\nu_1 = 1$ and $\nu_2 = 3$. Note that the explicit expression for ν_i is of the form $A(1+\sqrt{2})^i + B(1-\sqrt{2})^i$ where $A \neq 0$ and B are easily determined constants, and $\nu_i \in \Theta((1+\sqrt{2})^i)$.

Theorem 3 *Let C be a layered depth d threshold circuit with n inputs d and nk edges, where $k \geq 1$. Let f denote the function computed by C. Then there exists a partial assignment α that leaves at least $\lfloor \frac{n}{4(11k)^{\nu_d - 1}} \rfloor$ variables free such that $f(\alpha)$ is a constant function.*

As before, this theorem immediately implies a size-depth trade-off for the parity function.

Corollary 4 *Any threshold circuit of depth $d \geq 2$ that computes parity of n variables has at least $(n/11)^{1+\frac{1}{\nu_d - 1}}$ edges.*

To motivate the proof of Theorem 3, we first summarize the main inductive argument of the previous proof. In each inductive step the depth of the circuit is decreased by one, by fixing some variables in order to eliminate the first level. The fraction of variables left unfixed after each step is inversely proportional to the square of the parameter δ, the ratio of the number of edges at the first level to the number of unfixed variables before the step. In analyzing the resulting recurrence, we upper bounded the number of edges at the first level by the total number of edges in the circuit.

The idea for improving this analysis is to improve substantially this upper bound on the number of edges at the first level, thereby increasing the fraction of variables that are known to survive each reduction step. It might seem that since the circuit is arbitrary, we can not do better than to bound the number of edges at the first level by the total number of edges in the circuit. This is indeed true the first time the reduction is applied. However, it turns out that for all subsequent reduction steps, there is a better bound available. This is because the partial assignment produced by $A(L)$ in the proof of Lemma 1 has a very useful side effect: for each first level gate whose output is fixed to a constant by the partial assignment, the edges leaving that gate can be eliminated from the circuit. We will show that, with high probability, the number of edges in the second level of the circuit (which becomes the first level) is decreased by a large amount. This allows us to keep a larger fraction of variables unassigned when we recursively perform the reduction on the first level of the resulting circuit.

To make this idea precise we need a modified version of Lemma 1. The proof of Lemma 1 will appear in the full version of the paper.

Lemma 7 *Let F be a collection of generalized monotone functions on n variables, and suppose that each function $f \in F$ has a nonnegative weight $w(f)$. Let $\delta = \frac{1}{n}\sum_{f\in F} s(f) \geq 1$ and $W = \sum_{f \in F} w(f)$. Then, assuming that $n/(9\delta)^2 \geq 4$, there exists a partial assignment α that leaves at least $n/(9\delta)^2$ variables free such that:*

1. *For every $f \in F$, $f(\alpha)$ depends on at most one variable.*
2. $$\sum_{f : f(\alpha) \text{ is not constant}} w(f) \leq W/8\delta.$$

Note that when we apply this lemma in the inductive argument, the weights of the functions will correspond to the out-degree of the corresponding gates. The point is that the total number of edges remaining on the new first level can then be bounded above by $1/8\delta$ times the number of edges in the circuit.

Corollary 5 *Let C be a depth d layered threshold circuit with n inputs and nk edges, where $k \geq 1$. Let f denote the function computed by C. For $i \geq 0$, let $\rho_i = (11k)^{\nu_{i+1}-1}$. Then for each $i \in \{0, 1, \ldots, d-1\}$ such that $n \geq 4\rho_i$, there exists a partial assignment α^i that leaves at least $\frac{n}{\rho_i}$ variables free such that $f(\alpha^i)$ can be computed by a layered circuit C_i of depth $d - i$.*

We can now finish the proof of Theorem 3. If $n < 4\rho_{d-1}$ then we can choose any total assignment for α and the conclusion holds trivially. Otherwise, we may apply Corollary 5 with $i = d-1$ to find a partial assignment α^{d-1} with at least n/ρ_{d-1} unfixed variables such that the resulting restricted function can be computed by a single threshold gate. Applying Corollary 3, we need to fix at most half the remaining variables to make the function constant. ♣

7 Final Remarks and Open Problems

This paper gives the first non-trivial lower bounds for threshold circuits with arbitrary weights and any fixed depth, on the number of edges and gates needed to compute an explicit function. The results show that there are functions $\epsilon(d)$ and $\gamma(d)$ such that any depth d threshold circuit that computes parity on n variables must have at least $n^{1+\epsilon(d)}$ edges and $n^{\gamma(d)}$ gates. In our case, the functions $\epsilon(d)$ and $\gamma(d)$ tend to 0 as d tends to ∞. An apparently difficult challenge would be to prove an n^ϵ lower bound, with $\epsilon > 1$ a constant independent of depth, on the number of gates needed to compute some explicit function.

For each fixed depth, there is a gap between the bounds provided by our results and the best constructions for parity circuits. For instance, for depth 2 circuits, the result in the present paper gives an $\Omega(n^{3/2})$ bound on the number of edges and an $n^{1/2}$ bound on the number of gates, while the best construction requires $O(n^2)$ edges and $O(n)$ gates. One way to reduce this gap is to improve Lemma 1 by increasing the number of variables left free in the restrictions.

Problem 1 *What is the smallest exponent r such that the conclusion of Lemmas 1 and 7 holds with $n/(4\delta^2 + 2)$ replaced by $\Omega(n/\delta^r)$?*

The best possible r is at least 1, as is shown by the family $F = \{T_i : 0 \leq i \leq n\}$ of n-variable functions, where T_i is the function which is 1 on inputs with at least i 1's. If the conclusion holds for $r = 1$, then this would lead to an $\Omega(n^2)$ edge lower bound for depth 2 circuits that compute parity, and more generally to an improvement in the value of θ in the main Theorem to $\theta = (1+\sqrt{5})/2$. This would exactly match the value of θ in the known upper bounds. Note that for purposes of applications to circuits, it would suffice to consider the above problem for families of threshold functions, rather than for generalized monotone functions.

It is interesting also to look for a similar improvement to Lemma 2.

Problem 2 *What is the smallest exponent r such that the conclusions of Lemmas 2 hold with $n/(4\delta^2 + 2)$ replaced by $\Omega(n/|F|^r)$.*

Again, the best lower bound on r we have is 1. Any value of $r < 2$ would give a corresponding improvement in Theorem 2: the number of variables left free would be $\Omega(n/N^{r(d-1)})$.

For the special case of monotone functions, it is easy to show that Lemma 2 has such a strengthening:

Proposition 1 *Let F be a collection of monotone functions on n variables. Then there exists a partial assignment α that leaves at least $\lfloor n/(|F|+1) \rfloor$ variables free such that for each $f \in F$, $f(\alpha)$ is a constant function.*

Proof. Fix an ordering Γ for the variables X and for each $f \in F$, let $j(f)$ be the index promised by Lemma 3. Order the functions as $f_1, f_2, \ldots, f_m$ so that $j_1 \leq j_2 \leq \ldots \leq j_m$, where $j_i = j(f_i)$ and let $j_0 = 0$ and $j_{m+1} = n+1$. Let i be an index such that $j_{i+1} - j_i$ is maximum (and hence at least $\frac{n+1}{m+1}$) and let α be the assignment which sets all variables in $\Gamma(\leq j_i)$ to 0 and $\Gamma(\leq j_{i+1})$ to 1. Then $f_h(\alpha)$ is identically 0 for all $h \leq i$ and $f_h(\alpha)$ is identically 1 for all $h \geq i+1$. The number of free variables of α is $j_{i+1} - j_i - 1 \geq \lceil (n-m)/(m+1) \rceil = \lfloor n/(m+1) \rfloor$. ♣

References

[1] Ajtai M., Σ_1^1-Formulae on Finite Structures, *Annals of Pure and Applied Logic*, **24**, pp 1–48, 1983.

[2] Beame, P., Cook, S.A. and Hoover, H.J. (1986), Log Depth Circuits for Division and

Related Problems, *SIAM Journal on Computing*, **15**, pp. 994–1003.

[3] Blum, N. (1984), A Boolean Function requiring $3n$ Network Size, *Theoretical Computer Science*, **28**, pp. 337–345.

[4] Boppana, R. and Sipser, M. (1990), The Complexity of Finite Functions, *in* "The Handbook of Theoretical Computer science", Vol. A, Elsevier Science Publishers.

[5] Bruck, J. and Smolensky, R. (1990), Polynomial Threshold Functions, AC^0 Functions and Spectral Norms, *in* "31st IEEE Symposium on Foundations of Computer Science", pp. 632–641.

[6] Chandra, A. and Stockmeyer, L. (1984), Constant Depth Reducibility, *SIAM Journal on Computing*, **13**, pp. 423–439.

[7] Dunne, P.E. (1988), "The Complexity of Boolean Functions", Academic Press.

[8] Furst, M., Saxe, J.B. and Sipser, M. (1984), Parity, Circuits, and the Polynomial Time Hierarchy, *Mathematical Systems Theory*, **17**, pp. 13–28.

[9] Hastad, J., (1986), Almost Optimal Lower Bounds for Small Depth Circuits, *in Proceedings of the 18th ACM Symposium on Theory of Computing*, pp 6–20.

[10] Hopcroft, J.E. and Ullman, J.D. (1979), "Introduction to Automata Theory, Languages and Computation", Addison–Wesley.

[11] Hinton, G.E. (1987), Connectionist Learning Procedures, CMU–CS–87–115.

[12] Hofmeister, T., Hohberg, W. and Köhling, S. (1991), Some notes on threshold circuit and multiplication in depth 4, *Information Processing Letters*, **39**, pp 219–225.

[13] Minsky, M. and Papert, S.A. (1969), "Perceptrons", MIT Press, Cambridge, Massachusetts.

[14] Mead, C. (1989), "Analog VLSI and Neural Systems", Addison–Wesley.

[15] Maass, W., Schnitger, G. and Sontag, E.D. (1991), On the Computational Power of Sigmoid versus Boolean Threshold Circuits, *in* "Proceedings of the 32nd Annual Symposium on Foundations of Computer Science ", pp 767–776.

[16] Parberry, I. and Schnitger, G. (1988), Parallel Computation with Threshold Functions, *Journal of Computer and System Science* **36** pp. 278–302.

[17] Paturi, R. and Saks, M.E., Approximating Threshold Circuits by Rational Functions, *Information and Computation, in press.*

[18] Raghavan, P., (1986), Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs, *in* "27th IEEE Symposium on Foundations of Computer Science", pp. 10–18.

[19] Red'kin, N.P. (1973), Proof of Minimality of Circuits consisting of Functional Elements, Problemy Kibernetika, **23**, pp. 83–102 (in Russian) (Translation: *Systems Theory Research*, **23**, pp. 85–103.)

[20] Razborov, A.A. (1986), Lower Bounds on the Size of Bounded Depth Networks over a Complete Basis with Logical Addition, *Mathematische Zametki* **41** pp. 598–607 (in Russian). English Translation in*Mathematical Notes of the Academy of Sciences of the USSR* **41**, pp. 333–338.

[21] Reif, J. (1987), On Threshold Circuits and Polynomial Computations, *in* "Proceedings of 2nd Structure in Complexity Theory Conference", pp. 118–125.

[22] Siu, K., Roychowdury, V., and Kailath, T. (1990). Computing with Optimal Size Threshold Circuits, Technical Report, Stanford University.

[23] Siu, K. -Y, Bruck, J. and Kailath, T. Depth–Efficient Neural Networks for Division and Related Problems, *IEEE Transactions on Information Theory*, to appear.

[24] Smolensky, R. (1987), Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity *in* "Proceedings of 19th Annual ACM symposium on Theory of Computing", pp. 77–82.

[25] Yao, A. C–C. (1989), Circuits and Local Computation, *in* "Proceedings of the 21st Annual ACM Symposium on Theory of Computing", pp. 186–196.

[26] Yao, A. C–C. (1990), On ACC and Threshold Circuits, *in* "31st IEEE Symposium on Foundations of Computer Science", pp. 619–627.