

## ALGEBRAIC TECHNIQUES FOR CONSTRUCTING MINIMAL WEIGHT THRESHOLD FUNCTIONS\*

VASKEN BOHOSSIAN<sup>†</sup> AND JEHOSSUA BRUCK<sup>†</sup>

**Abstract.** A linear threshold element computes a function that is a sign of a weighted sum of the input variables. The best known lower bounds on the size of threshold circuits are for depth-2 circuits with small (polynomial-size) weights. However, in general, the weights are arbitrary integers and can be of exponential size in the number of input variables. Namely, obtaining progress in lower bounds for threshold circuits seems to be related to understanding the role of large weights. In the present literature, a distinction is made between the two extreme cases of linear threshold functions with polynomial-size weights, as opposed to those with exponential-size weights. Our main contributions are in devising two novel methods for constructing threshold functions with minimal weights and filling up the gap between polynomial and exponential weight growth by further refining the separation. Namely, we prove that the class of linear threshold functions with polynomial-size weights can be divided into subclasses according to the degree of the polynomial. In fact, we prove a more general result—that there exists a minimal weight linear threshold function for any arbitrary number of inputs and any weight size.

**Key words.** threshold functions, computational complexity, neural networks

**AMS subject classifications.** 03D15, 68Q15, 68Q17, 92B20

**PII.** S0895480197326048

**1. Introduction.** The present paper focuses on the study of a single linear threshold gate with binary inputs and output as well as integer weights. Such a gate is mathematically described by a *linear threshold function*.

**DEFINITION 1.1** (linear threshold function). *A linear threshold function of  $n$  variables is a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  that can be written, for any  $\mathbf{x} \in \{0, 1\}^n$  and a fixed  $\mathbf{w} \in \mathbb{Z}^{n+1}$ , as*

$$f(\mathbf{x}) = \text{sgn}(F(\mathbf{x})) = \begin{cases} 1 & \text{for } F(\mathbf{x}) \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{where } F(\mathbf{x}) = \mathbf{w} \cdot (-1, \mathbf{x}) = -w_0 + \sum_{i=1}^n w_i x_i.$$

Although we could allow the weights,  $w_i$ , to be real numbers, it is known [Muroga 71] that one needs only  $O(n \log n)$  bits per weight, where  $n$  is the number of inputs. So in the rest of the paper, we will assume without loss of generality that all weights are integers. Also, notice that a linear threshold function can be implemented as

$$f : \{-1, 1\}^n \rightarrow \{0, 1\}.$$

We will address both the  $\{0, 1\}$  and the  $\{-1, 1\}$  representations.

Note that, given a function  $f$ , the weight vector  $\mathbf{w}$  is not unique (see Example 1 below).

---

\*Received by the editors August 13, 1997; accepted for publication (in revised form) July 22, 2002; published electronically January 3, 2003. Part of this work was presented in Proceedings of Neural Information Processing Systems 8, 1995, under the title *On Neural Networks with Minimal Weights*, by the authors of this paper.

<http://www.siam.org/journals/sidma/16-1/32604.html>

<sup>†</sup>California Institute of Technology, Mail Code 136-93, Pasadena, CA 91125 (vincent@paradise.caltech.edu, bruck@paradise.caltech.edu).

DEFINITION 1.2 (weight space). *Given a linear threshold function  $f$  we define  $\mathcal{W}$  as the set of all weights that satisfy Definition 1.1, that is,*

$$\mathcal{W} = \{\mathbf{w} \in \mathbb{Z}^n : \forall \mathbf{x} \in \{0, 1\}^n, \text{sgn}(\mathbf{w} \cdot (-1, \mathbf{x})) = f(\mathbf{x})\}.$$

Here follows a measure of the size of the weights.

DEFINITION 1.3 (minimal weight size). *We define the size of a weight vector as the sum of the absolute values of the weights. The minimal weight size of a linear threshold function is defined as*

$$S[f] = \min_{\mathbf{w} \in \mathcal{W}} \left( \sum_{i=0}^n |w_i| \right).$$

*The particular vector that achieves the minimum is called a minimal weight vector.*

Naturally,  $S[f]$  is a function of  $n$ .

**1.1. Motivation.** Why do we care about the size of the weights in threshold circuits?

Threshold circuits have been shown to be surprisingly powerful. For example, integer division can be implemented by a polynomial-size threshold circuit of constant depth [Beame 84], [Siu 93]. It is also proved in [Allender 89] that any function in  $AC^0$  can be computed by depth-3 majority circuits of quasi-polynomial size; in fact, it is true for all of  $ACC^0$  [Yao 90]. For a general survey about the representation of Boolean functions by threshold functions, see [Saks 93].

Given the foregoing impressive upper bounds, it is not surprising that we face difficulties in obtaining lower bounds. In fact, the best general lower bound for threshold circuits is the result that the inner-product mod 2 (IP2) requires exponential size for depth 2 [Hajnal 93]. However, this lower bound assumes that the circuits involve small weights, and it is not known whether IP2 can be computed by a depth-2 polynomial size threshold circuit with arbitrary weights. Obtaining progress in lower bounds for threshold circuits therefore seems to be related to understanding the role of large weights.

Hence, it is natural to ask how limited the computational power of the circuit is if one limits oneself to threshold elements with only “small” growth in the size of the coefficients. It has been shown [Anthony 93], [Hampson 86], [Hastad 94], [Myhill 61], [Muroga 71], [Siu 91] that there exist linear threshold functions that can be implemented by a single threshold element with exponentially growing weights,  $S[f] \sim 2^n$ , but cannot be implemented by a threshold element with smaller polynomially growing weights,  $S[f] \sim n^d$ ,  $d$  constant. In light of that result, the above question was dealt with by defining a class within the set of linear threshold functions, the class of functions with “small” (i.e., polynomially growing) weights [Siu 91]. Most of the recent research focused on the power of circuits with small weights, relative to circuits with arbitrary weights [Goldmann 92], [Goldmann 98]. In particular, it showed that increasing the depth of the circuit by one is sufficient to reduce all the weights to be of polynomial size. However, these impressive upper bounds were still not helpful in improving the lower bounds.

In this paper we take a different approach. Rather than dealing with circuits we focus on the modest task of studying a single threshold gate. The main contribution of the present paper is to further refine the division of small versus arbitrary weights. We separate the set of functions with small weights into classes indexed by  $d$ , the degree of polynomial growth, and show that all of them are nonempty. In particular, we develop

a technique for proving that a weight vector is minimal. We use that technique to construct a function of size  $S[f] = s$  for an arbitrary  $s$ . The natural future direction is to extend our techniques for constructing minimal weight threshold functions to circuits of depth 2. This might help in defining explicit functions that cannot be computed by depth-2, polynomial size threshold circuits with specific weight size.

**1.2. Organization.** Here follows a brief outline of the rest of the paper. In section 2 we show some of the difficulties one faces when minimizing the weights as well as how they are affected by the choice of input domain. In section 3 we consider functions defined over  $\{-1, 1\}$ . We limit ourselves to functions with no threshold (generalized majority function), and we show how to construct such functions with minimal weights. In section 4 we present another way of constructing minimal functions that allows us to deal with any threshold function defined over  $\{0, 1\}$ .

**2. Preliminaries and examples.** In this section we illustrate some of the difficulties one faces when trying to minimize the weights of a threshold function. We also show how the input domain (i.e.,  $\{0, 1\}$  versus  $\{-1, 1\}$ ) affects the size of the weights. See [Krause 95] for related results.

**2.1. Minimizing the weights.** The main difficulty in analyzing the size of the weights of a threshold element is due to the fact that a single linear threshold function can be implemented by different sets of weights as shown in the following example.

EXAMPLE 1 (a threshold function with minimal weights). *Let us consider the following two sets of weights (weight vectors):*

$$\mathbf{w}_1 = (4 \ 1 \ 2 \ 5), \quad F_1(\mathbf{x}) = -4 + x_1 + 2x_2 + 5x_3,$$

$$\mathbf{w}_2 = (8 \ 2 \ 4 \ 10), \quad F_2(\mathbf{x}) = -8 + 2x_1 + 4x_2 + 10x_3.$$

*They both implement the same threshold function*

$$f(\mathbf{x}) = \text{sgn}(F_2(\mathbf{x})) = \text{sgn}(2F_1(\mathbf{x})) = \text{sgn}(F_1(\mathbf{x})).$$

*A closer look reveals that  $f(\mathbf{x}) = \text{sgn}(-1 + x_3)$ , implying that none of the above weight vectors has minimal size. Indeed, the minimal one is  $\mathbf{w}_3 = (1 \ 0 \ 0 \ 1)$  and  $S[f] = 2$ .*

To determine if a given set of weights is minimal is in general a difficult problem [Willis 63]. Our technique consists of constructing weight vectors whose minimality is easily established. We then show how to modify them, while keeping them minimal, in order to get to a larger set of functions.

**2.2.  $\{0, 1\}$  versus  $\{-1, 1\}$ .** Suppose we implement the same function over  $\{0, 1\}$  and over  $\{-1, 1\}$ . How are the weights affected? Let us look at an example.

EXAMPLE 2 (the OR function).

1. Let  $x_i \in \{0, 1\}$ ,

$$OR(x_1, \dots, x_n) = \text{sgn}(-1 + x_1 + \dots + x_n).$$

*The size of the weights is  $s = n + 1$ . Those weights are minimal.*

*Proof.* The weights are integers. Reducing their size implies resetting one or more of them to 0, which will violate the definition of OR.  $\square$

2. Now let  $x_i \in \{-1, 1\}$ ,

$$OR(x_1, \dots, x_n) = \text{sgn}(n - 2 + x_1 + \dots + x_n).$$

*The size of the weights is  $s = 2n - 2$ . Those weights are minimal as well.*

*Proof.* Any weights that implement *OR* have to be positive. Suppose there exist weights of size  $s' < 2n - 2$ . No weight can be 0, so  $\sum_1^n w' \geq n$ , implying that the threshold  $-w_0 < (2n - 2) - n = n - 2$ . Let  $w'_i$  be the smallest weight. Set  $x_i = 1$  and all other inputs to  $-1$ .  $\sum_1^n w' < -w_i(n - 2)$  so that  $F(\mathbf{x}) < 0$  violating the definition of *OR*.  $\square$

It appears from this example that the  $\{0, 1\}$  implementation has smaller weight size than the  $\{-1, 1\}$  representation. Is that true in general?

EXAMPLE 3 (the majority (*MAJ*) function). *Let the number of variables,  $n$ , be odd. The majority function outputs true if more than half of its inputs are true.*

1. Let  $x_i \in \{0, 1\}$ ,

$$MAJ(x_1, \dots, x_n) = \text{sgn} \left( -\frac{n+1}{2} + x_1 + \dots + x_n \right).$$

The size of the weights is  $s = \frac{3n+1}{2}$ . They can be shown to be minimal by a proof similar to case 2 in Example 2.

2. Now let  $x_i \in \{-1, 1\}$ ,

$$MAJ(x_1, \dots, x_n) = \text{sgn}(x_1 + \dots + x_n).$$

Those weights are minimal, since reducing them would imply resetting one or more of them to 0, which will violate the definition of *MAJ*. The size of the weights is  $s = n$ .

Example 3 shows that in general we cannot tell which implementation  $\{0, 1\}$  or  $\{-1, 1\}$  will produce a function with smaller weights. However, the weight sizes for each of those functions are always within a constant factor of each other, since the  $\{0, 1\}$  weights are related to a set of  $\{-1, 1\}$  weights by a simple linear transformation.

**3. Generalized majority function over  $\{-1, 1\}$ .** In this section we study the following model:

$$f : \{-1, 1\} \rightarrow \{0, 1\},$$

$$f(X) = \text{sgn} \left( \sum_1^n w_i x_i \right).$$

Notice that there is no threshold; we are looking at a majority function with arbitrary weights. We address the problem of constructing functions with minimal weights. In particular, our goal is that for a given number of inputs  $n$  and size  $s$  we find a function.

**3.1. Mathematical setting.** We are interested in constructing functions for which the minimal weight is easily determined. Finding the minimal weight involves a search, and we are therefore interested in finding functions with constrained weight spaces. The following tools allow us to put constraints on  $\mathcal{W}$ .

DEFINITION 3.1 (root space of a Boolean function). *A vector  $\mathbf{v} \in \{-1, 1\}^n$  such that  $f(\mathbf{v}) = f(-\mathbf{v})$  is called a root of  $f$ . We define the root space,  $\mathcal{R}$ , as the set of all roots of  $f$ . Note that a vector  $\mathbf{v}$  is a root if and only if  $\sum w_i v_i = 0$ .*

DEFINITION 3.2 (root generator matrix). *For a given weight vector  $\mathbf{w} \in \mathcal{W}$  and a root  $\mathbf{v} \in \mathcal{R}$ , the root generator matrix,  $G = (g_{ij})$ , is a  $(k \times n)$ -matrix, with entries in  $\{-1, 0, 1\}$ , whose rows  $\mathbf{g}$  are orthogonal to  $\mathbf{w}$  and equal to  $\mathbf{v}$  at all nonzero coordinates, namely,*

1.  $G\mathbf{w}^T = \mathbf{0}$ ;

2.  $g_{ij} = 0$  or  $g_{ij} = v_j$  for all  $i$  and  $j$ .

The root generator matrix is used to generate linearly independent root vectors for  $f$ . Each row of  $G$  corresponds to a new root vector.

EXAMPLE 4 (root generator matrix). *Suppose that we are given a linear threshold function specified by a weight vector  $\mathbf{w} = (1, 1, 2, 4, 1, 1, 2, 4)$ . By inspection we determine one root  $\mathbf{v} = (1, 1, 1, 1, -1, -1, -1, -1)$ . Notice that  $w_1 + w_2 - w_7 = 0$  which can be written as  $\mathbf{g} \cdot \mathbf{w} = 0$ , where  $\mathbf{g} = (1, 1, 0, 0, 0, 0, -1, 0)$  is a row of  $G$ . Set  $\mathbf{r} = \mathbf{v} - 2\mathbf{g}$ . Since  $\mathbf{g}$  is equal to  $\mathbf{v}$  at all nonzero coordinates,  $\mathbf{r} \in \{-1, 1\}^n$ . Also  $\mathbf{r} \cdot \mathbf{w} = \mathbf{v} \cdot \mathbf{w} - 2\mathbf{g} \cdot \mathbf{w} = 0$ . We have generated a new root:  $\mathbf{r} = (-1, -1, 1, 1, -1, -1, 1, -1)$ .*

LEMMA 3.3 (orthogonality of  $G$  and  $\mathcal{W}$ ). *For a given weight vector  $\mathbf{w} \in \mathcal{W}$  and a root  $\mathbf{v} \in \mathcal{R}$ ,  $G\mathbf{u}^T = \mathbf{0}$  holds for any weight vector  $\mathbf{u} \in \mathcal{W}$ .*

*Proof.* For an arbitrary  $\mathbf{u} \in \mathcal{W}$  and an arbitrary row,  $\mathbf{g}_i$ , of  $G$ , let  $\mathbf{v}' = \mathbf{v} - 2\mathbf{g}_i$ . By definition of  $\mathbf{g}_i$ ,  $\mathbf{v}' \in \{-1, 1\}^n$  and  $\mathbf{v}' \cdot \mathbf{w} = 0$ . This implies  $f(\mathbf{v}') = f(-\mathbf{v}')$ :  $\mathbf{v}'$  is a root of  $f$ . For any weight vector  $\mathbf{u} \in \mathcal{W}$ ,  $\text{sgn}(\mathbf{u} \cdot \mathbf{v}') = \text{sgn}(-\mathbf{u} \cdot \mathbf{v}')$ . Therefore  $\mathbf{u} \cdot (\mathbf{v} - 2\mathbf{g}_i) = 0$  and finally, since  $\mathbf{v} \cdot \mathbf{u} = 0$ , we get  $\mathbf{u} \cdot \mathbf{g}_i = 0$ .  $\square$

LEMMA 3.4 (minimality). *For a given weight vector  $\mathbf{w} \in \mathcal{W}$  and a root  $\mathbf{v} \in \mathcal{R}$  if  $\text{rank}(G) = n - 1$  (i.e.,  $G$  has  $n - 1$  independent rows) and  $|w_i| = 1$  for some  $i$ , then  $\mathbf{w}$  is the minimal weight vector.*

*Proof.* From Lemma 3.3 any weight vector  $\mathbf{u}$  satisfies  $G\mathbf{u}^T = \mathbf{0}$ .  $\text{rank}(G) = n - 1$  implies that  $\dim(\mathcal{W}) = 1$ ; i.e., all possible weight vectors are integer multiples of each other. Since  $|w_i| = 1$ , all vectors are of the form  $\mathbf{u} = k\mathbf{w}$  for  $k \geq 1$ . Therefore  $\mathbf{w}$  has the smallest size.  $\square$

We complete Example 4 with an application of Lemma 3.4.

EXAMPLE 5 (minimality). *Given the following weights  $\mathbf{w}$  and a root  $\mathbf{v}$ ,*

$$\mathbf{w} = (1, 1, 2, 4, 1, 1, 2, 4), \quad \mathbf{v} = (1, 1, 1, 1, -1, -1, -1, -1),$$

we can construct  $G$ :

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

It is easy to verify that  $\text{rank}(G) = n - 1 = 7$  and therefore, by Lemma 3.4,  $\mathbf{w}$  is minimal and  $S[f] = 16$ .

**3.2. Weight vectors.** In Example 5 we saw how, given a weight vector, one can show that it is minimal. In this section we present an example of a linear threshold function with minimal weight size, with an arbitrary number of input variables.

We would like to construct a weight vector and show that it is minimal. Let the number of inputs,  $n$ , be even. Let  $\mathbf{w}$  consist of two identical blocks :

$$\mathbf{w} = (w_1, w_2, \dots, w_{n/2}, w_1, w_2, \dots, w_{n/2}).$$

Clearly,  $\mathbf{v} = (1, 1, \dots, 1, -1, -1, \dots, -1)$  is a root and  $G$  is the corresponding genera-

tor matrix.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \end{pmatrix}$$

**3.3. Construction.** The following theorem states that given an integer  $s$  and a number of variables  $n$  there exists a function of  $n$  variables and minimal weight size  $s$ .

**THEOREM 3.5 (main result).** *For any pair  $(s, n)$ , with both  $s$  and  $n$  even, satisfying  $n \leq s \leq 2^{\frac{n}{2}}$ , there exists a linear threshold function of  $n$  variables,  $f$ , with minimal weight size  $S[f] = s$ .*

*Proof.* Given a pair  $(s, n)$  that satisfies the above conditions we first construct a weight vector  $\mathbf{w}$  that satisfies  $\sum_{i=1}^n |w_i| = s$ ; then we show that it is the minimal weight vector of the function  $f(x) = \text{sgn}(\mathbf{w} \cdot \mathbf{x})$ . The proof is shown only for  $n$  even.

*Construction.*

1. Define  $(a_1, a_2, \dots, a_{n/2}) = (1, 1, \dots, 1)$ .
2. If  $\sum_{i=1}^{n/2} a_i < s/2$ , then increase by one the smallest  $a_i$  such that  $a_i < 2^{i-2}$ . (In the case of a tie take the  $a_i$  with smallest index  $i$ ).
3. Repeat the previous step until  $\sum_{i=1}^{n/2} a_i = s/2$  or  $(a_1, a_2, \dots, a_{n/2}) = (1, 1, 2, 4, \dots, 2^{\frac{n}{2}-2})$ .
4. Set  $\mathbf{w} = (a_1, a_2, \dots, a_{n/2}, a_1, a_2, \dots, a_{n/2})$ .

Because we increase the size by one unit at a time the algorithm will converge to the desired result for any integer  $s$  that satisfies  $n \leq s \leq 2^{\frac{n}{2}}$ . We have a construction for any valid  $(s, n)$  pair. Let us show that  $\mathbf{w}$  is minimal.

*Minimality.* Given that  $\mathbf{w} = (a_1, a_2, \dots, a_{n/2}, a_1, a_2, \dots, a_{n/2})$  we find a root  $\mathbf{v}$ ,

$$\mathbf{v} = (1, 1, \dots, 1, -1, -1, \dots, -1),$$

and  $n/2$  rows of the generator matrix  $G$  corresponding to the equations  $w_i = w_{i+\frac{n}{2}}$ . To form additional rows note that the first  $k$   $a_i$ 's are powers of two (where  $k$  depends on  $s$  and  $n$ ). Those can be written as  $a_i = \sum_{j=1}^{i-1} a_j$  and generate  $k - 1$  rows. And finally note that all other  $a_i, i > k$ , are smaller than  $2^{k+1}$ . Hence, they can be written as a binary expansion  $a_i = \sum_{j=1}^k \alpha_{ij} a_j$ , where  $\alpha_{ij} \in \{0, 1\}$ . There are  $\frac{n}{2} - k$  such weights.  $G$  has a total of  $n - 1$  independent rows.  $\text{rank}(G) = n - 1$  and  $w_1 = 1$ ; therefore, by Lemma 3.4,  $\mathbf{w}$  is minimal and  $S[f] = s$ .  $\square$

**EXAMPLE 6** (a function of 10 variables and size 26). *We start with  $\mathbf{a} = (1, 1, 1, 1, 1)$ . We iterate  $(1, 1, 2, 1, 1)$ ,  $(1, 1, 2, 2, 1)$ ,  $(1, 1, 2, 2, 2)$ ,  $(1, 1, 2, 3, 2)$ ,  $(1, 1, 2, 3, 3)$ ,  $(1, 1, 2, 4, 3)$ ,  $(1, 1, 2, 4, 4)$ , and finally the algorithm converges to  $\mathbf{a} = (1, 1, 2, 4, 5)$ . We claim that  $\mathbf{w} = (\mathbf{a}, \mathbf{a}) = (1, 1, 2, 4, 5, 1, 1, 2, 4, 5)$  is minimal. Indeed,*

$\mathbf{v} = (1, 1, 1, 1, 1, -1, -1, -1, -1, -1)$  and

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

is a matrix of rank 9.

EXAMPLE 7 (functions with polynomial size). *This example shows an application of Theorem 3.5. We define  $\widehat{LT}^{(d)}$  as the set of linear threshold functions for which  $S[f] \leq n^d$ . The theorem states that for any even  $n$  there exists a function  $f$  of  $n$  variables and minimum weight  $S[f] = n^d$ . The implication is that for all  $d$ ,  $\widehat{LT}^{(d-1)}$  is a proper subset of  $\widehat{LT}^{(d)}$ .*

**4. Arbitrary threshold function over  $\{0, 1\}$ .** In this section we present a different technique for constructing threshold functions with minimal weights. It allows us to construct functions with any weight size and number of variables. We consider functions with input domain  $\{0, 1\}$ , but, as mentioned below, the argument holds for an arbitrary input space  $\{a, b\}$ .

**4.1. Approach.** The method we use is based on a result from [Willis 63]. We assume, without loss of generality, that the weights are strictly positive integers. Our goal is to minimize  $s = \sum_0^n |w_i| = \sum_0^n w_i$ . We know from [Muroga 71] that any other weights,  $\mathbf{u}$ , implementing the same function have to be strictly positive. We will show that under certain conditions on  $\mathbf{w}$ ,  $\sum_0^n w_i \leq \sum_0^n u_i$  for any  $\mathbf{u}$ .

Consider input vectors  $\mathbf{x}$  and  $\mathbf{y}$  for which the following equations hold:

$$F(\mathbf{x}) = -w_0 + \sum_1^n w_i x_i = 0, \quad F(\mathbf{y}) = -w_0 + \sum_1^n w_i y_i = -1.$$

Let them define the rows of a matrix that we call  $A$ . Using  $p$   $\mathbf{x}$ -type and  $q$   $\mathbf{y}$ -type vectors we get

$$A = \begin{pmatrix} -1 & \mathbf{x}^{(1)} \\ -1 & \mathbf{x}^{(2)} \\ \vdots & \vdots \\ -1 & \mathbf{x}^{(p)} \\ 1 & -\mathbf{y}^{(1)} \\ 1 & -\mathbf{y}^{(2)} \\ \vdots & \vdots \\ 1 & -\mathbf{y}^{(q)} \end{pmatrix} = \begin{pmatrix} -1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ -1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ -1 & x_1^{(p)} & x_2^{(p)} & \cdots & x_n^{(p)} \\ 1 & -y_1^{(1)} & -y_2^{(1)} & \cdots & -y_n^{(1)} \\ 1 & -y_1^{(2)} & -y_2^{(2)} & \cdots & -y_n^{(2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & -y_1^{(q)} & -y_2^{(q)} & \cdots & -y_n^{(q)} \end{pmatrix}$$

EXAMPLE 8 (the matrix  $A$ ). *Suppose we are given the following weights:*

$$\mathbf{w} = (16 \ 1 \ 2 \ 4 \ 8 \ 1 \ 2 \ 4 \ 8).$$

Our goal is to show they are minimal. We need to first construct the matrix  $A$ . Here follows a candidate:

$$A = \begin{pmatrix} -1 & \mathbf{x}^{(1)} \\ -1 & \mathbf{x}^{(2)} \\ 1 & -\mathbf{y}^{(1)} \\ 1 & -\mathbf{y}^{(2)} \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

There are many possible choices for  $A$ , depending on which of the  $\mathbf{x}$ - and  $\mathbf{y}$ -type vectors are used. The one shown above is not a good one, as we will see. Additional  $\mathbf{x}$ -type vectors need to be included in the construction of  $A$  in order to satisfy the requirements in Theorem 4.1.

**THEOREM 4.1** (condition for minimality). *Given a weight vector  $\mathbf{w}$ , we construct  $A$  as described above. If there is a nonnegative row vector  $\mathbf{a}$  (that is,  $a_i \geq 0$  for all  $i$ ) such that  $A$  satisfies*

$$\mathbf{a}A = \overbrace{(1 \dots 1)}^{n+1},$$

the weight vector  $\mathbf{w}$  is minimal.

*Proof.* By definition of the  $\mathbf{x}$ 's and the  $\mathbf{y}$ 's the matrix  $A$  satisfies

$$(4.1) \quad A \cdot (w_0 \ w_1 \ w_2, \dots, w_n)^T = \overbrace{(0 \ 0 \ \dots \ 0 \ 0)}^p \overbrace{(1 \ 1 \ \dots \ 1 \ 1)}^q)^T.$$

Because  $\text{sgn}(0) = 1$  and  $\text{sgn}(-1) = 0$ , any other weight vector,  $\mathbf{u}$ , implementing the same function has to verify the above equalities with “ $\geq$ ” instead of “ $=$ ”:

$$(4.2) \quad A \cdot (u_0 \ u_1 \ u_2, \dots, u_n)^T \geq \overbrace{(0 \ 0 \ \dots \ 0 \ 0)}^p \overbrace{(1 \ 1 \ \dots \ 1 \ 1)}^q)^T.$$

Let  $\mathbf{v} = \mathbf{u} - \mathbf{w}$ , and subtracting equations (4.1) from inequalities (4.2) we get

$$(4.3) \quad A \cdot (v_0 \ v_1 \ v_2, \dots, v_n)^T \geq \overbrace{(0 \ 0 \ \dots \ 0 \ 0)}^{p+q})^T$$

Now suppose  $A$  is such that

$$(4.4) \quad (a_0 \ a_1, \dots, a_{p+q-1}) \cdot A = \overbrace{(1 \ 1 \ \dots \ 1 \ 1)}^{n+1}$$

Where the  $a_i$  are strictly positive. We multiply inequalities (4.3) by  $\mathbf{a}$  from the left and get

$$(a_0 \ a_1, \dots, a_{p+q-1}) \cdot A \cdot (v_0 \ v_1 \ v_2, \dots, v_n)^T \geq (a_0 \ a_1, \dots, a_{p+q-1}) \cdot \overbrace{(0 \ 0 \ \dots \ 0 \ 0)}^{p+q})^T,$$

$$\overbrace{(1 \ 1 \ \dots \ 1 \ 1)}^{n+1} \cdot (v_0 \ v_1 \ v_2, \dots, v_n)^T \geq 0,$$

$$\sum_0^n v_i \geq 0.$$

Since  $w_i \geq 0$ ,  $u_i \geq 0$  for all  $i = 0, \dots, n$  we know that  $\sum_0^n u_i \geq \sum_0^n w_i$ .  $\square$

Notice that nowhere in the proof did we use the fact that the input domain is  $\{0, 1\}$ . Indeed, the above proof is valid for any input domain  $\{a, b\}$ . As you can see the proof relies on constructing  $A$  so that (4.4) holds. To construct  $A$  we need appropriate  $\mathbf{x}$ 's and  $\mathbf{y}$ 's which in turn depend on the choice  $\mathbf{w}$ .

**4.2. Basic construction.** In this section we introduce  $\mathbf{w}$ , the weight vector for the general construction, and prove it is minimal by finding an appropriate matrix  $A$ . We use a construction similar to the one in section 3, based on powers of two.

*Construction.* Given a pair  $(s, n)$ , where  $n + 1 \leq s \leq 3 * 2^{\lfloor \frac{n}{2} \rfloor} - 2$ , and  $s = 3m - l$ , with  $l \in \{0, 1, 2\}$ , we have the following:

1. Define  $s' = 3m - 2$  and  $n' = n - (s - s')$ .
2. Define  $k$  as the largest integer such as  $s' > 3 * 2^{k-1} - 2$ .
3. Define  $s_0 = \frac{1}{3}(s' - 3 * 2^{k-1} + 2)$ .
4. Set

$$(w_0, w_1, \dots, w_{2k}) = (2^{k-1} + s_0, 1, 2, 4, \dots, 2^{k-2}, s_0, 1, 2, 4, \dots, 2^{k-2}, s_0).$$

At this point the size of  $\mathbf{w}$  is  $s'$ . In the following two steps additional weights are added in order to get to the desired number of variables  $n$  and the exact weight size  $s$ .

5. For every  $w_i$  with  $i \in \{2k + 1, \dots, n'\}$  let  $w_i = 1$  and subtract 1 from the largest weight  $w_j$ ,  $j \in \{1, \dots, 2k\}$ . In case of a tie select the weight with largest index.

6. For every  $w_i$  with  $i \in \{n' + 1, \dots, n\}$  let  $w_i = 1$ . No subtraction is needed. (Notice that  $n - n' \in \{0, 1, 2\}$ .)

Let us look at two examples.

EXAMPLE 9 (a function of 12 variables and size 35).  $s = 35 = 3 * 12 - 1$ , therefore  $s' = 34$ ,  $n' = 11$ ,  $k = 4$ ,  $s_0 = 4$ . The weight iterations are

$$\mathbf{w} = (12, 1, 2, 4, 4, 1, 2, 4, 4),$$

$$\mathbf{w} = (12, 1, 2, 4, 4, 1, 2, 4, 3, 1),$$

$$\mathbf{w} = (12, 1, 2, 4, 4, 1, 2, 3, 3, 1, 1),$$

$$\mathbf{w} = (12, 1, 2, 4, 3, 1, 2, 3, 3, 1, 1, 1),$$

$$\mathbf{w} = (12, 1, 2, 4, 3, 1, 2, 3, 3, 1, 1, 1, 1).$$

EXAMPLE 10 (base case:  $n = 2k$ ,  $s_0 = 2^{k-1}$ ). Let us show that the weights of Example 8 are minimal. Using the above notation  $n = 8$ ,  $s_0 = 8$ , and  $k = 4$ .

$$\mathbf{w} = (16, 1, 2, 4, 8, 1, 2, 4, 8).$$

Here follow the  $X$ - and  $Y$ -type rows for  $A$ :

$$\left\{ \begin{array}{cccccccc} -1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array} \right\} \text{sum}X_1 = (-2, 2, 1, 1, 1, 2, 1, 1, 1)$$

$$\left\{ \begin{array}{cccccccc} -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array} \right\} \text{sum}X_2 = (-2, 0, 2, 1, 1, 0, 2, 1, 1)$$

$$\left\{ \begin{array}{cccccccc} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array} \right\} \text{sum}X_3 = (-2, 0, 0, 2, 1, 0, 0, 2, 1)$$

$$\left\{ \begin{array}{cccccccc} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right\} \text{sum}X_4 = (-2, 0, 0, 0, 2, 0, 0, 0, 2)$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\text{sum} Y_1 = (2 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1)}$$

We replicate rows and add them in order to get to the all 1 vector. Only the first five columns are shown.

$$\begin{pmatrix} -2 & 2 & 1 & 1 & 1 \\ -2 & 0 & 2 & 1 & 1 \\ -2 & 0 & 0 & 2 & 1 \\ -2 & 0 & 0 & 0 & 2 \\ 2 & -1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -16 & 16 & 8 & 8 & 8 \\ -8 & 0 & 8 & 4 & 4 \\ -4 & 0 & 0 & 4 & 2 \\ -2 & 0 & 0 & 0 & 2 \\ 2 & -1 & -1 & -1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} -24 & 24 & 12 & 12 & 12 \\ -12 & 0 & 12 & 6 & 6 \\ -6 & 0 & 0 & 6 & 3 \\ -3 & 0 & 0 & 0 & 3 \\ 46 & -23 & -23 & -23 & -23 \end{pmatrix}$$

The last matrix was obtained by multiplying the first four rows by 3/2, and the last row by 23. Its rows add up to the all 1 vector. Using the notation of Theorem 4.1, given the matrix  $A$ , as defined above,

$$\mathbf{a} = \left( 12, 12, 6, 6, 3, 3, \frac{3}{2}, \frac{3}{2}, 23, 23 \right).$$

**THEOREM 4.2** (minimality of the construction). *For any pair  $(s, n)$  satisfying*

$$n + 1 \leq s \leq 3 * 2^{\lfloor \frac{n}{2} \rfloor} - 2$$

*one can construct an  $n$ -variable threshold function with minimal weights of size  $s$ .*

We will first show that steps 1–4 of the construction produce minimal weights. The second part of the proof focuses on adding a padding of unit weights in order to achieve the desired number of variables  $n$ .

*Proof* (part 1: no padding). As of step 4 of the construction,

$$\mathbf{w} = (2^{k-1} + s_0, 1, 2, 4, \dots, 2^{k-2}, s_0, 1, 2, 4, \dots, 2^{k-2}, s_0).$$

We are going to construct  $A$ , show that it satisfies  $\mathbf{a}A = \mathbf{1}$ , and apply Theorem 4.1. Only two  $Y$ -type vectors are needed for the construction of  $A$ :

$$\begin{pmatrix} 1 & 0 & \dots & 0 & -1 & \dots & -1 \\ 1 & -1 & \dots & -1 & 0 & \dots & 0 \end{pmatrix}$$

They add up to  $(2 \ -1 \ \dots \ -1)$ . The  $X$ -type vectors, summed two by two, produce the following matrix (only the first  $k+1$  columns are shown, the remaining  $k$  columns are identical to columns 2 to  $k+1$ ):

$$A_X = \begin{pmatrix} -2 & 2 & 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ -2 & 0 & 2 & 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ -2 & 0 & 0 & 2 & 1 & 1 & \dots & 1 & 1 & 1 \\ -2 & 0 & 0 & 0 & 2 & 1 & \dots & 1 & 1 & 1 \\ \vdots & & & & & & & & & \vdots \\ -2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 2 & 1 \\ -2 & t_0 & t_1 & t_2 & t_3 & t_4 & \dots & t_{k-2} & t_{k-1} & 2 \end{pmatrix}$$

The  $t_i$ , ( $t_i \in \{0, 1\}$ ), are the binary expansion of  $2^{k-1} - s_0$ ,

$$2^{k-1} - s_0 = \sum_{i=0}^{k-1} 2^i t_i.$$

One can verify that the last row is indeed the sum of two  $X$ -type vectors. Given the above choice of  $A$  we need to compute the  $a_i$  in the following set of equations:

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{k-2} \\ a_{k-1} \\ a_k \end{pmatrix}^T \begin{pmatrix} -2 & 2 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ -2 & 0 & 2 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ -2 & 0 & 0 & 2 & 1 & \cdots & 1 & 1 & 1 & 0 \\ \vdots & & & & & & & & & \vdots \\ -2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 2 & 1 \\ -2 & t_0 & t_1 & t_2 & t_3 & \cdots & t_{k-4} & t_{k-3} & t_{k-2} & 2 \\ 2 & -1 & -1 & -1 & -1 & \cdots & -1 & -1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{pmatrix}^T$$

It is possible to get an explicit formula for  $a_i$  as a function of the  $t_i$ , but it is not necessary. All that is needed is to show that the  $a_i$  are nonnegative. Consider the following set of equations:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{k-2} \\ 1 \end{pmatrix}^T \begin{pmatrix} 2 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ 0 & 0 & 2 & 1 & \cdots & 1 & 1 & 1 & 0 \\ \vdots & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 2 & 1 \\ t_0 & t_1 & t_2 & t_3 & \cdots & t_{k-4} & t_{k-3} & t_{k-2} & 2 \end{pmatrix} = \begin{pmatrix} h \\ h \\ h \\ \vdots \\ h \\ h \end{pmatrix}^T$$

Notice that  $b_{k-1}$  is set to 1. This is a system of  $k$  equations with  $k$  unknowns. Solving for the  $b_i$  and  $h$  we get  $2b_0 = h - t_0$ ,  $2b_i = h - t_i - \sum_{j=0}^{i-1} b_j$ , and  $h = 2 + \sum_{j=0}^{k-2} b_j$ . The last two equations can be combined into  $b_{k-2} = 2 - t_{k-2}$ . Using the recurrence formula,  $2b_i = b_{i-1} - (t_i - t_{i-1})$ , the remaining values are obtained:

$$\begin{aligned} b_{k-3} &= 4 - t_{k-2} - t_{k-3}, \\ b_{k-4} &= 8 - 2t_{k-2} - t_{k-3} - t_{k-4}, \\ b_{k-5} &= 16 - 4t_{k-2} - 2t_{k-3} - t_{k-4} - t_{k-5}, \\ &\vdots \\ b_0 &= 2^{k-1} - 2^{k-3}t_{k-2} - 2^{k-4}t_{k-3} - \cdots - t_1 - t_0, \\ h &= 2^k - 2^{k-2}t_{k-2} - 2^{k-3}t_{k-3} - \cdots - 2t_1 - t_0. \end{aligned}$$

Notice that all the  $b_i$  and  $h$  are nonnegative because  $t_i \in \{0, 1\}$ .

Let  $a_i = \alpha b_i$  for  $i = 0, \dots, k-1$ . We need to show that  $\alpha$  and  $a_k$  are nonnegative. Going back to  $\mathbf{aA} = \mathbf{1}$ , the remaining two equations are

$$2a_k - 2\alpha \sum_{i=0}^{k-1} b_i = 1 \quad \text{and} \quad \alpha h - a_k = 1.$$

Solving for  $\alpha$  and  $a_k$  we get  $\alpha = 3/2(h - \sum b_i)$  and  $a_k = (h + 2 \sum b_i)/2(h - \sum b_i)$ . Substituting for  $h = 1 + \sum_{i=0}^{k-1} b_i$  we get

$$\alpha = \frac{3}{2} \quad \text{and} \quad a_k = \frac{1}{2} + \frac{3}{2} \sum_{i=0}^{k-1} b_i.$$

Since all  $b_i$  are nonnegative,  $a_k \geq 0$ , which completes the proof.  $\square$

*Proof* (part 2: extra padding of ones). The second part of the proof will focus on steps 5 and 6 of the construction. The following two lemmas are needed.

LEMMA 4.3 (splitting a weight). *Let  $\mathbf{w} = (w_0, w_1, \dots, w_n)$  be minimal. Then  $\tilde{\mathbf{w}} = (w_0, w_1, \dots, w_{n-1}, a, b)$ , where  $a + b = w_n$  is also minimal.*

*Proof.* Construct the matrix  $A$  while duplicating the last column.  $\square$

LEMMA 4.4 (adding an input with unit weight). *If  $\mathbf{w} = (w_0, w_1, \dots, w_n)$  is minimal, and  $w_0 > 0$ , then  $\tilde{\mathbf{w}} = (w_0, w_1, w_2, w_3, \dots, w_{n+1})$ , where  $w_{n+1} = 1$ , is also minimal.*

*Proof.* Suppose it is not minimal, implying there exists a better choice for  $\tilde{\mathbf{w}}$ ; let us call it  $\mathbf{w}'$ . There are two possibilities. Either  $w'_{n+1} = 0$  or some of the  $w'_i$  for  $i < n + 1$  is smaller than the corresponding  $w_i$ . In the latter case, we set  $x_{n+1} = 0$  and obtain the original function implemented with smaller weights, contradicting the hypothesis. Now suppose  $w'_{n+1} = 0$ , implying that  $\tilde{f}$  does not depend on  $x_{n+1}$ . That in turn implies  $\sum_0^n w_i x_i \geq 0$  or  $\sum_0^n w_i x_i \leq -2$  for all inputs  $X$ . We can reduce  $w_0$  by 1, implying the original function was not minimal.  $\square$

In step 5 of the construction, starting with the following weights,

$$\mathbf{w} = (2^{k-1} + s_0, 1, 2, 4, \dots, 2^{k-2}, s_0, 1, 2, 4, \dots, 2^{k-2}, s_0).$$

Lemma 4.3 is used to increase the number of weights while keeping their size constant. In step 6, a final adjustment is done for the cases  $s = 3m - 1$  and  $s = 3m$ . Applying Lemma 4.4, an additional one, or two, unit weights are added to achieve the desired pair  $(s, n)$ . The smallest weights achievable are  $\mathbf{w} = (1 \dots 1)$ . Any smaller weights will produce a function of less variables. The upper bound  $3 * 2^{\lfloor \frac{n}{2} \rfloor} - 2$  is achieved when  $s_0 = 2^{k-1}$  and there is no padding of ones.  $\square$

EXAMPLE 11 (functions with polynomial size). *Just as in section 3, we can define  $\widehat{LT}^{(d)}$  as the set of linear threshold functions for which  $S[f] \leq n^d$ . Theorem 4.2 states that for any  $n$  there exists a function  $f$  of  $n$  variables and minimum weight  $S[f] = n^d$ . The implication is that for all  $d$ ,  $\widehat{LT}^{(d-1)}$  is a proper subset of  $\widehat{LT}^{(d)}$ .*

**5. Conclusions.** We presented two techniques for constructing minimal weight threshold functions of arbitrary weight size and number of inputs. We considered both the  $\{0, 1\}$  and  $\{-1, 1\}$  input domains. Using these techniques we further refined the separation between polynomially and exponentially growing weights. The natural open problem is to find out if these new techniques are useful in extending the existing lower bounds [Hajnal 93] on circuit size to functions with arbitrary weights.

REFERENCES

[Allender 89] E. ALLENDER, *A note on the power of threshold circuits*, in Proceedings of the 30th IEEE Symposium on Foundations of Computer Science, Research Triangle Park, NC, 1989, pp. 580 - 584.

[Anthony 93] M. ANTHONY AND J. SHAWE-TAYLOR, *Using the perceptron algorithm to find consistent hypotheses*, *Combin. Probab. Comput.*, 2 (1993), pp. 385-387.

- [Beame 84] P.W. BEAME, S.A. COOK, AND H.J. HOOVER, *Log depth circuits for division and related problems*, in Proceedings of the 25th IEEE Symposium on Foundations of Computer Science, Singer Island, FL, 1984, pp. 1–6.
- [Goldmann 92] M. GOLDMANN, J. HASTAD, AND A. RAZBOROV, *Majority gates vs. general weighted threshold gates*, *Comput. Complexity*, 2 (1992), pp. 277–300.
- [Goldmann 98] M. GOLDMANN AND M. KARPINSKI, *Simulating threshold circuits by majority circuits*, *SIAM J. Comput.*, 27 (1998), pp. 230–246.
- [Hajnal 93] A. HAJNAL, W. MAASS, P. PUDLAK, M. SZEGEDY, AND G. TURAN, *Threshold circuits of bounded depth*, *J. Comput. System Sci.*, 46 (1993), pp. 129–154.
- [Hampson 86] S.E. HAMPSON AND D.J. VOLPER, *Linear function neurons: Structure and training*, *Biol. Cybernet.*, 53 (1986), pp. 203–217.
- [Hastad 94] J. HÅSTAD, *On the size of weights for threshold gates*, *SIAM. J. Discrete Math.*, 7 (1994), pp. 484–492.
- [Krause 95] M. KRAUSE AND P. PUDLAK, *On computing boolean functions by sparse real polynomials*, in Proceedings of the 36th IEEE Symposium on Foundations of Computer Science, Milwaukee, WI, 1995, pp. 682–691.
- [Muroga 71] M. MUROGA, *Threshold Logic and Its Applications*, Wiley-Interscience, New York, 1971.
- [Myhill 61] J. MYHILL AND W. H. KAUTZ, *On the size of weights required for linear-input switching functions*, *IRE Trans. Electronic Computers*, EC10 (1961), pp. 288–290.
- [Saks 93] M. SAKS, *Slicing the hypercube*, in *Surveys in Combinatorics*, London Math. Soc. Lecture Note Ser. 187, 1, K. Walker, ed., Cambridge University Press, Cambridge, UK, 1993, pp. 211–256.
- [Siu 93] K. SIU, J. BRUCK, T. KAILATH, AND T. HOFMEISTER, *Depth efficient neural networks for division and related problems*, *IEEE Trans. Inform. Theory*, 39 (1993), pp. 423–435.
- [Siu 91] K.-Y. SIU AND J. BRUCK, *On the power of threshold circuits with small weights*, *SIAM J. Discrete Math.*, 4 (1991), pp. 423–435.
- [Willis 63] D.G. WILLIS, *Minimum weights for threshold switches*, in *Switching Theory in Space Techniques*, Stanford University Press, Stanford, CA, 1963.
- [Yao 90] A.C. YAO, *On ACC and threshold circuits*, in Proceedings of the 31th IEEE Symposium on Foundations of Computer Science, St. Louis, MO, 1990, pp. 619–627.