Monotone circuits for the majority function

Shlomo Hoory, Avner Magen, and Toniann Pitassi

IBM Research Laboratory in Haifa, Israel, * Department of Computer Science **
University of Toronto

Abstract. We present a simple randomized construction of size $O(n^3)$ and depth $5.3 \log n + O(1)$ monotone circuits for the majority function on n variables. This result can be viewed as a reduction in the size and a partial derandomization of Valiant's construction of an $O(n^{5.3})$ monotone formula, [15]. On the other hand, compared with the deterministic monotone circuit obtained from the sorting network of Ajtai, Komlós, and Szemerédi [1], our circuit is much simpler and has depth $O(\log n)$ with a small constant. The techniques used in our construction incorporate fairly recent results showing that expansion yields performance guarantee for the belief propagation message passing algorithms for decoding low-density parity-check (LDPC) codes, [3]. As part of the construction, we obtain optimal-depth linear-size monotone circuits for the promise version of the problem, where the number of 1's in the input is promised to be either less than one third, or greater than two thirds. We also extend these improvements to general threshold functions. At last, we show that the size can be further reduced at the expense of increased depth, and obtain a circuit for the majority of size and depth about $n^{1+\sqrt{2}}$ and $9.9 \log n$.

1 Introduction

The complexity of monotone formulas/circuits for the majority function is a fascinating, albeit perplexing, problem in theoretical computer science. Without the monotonicity restriction, majority can be solved with simple linear-size circuits of depth $O(\log n)$, where the best known depth (over binary AND, OR, NOT gates) is $4.95 \log n + O(1)$ [12]. There are two fundamental algorithms for the majority function that achieve logarithmic depth. The first is a beautiful construction obtained by Valiant in 1984 [15] that achieves monotone formulas of depth $5.3 \log n + O(1)$ and size $O(n^{5.3})$. The second algorithm is obtained from the celebrated sorting network constructed in 1983 by Ajtai, Komlós, and Szemerédi [1]. Restricting to binary inputs and taking the middle output bit (median), reduces this network to a monotone circuit for the majority function of depth $K \log n$ and size $O(n \log n)$. The advantage of the AKS sorting network for majority is that it is a completely uniform construction of small size. On the negative side, its proof is quite complicated and more importantly, the constant

* shlomoh@il.ibm.com
** {avner|*toni*}@cs.toronto.edu

K is huge: the best known constant K is about 5000 [11], and as observed by Paterson, Pippenger, and Zwick [12], this constant is important. Further converting the circuit to a formula yields a monotone formula of size $O(n^K)$, which is roughly n^{5000}.

In order to argue about a quality of a solution to the problem, one should be precise about the different resources and the tradeoffs between them. We care about the depth, the size, the number of random bits for a randomized construction, and formula vs circuit question. Finally, the conceptual simplicity of both the algorithm and the correctness proof is also an important goal. Getting the best depth-size tradeoffs is perhaps the most sought after goal around this classical question, while achieving uniformity comes next.

An interesting aspect of the problem is the natural way it splits into two subproblems, the solution to which gives a solution to the original problem. Problem I takes as input an arbitrary n-bit binary vector, and outputs an m-bit vector. If the input vector has a majority of 1's, then the output vector has at least a $2/3$ fraction of 1's, and if the input vector does not have a majority of 1's, then the output vector has at most a $1/3$ fraction of 1's. Problem II is a promise problem that takes the m-bit output of problem I as its input. The output of Problem II is a single bit that is 1 if the input has at least a $2/3$ fraction of 1's, and is a 0 if the input has at most a $1/3$ fraction of 1's. Obviously the composition of these two functions solves the original majority problem.

There are several reasons to consider monotone circuits that are constructed via this two-phase approach. First, Valiant's analysis uses this viewpoint. Boppana's later work [2] actually lower bounds each of these subproblems separately (although failing to provide lower bound for the entire problem). Finally, the second subproblem is of interest in its own right. Problem II can be viewed as an approximate counting problem, and thus plays an important role in many areas of theoretical computer science. Non monotone circuits for this promise problem have been widely studied.

Results The contribution of the current work is primarily in obtaining a new and simple construction of monotone circuits for the majority function of depth $5.3 \log n$ and size $O(n^3)$, hence significantly reducing the size of Valiant's formula while not compromising at all the depth parameter. Further, for subproblem II as defined above, we supply a construction of a circuit size that is of a *linear size*, and it too does not compromise the depth compared to Valiant's solution. A very appealing feature of this construction is that it is uniform, conditioned on a reasonable assumption about the existence of good enough expander graphs. To this end we introduce a connection between this circuit complexity question and another domain, namely *message passing algorithms*. The depth we achieve for the promise problem nearly matches the 1954 lower bound of Moore and Shannon [10].

We further show how to generalize our solution to a general threshold function, and explore another option in the tradeoffs between the different resources we use; specifically, we show that by allowing for a depth of roughly twice that of Valiant's construction, we may get a circuit of size $O(n^{1+\sqrt{2}+o(1)}) = O(n^{2.42})$.

Techniques In obtaining our result we introduce the concept of *deterministic amplification*, replacing the *probabilistic amplification* used by Valiant. In probabilistic amplification, given a monotone boolean function $f : \{0,1\}^n \to \{0,1\}$, one considers the probability $A_f(p)$ that f is one when the n input variables are independently one with probability p. We say that f probabilistically amplifies (p_l, p_h) to (q_l, q_h) if $A_f(p_l) \leq q_l$ and $A_f(p_h) \geq q_h$. We say that a monotone function $f : \{0,1\}^n \to \{0,1\}^m$ deterministically amplifies (p_l, p_h) to (q_l, q_h) if for *every* input with up to $p_l n$ (at least $p_h n$) ones the proportion of ones in the output is at most q_l (at least q_h).

With this terminology splitting the problem into the two subproblems mentioned above can be easily described. We seek two function f_1 and f_2 so that $f_1 : \{0,1\}^n \to \{0,1\}^m$ deterministically amplifies $(1/2 - 1/n, 1/2)$ to $(\delta, 1-\delta)$ for some small constant $\delta > 0$m and $f_2 : \{0,1\}^m \to \{0,1\}$ deterministically amplifies $(\delta, 1-\delta)$ to $(0,1)$. In the sequel, we will call the problem of constructing f_1 *phase I* and that of constructing f_2 *phase II.*

Our circuit for phase I is quite simple. Starting with the n input variables at level zero, we have alternating layers of AND/OR gates, where each gate independently chooses its two inputs from the previous layer. We prove that such a circuit satisfies the requirements if the number of layers is $3.3 \log n$, and if the layers are sufficiently large (width decreasing with depth from $O(n^3)$ to $O(n)$).

We give two constructions of circuits for phase II. Both constructions yield circuits for $f_2 : \{0,1\}^m \to \{0,1\}$ of size $O(m)$ and depth $(2+\epsilon) \cdot \log m + O(1)$, for arbitrarily small $\epsilon > 0$, almost matching the depth lower bound of $2 \log \delta m$ of Moore-Shannon [10]. The first construction is a probabilistic argument similar to our phase I construction but with different parameters. In it we explore the somewhat surprising benefits gained when changing the fanin of the gates to a large enough parameter d.

In the second construction we derandomize our construction using good expander graphs. The construction is an application of a well-known message-passing belief-propagation algorithm on an expander graph. To compute the promise problem, we simulate a logarithmic number of rounds of the message-passing algorithm on a d-regular bipartite graph that is a sufficiently good expander. The message passing algorithm is similar to the belief propagation algorithm used to decode LDPC codes on the erasure channel, and the analysis is based on adaptation of a result of Burshtein and Miller [3] to our setting. For the construction to be completely uniform, we must assume the existence of an explicit construction of sufficiently good expanders. While not known to date, finding such expanders is the focus of a rapidly developing research area, which hopefully will produce the required good expanders.

One crucial parameter used in our analysis, is the number of different inputs the circuit must handle. It is appealing from a computational point of view, as it gives a progress measure toward the final goal of the circuit. One interesting aspect of our probabilistic construction is that it can translate an improvement in this parameter into a reduction in the circuit size. We obtain a variant to our construction by exploiting this property. This variant, has a preprocessing stage that partially sorts its input, and consequently has a smaller size, at the expense of an increased depth.

The organization of the rest of the paper is as follows. In Section 2, we define the two notions of amplification that we will be considering, and review Valiant's argument. In Section 3, we present our new monotone circuits for majority. In Section 4, we adapt our construction to obtain efficient monotone circuits for all threshold functions. In Section 5, we obtain smaller size monotone circuits for the majority, at the expense of increasing the depth. In the last section, we discuss the known lower bounds, and open problems.

2 Definitions and amplification

For a monotone boolean function H on k inputs, we define its *amplification function* $A_H : [0,1] \to [0,1]$ as $A_H(p) = \Pr[H(X_1,\ldots,X_k)=1]$, where X_i are independent boolean random variables that are one with probability p. Valiant [15], considered the function H on four variables, which is the OR of two AND gates, $H(x_1,x_2,x_3,x_4) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$. The amplification function of H, depicted in Figure 1, is $A_H(p) = 1-(1-p^2)^2$, and has a non-trivial fixed point at $\beta = (\sqrt{5}-1)/2 \simeq 0.61$.

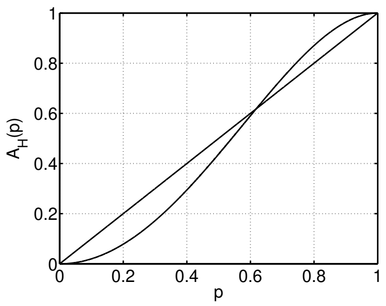


Figure 1: $A_H(p)$ for $H(x_1,x_2,x_3,x_4) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$.

We say that a monotone function $F : \{0,1\}^n \to \{0,1\}$ *probabilistically amplifies* (p_l,p_h) to (q_l,q_h), if $q_l \geq A_F(p_l)$ and $q_h \leq A_F(p_h)$. In other words, applying F to independent boolean random variables that are one with probability p will amplify a promise that p is less than p_l or more than p_h, to a promise that F's output is one with probability less than q_l or more than q_h respectively. Looking at the linear approximation of A_H around β we get that the A_H "almost" amplifies $(\beta-\Delta,\beta+\Delta)$ to $(\beta-\gamma\Delta,\beta+\gamma\Delta)$ for Δ small enough, where $\gamma = A_H(\beta)' = (\sqrt{5}-1)^2 \simeq 1.52$. A more precise statement is that for any $\epsilon > 0$ there exists $\Delta_0 > 0$ such that H probabilistically amplifies $(\beta-\Delta,\beta+\Delta)$ to $(\beta-(\gamma-\epsilon)\Delta,\beta+(\gamma-\epsilon)\Delta)$ for all $\Delta < \Delta_0$. Let H_k be the depth $2k$ binary tree with alternating layers of AND and OR gates, where the root is labeled OR. Valiant's construction uses the fact that A_{H_k} is the composition of A_H with itself k times. Therefore, H_k probabilistically amplifies $(\beta-\Delta,\beta+\Delta)$ to

$(\beta - (\gamma-\epsilon)^k \Delta, \beta + (\gamma-\epsilon)^k \Delta)$, as long as $(\gamma-\epsilon)^k \Delta < \Delta_0$. This implies that for any constant $\epsilon > 0$ we can take $2k = 3.3 \log n + O(1)$ to probabilistically amplify $(\beta - \Omega(1/n), \beta + \Omega(1/n))$ to $(\epsilon, 1-\epsilon)$, where 3.3 is any constant bigger than $\alpha = \log_{\sqrt{5}-1} 2 \simeq 3.27$. Further analysis shows that for $2k = 5.3 \log n + O(1)$, the tree H_k probabilistically amplifies $(\beta - \Omega(1/n), \beta + \Omega(1/n))$ to $(2^{-n-1}, 1-2^{-n-1})$, implying the existence a formula of depth $5.3 \log n + O(1)$ and size $O(n^{5.3})$ for the $\lceil \beta n \rceil$-th threshold function. Results of Boppana [2] and Dubiner and Zwick [5] show that no smaller formula can produce such an amplification.

One aspect of Valiant's construction that we are going to exploit, is that the use of a binary tree in the last $2 \log n$ layers is rather arbitrary. Similar analysis shows that replacing those layers by $2 \log_r n$ layers of an r-ary tree result with the similar probabilistic amplification. Replacing the r-ary AND, OR gates by formulas using binary gates results in a depth blowup of factor $\lceil \log r \rceil$. Therefore, the same depth as Valiant's construction can be obtained when r is a power of two, and taking any large value of r results in an arbitrarily small degradation in the constant before the log.

The approach of this paper, is to follow the same general scheme suggested by Valiant. However, instead of an $O(n^{5.3})$ formula, we produce an $O(n^3)$ circuit of similar depth. Because of the smaller size we cannot use a tree and maintain complete independence between the results computed at a certain layer, as is done in Valiant's tree. Instead we define a random circuit such that the values in a layer are completely independent, given the number of 1's of the previous a layer. In order that the portion of ones in each layer behaves as we would like, we need to make layer sizes sufficiently large. The crucial simple observation that enables us to keep layer sizes small, is that the circuit need only handle 2^n scenarios.

Definition 1. *Let F be a boolean function $F : \{0,1\}^n \to \{0,1\}^m$, and let $\mathcal{S} \subseteq \{0,1\}^n$ be some subset of the inputs. We say that F* deterministically amplifies *(p_l, p_h) to (q_l, q_h) with respect to $\mathcal{S}$, if for all inputs $x \in \mathcal{S}$, the following promise is satisfied (we denote by $|x|$ the number of ones in the vector x):*

$$|F(x)| \le q_l m \quad \text{if} \quad |x| \le p_l n$$
$$|F(x)| \ge q_h m \quad \text{if} \quad |x| \ge p_h n.$$

Note that unlike the probabilistic amplification, deterministic amplification has to work for all inputs or scenarios in the given set $\mathcal{S}$. From here on, whenever we simply say "amplification" we mean deterministic amplification.

For an arbitrary small constant $\epsilon > 0$, the construction we give is composed of two independent phases that may be of independent interest.

- A circuit $C_1 : \{0,1\}^n \to \{0,1\}^m$ for $m = O(n)$ that deterministically amplifies $(\beta - \Omega(1/n), \beta + \Omega(1/n))$ to $(\delta, 1-\delta)$ for an arbitrarily small constant $\delta > 0$. This circuit has size $O(n^3)$ and depth $(\alpha + \epsilon) \cdot \log n + O(1)$.
- A circuit $C_2 : \{0,1\}^m \to \{0,1\}$, such that $C_2(x) = 0$ if $|x| < \delta m$ and $C_2(x) = 1$ if $|x| > (1-\delta)m$, where $\delta > 0$ is a sufficiently small constant. This circuit has size $O(m)$ and depth $(2+\epsilon) \cdot \log m + O(1)$.

The first circuit C_1 is achieved by a simple probabilistic construction that resembles Valiant's construction. We present two constructions for the second circuit, C_2. The first construction is probabilistic; the second construction is a simulation of a logarithmic number of rounds of a certain message passing algorithm on a good bipartite expander graph. The correctness is based on the analysis of a similar algorithm used to decode a low density parity check code (LDPC) on the erasure channel [3].

Combining the two circuits together yields a circuit $C : \{0,1\}^n \to \{0,1\}$ for the $\lceil \beta n \rceil$-th threshold function. The circuit is of size $O(n^3)$ and depth $(\alpha + 2 + 2\epsilon)\log n + O(1)$.

3 Monotone circuits for Majority

In this section we give a randomized construction of the circuit $C : \{0,1\}^n \to \{0,1\}$ such that $C(x)$ is one if the portion of ones in x is at least βn and zero otherwise. The circuit C has size $O(n^3)$ and depth $(2 + \alpha + \epsilon) \cdot \log n + O(1)$ for an arbitrary small constant $\epsilon > 0$. As we described before, we will describe C as the compositions of the circuits C_1 and C_2 whose parameters are given by the following two theorems:

Theorem 1. *For every $\epsilon, \epsilon', c > 0$, there exists a circuit $C_1 : \{0,1\}^n \to \{0,1\}^m$ for $m = O(n)$, of size $O(n^3)$ and depth $(\alpha+\epsilon)\cdot\log n+O(1)$ that deterministically amplifies all inputs from $(\beta - c/n, \beta + c/n)$ to $(\epsilon', 1 - \epsilon')$.*

Theorem 2. *For every $\epsilon > 0$, there exists $\epsilon' > 0$ and a circuit $C_2 : \{0,1\}^n \to \{0,1\}$, of size $O(n)$ and depth $(2+\epsilon)\cdot\log n+O(1)$ that deterministically amplifies all inputs from $(\epsilon', 1 - \epsilon')$ to $(0, 1)$.*

The two circuits use a generalization of the four input function H used in Valiant's construction. For any integer $d \geq 2$, we define the function $H^{(d)}$ on d^2 inputs as the d-ary OR of d d-ary AND gates, i.e $\vee_{i=1}^{d} \wedge_{j=1}^{d} x_{ij}$. Note that Valiant's function H is just $H^{(2)}$.

Each of the circuits C_1 and C_2 is a layered circuit, where layer zero is the input, and each value at the i-th layer is obtained by applying $H^{(d)}$ to d^2 independently chosen inputs from layer $i - 1$. However, the values of d we choose for C_1 and C_2 are different. For C_1 we have $d = 2$, while for C_2 we choose sufficiently large $d = d(\epsilon)$ to meet the depth requirement of the circuit. We let $\mathcal{F}_{n,m,F}$ denote a random circuit mapping n inputs to m outputs, where F is a fixed monotone boolean circuit with k inputs, and each of the m output bits is calculated by applying F to k independently chosen random inputs.

We start with a simple lemma that relates the deterministic amplification properties of $\mathcal{F}$ to the probabilistic amplification function A_F. [1]

[1] Note that we talk about the deterministic amplification properties of a random function.

Lemma 1. *For any $\epsilon, \delta > 0$, the random function $\mathcal{F}$ deterministically amplifies (p_l, p_h) to $(A_F(p_l) \cdot (1+\delta), A_F(p_h) \cdot (1-\delta))$ with respect to $\mathcal{S} \subseteq \{0,1\}^n$ with probability at least $1-\epsilon$, if:*

$$m = \Omega\left(\frac{\log(|\mathcal{S}|) + \log(1/\epsilon)}{A_F(p_l) \cdot \delta^2}\right).$$

Proof. It is sufficient to prove that for any input $x \in \mathcal{S}$, the probability of failure of F is bounded by $\epsilon/|\mathcal{S}|$. By definition, for any application of $\mathcal{F}$, the probability to get 1 is $A_F(p)$, where $p = |x|/n$ is the portion of ones in x. By monotonicity, we may assume that $p = p_l$ or $p = p_h$. A straightforward application of the Chernoff bound is all we need here. For the case $p = p_l$, we have $\Pr[|\mathcal{F}(x)| > A_F(p_l)(1+\delta)m] < \exp(-mA_F(p)\delta^2/3)$, which is less than $\epsilon/|\mathcal{S}|$ for $m \geq 3(\log|\mathcal{S}| + \log(1/\epsilon))/(A_F(p_l) \cdot \delta^2)$. The case $p = p_h$ is handled similarly.

Proof (Proof of Theorem 1).
The circuit C_1 is a composition of $\mathcal{F}_{n,m_1,H}, \mathcal{F}_{m_1,m_2,H}, \ldots, \mathcal{F}_{m_{t-1},m_t,H}$, where the parameters $n = m_0, m_1, \ldots, m_t = m$ are positive integers to be fixed later, and are the sizes of the layers of the circuit. Since $\mathcal{F}_{\cdot,\cdot,H}$ is a random function, this describes a random construction of a circuit. We prove that with high probability such a circuit deterministically amplifies all inputs from $(\beta - c/n, \beta + c/n)$ to $(\epsilon', 1-\epsilon')$. For simplicity, we only prove that with high probability for all inputs with portion of ones smaller than $\beta - c/n$ the output has fewer than $\epsilon' m$ ones. The proof of the other case is similar. For convenience of notation, we say that some circuit (deterministically or probabilistically) amplifies p to q as a short hand for amplifying $(p, \cdot)$ to $(q, \cdot)$ where the dot stands for the unspecified upper bounds.

The basic idea is that if layers have large size, we expect this circuit to have similar behavior to Valiant's tree. As observed before, for every constant $\epsilon > 0$ there is a constant $\Delta_0 > 0$ such that for any $p = \beta - \Delta$ with $0 < \Delta < \Delta_0$, we have $A_H(p) < \beta - (\gamma - \epsilon)\Delta$, where $\gamma = A_H(\beta)'$. This implies that if the portion of ones at some level i is p, then the expected portion of ones at level $i+1$ is $A_H(p) < \beta - (\gamma - \epsilon)\Delta$. We will set δ in Lemma 1 so that the deterministic amplification of $\mathcal{F}_{m_i,m_{i+1}}$ guarantees that the portion of ones at level $i+1$ will be at most $\beta - (\gamma - 2\epsilon)\Delta$. The details follow.

Let G_i be the be the circuit truncated to the first i layers. We prove that with high probability G_i deterministically amplifies the initial promise $\beta - c/n$ to $\beta - (\gamma - 2\epsilon)^i \cdot (c/n)$, as long as $(\gamma - 2\epsilon)^i \cdot (c/n) < \Delta_0$. The proof proceeds by inductions on i, where the basis $i = 0$ trivially holds. So, assume that $i > 0$, and that $(\gamma - 2\epsilon)^i \cdot (c/n) < \Delta_0$. Furthermore, assume that the first $i-1$ layers are some *fixed circuit* G_{i-1} satisfying the hypothesis. Namely, deterministically amplifies $\beta - c/n$ to $\beta - (\gamma - 2\epsilon)^{i-1} \cdot (c/n)$, for all inputs. Let G_i be obtained by composing the fixed circuit G_{i-1} with the random circuit $\mathcal{F}_{m_{i-1},m_i,H}$. Then, as G_{i-1} is fixed, it has at most 2^n possible outputs. The crucial observation, is that it suffices for $\mathcal{F}_{m_{j-1},m_j,H}$ to deterministically amplify $\beta - (\gamma - 2\epsilon)^{i-1} \cdot (c/n)$ to $\beta - (\gamma - 2\epsilon)^i \cdot (c/n)$, *only* with respect to the 2^n outputs of G_{i-1}.

Then, it suffice to choose the values δ in Lemma 1, as

$$\frac{\epsilon \cdot (\gamma - 2\epsilon)^{i-1} \cdot (c/n)}{\beta - (\gamma - \epsilon) \cdot (\gamma - 2\epsilon)^{i-1} \cdot (c/n)} = \Theta\left((\gamma - 2\epsilon)^{i-1} \cdot (c/n)\right).$$

That is, we can choose δ as an increasing geometric sequence, starting from $\Theta(1/n)$ for $i = 1$, up to $\Theta(1)$ for $i = \log_{\gamma-2\epsilon} n$. The implied layer size for error probability 2^{-n} (which is much better than we need), is $\Theta(n/\delta^2)$. Therefore, it decreases geometrically from $\Theta(n^3)$ down to $\Theta(n)$.

It is not difficult to see that after achieving the desired amplification from $\beta - c/n$ to $\beta - \Delta_0$, only a constant number of layers is needed to get down to ϵ'. The corresponding value of δ in these last steps is a constant (that depends on ϵ'), and therefore, the required layer sizes are all $\Theta(n)$.

Proof (Proof of Theorem 2).

The circuit C_2 is a composition of $\mathcal{F}_{n,m_1,H^{(d)}}, \mathcal{F}_{m_1,m_2,H^{(d)}}, \ldots, \mathcal{F}_{m_{t-1},m_t,H^{(d)}}$, where d and the layer sizes $n = m_0, m_1, \ldots, m_t = 1$ are suitably chosen parameters depending on ϵ. We prove that for each layer, with nonzero probability such a circuit deterministically amplifies all inputs from $(\epsilon', 1-\epsilon')$ to $(0, 1)$. As before, we restrict our attention to the lower end of the promise problem and prove that C_2 outputs zero on all inputs with portion of ones smaller than ϵ'.

As in the circuit C_1, the layer sizes must be sufficiently large to allow accurate computation. However, for the circuit C_2, accurate computation does not mean that the portion of ones in each layer is close to its expected value. Rather, our aim is to keep the portion of ones bounded by a fixed constant ϵ', while making each layer smaller than the preceding one by approximately a factor of d. We continue this process until the layer size is constant, and then use a constant size circuit to finish the computation. Therefore, since the number of layers of such a circuit is about $\log n / \log d$, and the depth of the circuit for $H^{(d)}$ is $2\lceil \log d \rceil$, the total depth is about $2 \log n$ for large d.

By the above discussion, it suffices to prove the following: For every $\epsilon > 0$ there exists a real number $\delta > 0$ and two integers d, n_0, such that for all $n \geq n_0$ the random circuit $\mathcal{F}_{n,m,H^{(d)}}$ with $m = (1+\epsilon) \cdot n/d$, deterministically amplifies δ to δ with respect to all inputs, with failure probability at most $1/n$. Since $A_H(\delta) = 1 - (1 - \delta^d)^d \leq d \cdot \delta^d$, the probability of failure for any specific input with portion of ones at most δ, is bounded by:

$$\binom{m}{\delta m} \cdot (A_H(\delta))^{\delta m} \leq \left(\frac{e}{\delta} \cdot d \cdot \delta^d\right)^{\delta m} = \left(de \cdot \delta^{d-1}\right)^{\delta m}.$$

Therefore, by a union bound the probability that $\mathcal{F}_{n,m,H^{(d)}}$ fails is bounded by:

$$\sum_{i=0}^{\delta n} \left(de \cdot i^{d-1}\right)^{im} \cdot \binom{n}{i} \leq 2\left(de \cdot \delta^{d-1}\right)^{\delta m} \cdot \binom{n}{\delta n} \leq 2\left[\left(de \cdot \delta^{d-1}\right)^{(1+\epsilon)/d} \cdot (e/\delta)\right]^{\delta n}$$

The last term above is equal to

$$2\left[c(d, \epsilon) \cdot \delta^{(1+\epsilon)\cdot(d-1)/d-1}\right]^{\delta n},$$

where $c(d, \epsilon)$ is some function of d and ϵ. Given, ϵ, we choose a sufficiently large d so that $(1+\epsilon) \cdot (d-1)/d - 1$ is positive. Then we take sufficiently small δ, so that the expression in the square brackets is smaller than one. Finally, we take a sufficiently large n_0 to guarantee that the exponentially small upper bound on the error probability, is smaller than 1.

3.1 Derandomizing the construction of phase II

In this subsection we present a second construction of a small monotone circuit C that deterministically amplifies $(a, 1-a)$ to $(0, 1)$ with respect to $\{0,1\}^n$.

Our construction uses recent ideas and algorithms from belief propagation decoding, applied to solving majority. Underlying both belief propagation and algorithms for majority is the concept of amplification, first introduced in the classical 1954 paper of Moore and Shannon. Since then, the amplification method has been generalized and used in a variety of contexts. Luby, Mitzenmacher and Shokrollahi [8] used the amplification method to analyze the performance of a belief propagation message passing algorithm for decoding low density parity check (LDPC) codes. Today the use of belief propagation for decoding LDPC codes is one of the hottest topics in error correcting codes [9, 14, 13].

Let $G = (V_L, V_R; E)$ be a d regular bipartite graph with n vertices on each side, $V_L = V_R = [n]$. Consider the following message passing algorithm, where we think of the left and right as two players. The left player "plays AND" and the right player "plays OR". At time zero the left player starts by sending one boolean message through each left to right edge, where the value of the message m_{uv} from $u \in V_L$ to $v \in V_R$ is the input bit x_u. Subsequently, the messages at time $t > 0$ are calculated from the messages at time $t-1$. At odd times, given the left to right messages m_{uv}, the right player calculates the right to left messages m'_{vw}, from $v \in V_R$ to $w \in V_L$ by the formula $m'_{vw} = \vee_{u \in N(v) \setminus w} m_{uv}$. That is, the right player sends a 1 along the edge from $v \in V_R$ to $w \in V_L$ if and only if *at least one* of the incoming messages/values (not including the incoming message from w) is 1. Similarly, at even times the algorithm calculates the left to right messages m'_{vw}, $v \in V_L$, $w \in V_R$, from the right to left messages m_{uv}, by the formula $m'_{vw} = \wedge_{u \in N(v) \setminus w} m_{uv}$. That is, the left player sends a 1 along the edge from $v \in V_L$ to $w \in V_R$ if and only if *all* of the incoming messages/values (not including the incoming message from w) are 1. We further need the following definitions. We call a *left* vertex *bad* at even time t if it transmits at least one message of value one at time t. Similarly, a *right* vertex is *bad* at odd time t if it is a right vertex that transmits at least one message of value zero at time t. We let $b(t)$ be the number of bad vertices at time t. These definitions will be instrumental in providing a potential function measuring the progress of the message passing algorithm which is expressed in Lemma 2.

We say that a bipartite graph $G = (V_L, V_R; E)$ is (λ, e)-expanding, if for any vertex set $S \subseteq V_L$ (or $S \subseteq V_R$) of size at most λn, $|N(S)| \geq e|S|$. It will be convenient to denote the expansion of the set S by $e_S = |N(S)|/|S|$.

Lemma 2. *Consider the message passing algorithm using a $d \geq 4$ regular expander graph with $d-1 > e \geq (d+1)/2$. If $b(t) \leq \lambda n/d^2$ then $b(t+2) \leq b(t)/\eta$, where $\eta = \frac{d-1}{2(d-e)}$.*

We postpone the proof of the lemma, and show its use for constructing the circuit C_2. First, we show that, for any $\epsilon > 0$, the algorithm provides a circuit of depth $(2+\epsilon)\log n$ and of size $O(n \log n)$ for the promise problem with $a \leq \lambda(d-1)/d^3$. Suppose that there are at most an ones. Then $b(0) \leq an$. Therefore $b(2t) \leq an/\eta^t$, and so $b(2t) = 0$ for $t > \log(an)/\log \eta$ and all outputs are zero at that time. If there are at most an zeros, we analyze the number of bad *right* vertices as follows. Since each bad right vertex must be connected to at least $d-1$ left vertices associated with zero input bits, and since there are at most an left vertices transmitting zero, it follows that $b(1) \leq an \cdot d/(d-1) \leq \lambda n/d^2$ whence the conditions of Lemma 2 are satisfied and $b(2t+1) \leq \frac{and}{d-1}/\eta^t$ and so $b(2t+1) = 0$ for $t > \log(a\frac{d}{d-1}n)/\log \eta$.

The better the expanders we use, the bigger $\eta = \frac{d-1}{2(d-e)}$ gets, and the better the time guarantee above gets. How good are the expanders that we may use? One can show the existence of such expanders for sufficiently large d large, and $e > d - c$ for an absolute constant c.

The best known explicit construction that gets close to what we need, is the result of [4]. However, that result does not suffice here for two reasons. The first is that it only achieves expansion $(1-\epsilon)d$ for any $\epsilon > 0$ and sufficiently large d depending on ϵ. The second is that it only guarantees left-to-right expansion, while our construction needs both left-to-right and right-to-left expansion. We refer the reader to the survey [6] for further reading and background.

For such expanders, $\eta \geq \frac{d-1}{2c}$, and therefore, after $2\log(\frac{and}{d-1})/\log\frac{d-1}{2c} = (2+\epsilon)\frac{\log n}{\log d-1}$ iterations, all messages contain the right answer, where ϵ can be made arbitrarily small by choosing sufficiently large d. It remains to convert the algorithm into a monotone circuit, which introduces a depth-blowup of $\log\lceil d-1\rceil$ owing to the depth of a binary tree simulating a $(d-1)$-ary gate. Thus we get a $(2+\epsilon)\log n$-depth circuit for arbitrarily small $\epsilon > 0$. The size is obviously $dn \cdot \text{depth} = O(n \log n)$.

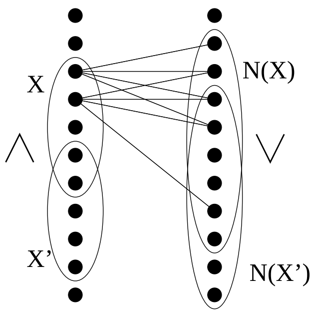
To get a linear circuit, further work is needed, which we now describe. The idea is to use a sequence of graphs $G_0 = G, G_1, \ldots$, where each graph is half the size of its preceding graph, but has the same degree and expansion parameters. We start the message passing algorithm using the graph $G = G_0$, and every t_0 rounds (each round consists of OR and then AND), we switch to the next graph in the sequence. Without the switch, the portion of bad vertices should decrease by a factor of η^{t_0}, every t_0 rounds. We argue that each switch can be performed, while losing at most a constant factor. To describe the switch from G_i to G_{i+1}, we identify $V_L(G_{i+1})$ with an arbitrary half of the vertices $V_L(G_i)$, and start the message passing algorithm on G_{i+1} with the left to right messages from each vertex in $V_L(G_{i+1})$, being the same as at the last round of the algorithm on G_i. As the number of bad left vertices cannot increase at a switch, their portion, at most doubles. For the right vertices, the exact argument is slightly more involved, but it is clear that the portion of bad right vertices in the first round in G_{i+1}, increases by at most a constant factor c, compared with what it should have been, had there been no switch. (Precise calculation yields $c = 2d\eta$.) To summarize, as the circuit consists of a geometrically decreasing sequence of blocks starting with a linear size block, the total size is linear as well. As for the depth, the amortized reduction in the portion of bad vertices per round, is by a factor of $\eta' = \eta/c^{1/t_0}$. Therefore, the resulting circuit is only deeper than the one described in the previous paragraph, by a factor of $\log \eta / \log \eta'$. By choosing a sufficiently large value for t_0, we obtain:

Theorem 3. *For any $\epsilon > 0$, there exists $a > 0$ such that for any n there exists a monotone circuit of depth $(2+\epsilon)\log n + O(1)$ and size $O(n)$ that solves a-promise problem.*

We note here that $O(\log n)$ depth monotone circuits for the a-promise problem can also be obtained from ϵ-halvers. These are building blocks used in the AKS network. However, our monotone circuits for the a-promise problem have two advantages. First, our algorithm relates this classical problem in circuit complexity to recent popular message passing algorithms. And second, the depth that we obtain is nearly tight. Namely, Moore and Shannon [10] prove that any monotone formula/circuit for majority requires depth $2\log n - O(1)$, and the lower bound holds for the a-promise problem as well.

Proof (Proof of Lemma 2). (builds on [3])

We consider only the case of bad left vertices. The proof for bad right vertices follows from the same proof, after exchanging ones with zeroes, ANDs with ORs, and lefts with rights. Let $B \subseteq V_L$ be the set of bad left vertices, and assume $|B| \leq \lambda n/d^2$ at some even time t and B' the set of bad vertices at time $t+2$. We bound the size of B' by considering separately $B' \setminus B$ and $B' \cap B$. Note that all sets considered in the proof have size at most λn, and therefore expansion at least e.



To bound $B' \setminus B$, consider the set $Q = N(B' \setminus B) \setminus N(B) = N(B' \cup B) \setminus N(B)$. Since vertices in Q are not adjacent to B, then at time $t+1$ they send right to left messages valued zero. On the other hand, any vertex in $B' \setminus B$ can receive at most one such zero message (otherwise all its messages at time $t+2$ will be valued zero and it cannot be in B'). Therefore, since each vertex in Q must have at least one neighbour in $B' \setminus B$, it follows that $|Q| \leq |B' \setminus B|$. Therefore, we have:

$$|N(B' \cup B)| = |N(B)| + |Q| \leq |N(B)| + |B' \setminus B| = e_B \cdot |B| + |B' \setminus B|.$$

On the other hand, $|N(B' \cup B)| \geq e \cdot |B' \cup B| = e \cdot (|B| + |B' \setminus B|)$.
Combining the above two inequalities, we obtain:

$$|B' \setminus B| \leq \frac{e_B - e}{e - 1} \cdot |B|. \tag{1}$$

To bound $B' \cap B$, consider the set $T = N(B' \cap B) \setminus N(B \setminus B') = N(B) \setminus N(B \setminus B')$. Let N_0 (resp. N_1) be the number of zero (resp. one) messages received by vertices in $B' \cap B$ at time $t+1$. Then obviously, $N_0 + N_1 = d \cdot |B' \cap B|$. As before, a vertex in $B' \cap B$ can receive at most one zero message and therefore $N_0 \leq |B' \cap B|$. Also, let T_0 be the vertices of T that transmit at least one zero message at time $t+1$ to $B' \cap B$, and $T_1 = T \setminus T_0$. Clearly $|T_0| \leq N_0$. On the other hand, each vertex in T_1 transmits a one to some vertex in $B' \cap B$, and therefore must have at least two neighbors in $B' \cap B$, implying that $|T_1| \leq N_1/2$. Hence

$$\begin{aligned} |T| \leq N_0 + N_1/2 &= (N_0 + N_1)/2 + N_0/2 \\ \leq \quad & \frac{d}{2}|B' \cap B| + \frac{1}{2}|B' \cap B| = |B' \cap B| \cdot (d+1)/2. \end{aligned}$$

Therefore

$$\begin{aligned} e_B \cdot |B| = |N(B)| &= |N(B \setminus B')| + |T| \leq d \cdot |B \setminus B'| + |B' \cap B| \cdot (d+1)/2 \\ &= d \cdot |B| - |B' \cap B| \cdot (d-1)/2. \end{aligned}$$

This implies:

$$|B' \cap B| \leq \frac{d - e_B}{(d-1)/2} \cdot |B|. \tag{2}$$

Combining inequalities (1) and (2) we get that:

$$|B'|/|B| \leq \frac{e_B - e}{e - 1} + \frac{d - e_B}{(d-1)/2}.$$

Since $e \geq (d+1)/2$, and $e_B \geq e$, this yields the required bound:

$$|B'|/|B| \leq 2(d-e)/(d-1).$$

As noted before in Section 2, replacing the last $2 \log n$ layers of Valiant's tree with $2 \log_r n$ layers of r-ary AND/OR gates, results in an arbitrarily small increase in the depth of the corresponding formula for a large value of r. It is interesting to compare the expected behavior of the suggested belief-propagation algorithm to the behavior of the $(d-1)$-ary tree. Assume that the graph G is chosen at random (in the configuration model), and that the number of rounds k is sufficiently small, $(d-1)^{2k} \ll n$. Then, almost surely the computation of all but $o(1)$ fraction of the k-th round messages is performed by evaluating a $(d-1)$-ary depth k trees. Moreover, introducing an additional $o(1)$ error, one may assume that the leaves are independently chosen boolean random variables that are one with probability p, where p is the portion of ones in the input. This observation sheds some light on the performance of the belief propagation algorithm. However, our analysis proceeds far beyond the number of rounds for which a cycle free analysis can be applied.

4 Monotone formulas for threshold-k functions

Consider the case of the k-th threshold function, $T_{k,n}$, i.e. a function that is one on $x \in \{0,1\}^n$ if $|x| \geq (k+1)$ and zero otherwise. We show that, by essentially the same techniques of Section 3, we can construct monotone circuits to this more general problem. We assume henceforth that $k < n/2$, since otherwise, we construct the circuit $T_{n-1-k,n}$ and switch AND with OR gates. For $k/n = \Theta(1)$, the construction yields circuits of depth $5.3 \log n + O(1)$ and size $O(n^3)$. However, when $k = o(n)$, circuits are shallower and smaller (this not surprising fact is also discussed in [2] in the context of formulas).

The construction goes as follows: (i) Amplify $(k/n, (k+1)/n)$ to $(\beta - \Omega(1/k), \beta + \Omega(1/k))$ by randomly applying to the input a sufficiently large number of OR gates with arity $\Theta(n/k)$ (ii) Amplify $(\beta - \Omega(1/k), \beta + \Omega(1/k))$ to $(O(1), 1 - O(1))$ using a variation of phase I, and (iii) Amplify $(O(1), 1 - O(1))$ to $(0,1)$ using phase II.

We now give a detailed description. For the sake of the section to follow, we require the following lemma which is more general than is needed for the results of this section. The proof is deferred to the appendix.

Lemma 3. *Let $\mathcal{S} \subseteq \{0,1\}^n$, and $\epsilon > 0$. Then, for any k, there is a randomized construction of a monotone circuit that evaluates $T_{k,n}$ correctly on all inputs from $\mathcal{S}$ and has*

$$\begin{aligned} \textit{depth} &\leq \log(n) + 2.3\log(k') + (2+\epsilon) \cdot \log\log|\mathcal{S}| + O(1), \\ \textit{size} &\leq O(\log|\mathcal{S}| \cdot k'n). \end{aligned}$$

Here $k' = \min(k, n-1-k)$, and the constants of the O depend only on ϵ.

To guarantee the correctness of a monotone circuit for $T_{n,k}$, it suffices to check its output on inputs of weight $k, k+1$ (as the circuit is monotone). Therefore, $|\mathcal{S}| = \binom{n}{k} + \binom{n}{k+1}$, implying that $\log|\mathcal{S}| = O(k\log(n/k))$. Therefore, we have:

Theorem 4. *There is a randomized construction of a monotone circuit for $T_{k,n}$ with:*

$$\begin{aligned} \textit{depth} &\leq \log(n) + 4.3\log(k') + O(\log\log(n/k)), \\ \textit{size} &\leq O((k')^2 n \log(n/k')), \end{aligned}$$

where $k' = \min(k, n-1-k)$, and the constants of the O are absolute.

5 Reducing the circuit size

The result obtained so far for the majority, is a monotone circuit of depth $5.3\log n + O(1)$ and size $O(n^3)$. In this section, we would like to obtain smaller circuit size, at the expense of increasing the depth somewhat. The crucial observation is that the size of our circuit depends linearly on the logarithm of the number of scenarios it has to handle. Therefore, applying a preprocessing stage to reduce the wealth of scenarios may save up to a factor of n in the circuit size. We propose a recursive construction that reduces the circuit size to about $n^{1+\sqrt{2}}$.

Initially, by Theorem 4, we have monotone circuits $C^0_{k,n}$ for the threshold functions $T_{k,n}$ with size, depth:

$$s_0(n) = O(n^3), \quad d_0(n) = 5.3\log n + O(1).$$

Given circuits $C^{(i)}_{k,n}$ for $T_{k,n}$ of size and depth bounded by $s_i(n)$, $d_i(n)$, one can calculate all threshold functions in parallel and obtain a sorting circuit $C^{(i)}_n$: $\{0,1\}^n \to \{0,1\}^n$ of size and depth bounded by $ns_i(n)$, $d_i(n)$. The circuit $C^{(i+1)}_{k,n}$ is built of two stages. First, the n-bit input is partitioned into n/a_i blocks of size a_i, and each block is sorted in parallel using the circuit $C^{(i)}_a$. Second, the kth threshold function is calculated on the partially sorted n-bit string by the family of circuits with parameters given by Lemma 3. When the n/a blocks are sorted, there are only $(a_i+1)^{n/a_i}$ possible inputs, as the number of ones in each blocks completely specifies the input. Therefore, the first stage reduces the number of scenarios to $(a_i+1)^{n/a_i} \le n^{n/a_i}$ and we have

$$s_{i+1}(n) = (n/a_i)\cdot(a_i+1)\cdot s_i(a_i) + O\left((n/a_i)\cdot \log n \cdot n^2\right) = n\cdot s_i(a_i) + n^{3+o(1)}/a_i,$$
$$d_{i+1}(n) = d_i(a_i) + 5.3\log n - 2\log(a_i) + O(\log\log n).$$

Let $a_i = n^{\alpha_i}$ for some constants α_i, and assume that $s_i(n) = n^{\sigma_i + o(1)}$, and that $d_i(n) = \delta_i \log n + O(\log\log n)$. Then we obtain the following recurrence for σ_i and δ_i:

$$\begin{aligned}\sigma_{i+1} &= \max(1+\alpha_i\sigma_i,\, 3-\alpha_i)\\ \delta_{i+1} &= \alpha_i\delta_i + 5.3\log n - 2\alpha_i.\end{aligned}$$

We choose $\alpha_i = 2/(\sigma_i+1)$ to equate $1+\alpha_i\sigma_i$ with $3-\alpha_i$. This implies that $\sigma_{i+1} = 3 - 2/(\sigma_i+1)$, and $\delta_{i+1} = 5.3 + (\delta_i - 2)\cdot 2/(\sigma_i+1)$, resulting in the following sequence:

i	0	1	2	3	4	5	6	7	8	9	10
α_i	0.500	0.571	0.583	0.585	0.586	0.586	0.586	0.586	0.586	0.586	0.586
σ_i	3.000	2.500	2.429	2.417	2.415	2.414	2.414	2.414	2.414	2.414	2.414
δ_i	5.271	6.906	8.074	8.814	9.259	9.522	9.677	9.768	9.821	9.852	9.870

The sequence of α_i tends to the positive solution of $x = 3 - 2/(x+1)$, which is $1+\sqrt{2}$, and the sequence of δ_i tends to $(1+\sqrt{2})(\alpha - 2 + 2\sqrt{2}) \simeq 9.896$. Therefore, we have:

Theorem 5. *There is a randomized construction of a monotone circuit for the majority of size* $n^{1+\sqrt{2}+o(1)}$*, and depth* $9.9 \log n + O(1)$*.*

6 Related Work and Open Problems

It is of great interest to improve upon the size or amount of randomness required by our construction. One approach, is to reduce the number of scenarios by preprocessing. The best result we have here is stated in Theorem 5. A second approach, is to improve the original bound (n^3 for size, and $n^3 \log n$ for the number of random bits). The obvious obstacle are the first few layers of the phase I circuit. The current n^3 size upper bound follows from a union bound, which we do not know to be tight. In fact, we do not even know how to save on the size or amount of randomness required to construct the first layer! This very problem can be cast as a discrepancy problem on hypergraphs. Indeed, if we restrict ourselves to repeated applications of $H(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$ so that each application is associated with the two pairs $\{x_1, x_2\}$ and $\{x_3, x_4\}$, we have the following discrepancy problems on 4-uniform hypergraphs. Find a hypergraph on n vertices with each edge is composed of two size-two sets $e = e_1 \cup e_2$. The graph should have as few edges as possible while satisfying that for every vertex subset $S \subset [n]$, the portion of edges that have at least one of their halves inside S is close to $A_H(|S|/n)$. This problem seems to generalize a similar problem for graphs, that is to construct a graph where for every vertex subset S, the portion of edges with both end points in S is close to $(|S|/n)^2$. Not, surprisingly, this turns out to be equivalent to expansion.

In seminal work, Karchmer and Wigderson [7] gave a precise characterization of both monotone and monotone formula/circuit size based on the complexity of related communication search problems. For the majority function, the monotone search problem is as follows. Let mMaj-search be the following two-player communication complexity problem. Player I is given a subset $A \subset [n]$ of size $n/2+1$. Player II is given a subset $B \subset [n]$ of size $n/2$. They want to determine an element $i \in [n]$ such that i is in their intersection. In the non-monotone version of the problem, Maj-search, the input is the same, but now they are allowed to find either an element i in their intersection, or an element j lying outside of both sets. By the main theorem of [7], the minimal monotone formula/circuit size for majority is equal to the communication complexity of mMaj-search, and the minimal formula/circuit size for majority is equal to the communication complexity of Maj-search. The monotone communication complexity problem for the promise problem is as above except that now Players I and II are given subsets A and B each of size $2n/3$ and again they want to find some element in their common intersection. Likewise, for the monotone version, they want to compute either an element in the common intersection, or an element j lying outside of both sets. We find it useful to consider the upper and lower bounds for majority, as well as for the promise version of majority within this communication complexity setting.

There are two central open problems related to this work. First, is the promise version really simpler than majority? A lower bound greater than $2\log n$ on the communication complexity of mMaj-search would settle this question. Boppana [2] and more recent work [5] show lower bounds on a particular method for obtaining monotone formulas for majority. However we are asking instead for lower bounds on the size/depth of unrestricted monotone formulas/circuits. Secondly, the original question remains unresolved. Namely, we would like to obtain explicit uniform formulas for majority of optimal or near optimal size. A related problem is to come up with a natural (top-down) communication complexity protocol for mMaj-Search that uses $O(\log n)$ many bits.

References

1. M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c\log n$ parallel steps. *Combinatorica*, 3(1):1–19, 1983.
2. R. B. Boppana. Amplification of probabilistic boolean formulas. *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 20–29, 1985.
3. D. Burshtein and G. Miller. Expander graph arguments for message-passing algorithms. *IEEE Trans. Inform. Theory*, 47(2):782–790, 2001.
4. M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree expansion beyond the degree 2 barrier. In *Proceedings 34th Symposium on Theory of Computing*, pages 659–668, 2002.
5. M. Dubiner and U. Zwick. Amplification by read-once formulas. *SIAM J. Comput.*, 26(1):15–38, 1997.
6. S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. to appear.
7. Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 539–550, Chicago, IL, May 1988.
8. M. Luby, M. Mitzenmacher, and A. Shokrollahi. Analysis of random processes via and-or tree evaluation. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1998.
9. M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman. Analysis of low density codes and improved designs using irregular graphs. *ACM Symposium on Theory of Computing (STOC)*, 1998.
10. E. F. Moore and C. E. Shannon. Reliable circuits using less reliable relays. I, II. *J. Franklin Inst.*, 262:191–208, 281–297, 1956.
11. M. S. Paterson. Improved sorting networks with $O(\log N)$ depth. *Algorithmica*, 5(1):75–92, 1990.
12. M. S. Paterson, N. Pippenger, and U. Zwick. Optimal carry save networks. In *Boolean function complexity (Durham, 1990)*, volume 169 of *London Math. Soc. Lecture Note Ser.*, pages 174–201. Cambridge Univ. Press, Cambridge, 1992.
13. T. Richardson and R. Urbanke. Modern coding theory. Draft of a book.
14. T. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, 2001.
15. L. G. Valiant. Short monotone formulae for the majority function. *J. Algorithms*, 5(3):363–366, 1984.