

Representing (0,1)-Matrices by Boolean Circuits[☆]

Stasys Jukna^{a,1}

^a*Institute of Mathematics and Computer Science, LT-80663 Vilnius, Lithuania*

Abstract

A boolean circuit *represents* an n by n (0,1)-matrix A if it correctly computes the linear transformation $\vec{y} = A\vec{x}$ over $\text{GF}(2)$ on all n unit vectors. If we only allow *linear* boolean functions as gates, then some matrices cannot be represented using fewer than $\Omega(n^2/\ln n)$ wires. We first show that using non-linear gates one can save a lot of wires: any matrix can be represented by a depth-2 circuit with $O(n \ln n)$ wires using multilinear polynomials over $\text{GF}(2)$ of relatively small degree as gates. We then show that this cannot be substantially improved: If any two columns of an n by n (0,1)-matrix differ in at least d rows, then the matrix requires $\Omega(d \ln n / \ln \ln n)$ wires in any depth-2 circuit, even if arbitrary boolean functions are allowed as gates.

Key words: Boolean circuits; Linear circuits; Hadamard matrices; Lower bounds; Sunflower lemma

1. Introduction

Every n by n (0,1)-matrix A defines a linear transformation $\vec{y} = A\vec{x}$ over $\text{GF}(2)$, where $\vec{x} \in \text{GF}(2)^n$ is an input. Our goal is to compute this transformation by a general depth-2 circuit using as few wires as possible. Such a circuit is a directed acyclic graph of depth 2 with n input nodes x_1, \dots, x_n , n output nodes y_1, \dots, y_n and every non-input node v assigned a gate g_v computing an *arbitrary* boolean function of its inputs; there is no bound on the fanin or on the fanout of the nodes.

It is clear that n^2 wires are always enough, even in depth-1 and even if all gates are *linear*, that is, compute sums mod 2 of their inputs: just let the i th output gate y_i to compute the scalar product of the input vector \vec{x} with the i th row of A .

The interest in general depth-2 circuits comes from a famous result of Valiant [10] showing that large lower bounds on the number of wires in a depth-2 circuit computing $A\vec{x}$ would give super-linear lower bounds on boolean log-depth circuits, thus resolving a more than 30 years old problem in boolean circuit complexity. This was one of the reasons why depth-2 circuits were considered by many authors, [1, 3, 5, 6, 7, 8, 9] among others.

[☆]Research supported by a DFG grant SCHN 503/4-1.

Email addresses: `jukna@thi.informatik.uni-frankfurt.de` (Stasys Jukna)

¹Current address: Universität Frankfurt, Institut für Informatik, Robert-Mayer-Str. 11-5, D-60054 Frankfurt, Germany.

Counting arguments show that, for a random matrix A , $\Omega(n^2/\ln n)$ wires are needed to compute $A\vec{x}$ using linear gates, even if there is no restriction on circuit depth. But no comparable lower bounds are known for *general* circuits computing linear transformations $A\vec{x}$: even in depth 2 the largest known lower bound remains of the form $\Omega(n \ln n)$ [7]. Hence, the following natural question arises:

Question 1. *Can the number of wires in a depth-2 circuit computing a linear transformation $A\vec{x}$ be substantially reduced by using non-linear boolean functions as gates?*

To approach this question, in this note we relax the problem and only require that the circuit correctly computes $A\vec{x}$ on unit vectors. We show that then, indeed, non-linear gates can substantially reduce the number of used wires.

Definition 1. A circuit *represents* a boolean matrix $A = (a_{ij})$ if it correctly computes the linear operator $A\vec{x}$ over $\text{GF}(2)$ on all n unit vectors² $\vec{e}_1, \dots, \vec{e}_n$; on other input vectors \vec{x} the circuit can output arbitrary values.

Hence, if $f = (f_1, \dots, f_n)$ is the operator computed by a circuit representing A , then the only requirement is that $f_i(\vec{e}_j) = a_{ij}$ must hold for all i and j .

Remark 1. This relaxation—instead of computing the whole transformation $A\vec{x}$ just try to correctly compute it on unit vectors—may be interesting in itself. Namely, if we would count *nodes* (not wires), then a proof that some explicit matrix A cannot be represented by a symmetric depth-2 circuit with $2^{(\ln \ln n)^{O(1)}}$ nodes on the middle layer would imply first super-polynomial lower bounds for so-called ACC circuits, and hence, re-solve another old problem in boolean circuit complexity (see [5], Problem 3). A depth-2 circuit is symmetric if each its output gate only depends on the number of 1's in its input; the gates on the middle layer are OR gates. An ACC circuit is a constant-depth circuit whose gates are NOT gates as well as unbounded fanin AND, OR and arbitrary modular gates MOD_m accepting the input iff the number of 1s in it is divisible by m .

Let us observe that, in the class of linear circuits, there is no difference between the representation of A and the computation of $A\vec{x}$: a linear circuit represents a matrix A if and only if it computes the entire linear transformation $A\vec{x}$. This holds, because the behavior of a linear circuit on all 2^n input vectors \vec{x} is completely determined by its behavior on n unit vectors: just write each input vector $\vec{x} = (x_1, \dots, x_n)$ as the sum $\vec{x} = x_1\vec{e}_1 \oplus \dots \oplus x_n\vec{e}_n$ and use the linearity of gates.

The observation implies that some $n \times n$ matrices A cannot be represented by *linear* circuits with fewer than $\Omega(n^2/\ln n)$ wires.

In this note we show that the situation changes drastically if we allow non-linear gates. For a matrix A , let $\text{dist}(A)$ denote the smallest Hamming distance between the columns of A .

Theorem 1. *Every an n by n $(0, 1)$ -matrix can be represented by a depth-2 circuit with $O(n \ln n)$ wires, and at least*

$$\Omega\left(\text{dist}(A) \cdot \frac{\ln n}{\ln \ln n}\right)$$

wires are always necessary.

²The j th unit vector is the vector $\vec{e}_j = (0, \dots, 0, 1, 0, \dots, 0)$ with precisely one 1 in the j th position.

Remark 2. Circuits we construct do not use the whole power of general depth-2 circuits: they contain $O(\ln n)$ parity gates on the middle layer and n multilinear polynomials over $\text{GF}(2)$ of degree $O(\log n)$ as gates on the output layer.

Remark 3. A $(0, 1)$ -Hadamard matrix H_n , for n being a power of 2, is defined inductively by

$$H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & \overline{H_n} \end{bmatrix},$$

where $\overline{H_n}$ is the matrix obtained from H_n by flipping all its entries. It is well known (and can be easily shown by induction on n) that every two columns of H_n differ in $d = n/2$ positions. By Theorem 1, $\Omega(n \ln n / \ln \ln n)$ wires are necessary to represent this matrix by a depth-2 circuit with arbitrary gates.

Remark 4. Although, in the case of non-linear circuits, the problem of matrix representation is significantly simpler, and indeed, the proof of the upper bound in Theorem 1 shows that it is simpler than the general question of computing the whole linear transformation $A\vec{x}$, we still get roughly the same lower bounds. This shows the weakness of the known lower bounds for the general case.

2. Proof of Theorem 1

2.1. Upper bound

We construct the desired depth-2 circuit representing $A = (a_{ij})$ as follows. Let m be the smallest even integer such that $\binom{m}{m/2} \geq n$; hence $m = O(\ln n)$. Take m middle nodes $V = \{v_1, \dots, v_m\}$. To each input variable x_j assign its own subset $S_j \subseteq V$ of $|S_j| = m/2$ middle nodes; hence, $S_{j_1} \subseteq S_{j_2}$ iff $j_1 = j_2$. Join x_j with all nodes in S_j . Finally, connect each $v \in V$ with all output nodes. The total number of wires is then $n(m/2) + nm = O(n \ln n)$.

Now we assign gates to the nodes. If v is a node on the middle layer connected to inputs x_{j_1}, \dots, x_{j_k} , then assign to v the gate $g_v = x_{j_1} \oplus \dots \oplus x_{j_k}$. To the i th output node we assign the gate

$$\phi_i = a_{i1}h_1 \oplus a_{i2}h_2 \oplus \dots \oplus a_{in}h_n, \quad \text{where } h_k = \prod_{v \in S_k} g_v.$$

Then

$$\begin{aligned} h_k(\vec{e}_j) &= 1 \text{ iff } g_v(\vec{e}_j) = 1 \text{ for all } v \in S_k \\ &\text{iff } x_j \text{ is connected to all nodes in } S_k \\ &\text{iff } S_k \subseteq S_j \\ &\text{iff } k = j. \end{aligned}$$

Hence, $h_j(\vec{e}_j) = 1$ and $h_k(\vec{e}_j) = 0$ for all $k \neq j$. Thus, if $f_i(\vec{x})$ is the function computed at the i th output gate then, for all $j = 1, \dots, n$, we have that

$$f_i(\vec{e}_j) = \phi_i(\vec{e}_j) = a_{i1} \cdot 0 \oplus \dots \oplus a_{ij} \cdot 1 \oplus \dots \oplus a_{in} \cdot 0 = a_{ij},$$

as desired. □

2.2. Lower bound

Let A be a fixed n by n $(0, 1)$ matrix. Our goal is to prove that then any general depth-2 circuit representing A must have $\Omega(\text{dist}(A) \cdot \ln n / \ln \ln n)$ wires, where $\text{dist}(A)$ is the smallest Hamming distance between any two columns of A .

A *sunflower* with k petals is a family S_1, \dots, S_k of k finite sets, each two of which share precisely the same set of common elements, called the *core* of the sunflower. That is, there is a set C (the core of the sunflower) such that

$$S_i \cap S_j = C \quad \text{for all } 1 \leq i < j \leq k.$$

In other words, each element belongs either to *none*, or to *exactly one*, to *all* of the S_i . The following result of Erdős and Rado [4] is well known.

Sunflower Lemma (Erdős–Rado). *Every family of $r!k^r$ sets, each of which has cardinality less than r , contains a sunflower with k petals.*

This lemma was used by Alon, Karchmer and Wigderson [2] to prove a super-linear lower bound on the number of wires in *linear* depth-2 circuits. Our proof is an extension of their argument to the *non-linear* case.

Fix a minimal depth-2 circuit with arbitrary gates representing a given matrix A . Without loss of generality, we may assume that there are no direct wires from inputs to outputs: this can be easily achieved by adding at most n new wires. Let x_1, \dots, x_n be its input nodes, and S_1, \dots, S_n be sets of their neighbors on the middle layer. Let f_1, \dots, f_n be the functions computed at the output nodes. Since the circuit represents A , we must have that $f_i(\vec{e}_j) = a_{ij}$ for all $1 \leq i, j \leq n$.

Let L_1 be the number of wires leaving the input nodes, and L_2 the number of wires entering the output nodes. Hence, $L_1 = \sum_{i=1}^n |S_i|$, and $L_1 + L_2$ is the total number of wires. Set

$$m := c \ln n / \ln \ln n \tag{1}$$

for a sufficiently small absolute constant $c > 0$. If we have $L_1 > mn$ wires leaving the input nodes, then we are done. So, assume that $L_1 \leq mn$. Our goal is to show that then we must have $L_2 \geq m \cdot \text{dist}(A)$ wires entering the output nodes.

Our assumption $\sum_{i=1}^n |S_i| \leq mn$ implies that at least $n/2$ of the sets S_i must be of size at most $r = 2m$. Hence, if the constant c in (1) is small enough then, by Sunflower Lemma, these sets must contain a sunflower with $s = 2m$ petals. Having such a sunflower with a core C , we can pair its members arbitrarily $(S_{p_1}, S_{q_1}), \dots, (S_{p_m}, S_{q_m})$; hence, $S_{p_i} \cap S_{q_i} = C$ for all $i = 1, \dots, m$. Important for us will only be that the symmetric differences

$$S_{p_i} \oplus S_{q_i} = (S_{p_i} \setminus S_{q_i}) \cup (S_{q_i} \setminus S_{p_i}) = (S_{p_i} \cup S_{q_i}) \setminus C$$

of these pairs of sets are mutually disjoint. Hence, we have m mutually disjoint subsets $S_{p_i} \oplus S_{q_i}$ of nodes on the middle layer, and we only have to show that each of these sets has at least $\text{dist}(A)$ outgoing wires: then $L_2 \geq m \cdot \text{dist}(A)$.

Fix one of the pairs (S_p, S_q) . Since the circuit represents the matrix A , the value $f(\vec{e}_j)$ of the computed operator $f = (f_1, \dots, f_n)$ on the j th unit vector must be the j th column of A . Since the Hamming distance between the p th and the q th columns of A must be at least $\text{dist}(A)$, there must exist a set I of $|I| \geq \text{dist}(A)$ rows such that

$$f_i(\vec{e}_p) \neq f_i(\vec{e}_q) \quad \text{for all } i \in I. \tag{2}$$

Claim 1. Every output f_i with $i \in I$ must be adjacent to at least one node in $S_p \oplus S_q$.

Proof. Let V be the set of all nodes on the middle layer. For a node $v \in V$, let $g_v(x_1, \dots, x_n)$ be the boolean function computed at this node. Claim 1 is a direct consequence of the following two simple observations about the behavior of the gates g_v on unit vectors. Let $\vec{0}$ denote the all-0 vector.

Observation 1. For every input node $j \in \{1, \dots, n\}$, we have that $g_v(\vec{e}_j) = g_v(\vec{0})$ iff the wire (j, v) is not present.

Proof. (\Leftarrow): If the wire (j, v) is not present, then g_v cannot depend on j th input variable x_j , and this is the only variable set to 1 by \vec{e}_j .

(\Rightarrow): Suppose that the wire (j, v) is present. To show that then $g_v(\vec{e}_j) \neq g_v(\vec{0})$, assume that $g_v(\vec{e}_j) = g_v(\vec{0})$. Then we can remove the wire (j, v) and replace g_v by a new boolean function g'_v obtained from g_v by fixing the j th variable x_j of g_v to 0. By our assumption $g_v(\vec{e}_j) = g_v(\vec{0})$, we have that $g'_v(\vec{e}_j) = g_v(\vec{0}) = g_v(\vec{e}_j)$, as \vec{e}_j has only one 1 in position j and the j th variable x_j is already set to 0 in g'_v . For the remaining unit vectors \vec{e}_k with $k \neq j$, we also have that $g'_v(\vec{e}_k) = g_v(\vec{e}_k)$, just because the j th position of \vec{e}_k is 0. Hence, we have removed one wire (j, v) , and the resulting circuit still represents A . This contradicts the minimality of our original circuit. \square

Observation 2. For all $v \notin S_p \oplus S_q$, we have that $g_v(\vec{e}_p) = g_v(\vec{e}_q)$.

Proof. If $v \notin S_p \cup S_q$, then neither the wire (p, v) nor the wire (q, v) is present. Observation 1 implies that then $g_v(\vec{e}_p) = g_v(\vec{0}) = g_v(\vec{e}_q)$.

If $v \in S_p \cap S_q$, then both wires (p, v) and (q, v) must be present. Observation 1 implies that then $g_v(\vec{e}_p) \neq g_v(\vec{0})$ as well as $g_v(\vec{e}_q) \neq g_v(\vec{0})$. Hence, in this case we also have that $g_v(\vec{e}_p) = g_v(\vec{e}_q)$, just because g_v can take only two values. \square

To finish the proof of Claim 1, take the boolean function f_i computed at the i th output gate with $i \in I$. The value of f_i only depends on the values of gates g_v computed at the nodes on the middle layer. Hence, if there would be no wire from a node in $S_p \oplus S_q$ to the i th output f_i , then Observation 2 would imply that all gates on the middle layer, connected to f_i , would output the *same* values on input vectors \vec{e}_p and \vec{e}_q . But this would imply $f_i(\vec{e}_p) = f_i(\vec{e}_q)$, a contradiction with (2).

This completes the proof of Claim 1. \square

By Claim 1, for each of m pairs (S_{p_i}, S_{q_i}) of subsets of nodes on the middle layer, there must be at least $|I| \geq \text{dist}(A)$ wires going from the vertices in $S_{p_i} \oplus S_{q_i}$ to the output layer. Since the sets $S_{p_i} \oplus S_{q_i}$, $i = 1, \dots, m$, are mutually disjoint, the total number of wires going from the middle layer to the output layer must be at least $m \cdot \text{dist}(A)$, as desired. \square

3. Conclusion

Motivated by there being no higher than $\Omega(n \ln n)$ lower bounds on the number of wires in a general (and even, linear) depth-2 circuit computing an explicit linear operator $A\vec{x}$, we relax the problem and only require that the circuit correctly computes the operator on n unit vectors. Although for *linear* circuits this is, in fact, no relaxation, we show that the

situation changes drastically if we allow *arbitrary* boolean functions as gates. We show that then about $n \ln n$ wires—instead of about $n^2 / \ln n$ as in the linear case—are enough to represent any matrix A by a depth-2 circuit. This does not answer Question 1 because in our construction essential was that the circuit only needs to correctly compute $A\vec{x}$ on n vectors \vec{x} . This way only $O(\ln n)$ on the middle layer are enough. Would we require the circuit to compute $A\vec{x}$ on all vectors \vec{x} then $\text{rank}(A)$ nodes on the middle layer would be necessary: the operator computed at these nodes must take at least $2^{\text{rank}(A)}$ distinct values. Still, our result indicates that non-linear gates can (apparently) help to compute linear operators.

We also show that some (explicit) matrices A cannot be represented by depth-2 circuits with fewer than $n \ln n / \ln \ln n$ wires, even if arbitrary boolean functions are allowed as gates. Since the highest known lower bounds for circuits computing the whole transformation $A\vec{x}$ are also of the form $\Omega(n \ln n)$, this shows that larger lower bounds can only be achieved when analyzing the behavior of a circuit on larger sets of input vectors, not just on n unit vectors.

Our upper bound holds over any finite field, but in the proof of the lower bound (see Observation 2) the underlying field being $\text{GF}(2)$ was important. It would be interesting to extend the lower bound to other fields. It would be also interesting to eliminate the “annoying” $1 / \ln \ln n$ factor between the upper and lower bounds.

References

- [1] N. Alon, P. Pudlák, Superconcentrators of depth 2 and 3; odd levels help (rarely), *J. Comp. Sys. Sci.* 48 (1994), 194-202.
- [2] N. Alon, M. Karchmer, A. Wigderson, Linear circuits over $\text{GF}(2)$, *SIAM. J. Comput.* 19(6) (1990) 1064-1067.
- [3] D. Y. Cherukhin, The lower estimate of complexity in the class of schemes of depth 2 without restrictions on a basis, *Moscow Univ. Math. Bull.* 60(4) (2005) 42–44.
- [4] P. Erdős, R. Rado, Intersection theorems for systems of sets, *J. London Math. Soc.* 35 (1960) 85–90.
- [5] S. Jukna, Entropy of operators or why matrix multiplication is hard for depth-two circuits, *Theory of Comp. Syst.* (2008) doi: 10.1007/s00224-008-9133-y.
- [6] N. Pippenger, Superconcentrators of depth 2, *J. Comput. Syst. Sci.* 24 (1982) 82-90.
- [7] P. Pudlák, Communication in bounded depth circuits, *Combinatorica* 14 (2) (1994) 203-216.
- [8] P. Pudlák, P. Savický, On shifting networks, *Theor. Comput. Sci.* 116 (1993) 415-419.
- [9] J. Radhakrishnan, A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators, *SIAM J. Discrete Math.* 13(1) (2000) 2–24.
- [10] L. Valiant, Graph-theoretic methods in low-level complexity, in: *Proc. 6th Conf. on Math. Foundations of Comput. Sci.*, Springer Lect. Notes in Comput. Sci., vol. 53 (1977) pp. 162–176.