Lower Bounds on Formula Size of Error-Correcting Codes *

Arist Kojevnikov and Alexander S. Kulikov

St. Petersburg Department of Steklov Institute of Mathematics
27 Fontanka, 191023 St.Petersburg, Russia
http://logic.pdmi.ras.ru/{~arist,~kulikov}/

Abstract. We show that every formula over the basis $\{\wedge, \vee, \neg\}$ for a function $f : \{0,1\}^n \to \{0,1\}$, such that $\forall x, y \in f^{-1}(1), d(x,y) \geq 2d+1$, has size

$$\Omega(n^{d+2}\frac{|f^{-1}(1)|}{|f^{-1}(0)|}) \ .$$

This immediately implies a lower bound $\Omega(n^2)$ for a characteristic function of a BCH code of distance $2d+1$. The main technique used is estimating the number of monochromatic rectangles needed to cover a matrix.

1 Introduction

One of the most important problems in theoretical computer science is proving lower bounds for various computational models. Boolean circuits is probably the simplest such model. A Boolean circuit has n inputs, one output, interior gates that are labeled by Boolean functions (usually, these are $\wedge$, $\vee$ and $\neg$) and computes in a natural way a function $f : \{0,1\}^n \to \{0,1\}$. By general counting arguments it is possible to show that almost every Boolean function has exponential circuit complexity. Despite of this no nonlinear lower bound on the circuit size of an explicit Boolean function is known. Some progress however has been made in restricted settings.

Razborov [2] proved a superpolynomial lower bound on the monotone complexity of the clique function. Exponential lower bounds are also known for constant depth circuits (see, e.g., [3] and references therein).

In this paper we consider another restricted case of Boolean circuits, namely Boolean formulas. A formula is just a circuit whose underlying graph is a tree. While a formula is weaker than a circuit, it is known [4] that for any Boolean function the minimal depth $D(f)$ of a circuit and the logarithm of the minimal size $L(f)$ of a formula computing the same function f have the same asymptotic behavior, i.e., $\forall f : \{0,1\}^n \to \{0,1\}$,

$$D(f) = \Theta(\log L(f)) \ .$$

* After we finished this paper, we were told that the result was proved (by more simple method) in [1]. However, you may still want to read our draft to get the details of the proof (if, for example, you do not know Russian).

There are several general methods for proving lower bounds for the formula size. For example, the method of Fischer, Meyer and Paterson [5] uses the fact that simple functions have simple subfunctions, the Nečiporuk's method [6] gets rid of the fact that a simple function has only a few subfunctions, the method by Khrapchenko [7] counts the number of pairs (x, y) such that $f(x) = 0$ and $f(x) = 1$ and $d(x, y) = 1$ (where $d(x, y)$ is the Hamming distance of words x and y). See [8] for a survey of these and other methods.

In this paper, we prove lower bounds for functions f with $f^{-1}(1)$ having minimal distance bounded below by a constant. Namely, we prove an $\Omega(n^{d+2}\frac{|f^{-1}(1)|}{|f^{-1}(0)|})$ lower bound on the formula size of any Boolean function $f : \{0,1\}^n \to \{0,1\}$ such that $\forall x, y \in f^{-1}(1), d(x, y) \geq 2d+1$. A natural candidate for a lower bound given by our method is the characteristic function of a BCH code. We get an $\Omega(n^2)$ lower bound for this function (and this is actually the best bound that can be obtained by using our theorem).

Our result is inspired by a recent paper by Lee [9]. He presented a new rank technique for proving lower bounds on the formula size. It is based on estimating the rank of a certain relation matrix. We generalize Lee's method of estimating the number of monochromatic rectangles needed to cover a matrix.

2 General Setting

2.1 Formula Size

By B_n we denote the set of Boolean functions $g: \{0,1\}^n \to \{0,1\}$. Let $V = \{x_1, \ldots, x_n\}$ be a set of Boolean variables. A *formula* over the *basis* $\Omega \subseteq B_1 \cup B_2$ is a rooted tree whose leaves are labelled with variables from V and nodes are labelled with functions from Ω. The size of a formula is the number of leaves in the tree. For a function $f \in B_n$, the *formula size* of f, denoted by $L_\Omega(f)$, is the minimum size of a formula over Ω which computes f. The two frequently used bases are

- $U = \{\vee, \wedge, \neg\}$,
- $B = \{$all the functions of 1 and 2 variables$\}$.

In this paper we consider the basis U.

2.2 Communication Complexity

There is a strong connection between formula size and communication complexity of a function [10]. Let X and Y be two disjoint subsets of $\{0,1\}^n$. A communication protocol between Alice and Bob on $X \times Y$ is a binary tree, where each internal node v is labelled either by a function $a_v: X \to \{0,1\}$ or by a function $b_v: Y \to \{0,1\}$ and each leaf is labelled by an integer $i \in [1..n]$. By $C(X, Y)$ we denote the minimum number of leaves in a protocol that for any $(x, y) \in X \times Y$ outputs a coordinate i such that $x_i \neq y_i$.

Theorem 1 ([10]). *For every function $f \in B_n$,*

$$L_U(f) = C(f^{-1}(0), f^{-1}(1)) \ .$$

2.3 Covering by Rectangles

By *selection function* (or *selection matrix*) for $X \times Y$ we mean a function that maps a pair (x, y) to a coordinate at which x and y differ. By *rectangle* in $X \times Y$ we mean a set that can be represented as $X' \times Y'$ for some $X' \subseteq X$, $Y' \subseteq Y$. We say that a rectangle is *monochromatic* w.r.t. a selection function S, if S is constant on this rectangle. By $R_S(X, Y)$ we denote the minimal number of monochromatic rectangles needed to cover $X \times Y$. Let also $R(X, Y) = \min_S R_S(X, Y)$. It is not difficult to show [11] that $R(X, Y) \leq C(X, Y)$ (as every communication protocol defines a covering in a natural way).

For a 0/1-matrix M, by $R_1(M)$ we denote the minimal number of rectangles needed to cover all 1's of M and by $T_1(M)$ we denote the total number of 1's in M. We say that a 0/1-matrix $M[X, Y]$ contains a 0/1-matrix $M_0[X', Y']$ if $X' \subseteq X$, $Y \subseteq Y'$ and $\forall (x, y) \in X' \times Y'$, $M_0[x, y] = 1$ implies $M[x, y] = 1$. Clearly, $R_1(M[X, Y]) \geq R_1(M_0[X', Y'])$.

Lemma 1. *Let $M[X, X]$ be a 0/1-matrix and suppose that it contains a matrix $M_0[X, X]$, such that each column and each row of M_0 contain exactly one 1 (i.e., M_0 is a permutation matrix). Then*

$$R_1(M) \cdot T_1(M) \geq |X|^2 \ .$$

Proof. The proof is by induction on n. The base case ($n = 1$) is trivial. For the induction step consider some covering of M by rectangles. Let $X' \times Y'$ be a rectangle of this covering and suppose w.l.o.g. that $|X'| \leq |Y'|$. Note that the matrix $M[X \backslash X', X \backslash X']$ satisfies the condition of the lemma. Let $|X| = n$, $|X'| = n_0$, $R_1(M[X, X]) = r$, $T_1(M[X, X]) = t$. Then, by induction,

$$\begin{aligned} rt &= R_1(M[X, X]) \cdot T_1(M[X, X]) \geq \\ &\geq (1 + R_1(M[X \backslash X', X \backslash X'])) \cdot T_1(M[X, X]) \geq (1 + \frac{(n - n_0)^2}{t - {n_0}^2})t \ . \end{aligned}$$

Thus, $rt \geq n^2$ if $t(t - {n_0}^2) + (n - n_0)^2 \geq n^2(t - n_0^2)$. It is easy to see that the last inequality is equivalent to $(t - n_0 n)^2 \geq 0$. □

Intuitively, this lemma says that by adding a few 1's to a permutation matrix it is not possible to reduce greatly the number of rectangles needed to cover all its 1's. The following lemma is a simple extension of this fact.

Lemma 2. *Let $M[X, Y]$, where $k|X| = |Y|$, be a 0/1-matrix and suppose that it contains a matrix $M_0[X, Y]$, such that each column of M_0 contains exactly one 1 and each row of M_0 contains exactly k 1's. Then*

$$R_1(M[X, Y]) \cdot T_1(M[X, Y]) \geq k|X|^2 \ .$$

Proof. It is easy to see that Y can be represented as $Y_1 \cup Y_2 \cup \ldots \cup Y_k$, such that for any i, $|Y_i|{=}|X|$ and the matrix $M[X, Y_i]$ satisfies the condition of Lemma 1 (see Fig. 1). Thus, for each Y_i,

$$R_1(M[X,Y]) \cdot T_1(M[X,Y_i]) \geq R_1(M[X,Y_i]) \cdot T_1(M[X,Y_i]) \geq |X|^2 \ .$$

By summing up all these inequalities one gets the required inequality. □

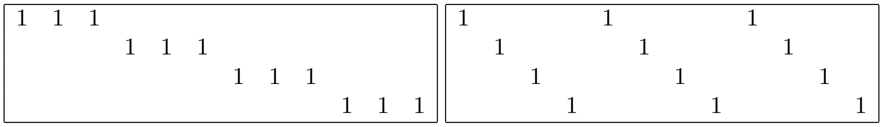

Fig. 1. Matrices with blocks of 1's.

For a selection function S, we define functions $S_1, \ldots, S_n \colon X \times Y \to \{0,1\}$ as follows: $S_i(x,y) = 1$ iff $S(x,y) = i$. Clearly, $R_S(X,Y) = \sum_{i=1}^{n} R_1(S_i)$ and so $R(X,Y) = \min_S \sum_{i=1}^{n} R_1(S_i)$. In our main result we apply Lemma 2 to estimate this sum.

3 The Main Result

For $x, y \in \{0,1\}^n$, let $\mathrm{diff}(x,y) = |\{i : x_i \neq y_i\}|$; $d(x,y) = |\mathrm{diff}(x,y)|$ is the Hamming distance of x and y. We say that a set $C \subseteq \{0,1\}^n$ has *minimal distance* d, if $\forall x, y \in C, d(x,y) \geq d$.

Theorem 2. *Let $d \geq 1$ be an integer constant, $f \in B_n$, $A_0 \subseteq f^{-1}(0)$, $A_1 \subseteq f^{-1}(1)$. If A_1 has minimal distance $2d+1$, then*

$$L_U(f) = \Omega(n^{d+2}\frac{|A_1|}{|A_0|}) \ .$$

Let us first give some informal ideas for proving this theorem. Let $C \subset \{0,1\}^n$ be an error-correcting code of distance 3, i.e., $\forall x, y \in C, d(x,y) \geq 3$. For example, we can consider the Hamming code. Then, for $n = 2^t - 1$, $|\overline{C}| = n|C|$. Let $f \in B_n$ be the characteristic function of the set C. Note that there are $\Omega(n^2|C|)$ pairs (x,y), such that $x \in C$, $y \notin C$ and $d(x,y) = 2$ (as by flipping any two bits in a Hamming word one gets a non-Hamming word). Now consider a selection matrix S and let 0/1-matrices M_i, $1 \leq i \leq n$, be defined as follows: $M_i[x,y] = 1$ iff $S[x,y] = i$ and $d(x,y) = 2$. Obviously, for each i, S_i contains M_i. Moreover, each column of M_i contains at most one 1: if $M_i[x_1,y] = M_i[x_2,y] = 1$, then $\mathrm{diff}(x_1,y) = \{i,j\}$, $\mathrm{diff}(x_2,y) = \{i,k\}$ (for some j,k) and hence $\mathrm{diff}(x_1,x_2) \subseteq \{j,k\}$. Fig. 2 shows the selection matrix for the Hamming code for $n = 7$, marked cells indicate all possible places where the matrix M_1 for this code can have 1's.

	0000001	0000010	0000011	0000100	0000101	0000110	0000111	0001000	0001001	0001010	0001011	0001100	0001101	0001110	0010000	0010001	0010010	0010011	0010100	0010101	0010111	0011000	0011010	0011011	0011100
0000000																									
0001111																									
0010110																									
0011001																									
0100101																									
0101010																									
0110011																									
0111100																									
1000011	●	●					●				●							●							
1001100				●				●					●	●											●
1010101					●											●			●		●				
1011010										●							●					●		●	
1100110						●																			
1101001									●																
1110000															●										
1111111																									

	0011101	0011110	0011111	0100000	0100001	0100010	0100011	0100100	0100110	0100111	0101000	0101001	0101011	0101100	0101101	0101110	0101111	0110000	0110001	0110010	0110100	0110101	0110110	0110111	0111000
0000000																									
0001111																									
0010110																									
0011001																									
0100101																									
0101010																									
0110011																									
0111100																									
1000011							●																		
1001100														●											
1010101	●																					●			
1011010		●																							
1100110						●		●		●						●							●		
1101001					●						●		●		●										
1110000				●															●	●	●				●
1111111			●														●							●	

	0111001	0111010	0111011	0111101	0111110	0111111	1000000	1000001	1000010	1000100	1000101	1000110	1000111	1001000	1001001	1001010	1001011	1001101	1001110	1001111	1010000	1010001	1010010	1010011	1010100
0000000								●	●	●				●							●				
0001111													●				●	●	●						
0010110												●											●		●
0011001															●							●			
0100101											●														
0101010																●									
0110011																								●	
0111100																									
1000011																									
1001100																									
1010101																									
1011010		●																							
1100110																									
1101001	●																								
1110000																									
1111111			●	●	●																				

	1010110	1010111	1011000	1011001	1011011	1011100	1011101	1011110	1011111	1100000	1100001	1100010	1100011	1100100	1100101	1100111	1101000	1101010	1101011	1101100	1101101	1101110
0000000										●												
0001111									●													
0010110		●						●														
0011001			●		●		●															
0100101											●			●		●					●	
0101010												●					●		●			●
0110011													●									
0111100						●														●		
1000011																						
1001100																						
1010101																						
1011010																						
1100110																						
1101001																						
1110000																						
1111111																						

By the above arguments we know that the total number of 1's in all M_i's is $\Omega(n^2|C|)$. This means that some constant fraction of M_i's contains $\Omega(n|C|)$ 1's. From this we can conclude that in these M_i's we can find matrices of size approximately $|C| \times n|C|$ satisfying the condition of Lemma 2 (more precisely, containing exactly n 1's in each string and exactly one 1 in each column). Fig. 3 shows an example of such a matrix.

Now, the total number of 1's in all S_i's is $|C||\overline{C}| = n|C|^2$ (as we need to cover by monochromatic rectangles a matrix of size $|C| \times |\overline{C}|$). Then some constant fraction of S_i contains $O(|C|^2)$ 1's. Overall, we have a constant fraction of S_i, each of which contains at most $O(|C|^2)$ 1's and contains also a matrix of size $|C| \times n|C|$ satisfying the condition of Lemma 2. Thus, each of this matrices needs $\Omega(n)$ monochromatic rectangles and the $\Omega(n^2)$ lower bound for f follows.

Before formalizing these ideas we prove two simple technical facts. Stated informally, they say that if a sum of several elements is fixed, then many of these elements are not big (compared to the sum) and vice versa.

Lemma 3. *1. Let $\sum_{i=1}^m x_i \leq S$, $0 < \Delta < 1$. If $c(1-\Delta)m \geq 1$, then*

$$|\{i : x_i \leq cS\}| \geq \Delta m \ .$$

2. Let $\sum_{i=1}^m x_i \geq S$, $0 < \Delta < 1$. If $\forall i, x_i \leq X$ and $\Delta m X + mcS \leq S$, then

$$|\{i : x_i \geq cS\}| \geq \Delta m \ .$$

Proof. 1. If $|\{i : x_i \leq cS\}| < \Delta m$, then $|\{i : x_i > cS\}| \geq (1-\Delta)m$ and $\sum_{i=1}^m x_i > cS(1-\Delta)m \geq S$.

2. If $|\{i : x_i \geq cS\}| < \Delta m$, then at most Δm of x_i's can be equal to X and all the remaining x_i's do not exceed cS. Thus, $\sum_{i=1}^m x_i < \Delta m X + mcS \leq S$.

□

Proof (Theorem 2). Consider a selection function S for $A_1 \times A_0$. It is sufficient to prove the stated lower bound for $\sum_{i=1}^n R_1(S_i)$.

Let $M_i[A_1, A_0]$ $(1 \leq i \leq n)$ be a 0/1-matrix defined as follows: $M_i(x,y) = 1$ iff $S_i(x,y) = 1$ and $d(x,y) = d+1$ (so, S_i contains M_i). Let also $s_i = T_1(S_i)$, $m_i = T_1(M_i)$. It is easy to see that each column of M_i contains at most one 1: if $M_i[x_1,y] = 1$ and $M_i[x_2,y] = 1$, then $d(x_1,x_2) \leq 2d$.

Clearly,

$$\sum_{i=1}^n s_i = |A_1||A_0| \ , \tag{1}$$

$$\sum_{i=1}^n m_i = C_n^{d+1}|A_1| \text{ and } \forall i, m_i \leq C_n^d|A_1| \ . \tag{2}$$

Let

$$\alpha = \frac{8d+8}{n-2d},\ \ \beta = \frac{1}{4n},\ \ \gamma = \frac{1}{8n} \ ,$$

$$\Delta_1 = \frac{(8d+7)n+2d}{(8d+8)n},\ \ \Delta_2 = \frac{2n-2d}{(8d+8)n},\ \ \Delta_3 = \frac{n-d}{(8d+8)n} \ .$$

By Lemma 3,

$$|\{i : s_i \leq \alpha|A_1||A_0|\}| \geq \Delta_1 n \ , \tag{3}$$

$$|\{i : m_i \geq \beta C_n^{d+1}|A_1|\}| \geq \Delta_2 n \ . \tag{4}$$

Let

$$I = \{i : s_i \leq \alpha|A_1||A_0| \text{ and } m_i \geq \beta C_n^{d+1}|A_1|\} \ . \tag{5}$$

It follows from (3) and (4) that $|I| \geq (\Delta_1 + \Delta_2 - 1)n$.

Now consider a matrix M_i for $i \in I$. For $1 \leq j \leq |A_1|$, let b_j denote the number of 1's in the j-th row of M_i. By (5) we know that $\sum_{j=1}^{|A_1|} b_j \geq \beta C_n^{d+1}|A_1|$. It is easy to see also that for any j, $b_j \leq C_n^d$. Thus,

$$|\{j : b_j \geq \gamma C_n^{d+1}\}| \geq \Delta_3|A_1| \ . \tag{6}$$

This means that M_i contains a submatrix of size $\Delta_3|A_1| \times \Delta_3|A_1|\gamma C_n^{d+1}$, such that each row of this submatrix contains exactly γC_n^{d+1} 1's and each column contains exactly one 1. By Lemma 2,

$$R_1(S_i) \geq \frac{\Delta_3^2|A_1|^2\gamma C_n^{d+1}}{s_i} \geq \frac{\Delta_3^2|A_1|\gamma C_n^{d+1}}{\alpha|A_0|} \ .$$

Thus,

$$\sum_{i=1}^{n} R_1(S_i) \geq \sum_{i\in I} R_1(S_i) \geq (\Delta_1 + \Delta_2 - 1)n\Delta_3^2 C_n^{d+1}\frac{\gamma}{\alpha}\frac{|A_1|}{|A_0|} \ . \tag{7}$$

Finally, since $(\Delta_1 + \Delta_2 - 1) = (8d + 8)^{-1}$, $\Delta_3 \sim 1$, $C_n^{d+1} \sim n^{d+1}$, $\gamma/\alpha \sim 1$, by (7),

$$\sum_{i=1}^{n} R_1(S_i) = \Omega(n^{d+2}\frac{|A_1|}{|A_0|}) \ .$$

□

It is easy to see that Theorem 2 cannot prove lower bounds greater than $\Omega(n^2)$, since if $f^{-1}(1)$ has minimal distance $2d + 1$, then $|f^{-1}(1)| = O(2^n/C_n^d)$. From the other hand, it is well-known that a BCH error-correcting code [12] has size $\Omega(2^n/C_n^d)$.

Corollary 1. *Let C_n be a BCH error-correcting code of minimal distance $2d+1$ and f be the characteristic function of C_n. Then $L_U(f) = \Omega(n^2)$.*

4 Open Problems and Further Directions

The question on exact formula complexity of error-correcting codes remains open. For example, a straightforward upper bound for the Hamming codes is $O(n^2 \log n)$, as these codes are defined by $\log n$ parities and it is known that the complexity of parity is $\Theta(n^2)$, while the lower bound given by our method is $\Omega(n^2)$.

Note that to prove lower bounds on the number of monochromatic rectangles we use quite a simple criteria (Lemma 2). It would be interesting to find more powerful ones.

Acknowledgments

We would like to thank Edward A. Hirsch and Pavel Pudlák for valuable comments.

References

1. Rychkov, K.L.: A modification of khrapchenko's method and its applications to bounds on the complexity of pi-schemes and coding functions. Metody Diskretnogo Analiza v theorii graphov i skhem **42** (1985) 91–98 (in Russian).
2. Razborov, A.A.: Lower bounds for the monotone complexity of some Boolean functions. Dokl. Akad. Nauk SSSR **281**(4) (1985) 798–801 (in Russian). English translation in *Soviet Math. Dokl.* 31:354–357, 1985.
3. Paturi, R., Saks, M.E., Zane, F.: Exponential lower bounds for depth three boolean circuits. Computational Complexity **9**(1) (2000) 1–15
4. Spira, P.M.: On time-hardware complexity tradeoffs for boolean functions. In: Proceedings of the 4th Hawaii Symposium on System Sciences, Western Periodicals Company, North Hollywood (1971) 525–527
5. Fischer, M.J., Meyer, A.R., Paterson, M.S.: $\Omega(n \log n)$ lower bounds on length of boolean formulas. SIAM Journal of Computation **11**(3) (1982) 416–427
6. Nečiporuk, E.I.: On a boolean function. Dokl. Akad. Nauk SSSR **169**(4) (1966) 765–767 (in Russian). English translation in *Soviet Math. Dokl.* 7:999-1000, 1966.
7. Khrapchenko, V.M.: A method of obtaining lower bounds for the complexity of π-schemes. Math. Zamet. **10**(1) (1971) 83–92 (in Russian). English translation in *Math. Notes Acad. Sci. USSR* 10:474–479, 1972.
8. Wegener, I.: The Complexity of Boolean Functions. John Wiley & Sons, Inc. (1987)
9. Lee, T.: A new rank technique for formula size lower bounds. In: Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science (STACS '2007). Volume 4393 of LNCS., Springer (2007)
10. Karchmer, M., Wigderson, A.: Monotone circuits for Connectivity require super-logarithmic depth. SIAM Journal on Discrete Mathematics **3**(2) (1990) 255–265
11. Kushilevitz, E., Nisan, N.: Communication complexity. Cambridge University Press (1996)
12. Chen, X., Reed, I.S.: Error-Control Coding for Data Networks. Kluwer Academic Publishers (1999)