Lower Bounds for Unrestricted Boolean Circuits: Open Problems

Alexander S. Kulikov(✉)

St. Petersburg Department of Steklov Institute of Mathematics,
St. Petersburg, Russia
kulikov@logic.pdmi.ras.ru

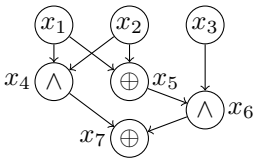
Abstract. To prove that P $\neq$ NP, it suffices to prove a superpolynomial lower bound on Boolean circuit complexity of a function from NP. Currently, we are not even close to achieving this goal: we do not know how to prove a $4n$ lower bound. What is more depressing is that there are almost no techniques for proving circuit lower bounds.

In this note, we briefly review various approaches that could potentially lead to stronger linear or superlinear lower bounds for unrestricted Boolean circuits (i.e., circuits with no restriction on depth, fan-out, or basis).

1 Computational Model: Boolean Circuits

A *straight-line program* is a simple and natural program for computing a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$. The input to such a program is variables $x_1, \ldots, x_n$ and each line of the program computes the value of a new Boolean variable by applying a binary Boolean operation to some of two previous variables. A *circuit* is a convenient way of representing a straight-line program as a directed acyclic graph. Below we show an example of a program and the corresponding circuit of size four for the majority function on three input bits x_1, x_2, x_3 (that outputs 1 iff $x_1 + x_2 + x_3 \geq 2$).

$$x_4 = x_1 \wedge x_3$$
$$x_5 = x_1 \oplus x_2$$
$$x_6 = x_5 \wedge x_3$$
$$x_7 = x_4 \oplus x_6$$



To prove that P $\neq$ NP, it suffices to find a Boolean function from NP that cannot be computed by polynomial size circuits (more precisely, a family of functions $\{f_n\}_{n=1}^{\infty}$ such that f_n has n inputs, $\bigcup_{n=1}^{\infty} f_n^{-1}(1) \in$ NP, and circuit size of f_n grows superpolynomially in n). This problem turned out to be extremely difficult: we do not know how to prove $4n$ lower bound, not to mention superlinear or superpolynomial lower bounds. Most of the known lower bounds are proved using the so-called gate elimination method which is difficult to use to

F. V. Fomin and V. V. Podolskii (Eds.): CSR 2018, LNCS 10846, pp. 15–22, 2018.
https://doi.org/10.1007/978-3-319-90530-3_2

beat the $4n$ barrier. In the rest of this note, we briefly review various approaches that could potentially lead to stronger linear or superlinear lower bounds. The focus of this note is unrestricted circuit model: we do not pose any restriction on the depth of a circuit, the fan-out of its gates, or the basis of allowed operations computed at gates. For various restricted circuit classes, such as monotone circuits (circuits using $\wedge$ and $\vee$ operations only), constant depth circuits, and formulas (circuit of fan-out 1), much stronger lower bounds are known. What is more important, various beautiful techniques for proving such lower bounds have been developed. An exposition of these bounds and techniques can be found in an excellent recent book by Jukna [1].

2 Lower Bounds: Approaches and Open Problems

Notation

We use $B_{n,m}$ to denote the set of all Boolean functions with n inputs and m outputs. By default, we will assume that $m = 1$, that is, we consider Boolean *predicates*. We use B_n as a shortcut for $B_{n,1}$. By a function f we mean (unless stated otherwise) a family of functions: $f = \{f_n \colon f_n \in B_n\}_{n=1}^{\infty}$.

2.1 Known Lower Bounds and Gate Elimination Method

How to prove, say, a $3n - o(n)$ lower bound for a Boolean function f? One way to do this is by induction: first show that f is resistant to $n - o(n)$ substitutions of some type (say, $x_i \leftarrow c$, where $c \in \{0, 1\}$, or $x_i \leftarrow \bigoplus_{j \in J} x_j \oplus c$); then show that for any circuit computing f one can find a substitution eliminating at least three gates. This type of argument is known as *gate elimination* and it is used in most of the known lower bounds proofs, in particular, in the proof of the currently strongest lower bound $(3 + 1/86)n - o(n)$ for affine dispersers by Find et al. [2]. A gate elimination proof usually consists of many cases depending on how the top part of a circuit looks like. The stronger is the lower bound the larger is the number of cases: if one wants to prove, say, $4n$ lower bound, one needs to carefully check that no two of the four gates eliminated at each iteration coincide. This makes gate elimination proofs quite tedious. Moreover, it was recently shown by Golovnev et al. [3] that certain formalizations of the gate elimination method are not able to prove stronger than cn lower bounds for a small constant c. For example, they constructed a simple function f such that no substitution of the form $x_i \leftarrow g$, where g is an *arbitrary* function on all the remaining variables, can reduce the circuit size of f by more than 5. *Can one at least prove a $4n$ lower bound using gate elimination?*

Further reading. An exposition of the proofs based on the gate elimination method is given by Wegener [4, Chap. 5]. A more recent survey is given by Golovnev et al. [3, Sect. 2].

2.2 Multi-output Functions

Can one prove stronger lower bounds for functions with multiple outputs? In this case, we assume that for each output of such a function, a circuit contains a gate computing this output. Computing several functions simultaneously is definitely not easier than computing any one of them. However, currently, we do not know how to exploit this fact in lower bounds proofs: the strongest lower bound for functions with $o(n)$ outputs is the same as for functions with a single output (up to additive $o(n)$ terms). When the number of outputs becomes linear, one can use the following observation by Lamagna and Savage [5]: the circuit complexity of computing k different functions $f_1, \ldots, f_k \in B_n$ simultaneously is at least $(\min_i \text{gates}(f_i) - 1) + k$. This is just because none of the topologically first $\min_i \text{gates}(f_i) - 1$ gates can compute any of the outputs and one needs at least k gates to compute all outputs. This allows one to prove $(c+1)n - O(1)$ lower bounds for functions from $B_{n,n}$ from cn lower bounds for functions from $B_{n,1}$: given $f \in B_n$, consider $g = (g_1, \ldots, g_n) \in B_{n,n}$ where $g_i(x) = f(x) \oplus x_i$; then, $\text{gates}(g_i) \geq \text{gates}(f) - 1$ and hence $\text{gates}(g) \geq \text{gates}(f) + n - 2$. *How to prove a 5n lower bound for a function from* $B_{n,n}$*?*

Further reading. A survey of lower bounds for multi-output functions is given by Hiltgen [6, Chap. 4].

2.3 Non-gate-Elimination Lower Bounds

Are there approaches other than gate elimination for proving lower bounds for unrestricted circuits. There are a few lower bounds that are *not* based on gate elimination techniques. Alas, none of them is currently known to give a stronger than $2n$ lower bound. Blum and Seysen [7] proved that *any* optimal circuit that computes simultaneously AND and NOR of n input bits consists of two formulas (that is, each output is computed by a tree) and hence has size $2n - 2$. Note that the gate elimination method with bit-fixing substitutions cannot be used for this particular function: assigning a constant to an input variable immediately trivializes one of the two output functions (and one loses a possibility to proceed by induction). Melanich [8] came up with a similar, but simpler argument. She considered the following multi-output function from $B_{n,o(n)}$: there are $n = \binom{k}{2}$ inputs $x_{\{i,j\}}$, where $1 \leq i \neq j \leq k$, and $k = o(n)$ outputs; the i-th output computes the AND of variables $\{x_{\{i,j\}}\}_{j \neq i}$. Each input contributes to two outputs and hence the function can be computed by a circuit of size $2n - o(n)$. Melanich proves that this straightforward circuit is optimal by showing that in any circuit (computing this function) one can reduce the number of gates shared between several outputs without increasing the size of the circuit. Chashkin [9] proved a $2n - o(n)$ for a function $f \in B_{n,\log_2 n}$ that has the form $f(x) = Ax$ where the matrix $A \in \{0,1\}^{\log_2 n \times n}$ has n pairwise distinct columns. He showed that any circuit computing this function has at least $n - o(n)$ branching gates (i.e., gates of out-degree at least 2). The lower bound then follows by counting the number of edges. *Can any of these non-gate-elimination methods be extended to get stronger than 2n lower bounds?*

2.4 Symmetric Functions

Can one prove a superlinear lower bound for a symmetric function (i.e., a function whose output depends on the sum of input bits only)? In fact, one cannot: while basic symmetric functions like parity, MOD_3, and majority are used to prove superpolynomial lower bounds in, e.g., constant depth circuit model, any symmetric function can be computed by a circuit of size $4.5n + o(n)$ as shown by Demenkov et al. [10]. The strongest known lower $2.5n - O(1)$ is proved by Stockmeyer [11]. Hence, it is not excluded that there exist symmetric functions of circuit size, say, $4n$. Note that the multi-output function $\mathrm{SUM}_n \in B_{n,\lceil \log_2(n+1) \rceil}$ that outputs the binary encoding of the sum of n input bits is not easier than any symmetric function $f \in B_n$: f can be computed by a circuit of size $\mathrm{gates}(\mathrm{SUM}_n) + o(n)$. *What is the circuit size of* SUM_n*?*

Further reading. Known lower and upper bounds on complexity of symmetric functions in various models are summarized in Jukna's book [1, end of Chap. 1].

2.5 Satisfiability Algorithms

Given a circuit with n inputs, how hard is it to find an assignment making this circuit to output 1? Williams [12] recently developed a general framework of getting circuit lower bounds from faster than brute force search satisfiability algorithms. Extending Williams' results, Jahanjou et al. [13] proved that one can prove a $2cn$ lower bound (for a function from $B_{n,2}$) by designing an $O(2^n/n^{\omega(1)})$-time algorithm for checking satisfiability of circuits of size $2cn$. In a sense, results like this show that designing fast satisfiability algorithms is not easier than proving circuit lower bounds. This also reflects the state-of-the-art on satisfiability algorithms: we only know how to beat the brute force search for circuits of size at most $2.99n$ [14]. Hence, the known satisfiability algorithms for small size (unrestricted) circuits currently do not give improved lower bounds. *Can one improve the brute force search for the satisfiability problem on circuits of size $4n$? Do non-trivial satisfiability algorithms for circuits of size cn imply cn lower bounds?*

Further reading. A good starting point is a recent survey by Williams [15].

2.6 Mass Production

Can one take a sufficiently hard function with constant number of inputs and cook out of it a family of functions of high circuit complexity? About 70 years ago, Shannon [16] showed that almost all functions from B_n have circuit complexity $\Omega(2^n/n)$ (by showing that the total number 2^{2^n} of functions is greater than the number of circuits of size $o(2^n/n)$). This implies that for any constant c, one can find a function $f_k \in B_k$, where $k = k(c)$, of circuit size at least ck just by enumerating functions one by one. A natural attempt to cook a family of functions out of f_k would be to define a function $f_n \in B_{n,\frac{n}{k}}$ as follows: split n input bits into $\frac{n}{k}$ blocks of k bits and apply f_k to each of the blocks. In other

words, we compute f_k on $\frac{n}{k}$ independent blocks of size k. The function f_n can be computed by a circuit of size $\frac{n}{k} \cdot \text{gates}(f_k)$. If this naive way of computing f_k was close to optimal, one would get a close to cn lower bound on the circuit size of f_n. We, however, do not know how to prove this. Still, this is what is known.

For a positive integer r and a function $f \in B_n$, by $r \times f$ we denote a function from $B_{rn,r}$ that applies f to r independent blocks of size n. We say that a *mass production* effect occurs for f when $\text{gates}(r \times f)$ is (much) smaller than $r \cdot \text{gates}(f)$. For *very simple* functions like $f = x_1 \oplus \cdots \oplus x_n$ (or any other function whose optimal circuit is a read-once formula) there is no mass production effect: $\text{gates}(r \times f) = r \cdot \text{gates}(f)$. This can be shown just by counting wires: f depends essentially on all its variables, hence there is at least one outgoing wire for every input; since each internal (non-output) gate reduces the number of outgoing wires at most by one, we conclude that $\text{gates}(f) = n - 1$ and $\text{gates}(r \times f) = rn - r = r \cdot (n - 1)$. Hiltgen [6] also shows that mass poduction effect occurs for many functions of circuit size about $2n$. On the other hand, for *very hard* function f one can show that $\text{gates}(r \times f)$ is almost the same as $\text{gates}(f)$ even if r is superpolynomial in n. More precisely, Ulig [17] showed that $\text{gates}(r \times f) \leq 2^n/n + o(2^n/n)$ for any $f \in B_n$ and $r = 2^{o(n/\log n)}$. *What are the functions avoiding mass production effect?*

Further reading. More on mass production can be found in Wegener's book [4, Sect. 10.2] and Hiltgen's PhD thesis [6, Sect. 4.4].

2.7 Logarithmic Depth Circuits

Can we at least prove superlinear lower bounds on circuits of logarithmic (i.e., $O(\log n)$) depth? Alas, currently, it is not known. However, if we further restrict the depth to be constant (in this case, one needs to allow arbitrary fan-in and to specify the operations allowed at gates), then one can prove even superpolynomial lower bounds! Moreover, Valiant [18] showed the following connection between these two models: if a function can be computed by a circuits of logarithmic depth and linear size, then it can also be computed by a subexponential depth 3 circuit, more precisely by an OR of CNF's of total size $2^{O(n/\log\log n)}$ (here, the constant inside $O(\cdot)$ depends on constants a, b where the size and depth of the original circuit is an and $b \log n$). *Currently, the strongest lower bounds known for such depth 3 circuits are of the form $2^{\Omega(n^{1/2})}$, though exponential lower bounds are known if we further restrict the length of clauses in CNF's to be constant.*

Further reading. An exposition of Valiant's reduction is given in the book by Viola [19, Chap. 2], while known results on constant depth circuits are summarized in the book by Jukna [1, Chaps. 11–12].

2.8 Linear Circuits and Matrix Rigidity

Can we at least prove superlinear lower bounds for circuits consisting of parity gates only? This question makes sense for multi-output functions. Specifically, let us focus on functions of the form $f(x) = Ax$ where $A \in \{0,1\}^{n\times n}$. Non-constructively, one can show that for almost all matrices A, the size of the smallest linear circuit computing Ax is $\Omega(n^2/\log n)$ (and there is a matching upper bound by Lupanov [20]). Alas, we do not have superlinear lower bounds even for this restricted model, even when we additionally restrict the depth to be $O(\log n)$. Interestingly, Valiant's depth reduction mentioned in Sec. 2.7 can be used to relate the circuit size to the notion of matrix rigidity introduced by Grigoriev [21] and Valiant [22]. Roughly speaking, for a parameter r, the rigidity of A, $R_A(r)$, is the Hamming distance from A to the set of matrices of rank (over $\mathbb{F}$) at most r. Valiant shows that if $R_A(\epsilon n) \geq n^{1+\delta}$ for positive constants ϵ, δ, then the function Ax cannot be computed by linear circuits of logarithmic depth of size $O(n)$. *So far, we have no such examples of explicit matrices.*

Further reading. More on circuit complexity and matrix rigidity can be found in the book by Lokam [23, Chap. 2]. Lower bounds for *constant depth linear circuits* (where superlinear lower bounds are known!) are summarized in the recent book by Jukna and Sergeev [24].

2.9 Multiplicative Complexity

What if some gates are given for free? Basically, each gate in a binary Boolean circuit is either an XOR-type gate, i.e., computes a binary operation of the form $x\oplus y\oplus a$ where $a \in \{0,1\}$, or an AND-type gate, i.e., computes $(x\oplus y)\wedge(y\oplus b)\oplus c$ where $a, b, c \in \{0,1\}$. It is well known that XOR-type gates are avoidable: any function can be computed by a circuit in the basis $U_2 = B_2\setminus\{\oplus, \equiv\}$. On the other hand, AND-type gates are unavoidable and it was shown by Nechiporuk [25] that almost all Boolean functions require about $2^{n/2}$ such gates. The minimum number of AND-type gates required to compute f is known as *multiplicative complexity* of f, $\mathrm{mc}(f)$. Of course, $\mathrm{mc}(f) \leq \mathrm{gates}(f)$ and the known lower bounds on multiplicative complexity are even weaker than those on circuit complexity. At the same time, one can prove lower bounds on mc without analyzing the structure of a circuit: as shown by Schnorr [26], a circuit with k AND-type gates computes a function of degree at most $k+1$. Here, the degree of a function is the degree of its polynomial over $\mathbb{F}_2$. This immediately gives a lower bound $n-1$ on multiplicative complexity of functions of full degree: e.g., $\mathrm{mc}(\mathrm{AND}) = n-1$. *Strangely enough, this is the strongest known lower bound: we do not know how to prove* $\mathrm{mc}(f) \geq n$, *let alone proving* $\mathrm{mc}(f) \geq (1+\epsilon)n$.

Acknowledgments. The research is supported by Russian Science Foundation (project 16-11-10123). The author is thankful to Alexander Golovnev and Edward A. Hirsch for fruitful discussions and many useful comments.

References

1. Jukna, S.: Boolean Function Complexity – Advances and Frontiers. Algorithms and Combinatorics. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24508-4
2. Find, M.G., Golovnev, A., Hirsch, E.A., Kulikov, A.S.: A better-than-3n lower bound for the circuit complexity of an explicit function. In: Dinur, I. (ed.) IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, Hyatt Regency, New Brunswick, NJ, USA, 9–11 October 2016, pp. 89–98. IEEE Computer Society (2016)
3. Golovnev, A., Hirsch, E.A., Knop, A., Kulikov, A.S.: On the limits of gate elimination. [27], pp. 46:1–46:13
4. Wegener, I.: The Complexity of Boolean Functions. Wiley-Teubner, Hoboken (1987)
5. Lamagna, E.A., Savage, J.E.: On the logical complexity of symmetric switching functions in monotone and complete bases. Technical report, Brown University (1973)
6. Hiltgen, A.P.: Cryptographically relevant contributions to combinational complexity theory. Ph.D. thesis, ETH Zurich, Zürich, Switzerland (1994)
7. Blum, N., Seysen, M.: Characterization of all optimal networks for a simultaneous computation of AND and NOR. Acta Inf. **21**, 171–181 (1984)
8. Melanich, O.: Technical report (2012)
9. Chashkin, A.V.: On complexity of Boolean matrices, graphs and corresponding Boolean matrices. Diskretnaya matematika **6**(2), 43–73 (1994). (in Russian)
10. Demenkov, E., Kojevnikov, A., Kulikov, A.S., Yaroslavtsev, G.: New upper bounds on the Boolean circuit complexity of symmetric functions. Inf. Process. Lett. **110**(7), 264–267 (2010)
11. Stockmeyer, L.J.: On the combinational complexity of certain symmetric Boolean functions. Math. Syst. Theory **10**, 323–336 (1977)
12. Williams, R.: Improving exhaustive search implies superpolynomial lower bounds. SIAM J. Comput. **42**(3), 1218–1244 (2013)
13. Jahanjou, H., Miles, E., Viola, E.: Local reductions. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015. LNCS, vol. 9134, pp. 749–760. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47672-7_61
14. Golovnev, A., Kulikov, A.S., Smal, A.V., Tamaki, S.: Circuit size lower bounds and #SAT upper bounds through a general framework. [27], pp. 45:1–45:16
15. Williams, R.R.: Some ways of thinking algorithmically about impossibility. SIGLOG News **4**(3), 28–40 (2017)
16. Shannon, C.E.: The synthesis of two-terminal switching circuits. Bell Syst. Tech. J. **28**(1), 59–98 (1949)
17. Ulig, D.: On the synthesis of self-correcting schemes from functional elements with a small number of reliable elements. Math. Notes Acad. Sci. USSR **15**(6), 558–562 (1974)
18. Valiant, L.G.: Exponential lower bounds for restricted monotone circuits. In: Johnson, D.S., Fagin, R., Fredman, M.L., Harel, D., Karp, R.M., Lynch, N.A., Papadimitriou, C.H., Rivest, R.L., Ruzzo, W.L., Seiferas, J.I. (eds.) Proceedings of the 15th Annual ACM Symposium on Theory of Computing, Boston, Massachusetts, USA, 25–27 April 1983, pp. 110–117. ACM (1983)
19. Viola, E.: On the power of small-depth computation. Found. Trends Theor. Comput. Sci. **5**(1), 1–72 (2009)

20. Lupanov, O.B.: On rectifier and switching-and-rectifier schemes. Dokl. Akad. Nauk SSSR **111**, 1171–1174 (1956)
21. Grigoriev, D.: An application of separability and independence notions for proving lower bounds of circuit complexity. Notes Sci. Semin. LOMI **60**, 38–48 (1976)
22. Valiant, L.G.: Graph-theoretic arguments in low-level complexity. In: Gruska, J. (ed.) MFCS 1977. LNCS, vol. 53, pp. 162–176. Springer, Heidelberg (1977). https://doi.org/10.1007/3-540-08353-7_135
23. Lokam, S.V.: Complexity lower bounds using linear algebra. Found. Trends Theor. Comput. Sci. **4**(1–2), 1–155 (2009)
24. Jukna, S., Sergeev, I.: Complexity of linear Boolean operators. Found. Trends Theor. Comput. Sci. **9**(1), 1–123 (2013)
25. Nechiporuk, E.I.: Complexity of schemes in certain bases containing nontrivial elements with zero weights. Dokl. Akad. Nauk SSSR **139**(6), 1302–1303 (1961)
26. Schnorr, C.P.: The multiplicative complexity of Boolean functions. In: Mora, T. (ed.) AAECC 1988. LNCS, vol. 357, pp. 45–58. Springer, Heidelberg (1989). https://doi.org/10.1007/3-540-51083-4_47
27. Faliszewski, P., Muscholl, A., Niedermeier, R. (eds.): 41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016. LIPIcs, vol. 58, Kraków, Poland, 22–26 August 2016. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)