Depth-Three Circuits for Inner Product and Majority Functions

Kazuyuki Amano
Gunma University, Kiryu, Japan

Abstract

We consider the complexity of depth-three Boolean circuits with limited bottom fan-in that compute some explicit functions. This is one of the simplest circuit classes for which we cannot derive tight bounds on the complexity for many functions. A Σ_3^k-circuit is a depth-three OR $\circ$ AND $\circ$ OR circuit in which each bottom gate has fan-in at most k.

First, we investigate the complexity of Σ_3^k-circuits computing the inner product mod two function IP_n on n pairs of variables for small values of k. We give an explicit construction of a Σ_3^2-circuit of size smaller than $2^{0.952n}$ for IP_n as well as a Σ_3^3-circuit of size smaller than $2^{0.692n}$. These improve the known upper bounds of $2^{n-o(n)}$ for Σ_3^2-circuits and $3^{n/2} \sim 2^{0.792n}$ for Σ_3^3-circuits by Golovnev, Kulikov and Williams (ITCS 2021), and also the upper bound of $2^{(0.965\ldots)n}$ for Σ_3^2-circuits shown in a recent concurrent work by Göös, Guan and Mosnoi (MFCS 2023).

Second, we investigate the complexity of the majority function MAJ_n aiming for exploring the effect of negations. Currently, the smallest known depth-three circuit for MAJ_n is a monotone circuit. A $\Sigma_3^{(+k,-\ell)}$-circuit is a Σ_3-circuit in which each bottom gate has at most k positive literals and ℓ negative literals as its input. We show that, for $k \leq 2$, the minimum size of a $\Sigma_3^{(+k,-\infty)}$-circuit for MAJ_n is essentially equal to the minimum size of a monotone Σ_3^k-circuit for MAJ_n. In sharp contrast, we also show that, for $k = 3, 4$ and 5, there exists a $\Sigma_3^{(+k,-\ell)}$-circuit computing MAJ_n (for an appropriately chosen ℓ) that is smaller than the smallest known monotone Σ_3^k-circuit for MAJ_n. Our results suggest that negations may help to speed up the computation of the majority function even for depth-three circuits. All these constructions rely on efficient circuits or formulas on a small number of variables that we found through a computer search.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases Circuit complexity, depth-3 circuits, upper bounds, lower bounds, computer-assisted proof

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.7

Supplementary Material *Dataset*: `https://gitlab.com/KazAmano/depth-3-circuits`
archived at `swh:1:dir:6c19f0a1a69eae1143d2270517dd6f46df08bfb7`

Funding This work was supported in part by JSPS Kakenhi No. JP21K19758, JP18K11152 and JP18H04090.

Acknowledgements The author would like to thank anonymous referees for their helpful comments.

1 Introduction

Deriving a strong lower bound on the size of a Boolean circuit computing an explicit function is one of the most challenging problems in theoretical computer science. Many different types of restricted circuits have been investigated, and this paper concentrates on depth-three circuits.

A Σ_3-circuit is a depth-three OR $\circ$ AND $\circ$ OR circuit consisting of unbounded fan-in AND/OR gates, with variables or their negations feeding into the bottom gates. In other words, a Σ_3-circuit is an OR of an arbitrary number of CNF formulas.

Despite its simplicity, there still is a considerable gap between the upper and lower bounds on the size of Σ_3-circuits. Here, the *size* of a Σ_3-circuit is defined as the number of *gates* in the circuit. A counting argument shows that a random function on n variable needs a Σ_3-circuit of size $\Theta(2^{n/2})$ [6, 25]. However, the strongest lower bound on the size of a Σ_3-circuit for an explicit Boolean function is $2^{\Omega(\sqrt{n})}$ (e.g., [16] for the proof for the parity and majority functions). Deriving a lower bound of $2^{\omega(\sqrt{n})}$ on the size of a Σ_3-circuit computing some explicit function has been open for over 30 years.

Such a situation motivates us to consider further restricted Σ_3-circuits. One natural restriction is to bound the bottom fan-in. For a natural number k, a Σ_3^k-circuit is a Σ_3-circuit with bottom fan-in bounded by k, or equivalently, an OR of k-CNF formulas.

When the value of k is small, stronger lower bounds are known. For example, Paturi, Saks and Zane [21] showed that the minimum size of a Σ_3^k-circuit computing the parity function on n variables is at least $2^{n/k}$, for any $k \leq O(\sqrt{n})$. See e.g., the introduction of [17] or [11] for more results on Σ_3^k-circuits. A recent work by Golovnev, Kulikov and Williams [11] showed that a $2^{n-o(n)}$ lower bound on the Σ_3^{16}-circuit size for an explicit function implies a $3.9n$ lower bound on the general circuit size, which would be a breakthrough on circuit complexity since the best known lower bound is much smaller, say, $3.1n - o(n)$ [20] (see also [8]).

Despite the simplicity of a model, for many functions, we still do not know the size of an optimal Σ_3^k-circuit even for small values of k. In this paper, we investigate Σ_3^k-circuits for two well-studied functions, namely, the inner product mod two function and the majority function.

1.1 Inner Product

The first target function we consider in this paper is the *inner product mod two* function $\mathsf{IP}_n(x_1, \ldots, x_n, y_1, \ldots, y_n) := \oplus_i x_i y_i$. The function IP_n has frequently appeared as a target for analyzing the complexity of shallow circuits (see e.g., [2, 9, 11, 15, 18]).

For a Boolean function f, we write the minimum size (i.e., number of gates) of a Σ_3^k-circuit computing f as $s_3^k(f)$. For many functions f, we do not have a technique for determining $s_3^k(f)$ even for small values of k. We sometimes consider the minimum fan-in of the top OR gate in a Σ_3^k-circuit computing f, which is denoted by $\tilde{s}_3^k(f)$. It is easy to see that $s_3^k(f)$ is at most polynomially larger than $\tilde{s}_3^k(f)$, when $k = O(1)$.

Recently, Golovnev, Kulikov and Williams [11] and Frankl, Gryaznov and Talebanfard [10] invesigated the complexity of IP_n for Σ_3^2 and Σ_3^3-circuits. The bounds described in [10, 11] are $2^{n/2} \leq s_3^2(\mathsf{IP}_n) \leq 2^{n-o(n)}$ and $2^{n/3} \leq s_3^3(\mathsf{IP}_n) \leq 3^{n/2} \sim 2^{0.792n}$. Both upper bounds are given in [11]. Both lower bounds are via a simple reduction to the parity function $\oplus_n$ on n variables and the fact that $s_3^k(\oplus_n) \geq 2^{n/k}$ [21]. The problem of determining $s_3^3(\mathsf{IP}_n)$ as well as $s_3^2(\mathsf{IP}_n)$ has been left as an open problem in these works. After the initial submission of this manuscript, we learned that Göös, Guan and Mosnoi [12] subsequently improved the upper and lower bounds on the size of Σ_3^2-circuits to $2^{(0.847\ldots)n} < s_3^2(\mathsf{IP}_n) < 2^{(0.965\ldots)n}$ using an LP-based technique.

In this work, we show that both upper bounds can be improved considerably. Namely, we present an explicit construction of Σ_3^2-circuits of size less than $2^{0.952n}$ and Σ_3^3-circuits of size less than $2^{0.692n}$ that compute IP_n (we will review this more carefully in Section 1.3).

1.2 Majority

The second target function we consider in this paper the size of is the *majority* function $\mathsf{MAJ}_n(x_1, \ldots, x_n) := [\sum_i x_i \geq n/2]$, where $[\cdot]$ denotes the Iverson bracket.

The best known upper bound on the size of a Σ_3-circuit for MAJ_n is $2^{O(\sqrt{n \log n})}$ [3, 19]. Note that their circuit is monotone, i.e., a circuit without negative literals. The best known lower bound on the size of a Σ_3-circuit for MAJ_n is $2^{d(\sqrt{n})-o(\sqrt{n})}$ where $d = 1/\sqrt{\ln 4} = 0.849\ldots$ due to Håstad, Jukna and Pudlák [16]. Hence, there still is $\sqrt{\log n}$ factor gap in the exponent and there is a possibility that the minimum size of a Σ_3-circuit for MAJ_n is $2^{\omega(\sqrt{n})}$. A recent work by Cavalar and Oliveira [4] reveals that a slightly weaker lower bound of $2^{\Omega(\sqrt{n/\log n})}$ can be shown via monotone simulations of non-monotone circuits.

Apparently, one natural question is whether an optimal Σ_3-circuit (or Σ_3^k-circuit) for MAJ_n is monotone or not, which is the main focus of the second part of this work.

The resolution of this problem would contribute to unraveling the mystery of NOT gates in the computation of Boolean functions. Despite more than 30 years have passed since exponential lower bounds on the size of monotone circuits were proven [1, 23], we cannot prove even $4n$ lower bound on a general circuit size for an explicit function in NP.

The depth-three is the simplest interesting case in the sense that negations are known to be useless for computing monotone functions in depth-two circuits. Namely, Quine [22] showed that, for any monotone Boolean function f, a smallest DNF (or CNF, respectively) computing f is a monotone DNF (or monotone CNF, respectively).

For two non-negative integers k and ℓ, a $(+k,-\ell)$-*CNF formula* is a CNF formula in which each clause contains at most k positive literals and at most ℓ negative literals. A $\Sigma^{(+k,-\ell)}$-circuit is a Σ_3-circuit in which all inputs of the top OR gate are $(+k,-\ell)$-CNF formulas. A $\Sigma_3^{(+k,-0)}$-circuit, which is a monotone Σ_3^k-circuit, is simply written as a Σ_3^{+k}-circuit. It is tempting to guess that, for any ℓ, an optimal $\Sigma_3^{(+k,-\ell)}$-circuit for MAJ_n is in fact a monotone, i.e., a Σ_3^{+k}-circuit.

In this work, we obtain some evidence suggesting that this hypothesis may not hold for $k \geq 3$. A detailed explanation of what we have shown is provided in the next subsection.

1.3 Our Contributions and Implications

In short, the contribution of this paper is to improve the upper bounds on the size of a Σ_3-circuit with limited bottom fan-in that compute IP_n and MAJ_n.

For IP_n, we give an explicit construction showing that (i) $\tilde{s}_3^2(\mathsf{IP}_n) < 2^{0.952n}$, and (ii) $\tilde{s}_3^3(\mathsf{IP}_n) < 2^{0.692n}$. As described in Section 1.1, these improve the known upper bounds in [11] and [12] . Note that the construction of our circuits includes components that are found by a computer search. As a result, some circuits look a bit exotic. This suggests that, even in the case of a simple circuit model and a simple target function, a good circuit may have an unintuitive form. We will give a detailed description of our circuits in Section 3.

For MAJ_n, we show that for $k = 3, 4$ and 5, there exist a $\Sigma_3^{(+k,-\ell)}$-circuit whose size is smaller than the currently known smallest monotone Σ_3^{+k}-circuit for an appropriately chosen ℓ. Namely, we show that

- **(i)** There exists a $\Sigma_3^{(+3,-6)}$-circuit of size $O(1.2768^n)$ for MAJ_n, which is smaller than the currently known smallest Σ_3^{+3}-circuit of size $O(1.2779^n)$.
- **(ii)** There exists a $\Sigma_3^{(+4,-4)}$-circuit of size $O(1.2040^n)$ for MAJ_n, which is smaller than the currently known smallest Σ_3^{+4}-circuit of size $O(1.2093^n)$.
- **(iii)** There exists a $\Sigma_3^{(+5,-7)}$-circuit of size $O(1.1751^n)$ for MAJ_n, which is smaller than the currently known smallest Σ_3^{+5}-circuit of size $O(1.1760^n)$.

Although the improvement in the size is relatively small, we believe that our results give some hints on how to use negations to speed up the computation of monotone functions in a shallow circuit. Note that, unlike the circuits for IP_n, the circuits for MAJ_n are probabilistic,

i.e., the proof is existential. As a complementary result, we also show that the size of a smallest $\Sigma_3^{(+k,-\infty)}$-circuit for MAJ_n and the size of a smallest a Σ_3^{+k}-circuit for MAJ_n are essentially equal when $k \leq 2$. We also give some non-trivial lower bounds on the size of a $\Sigma_3^{(+k,-\infty)}$-circuit computing MAJ_n for $k \geq 3$. All the results on the complexity of MAJ_n will be given in Section 4.

The constructions of all our circuits are based on a common methodology, which may be of independent interest. First, we obtain a small building block by a computer search, or more specifically by using an IP (integer programming) solver with some additional heuristics in some cases. Then, we use it to construct a circuit for general input size. As expected from the methodology, some circuits are complicated and difficult to explain why these work. All certificates of our circuits are available electronically at `https://gitlab.com/KazAmano/depth-3-circuits`. Note that we use Gurobi Optimizer [14] in our experiments.

1.4 Organization

The rest of the paper is organized as follows. In Section 2, we give preliminaries. In Section 3, we show the construction of depth-three circuits for the inner product functions. In Section 4, we analyze the size of depth-three circuits for the majority function focusing on the effect of negations. Finally, we close the paper with some concluding remarks in Section 5.

2 Preliminaries

For a vector $x \in \{0,1\}^n$, x_i denotes the i-th bit of x and $|x|$ denotes the number of ones in x, i.e., $|x| := \sum_{i=1}^n x_i$. For two vectors $x, y \in \{0,1\}^n$, we write $x \geq y$ if $x_i \geq y_i$ for every $i = 1, \ldots, n$.

As usual, a CNF formula in which each clause contains at most k literals is called a *k-CNF formula*. We also use a terminology *$(+k,-\ell)$-CNF formula* for two non-negative integers k and ℓ that represents a CNF formula in which each clause contains at most k positive literals and at most ℓ negative literals.

In this paper, we concentrate on depth-three OR $\circ$ AND $\circ$ OR circuits consisting of unbounded fan-in AND/OR gates. Each bottom OR gate has positive or negative literals as its input. We consider several subclasses of Σ_3-circuits where the inputs of the bottom gates are restricted.

A *Σ_3^k-circuit* is a Σ_3-circuit in which all inputs of the top OR gate are k-CNF formulas. Similarly, a *$\Sigma_3^{(+k,-\ell)}$-circuit* is a Σ_3-circuit in which all inputs of the top OR gate are $(+k,-\ell)$-CNF formulas. When $\ell = 0$, we simply write this as a Σ^{+k}-circuit. A *$\Sigma_3^{(+k,-\infty)}$-circuit* means a $\Sigma_3^{(+k,-\ell)}$-circuit with an unbounded value of ℓ. A circuit is said to be *monotone* if it does not contain negative literals.

Recall that the *size* of a Σ_3-circuit is defined as the number of *gates* in the circuit. For a Boolean function f, we write the minimum size of a Σ_3^k-circuit that computes f as $s_3^k(f)$. We sometimes consider the minimum fan-in of the top OR gate in a Σ_3^k-circuit that computes f, which is denoted by $\tilde{s}_3^k(f)$. When $k = O(1)$, $s_3^k(f)$ is at most polynomially (in the number of input variables) larger than $\tilde{s}_3^k(f)$. We also use the symbols $\tilde{s}_3^{(+k,-\infty)}$, which is defined analogously to $\tilde{s}_3^k$.

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. For an integer t such that $0 \leq t \leq n$, the number of input vectors $x \in \{0,1\}^n$ with $|x| = t$ such that $f(x) = 1$ is denoted by $|f|_t$. A Boolean function f is said to be *monotone* if $f(x) \geq f(y)$ for every pair of inputs x and y such that $x \geq y$.

Since the top gate of a Σ_3-circuit is an OR gate, every function g that feeds into the top gate satisfies $g^{-1}(1) \subseteq f^{-1}(1)$ when the circuit computes f. We refer to a function g satisfying this condition as being *consistent* with f.

The *inner product mod two function* over n pairs of input variables, denoted by IP_n, is defined as

$$\mathsf{IP}_n(x_1,\ldots,x_n,y_1,\ldots,y_n) := \begin{cases} 1, & \text{if } \sum_{i=1}^n x_i y_i \pmod 2 \equiv 1, \\ 0, & \text{otherwise.} \end{cases}$$

The *majority function* over n input variables, denoted by MAJ_n, is defined as

$$\mathsf{MAJ}_n(x_1,\ldots,x_n) := \begin{cases} 1, & \text{if } \sum_{i=1}^n x_i \geq n/2, \\ 0, & \text{otherwise.} \end{cases}$$

3 Improving Depth-3 Circuits for Inner Product

In this section, we show the upper bounds on the size of Σ_3^2-circuits and Σ_3^3-circuits for IP_n.

▶ **Theorem 1.** *For every sufficiently large n,*

1. $\tilde{s}_3^2(\mathsf{IP}_n) \leq 2^{d_2 n}$ *where* $d_2 = (\log_2 14)/4 = 0.9518\ldots$,
2. $\tilde{s}_3^3(\mathsf{IP}_n) \leq 2^{d_3 n}$ *where* $d_3 = (\log_2 11)/5 = 0.6918\ldots$.

The proof of Theorem 1 consists of two parts. First, we obtain a small Σ_3^k-circuit for IP_m and $\overline{\mathsf{IP}_m}$ for small values of m, where $\overline{\mathsf{IP}_m}$ denotes the negation of IP_m. Then, we use these circuits to construct a circuit computing IP_n for general values of n. The second part relies on the following lemma.

▶ **Lemma 2.** *Let m be some natural number. Suppose that $\tilde{s}_3^k(\mathsf{IP}_m) \leq s_1$ and $\tilde{s}_3^k(\overline{\mathsf{IP}_m}) \leq s_0$. Then, for all n that are multiples of m, $\tilde{s}_3^k(\mathsf{IP}_n) \leq 2^{dn}$ where $d = \log_2(s_0+s_1)/m$.*

Proof. Put $p = n/m$. Let IP_m^0 denote $\overline{\mathsf{IP}_m}$ and IP_m^1 denote IP_m. By the definition of IP_n, it is obvious that

$$\mathsf{IP}_n(x_1,\ldots,x_n,y_1,\ldots,y_n) = \bigvee_{\substack{i_1,\ldots,i_p\in\{0,1\} \\ i_1+\cdots+i_p\equiv 1 \pmod 2}} (\mathsf{IP}_m^{i_1} \wedge \mathsf{IP}_m^{i_2} \wedge \cdots \wedge \mathsf{IP}_m^{i_p}),$$

where we omit the input variables in the RHS of the above equation for simplicity. The t-th IP_m (or $\overline{\mathsf{IP}_m}$) in the brackets gets $(x_{(t-1)m+1},\ldots,x_{tm},y_{(t-1)m+1},\ldots,y_{tm})$.

Since IP_m and $\overline{\mathsf{IP}_m}$ can be represented by the OR of s_1 and s_0 k-CNF formulas, respectively, $(\mathsf{IP}_m^{i_1} \wedge \mathsf{IP}_m^{i_2} \wedge \cdots \wedge \mathsf{IP}_m^{i_p})$ can be represented by the OR of $(\prod_{j=1}^p s_{i_j})$ k-CNF formulas by expansion. Hence, we have

$$\tilde{s}_3^k(\mathsf{IP}_n) \leq \sum_{\substack{i_1,\ldots,i_p\in\{0,1\} \\ i_1+\cdots+i_p\equiv 1 \pmod 2}} \prod_{j=1}^p s_{i_j} < (s_0+s_1)^p = 2^{\log_2(s_0+s_1)n/m}.$$

This completes the proof of Lemma 2. ◀

By Lemma 2, our task now is to find a good Σ_3^k-circuit for IP_m and $\overline{\mathsf{IP}}_m$ for small values of m. In this work, we use an IP (integer programming) solver for this task.

Suppose that the fan-in of the top OR gate is T. We can formulate the condition that the OR of t k-CNF formulas computes the target function f as an integer programming problem.

For each $t = 1,\ldots,T$ and for every possible clause c, we introduce a Boolean variable $F_{t,c}$ that represents whether the clause c is appeared in the t-th CNF formula. For each $t = 1,\ldots,T$ and for every input vector x, we also introduce a Boolean variable $V_{t,x}$ which represents the output of the t-th CNF formula. Obviously, $V_{t,x} = 1$ iff $\sum_{c:c(x)=0} F_{t,c} = 0$ and $V_{t,x} = 0$ iff $\sum_{c:c(x)=0} F_{t,c} \geq 1$. Finally, we impose the additional constraint that $\bigvee_t V_{t,x} = f(x)$ for every x.

Proof of Theorem 1. In order to prove Theorem 1, it is sufficient to verify the following fact by Lemma 2. ◀

▶ **Fact 3.**
(i) $\tilde{s}_3^2(\mathsf{IP}_4) \leq 7$ *and* $\tilde{s}_3^2(\overline{\mathsf{IP}}_4) \leq 7$.
(ii) $\tilde{s}_3^3(\mathsf{IP}_5) \leq 6$ *and* $\tilde{s}_3^3(\overline{\mathsf{IP}}_5) \leq 5$.

Proof. A certificate for the first part of (i) is the OR of the following seven 2-CNF formulas.

```
f1:  (-x2 -y2)(-x3 -y3)(-x4 -y4)(x1)(y1)
f2:  (-x1 -y1)(-x3 -y3)(-x4 -y4)(x2)(y2)
f3:  (-x4 -y4)(x1 -y2)(x2 -y2)(y1 -y2)(-y1 y2)(x3)(y3)
f4:  (-x3 -y3)(x1 -x2)(-x1 x2)(y1 -x2)(-x2 y2)(x4)(y4)
f5:  (-x1 -y1)(-x2 -y2)(-y3 -y4)(x3 -y3)(y3 x4)(y3 y4)
f6:  (-x2 -y2)(x1 -x4)(x3 -y1)(-x1 x4)(-x3 y1)(y1 x4)(y3)(y4)
f7:  (-x1 -y1)(x2 -x4)(x3 -y2)(x3 x2)(-x2 x4)(-x3 y2)(-x3 y3)(y4)
```

Here, for example, `(x2 -y2)` represents a clause $(x_2 \vee \overline{y_2})$.

A certificate for the second part of (i), i.e., for $\overline{\mathsf{IP}}_4$, is the OR of the following seven 2-CNF formulas.

```
f1:  (-x1 -y1)(-x2 -y2)(x3 -x4)(-x3 x4)(y3 -x4)(y4)
f2:  (-x2 -y2)(-x4 -y4)(x1 -y3)(x3 -y1)(-x3 y1)(-x1 y3)
f3:  (-x2 -y2)(-x3 -y3)(x1 -y1)(-y1 x4)(y1 -x4)(-x4 y4)
f4:  (-x3 -y3)(-x4 -y4)(x1 -x2)(-x1 x2)(y1 -x2)(-x2 y2)
f5:  (-x1 -y1)(-x3 -y3)(x2 -y4)(x4 -y4)(y2 -y4)(-y2 y4)
f6:  (-x1 -y1)(-x4 -y4)(-y3 x2)(x3 -y3)(-y3 y2)(y3 -x2)
f7:  (x1)(x2)(x3)(x4)(y1)(y2)(y3)(y4)
```

The correctness of these circuits can be verified by hand or by using a computer. Currently, we do not have a simple explanation of why these circuits work, especially for the first set of formulas.

The certificates for statement (ii), i.e., for IP_5 and $\overline{\mathsf{IP}}_5$, are given in the appendix. ◀

Currently, we do not know whether the bounds in Fact 3 are optimal. Remark that we also observed that $\tilde{s}_3^3(\mathsf{IP}_4) \leq 4$ and $\tilde{s}_3^3(\overline{\mathsf{IP}}_4) \leq 3$, but these yield a weaker bound of $\tilde{s}_3^3(\mathsf{IP}_n) \leq 2^{0.702n}$. We include the certificates for these bounds in the appendix.

4 Negations may Help Depth-3 Circuits computing Majority

In this section, we consider the size of a $\Sigma_3^{(+k,-\ell)}$-circuit computing the majority function for small values of k.

4.1 Motivating Example

When we consider depth-three Σ_3^{+k}-circuits, i.e., an OR of monotone k-CNF formulas, the smallest size of such a circuit for MAJ_n is essentially determined by the maximum of $R_\phi := |\{x \mid \phi(x) = 1 \text{ and } |x| = n/2\}|$ over all monotone k-CNF formulas ϕ consistent with MAJ_n. We write this maximum as R_*.

Precisely, the minimum top fan-in of a Σ_3^{+k}-circuit for MAJ_n is at least $\binom{n}{n/2}/R_*$ and at most $n\binom{n}{n/2}/R_*$. The lower bound is obvious since the top gate is an OR gate. The upper bound can be proved by a standard technique combining random sampling and the union bound (similar to the proof of Theorem 10 in Section 4.3). An alternative proof based on the integral gap of a certain integer programming problem can be found in [17].

Let us consider Σ_3^{+2}-circuits, i.e., a disjunction of monotone 2-CNF formulas. Finding the value of R_* for 2-CNF formulas is closely related to the well-known Turán problem (see e.g., [17]). The value is known to be $R_* = 2^{n/2}$ and the unique extremal formula is given by the disjoint union of $n/2$ edges in an n-vertex graph. Hence the size of a smallest Σ_3^{+2}-circuit for MAJ_n is $\tilde{\Theta}(2^{n/2})$. In Section 4.2, we will verify that the minimum size of a $\Sigma_3^{(+2,-\infty)}$-circuit for MAJ_n is also $\tilde{\Theta}(2^{n/2})$, which means that negations are useless if each bottom gate gets at most two positive literals.

Let us now consider Σ_3^{+3} circuits. To the best of our knowledge, the value of R_* for monotone 3-CNF formulas is unknown. Through our computer experiments, we see that the maximum value of R_ϕ among all n-variable monotone 3-CNF formulas ϕ for $n = 4, 6, 8, 10$ and 12 are $6, 14, 36, 84$ and 216, respectively. Note that, for $n = 4, 8$ and 12, an extremal 3-CNF formula is given by $n/4$ independent copies of 3-uniform hypergraph on four vertices. This suggests that $R_* = 6^{n/4}$, which would imply that the smallest Σ_3^{+3}-circuit for MAJ_n has size $\tilde{\Theta}((2/6^{1/4})^n) = \tilde{\Theta}(1.2778\cdots^n)$. It is tempting to guess that the minimum size of a $\Sigma_3^{(+3,-\infty)}$-circuit is equal to this.

Surprisingly (at least for us), we discovered that this is not true. We found a CNF formula χ on 12 variables in which each clause contains at most three positive literals (and at most four negative literals), such that the value of R_χ is 217, exceeding the maximum value for monotone 3-CNF formulas by one. The description of χ will be given in Section 4.3. As expected, we can use χ to show that there exists a $\Sigma_3^{(+3,-4)}$-circuit of size $\tilde{O}((2/217^{1/12})^n) = O(1.2774^n)$ for MAJ_n. Although the improvement is small, this suggests that negations may be useful for computing MAJ_n by a $\Sigma_3^{(+3,-\infty)}$-circuit.

In the following subsections, we analyze such a phenomenon more carefully.

4.2 Negations are useless for $k \leq 2$

We first show a lower bound on the size a $\Sigma_3^{(+k,-\infty)}$-circuit for MAJ_n. This was essentially shown in [16]. We include the proof here for completeness.

▶ **Theorem 4.** *For every natural number k, $\tilde{s}_3^{(+k,-\infty)}(\mathsf{MAJ}_n) = \Omega(2^n/(k^{n/2}\sqrt{n}))$.*

The proof relies on the notion of the *lower limit* introduced in [16]. Here, for a vector $x \in \{0,1\}^n$ and a set of indices $S \subseteq \{1,\dots,n\}$, $x|_S$ denotes the restriction of x to the set S.

▶ **Definition 5.** *Let $B \subseteq \{0,1\}^n$ be a set of vectors. A vector $y \in \{0,1\}^n$ is a lower k-limit for a set B, if, for any subset of indices $S \subseteq \{1,\dots,n\}$ with $|S| = k$, there exists a vector $x \in B$ such that $x > y$ and $y|_S = x|_S$.* ⌟

▶ **Lemma 6** ([16])**.** *Let $\mathcal{F}$ be a family of s-element subsets of $\{1,\dots,n\}$. If $|\mathcal{F}| > k^s$, then there exists a lower k-limit y for $\mathcal{F}$.* ⌟

Proof of Theorem 4. Let C be any $\Sigma_3^{(+k,-\infty)}$-circuit computing MAJ_n. Let $g_1,\dots,g_m$ be the functions that feed into the top OR gate of C. Note that every g_i is consistent with MAJ_n (i.e., $g_i^{-1}(1) \subseteq \mathsf{MAJ}_n^{-1}(1)$) and that $\bigcup_{i=1}^m g_i^{-1}(1) = \mathsf{MAJ}_n^{-1}(1)$.

For every $i = 1,\dots,m$, let $B_i := \{x \mid x \in g_i^{-1}(1) \wedge |x| = n/2\}$. We claim that $|B_i| \leq k^{n/2}$ for every i, which immediately implies the lemma since $|\mathsf{MAJ}_n^{-1}(1)| = \Omega(2^n/\sqrt{n})$.

The claim can be verified using Lemma 6 as follows. Suppose for the contrary that $|B_i| > k^{n/2}$ for some i. By Lemma 6, there exists a lower k-limit y for B_i. For each clause c in a $(+k,-\infty)$-CNF g_i, let S_c be the set of indices of positive literals that appeared in c. By the definition of the lower limit, there exists a vector $x_c \in B_i$ with $x_c > y$ that coincides with y on S_c, which ensures that $c(y) = c(x_c) = 1$. This holds for every clause c in g_i, which implies that $g_i(y) = 1$. However, it should satisfy that $|y| \leq n/2 - 1$ and hence $y \in \mathsf{MAJ}_n^{-1}(0)$, a contradiction. ◀

▶ **Theorem 7.** *For $k = 1$ and 2, the minimum size of a $\Sigma_3^{(+k,-\infty)}$-circuit for MAJ_n and the minimum size of a Σ_3^{+k}-circuit for MAJ_n are both $\tilde{\Theta}(2^{n/k})$.*

Proof. Both lower bounds are shown by Theorem 4. The upper bound for $k = 1$ is trivial, and the upper bound for $k = 2$ was shown in e.g., [17], as discussed in Section 4.1. ◀

Note that Theorem 4 does not yield a non-trivial lower bound for $k \geq 4$. For such k, we can apply the following theorem that can be proved by a similar argument to the proof of Theorem 4. An explicit value of the lower bounds obtained from Theorem 8 is shown in Table 1 in Section 4.5.

▶ **Theorem 8.** *For every natural number k and for every real number s with $0 < s \leq 0.5$, $\tilde{s}_3^{(+k,-\infty)}(\mathsf{MAJ}_n) = \tilde{\Omega}(2^{dn})$ where $d = (0.5+s)H(s/(0.5+s)) - s\log_2 k$, where $H(\cdot)$ represents the binary entropy function.*

Proof (sketch). The proof is similar to the proof of Theorem 4. Suppose that a $\Sigma_3^{(+k,-\infty)}$-circuit C computes a MAJ_n. Let s be an arbitrary real number with $0 < s \leq 0.5$. We fix arbitrarily chosen $(0.5 - s)n$ input variables to the value 1 in C. The resulting circuit computes the threshold function on $(0.5 + s)n$ variables that output 1 iff the number of ones in an input vector is at least sn.

The rest of the proof is analogous to the proof of Theorem 4. By noticing that the number of vectors $x \in \{0,1\}^{(0.5+s)n}$ with $|x| = sn$ is

$$\binom{(0.5+s)n}{sn} \sim 2^{(0.5+s)nH(s/(0.5+s))},$$

we can complete the proof of Theorem 8 using Lemma 6. ◀

4.3 Negations may be useful for $k \geq 3$

In this subsection, we give some unintuitive construction of depth-three circuits for the majority function using negations. As to the construction for IP_n, we first obtain a good building block by a computer search and then extend it to a circuit for a general input size.

4.3.1 Blow-up Lemma

In the following, we give several lemmas that will be used in the blow-up process. If a "base" function g is monotone, then this step is easy. We can compute MAJ_n by taking an OR of an appropriate number of independent copies of g over random permutations on inputs. Because our base function is not monotone, we need a small twist to this argument.

▶ **Definition 9.** *Suppose that $n \geq 2$ is an even integer. We say that a list of n-variable Boolean functions $(g_{n/2}, g_{n/2+1}, \ldots, g_n)$ satisfies the increasing property, if (i) every g in the list is consistent with MAJ_n (i.e., $g(x) = 0$ for every $x \in \{0,1\}^n$ with $|x| < n/2$), and (ii) for every $t \geq n/2 + 1$, it holds that*

$$|g_t|_t \geq |g_{n/2}|_{n/2} \prod_{m=n/2+1}^{t} \frac{n-m+1}{m-1}.$$

Here, $g_{n/2}, \ldots, g_n$ are not necessarily distinct.

The following theorem gives a probabilistic construction of a depth-three circuit consistent with MAJ_n that outputs 1 on all inputs in the t-th layer of the Boolean cube, when we are given a function g on n variables consistent with MAJ_n such that $|g|_t$ is large.

▶ **Theorem 10.** *Suppose that an n-variable $(+k,-\ell)$-CNF formula g is consistent with MAJ_n. Then, for every $t \geq n/2+1$, there exists a $\Sigma_3^{(+k,-\ell)}$-circuit C of size at most $\frac{\binom{n}{t}}{|g|_t} \ln \binom{n}{t}$ such that (i) $C(x)=0$ for every $x \in \{0,1\}^n$ with $|x|<n/2$ and (ii) $C(x)=1$ for every $x \in \{0,1\}^n$ with $|x|=t$.*

Proof. Let $\mathbf{g}$ be the uniform distribution over all functions obtained from g by permuting the input variables of g.

Let $x \in \{0,1\}^n$ be an arbitrarily fixed input vector that contains t 1's. Then, we have

$$\Pr_{g \in \mathbf{g}}[g(x)=1] = \frac{|g|_t}{\binom{n}{t}}.$$

Let $v := \frac{\binom{n}{t}}{|g|_t} \ln \binom{n}{t}$. Then, we see that

$$\Pr_{g_1,\ldots,g_v \in \mathbf{g}}\left[\bigvee_{i=1}^{v} g_i(x)=0\right] = \left(1-\frac{|g|_t}{\binom{n}{t}}\right)^v < \frac{1}{\binom{n}{t}}.$$

By the union bound, this implies that there are $(+k,-\ell)$-CNF formulas $g_1,\ldots,g_v$ such that $\bigvee_{i=1}^{v} g_i(x)=1$ for every $x \in \{0,1\}^n$ with $|x|=t$. Obviously $\bigvee_{i=1}^{v} g_i(x)=0$ for every $x \in \{0,1\}^n$ with $|x|<n/2$. Hence, $\bigvee_{i=1}^{v} g_i$ gives a desired depth-three circuit, which completes the proof of Theorem 10. ◀

The following lemma intuitively says that we can mimic a monotone function by a list of non-monotone functions with the increasing property.

▶ **Lemma 11.** *Suppose that m is an even positive integer and n is a multiple of m. Let $\mathbf{g}$ be a list of m-variable $(+k,-\ell)$-CNF formulas $(g_{m/2},\ldots,g_m)$ satisfying the increasing property. For each integer t satisfying $n/2 \leq t \leq n$, we define an n-variable Boolean function f_t as*

$$f_t(x_1,\ldots,x_n) := g_{s_1}(x_1,\ldots,x_m) \wedge g_{s_2}(x_{m+1},\ldots,x_{2m}) \wedge \cdots \wedge g_{s_{n/m}}(x_{n-m-1},\ldots,x_n),$$

where $s_i \in \{\lfloor tm/n \rfloor, \lceil tm/n \rceil\}$ for $i=1,\ldots,n/m$ and satisfies $\sum_{i=1}^{n/m} s_i = t$. Then, it holds that

$$\frac{|f_t|_t}{\binom{n}{t}} \geq \frac{|f_{n/2}|_{n/2}}{\binom{n}{n/2}},$$

and $|f_t|_v = 0$ for every integer v satisfying $0 \leq v < n/2$. ⌟

For example, when $n=120$, $m=12$ and $t=63$, f_t is the AND of three g_7's and seven g_6's. The proof of Lemma 11 is postponed to Appendix.

The following theorem is the main body of our blow-up process.

▶ **Theorem 12.** *Let $m \geq 2$ be an even integer. Let $\mathbf{g}=(g_{m/2},\ldots,g_m)$ be a list of m-variable $(+k,-\ell)$-CNF formulas satisfying the increasing property. Then, there exists a $\Sigma_3^{(+k,-\ell)}$-circuit of size at most $O\left(n^{1.5} \cdot \left(\frac{2}{(|g_{m/2}|_{m/2})^{1/m}}\right)^n\right)$ that computes MAJ_n.*

Proof. For each t satisfying $n/2 \leq t \leq n$, we will construct a $\Sigma_3^{(+k,-\ell)}$-circuit C_t of size $O\left(\sqrt{n} \cdot \left(\frac{2}{(|g_{m/2}|_{m/2})^{1/m}}\right)^n\right)$ such that $C_t(x)=1$ for every $x \in \{0,1\}^n$ with $|x|=t$ and $C_t(x)=0$ for every $x \in \{0,1\}^n$ with $|x|<n/2$. By taking the OR of all C_t's, a desired depth-three circuit will be obtained.

For each t, let f_t be a function on n variables defined as in the statement of Lemma 11. Then, by Theorem 10, there exists a $\Sigma_3^{(+k,-\ell)}$-circuit C_t of size at most

$$\frac{\binom{n}{t}}{|f_t|_t} \ln \binom{n}{t} \tag{1}$$

such that $C_t(x) = 1$ for every x with $|x| = t$ and $C_t(x) = 0$ for every x with $|x| < n/2$.

By Lemma 11, Eq. (1) is upper bounded by

$$\frac{\binom{n}{n/2}}{|f|_{n/2}} \ln \binom{n}{n/2} = O\left(\frac{2^n}{\sqrt{n} \cdot |f|_{n/2}} \ln 2^n\right) = O\left(\sqrt{n} \cdot \left(\frac{2}{(|g|_{m/2})^{1/m}}\right)^n\right).$$

This completes the proof of Theorem 12. ◀

4.3.2 Construction for $k = 3$

By the blow-up lemma described in the previous section, what we need is to find a (list of) $(+3, \cdot)$-CNF formula that is consistent with the majority function satisfying the increasing property. For a Boolean function f on n variables, we call a list $(|f|_0, |f|_1, \ldots, |f|_n)$ as a *profile* of f.

▶ **Fact 13.** *There exists a* $(+3, -4)$*-CNF formula on 12 variables that is consistent with* MAJ_{12} *whose profile is* $(0, 0, 0, 0, 0, 0, 217, 394, 363, 196, 66, 12, 1)$*. There also exists a* $(+3, -6)$*-CNF formula on 16 variables that is consistent with* MAJ_{16} *whose profile is* $(0, \ldots, 0, 1314, 2933, 3547, 2710, 1424, 510, 120, 16, 1)$*.*

As to the case of IP_n, we use an IP (integer programming) solver to find these formulas. Essentially, our task is to find a CNF formula g consistent with MAJ_n that maximizes $|g|_{n/2}$. This can easily be formulated by an IP problem as was described in Section 4.2. In certain cases, we impose some additional constraints on the IP problem to narrow the search space. For this reason, most of our base functions have not been shown to be optimal.

Proof. A certificate for the first statement is the AND of the following 26 clauses.

```
(1 2 3 -4)(1 2 4 -3)(1 3 4 -2)(5 6 7 -8)(5 6 8 -7)
(2 3 4)(7 8 9)(7 8 10)
(2 9 10 -3 -4 -5 -6)(3 9 10 -2 -4 -5 -6)(4 9 10 -2 -3 -5 -6)
(5 9 10 -2 -3 -4)(6 9 10 -2 -3 -4)(7 9 10 -1 -5 -6)(8 9 10 -1 -5 -6)
(5 6 11 -9 -10)(5 6 12 -9 -10)(9 10 11 -12)(9 10 12 -11)
(2 11 12 -3 -4 -7 -8)(3 11 12 -2 -4 -7 -8)(4 11 12 -2 -3 -7 -8)
(5 11 12 -1)(6 11 12 -1)(5 11 12 -6)(6 11 12 -5)
```

Here, for example, `(1 2 3 -4)` denotes the clause $(x_1 \vee x_2 \vee x_3 \vee \overline{x_4})$. The verification is easy by using a computer (but not so easy by hand).

A certificate for the second statement, which has 99 clauses, is provided at the aforementioned GitLab repository. ◀

It is easy to check that both profiles satisfy the increasing property. Hence, the second statement of Fact 13 and Theorem 12 immediately imply the following bound.

▶ **Theorem 14.** *There is a* $\Sigma_3^{(+3,-6)}$*-circuit of size* $O(d^n)$ *that computes* MAJ_n*, where* $d = 2/(1314^{1/16}) < 1.2768$*.* ⌟

Here, the values 1314 and 16 in the statement of Theorem 14 represent the eighth element (counting from zero) of the profile and the number of variables in the base function, respectively.

4.3.3 Construction for $k = 4$

▶ **Fact 15.** *There exists a* $(+4, -4)$*-CNF formula on* 10 *variables consistent with* MAJ_{10} *whose profile is* $(0, 0, 0, 0, 0, 160, 120, 120, 45, 10, 1)$.

Proof. Unlike the case of $k = 3$, our certificate for Fact 15 is well-structured.

We show a CNF formula g consisting of two classes of clauses. To simplify the notation, we start the indexing of input variables from 0 instead of 1, i.e., g is a CNF formula over $\{x_0, x_1, \ldots, x_9\}$. The first class consists of $\binom{5}{2}$ clauses all of them are monotone:

$$(x_{2i} \vee x_{2i+1} \vee x_{2j} \vee x_{2j+1}) \qquad (\forall \{i, j\} \subseteq \{0, 1, \ldots, 4\}).$$

The second class consists of $\binom{5}{2} \cdot 2^4 = 80$ clauses each of which contains four positive literals and four negative literals:

$$(x_{2i_1}^{t_1} \vee x_{2i_1+1}^{1-t_1} \vee x_{2i_2}^{t_2} \vee x_{2i_2+1}^{1-t_2} \vee x_{2i_3}^{t_3} \vee x_{2i_3+1}^{1-t_3} \vee x_{2i_4}^{t_4} \vee x_{2i_4+1}^{1-t_4}) \\ (\forall \{i_1, i_2, i_3, i_4\} \subseteq \{0, 1, \ldots, 4\}, \forall \{t_1, t_2, t_3, t_4\} \in \{0, 1\}^4),$$

where x^0 denotes $\overline{x}$ and x^1 denotes x itself. It is not hard to verify that g is consistent with MAJ_{10} and has a profile $(0, 0, 0, 0, 0, 160, 120, 120, 45, 10, 1)$. ◀

Note that the maximum value of $|\phi|_5$ over all monotone 4-CNF formulas ϕ on 10 variables consistent with MAJ_{10} seems to be 136, which is much smaller than $|g|_5 = 160$.

Since $|g|_6$ is not large enough, the function g alone is not sufficient to form a list satisfying the increasing property. We need to introduce another function h on 10 variables. It is a monotone CNF consisting of 20 clauses, each of which contains four variables. The clauses of h are:

```
(0 1 2 4) (0 1 5 8) (0 1 6 9) (0 2 3 9) (0 2 7 8) (0 3 4 5) (0 4 6 7)
(0 5 7 9) (1 2 3 6) (1 2 5 7) (1 3 4 8) (1 4 7 9) (1 6 7 8) (2 3 5 8)
(2 4 5 9) (2 4 6 8) (3 4 6 9) (3 5 6 7) (3 7 8 9) (5 6 8 9).
```

We can see that h has a profile $(0, 0, 0, 0, 0, 132, 190, 120, 45, 10, 1)$. In fact, h is a function that maximizes $|h|_6$ over all monotone 4-CNF formulas on 10 variables that is consistent with MAJ_{10}. Now the list $\mathbf{g} = (g, h, h, h, h, h)$ satisfies the increasing property. Since $2/(|g|_5)^{1/10} = 2/160^{1/10} \sim 1.20398$, we have the following theorem.

▶ **Theorem 16.** *There is a* $\Sigma_3^{(+4,-4)}$*-circuit of size* $O(1.2040^n)$ *that computes* MAJ_n. ⌟

4.3.4 Construction for $k = 5$

For $k = 5$, our best bound relies on a CNF formula with the following property.

▶ **Fact 17.** *There exists a* $(+5, -7)$*-CNF formula on* 16 *variables that is consistent with* MAJ_{16} *whose profile is* $(0, \ldots, 0, 4958, 5312, 4890, 3353, 1820, 560, 120, 16, 1)$. ⌟

We provide a certificate for Fact 17, which has 6817(!) clauses, at the aforementioned GitLab repository. Since this profile satisfies the increasing property, we have the following bound.

▶ **Theorem 18.** *There exists a* $\Sigma_3^{(+5,-7)}$*-circuit of size* $O(d^n)$ *that computes* MAJ_n*, where* $d = 2/(4958^{1/16}) < 1.1751$. ⌟

4.4 Construction based on Covering Design

As we have seen in Introduction, an asymptotically tight bound for the majority function is not known even for monotone Σ_3^k-circuits. It is not hard to show that if we use a monotone k-CNF formula representing $\mathsf{MAJ}_{2(k-1)}$ as a building block, we have a Σ_3^k-circuit of size $\tilde{O}(d^n)$ where $d = 2/\binom{2(k-1)}{k-1}^{1/2(k-1)}$.

One possibility for improvement is to use a combinatorial object called *covering design*, which has a rich history of research (e.g., refer to [5, 7, 13, 24]). A (v, k, t)-*covering design* is a collection of k-element subsets, called blocks, of $\{1, 2, \ldots, v\}$, such that any t-element subset is contained in at least one block. Let $C(v, k, t)$ be the smallest possible number of blocks in a (v, k, t)-covering design.

We can use a covering design as a building block of depth-three circuits for the majority function. Given a $(2k, k, k-1)$-covering design $\mathcal{S}$, let $f_{\mathcal{S}}$ be a monotone k-CNF formula defined as

$$f_{\mathcal{S}} := \bigwedge_{S \in \mathcal{S}} \bigvee_{i \in \overline{S}} x_i,$$

where $\overline{S}$ denotes the set $\{1, 2, \ldots, 2k\} \backslash S$. It is easy to see that $f_{\mathcal{S}}$ is consistent with MAJ_{2k} and satisfies $|f_{\mathcal{S}}|_k = \binom{2k}{k} - |\mathcal{S}|$.

Generally, such a covering design gives a monotone Σ_3^k-circuit of size $\tilde{O}(d^n)$, where

$$d = \left(\binom{2k}{k} - C(2k, k, k-1) \right)^{\frac{1}{2k}}.$$

The size is smaller than the size of a circuit relying on the direct product of k-CNFs formula representing $\mathsf{MAJ}_{2(k-1)}$, if

$$d < \binom{2(k-1)}{k-1}^{\frac{1}{2(k-1)}}. \tag{2}$$

The exact value of $C(2k, k, k-1)$ is known for $k \leq 6$; $C(6, 3, 2) = 6$, $C(8, 4, 3) = 14$, $C(10, 5, 4) = 51$ and $C(12, 6, 5) = 132$ (see e.g., an online database by Gordon [13]). The situation is mixed for this range of k. InEq. (2) holds when $k = 4$ and 6, but it does not hold when $k = 3$ and 5. We think that the problem of determining the values of k that satisfy InEq. (2) is already an intriguing problem.

4.5 Summary

We summarize the upper bounds on the size of a Σ_3-circuit computing MAJ_n in which each bottom gate contains at most k positive literals for $k = 3, 4$ and 5 (see Table 1). In each entry in Table 1, (a,b) represents that the upper bound is $O((2/b^{1/a})^n)$ which is obtained from a base function ϕ on a-variable consistent with MAJ_a satisfying $|\phi^{-1}(1)|_{a/2} = b$. The first line shows the bounds by a direct product of k-CNF formulas representing $\mathsf{MAJ}_{2(k-1)}$. The second line shows the bounds based on a covering design described in Section 4.4. Both circuits are monotone. The third line shows the bounds by our construction using negations. It is very likely that these bounds can further be improved. The lower bounds given by Theorem 8 (for a suitable choice of s found by a numerical calculation) are shown in the last line.

Table 1 The upper bounds on the size of $\Sigma_3^{(+k,-\infty)}$-circuit for $k = 3, 4$ and 5.

	$\Sigma_3^{(+3,-\infty)}$	$\Sigma_3^{(+4,-\infty)}$	$\Sigma_3^{(+5,-\infty)}$
block threshold	$O(1.2779^n)$ (4,6)	$O(1.2140^n)$ (6,20)	$O(1.1760^n)$ (8,70)
covering design	$O(1.2883^n)$ (6,14)	$O(1.2093^n)$ (8,56)	$O(1.1769^n)$ (10,201)
with negations	$O(1.2768^n)$ (16, 1314)	$O(1.2040^n)$ (10,160)	$O(1.1751^n)$ (16,4958)
lower bound	$\Omega(1.2247^n)$	$\Omega(1.1547^n)$	$\Omega(1.1180^n)$

5 Concluding Remark

In this paper, we give some exotic but efficient constructions of Σ_3-circuits for IP_n and MAJ_n. Our construction relies on a computer search. As an outcome of this approach, in some cases, it seems hard to give a simple explanation on why the obtained circuits work. Extracting the reasoning from our circuits would be a good step for further research. Finally, we would like to emphasize that the question of whether an optimal circuit is inherently looking random would be an intriguing challenge.

References

1 Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Comb.*, 7(1):1–22, 1987. doi:10.1007/BF02579196.

2 Kazuyuki Amano. On the size of depth-two threshold circuits for the inner product mod 2 function. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, *Language and Automata Theory and Applications – 14th International Conference, LATA 2020*, volume 12038 of *Lecture Notes in Computer Science*, pages 235–247. Springer, 2020. doi:10.1007/978-3-030-40608-0_16.

3 Ravi B. Boppana. Threshold functions and bounded depth monotone circuits. *J. Comput. Syst. Sci.*, 32(2):222–229, 1986. doi:10.1016/0022-0000(86)90027-9.

4 Bruno Pasqualotto Cavalar and Igor C. Oliveira. Constant-depth circuits vs. monotone circuits. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 29:1–29:37. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CCC.2023.29.

5 Jeffrey H. Dinitz Charles J. Colbourn. *Handbook of Combinatorial Designs (Discrete Mathematics and Its Applications)*. Chapman and Hall/CRC, 2006.

6 Vlado Dancík. Complexity of boolean functions over bases with unbounded fan-in gates. *Inf. Process. Lett.*, 57(1):31–34, 1996. doi:10.1016/0020-0190(95)00182-4.

7 Paul Erdős and Joel Spencer. *Probabilistic Methods in Combinatorics*. Academic Press, 1974.

8 Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-3n lower bound for the circuit complexity of an explicit function. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 89–98. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.19.

9 Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. Syst. Sci.*, 65(4):612–625, 2002. doi:10.1016/S0022-0000(02)00019-3.

10 Peter Frankl, Svyatoslav Gryaznov, and Navid Talebanfard. A variant of the vc-dimension with applications to depth-3 circuits. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, volume 215 of *LIPIcs*, pages 72:1–72:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.72.

11 Alexander Golovnev, Alexander S. Kulikov, and R. Ryan Williams. Circuit depth reductions. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, volume 185 of *LIPIcs*, pages 24:1–24:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.24.

12 Mika Göös, Ziyi Guan, and Tiberiu Mosnoi. Depth-3 circuits for inner product. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France*, volume 272 of *LIPIcs*, pages 51:1–51:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.MFCS.2023.51.

13 Daniel M. Gordon. Covering designs. https://www.dmgordon.org/cover/.

14 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL: https://www.gurobi.com.

15 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993. doi:10.1016/0022-0000(93)90001-D.

16 Johan Håstad, Stasys Jukna, and Pavel Pudlák. Top-down lower bounds for depth-three circuits. *Comput. Complex.*, 5(2):99–112, 1995. doi:10.1007/BF01268140.

17 Shuichi Hirahara. A duality between depth-three formulas and approximation by depth-two. *CoRR*, abs/1705.03588, 2017. arXiv:1705.03588.

18 Daniel M. Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 633–643. ACM, 2016. doi:10.1145/2897518.2897636.

19 Maria M. Klawe, Wolfgang J. Paul, Nicholas Pippenger, and Mihalis Yannakakis. On monotone formulae with restricted depth (preliminary version). In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, STOC 1984*, pages 480–487. ACM, 1984. doi:10.1145/800057.808717.

20 Jiatu Li and Tianqi Yang. $3.1n - o(n)$ circuit lower bounds for explicit functions. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20–24, 2022*, pages 1180–1193. ACM, 2022. doi:10.1145/3519935.3519976.

21 Ramamohan Paturi, Michael E. Saks, and Francis Zane. Exponential lower bounds for depth 3 boolean circuits. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, STOC 1997*, pages 86–91. ACM, 1997. doi:10.1145/258533.258556.

22 W. V. Quine. Two theorems about truth-functions. *Boletín de la Sociedad Matemática Mexicana*, 10(1–2):64–70, 1953.

23 Alexander A. Razborov. On the method of approximations. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, STOC 1989*, pages 167–176. ACM, 1989. doi:10.1145/73007.73023.

24 Vojtech Rödl. On a packing and covering problem. *Eur. J. Comb.*, 6(1):69–78, 1985. doi:10.1016/S0195-6698(85)80023-8.

25 Igor' S. Sergeev. On the complexity of bounded-depth circuits and formulas over the basis of fan-in gates. *Discrete Mathematics and Applications*, 29:241–254, 2019. doi:10.1515/dma-2019-0022.

A Appendix

A.1 Certificates for IP_n

The following is a certificate of $\tilde{s}_3^3(\mathsf{IP}_5) \leq 6$.

```
   f1:  (-x1 -y1 -x2)(-x3 -x4 -y4)(x1 -y1 x2)(x3 x4 -x5)(-x3 -y3 x5)(-x4 -y4
x5)(y1 x2)(-x2 y2)(-x3 y3 -x5)(x3 y4 -x5)(-x5 y5)
   f2:  (-x1 -y1)(-x2 -y2 -y4)(x3 -x5 -y5)(x4 -y4)(-x3 -y3 x5) (y3 -x5 -y5)
(x2 -y2 y4)(y2 y4)(-x3 -y3 y5)
```

```
   f3:  (-x3 -y3)(-x2 -y2 -y5)(x1 -x4 -y4)(-x1 -y1 x4)(x5 -y5)(y1 -x4 -y4)
(-x1 -y1 y4)(x2 y5)(y2 y5)
   f4:  (-y3 -x4 -y4)(-y2 -x5 -y5)(x1)(x2)(-x1 x3)(y1 -x2 -x3)(y2 x5)(-x2 y3
x4)(-x1 y3 y4)(-x2 y2 y5)
   f5:  (-x1 -y1 -x3)(-y3 -x5 -y5)(x2 -x4 -y4)(x3 -x5 -y5)(x1 x3)(-x2 -y2
x4)(-x1 y1 x3)(y2 -x4 -y4)(-x3 y3 x5)(-x2 -y2 y4)(-x3 y3 y5)
   f6:  (x1)(y1)(-x2 -y2 x5)(x2 -x5 -y5)(x3 -x4 -y4)(-x3 -y3 x4)(y2 -x5 -y5)
(y3 -x4 -y4)(-x3 -y3 y4)(-x2 -y2 y5)
```

The following is a certificate of $\tilde{s}_3^3(\overline{\mathsf{IP}_5}) \leq 5$.

```
   f1:  (-x1 -y1)(x2 -x4 -y4)(x3 -x5 -y5)(-x2 -y2 x4)(-x3 -y3 x5)
(y2 -x4 -y4)(y3 -x5 -y5)(-x2 -y2 y4)(-x3 -y3 y5)
   f2:  (-x2 -y2)(x1 -x5 -y5)(x3 -x4 -y4)(-x3 -y3 x4)(-x1 -y1 x5)
(y1 -x5 -y5)(y3 -x4 -y4)(-x3 -y3 y4)(-x1 -y1 y5)
   f3:  (-x3 -y3)(x1 -x2 -y2)(-x1 -y1 x2)(x4 -x5 -y5)(-x4 -y4 x5)
(y1 -x2 -y2)(-x1 -y1 y2)(y4 -x5 -y5)(-x4 -y4 y5)
   f4:  (-x4 -y4)(x1 -x3 -y3)(x2 -x5 -y5)(-x1 -y1 x3)(-x2 -y2 x5)
(y1 -x3 -y3)(y2 -x5 -y5)(-x1 -y1 y3)(-x2 -y2 y5)
   f5:  (-x5 -y5)(x1 -x4 -y4)(x2 -x3 -y3)(-x2 -y2 x3)(-x1 -y1 x4)
(y1 -x4 -y4)(y2 -x3 -y3)(-x2 -y2 y3)(-x1 -y1 y4)
```

The first set of formulas looks quite random, whereas the second set of formulas is well-structured.

The following is a certificate for $\tilde{s}_3^3(\mathsf{IP}_4) \leq 4$.

```
   f1:  (x1)(y1)(-x2 -y2)(x3 -x4 -y4)(-x3 -y3 x4)(y3 -x4 -y4)(-x3 -y3 y4)
   f2:  (x2)(y2)(-x3 -y3)(x1 -x4 -y4)(-x1 -y1 x4)(y1 -x4 -y4)(-x1 -y1 y4)
   f3:  (x3)(y3)(-x4 -y4)(x1 -x2 -y2)(-x1 -y1 x2)(y1 -x2 -y2)(-x1 -y1 y2)
   f4:  (x4)(y4)(-x1 -y1)(x2 -x3 -y3)(-x2 -y2 x3)(y2 -x3 -y3)(-x2 -y2 y3)
```

The following is a certificate for $\tilde{s}_3^3(\overline{\mathsf{IP}_4}) \leq 3$.

```
   f1:  (x1 -x3 -y3)(-x1 -y1 x3)(y1 -x3 -y3)(-x1 -y1 y3)(x2 -x4 -y4)
(-x2 -y2 x4)(y2 -x4 -y4)(-x2 -y2 y4)
   f2:  (x1 -x4 -y4)(-x1 -y1 x4)(y1 -x4 -y4)(-x1 -y1 y4)(x2 -x3 -y3)
(-x2 -y2 x3)(y2 -x3 -y3)(-x2 -y2 y3)
   f3:  (x1 -x2 -y2)(-x1 -y1 x2)(y1 -x2 -y2)(-x1 -y1 y2)(x3 -x4 -y4)
(-x3 -y3 x4)(y3 -x4 -y4)(-x3 -y3 y4)
```

A.2 Proof of Lemma 11

Proof of Lemma 11. We first verify the second part of the statement of Lemma 11, i.e., $|f_v|_t = 0$ for every $v < n/2$. This is almost obvious, since if an input vector $x \in \{0,1\}^n$ contains less than $n/2$ ones, then some function g_{s_i} gets an input containing less than $m/2$ ones, and hence it outputs 0 for x.

We now show the first part of the statement of Lemma 11. To this end, it is sufficient to show that, for every $t \geq n/2 + 1$, it holds that

$$\begin{aligned} |f_t|_t &\geq |f_{n/2}|_{n/2} \cdot \frac{\binom{n}{t}}{\binom{n}{n/2}} \\ &= |f_{n/2}|_{n/2} \prod_{\ell=n/2+1}^{t} \frac{n-\ell+1}{\ell}. \end{aligned} \tag{3}$$

Let $p = n/m$ and $z = \lfloor t/p \rfloor$. Let $\alpha = t \pmod{p}$. Since $\mathbf{g}$ satisfies the increasing property, we have

$$\begin{aligned}
|f_t|_t &\geq (|g_{z+1}|_{z+1})^{\alpha} (|g_z|_z)^{p-\alpha} \\
&\geq (|g_{m/2}|_{m/2})^p \left(\prod_{\ell'=m/2+1}^{z} \frac{m-\ell'+1}{\ell'-1} \right)^p \left(\frac{m-z}{z} \right)^{\alpha} \\
&= |f_{n/2}|_{n/2} \prod_{\ell=n/2+1}^{t} \frac{n-\ell+1}{\ell}.
\end{aligned}$$

This completes the proof of Lemma 11. ◀