New Bounds on the Multiplicative Complexity of Boolean Functions

Meltem Sönmez Turan

National Institute of Standards and Technology

Presented at BFA2022 – September 2022

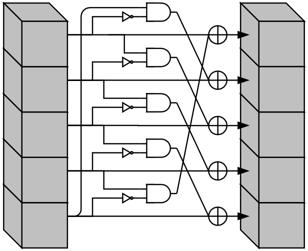
Overview

- Optimizing Boolean circuits
- Multiplicative Complexity (MC)
- Number of Boolean functions with MC k
- Open questions

Boolean Circuits

A *Boolean circuit* with n inputs and m outputs is a **directed acyclic graph** (DAG), where

- the inputs and the gates are *nodes*,
- the edges correspond to Boolean-valued *wires*,
- *fanin*/*fanout* of a node is the number of wires going in/out the node,
- the nodes with fanin zero are called *input nodes*,
- the nodes with fanout zero are called *output nodes*.



Circuit for Keccak s-box
https://keccak.team/figures.html

Straight Line Programs (SLPs)

An example SLP for the majority function:.

```
begin CIRCUIT MAJ3
# Description: The majority of x1,x2,x3
Inputs: x1:x3;
Outputs: y1;
GateSyntax: GateName Output Inputs
begin SLP
  XOR t1 x1 x2;
  XOR t2 x1 x3;
  AND t3 t1 t2;
  XOR y1 t3 x1
end SLP end CIRCUIT
```

Optimizing Boolean Circuits

Problem: Given a set of Boolean gates (e.g., `AND`, `NAND`, `XOR`, `NOR`), construct a circuit that computes a Boolean function that is optimal w.r.t. a target metric.

Target metric depends on the application.

- *Number of gates:* for *lightweight cryptography applications* running on constrained devices.
- *Number of nonlinear gates:* for *secure multi-party computation*, *zero-knowledge proofs* and *side channel protection*.
- *AND-depth:* for homomorphic encryption schemes.
- etc.

Benchmarking

Example circuits: [1]

Circuit	Gate count					Depth	
	All	AND	XOR	XNOR	NOT	Total	AND
AES S-Box	113	32	77	4	0	27	6
AES S-Box^{-1}	121	34	83	4	0	21	4
AES-128(k,m)	28 600	6400	21 356	844	0	326	60
AES-128(0,m)	21 392	5120	14 652	1620	0	325	60
SHA-256(m)	115 882	22 385	89 248	3894	355	5403	1604
SHA-256(cv,m)	118 287	22 632	92 802	2840	13	5458	1607

[1] NIST Circuit Complexity Team https://csrc.nist.gov/Projects/circuit-complexity

Multiplicative Complexity (MC)

Minimum number of nonlinear gates needed to implement f by a Boolean circuit

- Min. $\#$ of `AND` gates needed over the basis (`AND`, `XOR`, `NOT`).

Some known results:

- MC of a function with degree d is at least $d - 1$ (degree bound).
- Almost all $f \in B_n$ have MC at least $2^{n/2} - n - 1$ with high probability.
- MC of all Boolean functions with $n \leq 6$
- Results on special classes of Boolean functions: quadratic, cubic and symmetric functions etc.

Number of Boolean functions with MC k

$\lambda(n,k)$: the number of n variable Boolean functions with MC k

- useful to lower bound the MC of Boolean functions. e.g., 7-AND gates are not enough to compute 8-variable Boolean functions.

Boyar et al. Bound: $\lambda(n,k) \leq 2^{k^2+2k+2kn+n+1}$

Proof (sketch): The inputs of the *i*th AND gate, denoted a_i, is a subset of

$$\{x_1,\ldots,x_n,a_1,a_2,\ldots,a_{i-1},\mathbf{1}\},$$

with $2^{2(n+i)}$ possible input choices. The bound is achieved by summing all possible inputs to the AND gates, and adding all possible linear terms and a_i's to be final output.

$$\begin{aligned} 2^{n+k+1}\prod_{i=1}^{k}(2^{n+i})^2 =& 2^{n+k+1}2^{2kn}2^{\sum_{i=1}^{k}2i} \\ =& 2^{k^2+2k+2kn+n+1} \end{aligned}$$

Affine Equivalence

Boolean functions $f, g \in B_n$ are affine equivalent if there exists a transformation of the form

$$f(x) = g(Ax + a) + b \cdot x + c,$$

where A is a non-singular $n \times n$ matrix over $\mathbb{F}_2$; $a, b \in \mathbb{F}_2^n$, and $c \in \mathbb{F}_2$.

- The set of affine equivalent functions constitute an equivalence class denoted by $[f]$, where f is an arbitrary function from the class.
- *Sizes of equivalence classes*

$$\frac{\#\text{ affine transformations}}{\#\text{ self mappings}}$$

(self mappings of f is an affine transformation that outputs f).

MC is affine invariant.

Boolean functions with MC 1 and 2

Boolean functions with MC 1 [FP02]

- Functions with MC 1 are affine equivalent to x_1x_2.
- The number of n-variable Boolean functions with MC 1 is $2\binom{2^n}{3}$.

Boolean functions with MC 2 [FTT17]

- Functions with MC 2 are affine equivalent to one of these functions:
  $x_1x_2x_3$
  $x_1x_2x_3 + x_1x_4$
  $x_1x_2 + x_3x_4$
- The number of n-variable Boolean functions with MC 2 is

$$2^n(2^n-1)(2^n-2)(2^n-4)\left(\frac{2}{21}+\frac{2^n-8}{12}+\frac{2^n-8}{360}\right).$$

Affine Equivalence Classes with MC 3 [CTP19]

Dimension 4:

$x_1x_2x_3x_4$
$x_1x_2 + x_1x_2x_3x_4$
$x_2x_3 + x_1x_4 + x_1x_2x_3x_4$

Dimension 5:

$x_3x_4 + x_1x_5 + x_1x_2x_5 + x_1x_2x_3x_4$	$x_3x_4 + x_1x_3x_4 + x_1x_2x_5$
$x_2x_4 + x_1x_5 + x_1x_2x_3$	$x_4x_5 + x_1x_2x_3$
$x_1x_2x_5 + x_1x_2x_3x_4$	$x_1x_3x_4 + x_1x_2x_5$
$x_2x_3x_5 + x_1x_4x_5 + x_1x_2x_3x_4$	$x_3x_5 + x_1x_2x_5 + x_1x_2x_3x_4$
$x_1x_3 + x_1x_2x_5 + x_1x_2x_3x_4$	$x_3x_4 + x_1x_2x_5 + x_1x_2x_3x_4$
$x_1x_5 + x_1x_2x_3x_4$	$x_2x_3 + x_1x_5 + x_1x_2x_3x_4$
$x_2x_3 + x_2x_3x_5 + x_1x_4x_5 + x_1x_2x_3x_4$	$x_1x_5 + x_1x_2x_5 + x_1x_2x_3x_4$

Dimension 6:

$x_3x_4 + x_2x_5 + x_1x_6$	$x_1x_6 + x_1x_3x_4 + x_1x_2x_5$
$x_3x_4 + x_1x_6 + x_1x_3x_4 + x_1x_2x_5$	$x_4x_5 + x_1x_6 + x_1x_2x_3$
$x_1x_6 + x_1x_2x_5 + x_1x_2x_3x_4$	$x_5x_6 + x_3x_4x_5 + x_1x_2x_6 + x_1x_2x_3x_4$
$x_3x_4 + x_1x_6 + x_1x_2x_5 + x_1x_2x_3x_4$	

Number of Boolean functions with MC 3 [CTP19]

The number of n-variable Boolean functions with MC 3 is

$$\lambda(n,3) = \sum_{d=4}^{6} \left(2^{n-d} \prod_{i=0}^{d-1} \frac{2^n - 2^i}{2^d - 2^i} \beta(d,3) \right)$$

where

$$\begin{aligned} \beta(4,3) &= 32\,768, \\ \beta(5,3) &= 775\,728\,128, \\ \beta(6,3) &= 183\,894\,007\,808. \end{aligned}$$

Affine Equivalence Classes with MC 4 [CTP19]

The number of n-variable Boolean functions with MC 4 is

$$\lambda(n,4) = \sum_{d=5}^{8} \left(2^{n-d} \prod_{i=0}^{d-1} \frac{2^n - 2^i}{2^d - 2^i} \beta(d,4) \right)$$

where

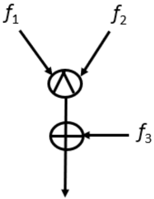
$$\begin{aligned}
\beta(5,4) &= 3\,515\,396\,096, \\
\beta(6,4) &= 7\,944\,313\,921\,970\,176, \\
\beta(7,4) &= 8\,217\,135\,092\,528\,316\,416, \\
\beta(8,4) &= 5\,502\,415\,308\,673\,798\,144.
\end{aligned}$$

Comparison of Boyar et al. bound and exact numbers

MC	Bound	n =6	n = 7	n =8	n =9	n =10	n =11	n =12	n =13	n =14	n =15	n =16
1	Exact	16.34	19.38	22.38	25.40	28.41	31.41	34.41	37.41	40.41	43.41	46.41
	Bound	22	25	28	31	34	37	40	43	46	49	52
2	Exact	26.13	31.30	36.38	41.42	46.44	51.45	56.45	61.45	66.46	71.46	76.46
	Bound	39	44	49	54	59	64	69	74	79	84	89
3	Exact	38.03	45.64	52.92	60.05	67.12	74.15	81.17	88.18	95.18	102.18	109.18
	Bound	58	65	72	79	86	93	100	107	114	121	128
4	Exact	52.81	63.15	71.94	80.29	88.46	96.56	104.63	112.70	120.82	129.02	137.35
	Bound	79	88	97	106	115	124	133	142	151	160	169

Table: Number of Boolean functions with MC 1, 2, 3, and 4 compared to the Boyar et al. bound on a log scale with base 2

Observation 1 - Elimination of equivalent inputs



$$
\begin{aligned}
(f_1, f_2, f_3) &\to f_1 f_2 + f_3 \\
(f_2, f_1, f_3) &\to f_1 f_2 + f_3 \\
(f_1 + f_2, f_2, f_3 + f_2) &\to f_1 f_2 + f_2 + f_2 + f_3 = f_1 f_2 + f_3 \\
(f_2, f_1 + f_2, f_3 + f_2) &\to f_2 f_1 + f_2 + f_2 + f_3 = f_1 f_2 + f_3 \\
(f_1, f_2 + f_1, f_3 + f_1) &\to f_2 f_1 + f_1 + f_3 + f_1 = f_1 f_2 + f_3 \\
(f_2 + f_1, f_1, f_3 + f_1) &\to f_2 f_1 + f_1 + f_3 + f_1 = f_1 f_2 + f_3
\end{aligned}
$$

All inputs generate the same output as $f_1 f_2 + f_3$, and counted separately in Boyar et al. bound.

Observation 2 - Elimination of the constant 1

$$
\begin{aligned}
(f_1, f_2, f_3) &\to f_1 f_2 + f_3 \\
(f_1 + \mathtt{1}, f_2, f_3 + f_2) &\to f_1 f_2 + f_2 + f_3 + f_2 = f_1 f_2 + f_3 \\
(f_1, f_2 + \mathtt{1}, f_3 + f_1) &\to f_1 f_2 + f_1 + f_3 + f_1 = f_1 f_2 + f_3 \\
(f_1 + \mathtt{1}, f_2 + \mathtt{1}, f_3 + f_1 + f_2) &\to f_2 f_1 + f_1 + f_2 + f_3 + f_1 + f_2 = f_1 f_2 + f_3
\end{aligned}
$$

All inputs generate the same output as $f_1 f_2 + f_3$, and counted separately in Boyar et al. bound.

Improved Bound

The number of n-variable Boolean functions that can be generated with k-AND gates is at most

$$\begin{aligned}\lambda(n,k) &\leq 2^{n+k+1}\prod_{i=1}^{k}\frac{1}{24}(2^{n+i+1})^2,\\ &\leq 2^{k^2+2nk+n-k+1}3^{-k}.\end{aligned}$$

Proof (sketch): Let f_1 and f_2 be the left and right inputs of an AND gate. For each AND:

- only count the lexicographically smallest among

  $$(f_1,f_2),(f_2,f_1),(f_1+f_2,f_2),(f_2,f_1+f_2),(f_1+f_2,f_1),(f_1,f_1+f_2)$$

  (improvement by a factor of 6)
- only consider f_1 and f_2 without the constant term (improvement by a factor of 4)

Comparison of Boyar et al. and improved bound

MC	Bound	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$	$n=12$	$n=13$	$n=14$	$n=15$	$n=16$
1	Exact	16.34	19.38	22.38	25.40	28.41	31.41	34.41	37.41	40.41	43.41	46.41
	Bound	22	25	28	31	34	37	40	43	46	49	52
	Improved	17.42	20.42	23.42	26.42	29.42	32.42	35.42	38.42	41.42	44.42	47.42
2	Exact	26.13	31.30	36.38	41.42	46.44	51.45	56.45	61.45	66.46	71.46	76.46
	Bound	39	44	49	54	59	64	69	74	79	84	89
	Improved	29.83	34.83	39.83	44.83	49.83	54.83	59.83	64.83	69.83	74.83	79.83
3	Exact	38.03	45.64	52.92	60.05	67.12	74.15	81.17	88.18	95.18	102.18	109.18
	Bound	58	65	72	79	86	93	100	107	114	121	128
	Improved	44.25	51.25	58.25	65.25	72.25	79.25	86.25	93.25	100.25	107.25	114.25
4	Exact	52.81	63.15	71.94	80.29	88.46	96.56	104.63	112.70	120.82	129.02	137.35
	Bound	79	88	97	106	115	124	133	142	151	160	169
	Improved	60.66	69.66	78.66	87.66	96.66	105.66	114.66	123.66	132.66	141.66	150.66

Table: The improved bound for the number of Boolean functions with MC 1, 2, 3, and 4 compared to the Boyar et al. bound on a log scale with base 2

Conclusion

- Studied the number of Boolean functions with a specific MC.
- Improved the Boyar et al. bound by a factor of 24 for each `AND` gate.

Open questions on the MC of Boolean functions:

- Generic heuristics to implement Boolean functions with $n \geq 7$ with small number of `AND` gates – Best known upper bound on the MC of 7-variable Boolean functions is 13.
- Extending the results to vectorial Boolean functions – Exhaustive list of affine equivalence classes for vectorial Boolean functions would be useful, e.g., 5-bit to 3-bit, 6-bit to 2-bits.

Thanks! Questions?

- NIST Circuit Complexity Project Webpage: https://csrc.nist.gov/Projects/Circuit-Complexity
- GitHubLink: https://github.com/usnistgov/Circuits/
- Contact emails:
  `meltem.turan@nist.gov`
  `circuit_complexity@nist.gov`

References

BPP00 J. Boyar, R. Peralta, and D. Pochuev, *On the multiplicative complexity of Boolean functions over the basis* $(\wedge, \oplus, 1)$ Theoretical Computer Science, vol. 235, no. 1, pp. 43 – 57, 2000.

TP14 M. Sönmez Turan and R. Peralta. *The Multiplicative Complexity of Boolean functions on Four and Five Variables* LightSec 2014, Turkey.

CTP18 Ç. Çalık, M. Sönmez Turan, R. Peralta, *The Multiplicative Complexity of 6-variable Boolean Functions* Cryptography and Communications 2018.

FTT17 M. G. Find, D. Smith-Tone, M. Sönmez Turan, *The Number of Boolean Functions with Multiplicative Complexity 2* International Journal of Information and Coding Theory, 2017.

CTP19 Ç. Çalık, M. Sönmez Turan, R. Peralta, *Boolean Functions with Multiplicative Complexity 3 and 4* Cryptography and Communications 2019.

STP21 M. Sönmez Turan and R. Peralta. *On the Multiplicative Complexity of Cubic Boolean Functions* IACR Cryptol. ePrint Arch. 2021: 1041 (2021)

BCS19 L. T. A. N. Brandão, Ç. Çalik, M. Sönmez Turan, R. Peralta, *Upper bounds on the multiplicative complexity of symmetric Boolean functions* Cryptogr. Commun. 11(6): 1339-1362 (2019)