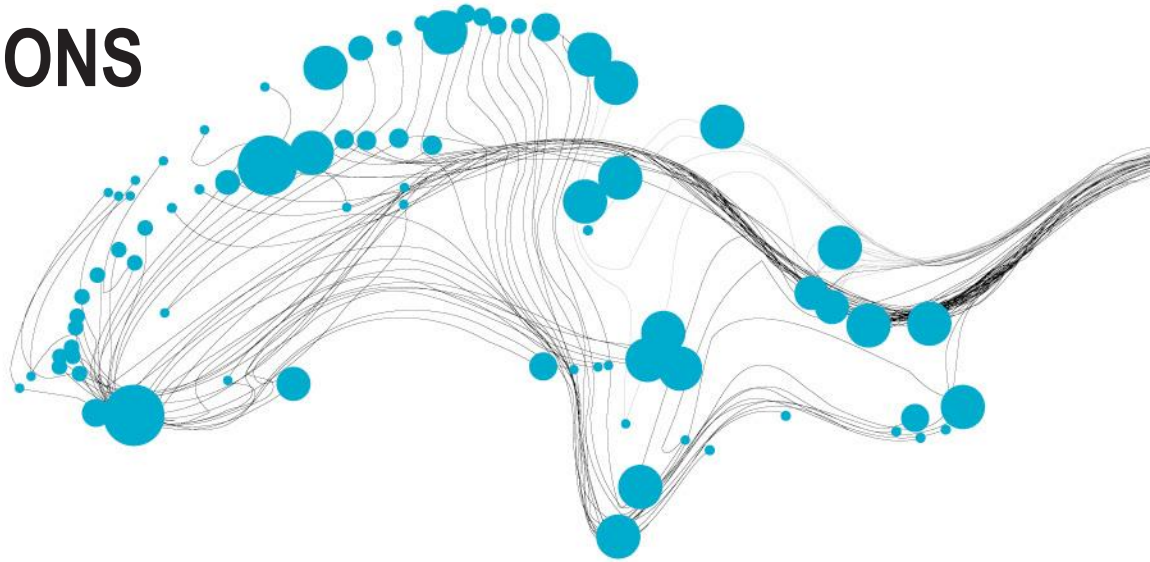
A SYSTEMATIC EVALUATION OF OPTIMIZING HIGHLY NONLINEAR BOOLEAN FUNCTIONS IN ODD SIZES WITH EA

CLAUDE CARLET, MARKO ĐURASEVIĆ, DOMAGOJ JAKOBOVIC, STJEPAN PICEK, LUCA MARIOT



EUROGP 2025

TRIESTE, 23 APRIL 2025

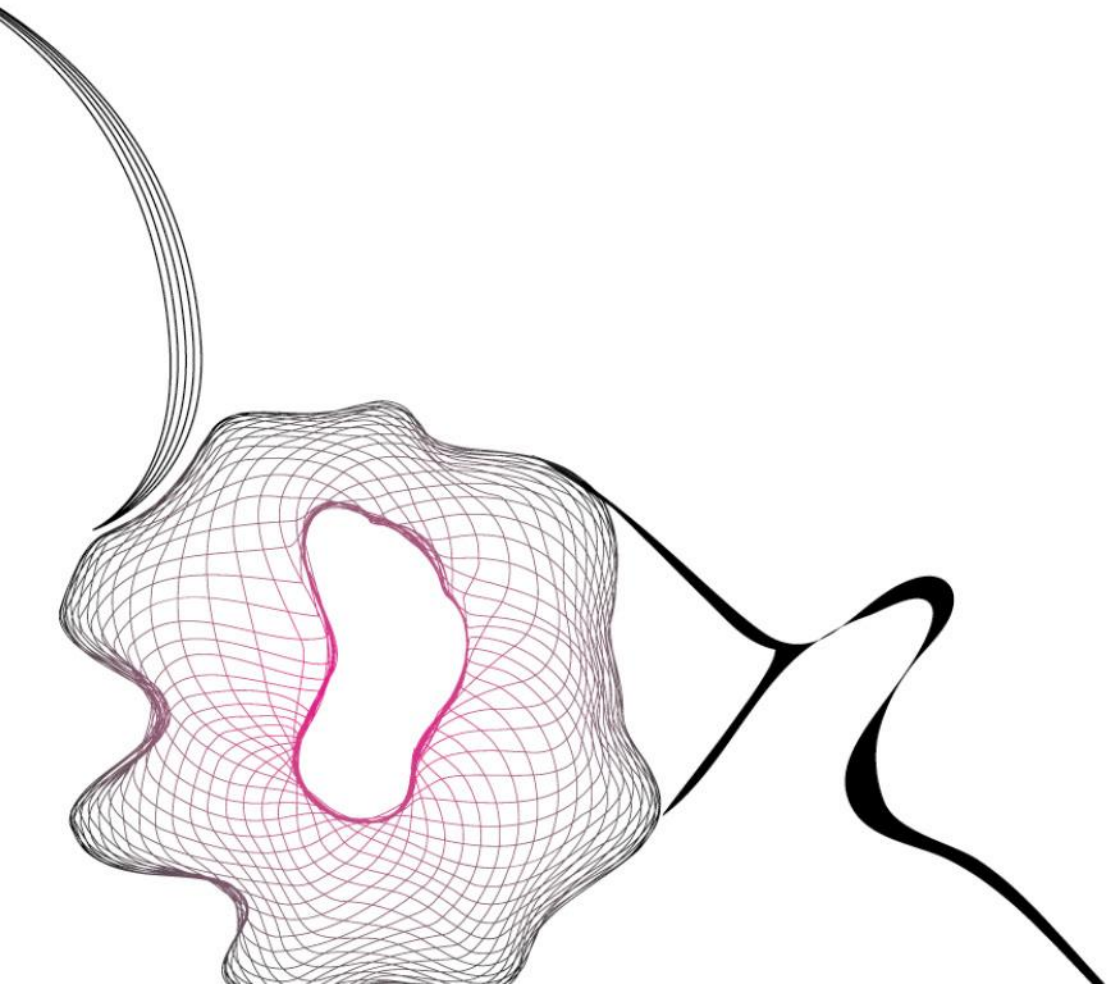
UNIVERSITAS BERGENSIS

UNIVERSITAS STUDIORUM ZAGRABIENSIS MDCLXIX

Radboud University

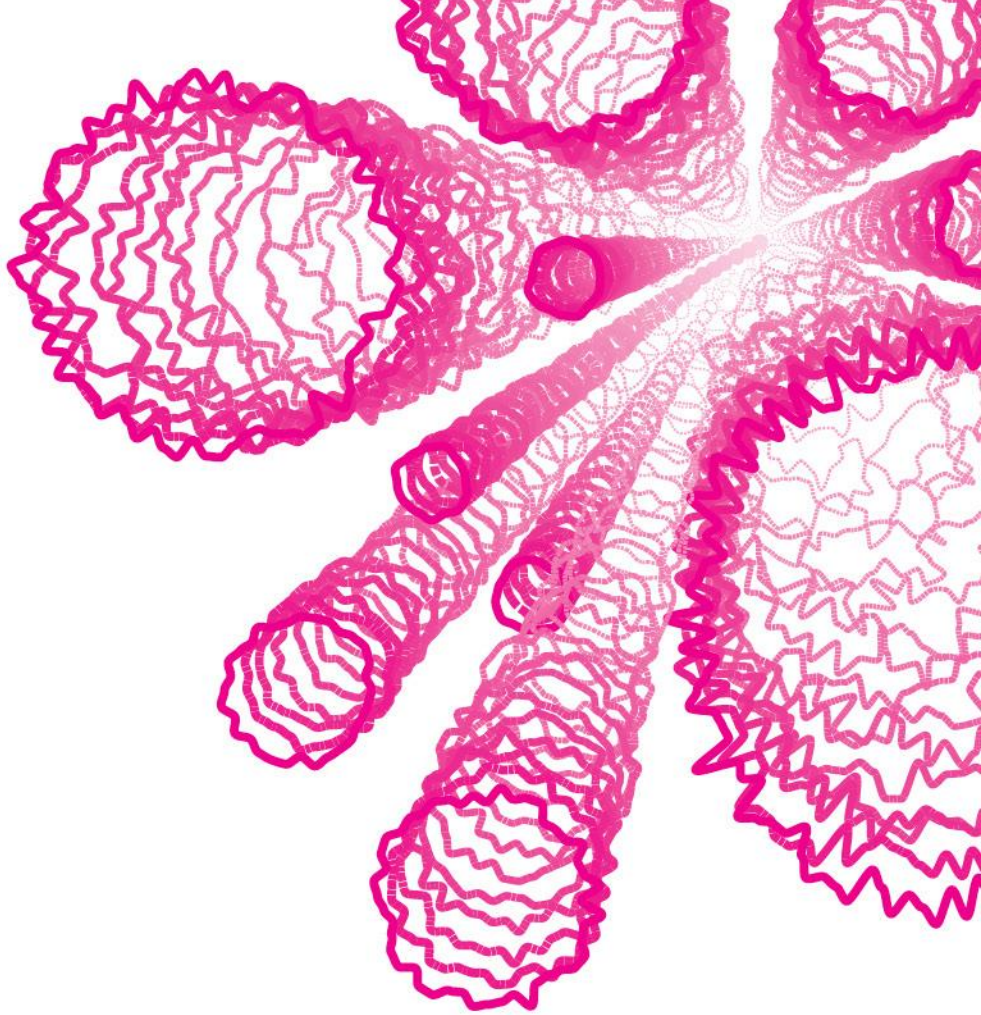
UNIVERSITY OF TWENTE.

SUMMARY



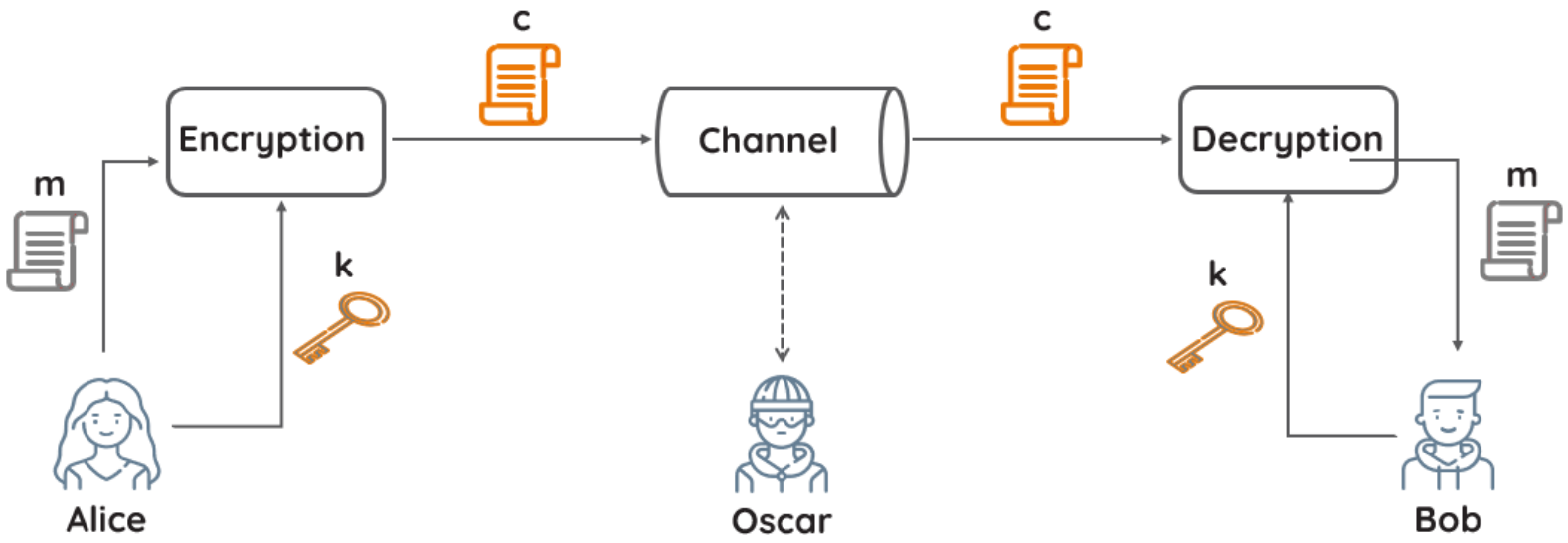
- Optimizing Nonlinearity of Boolean functions
- Evolutionary Algorithms Setup
- Experimental Results
- Conclusions and Insights

OPTIMIZING NONLINEARITY OF BOOLEAN FUNCTIONS

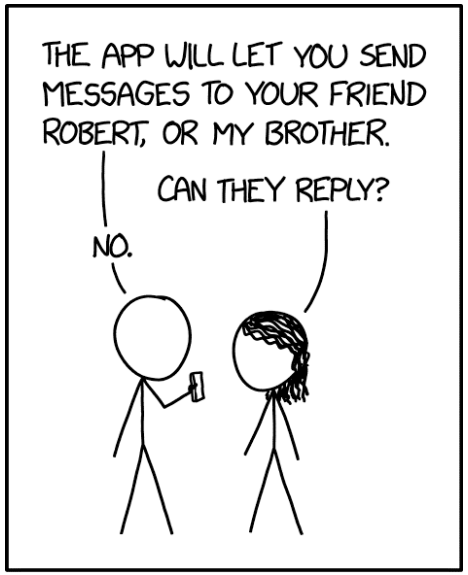


UNIVERSITY
OF TWENTE.

SYMMETRIC-KEY ENCRYPTION

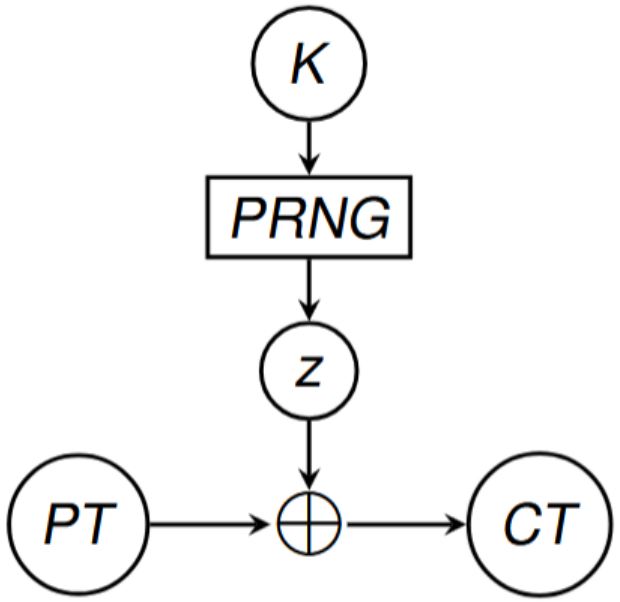


m: plaintext message **k:** encryption/decryption key **c:** ciphertext message



- The same key **k** is used both for encryption *and* decryption [5]

PRIMITIVES IN SYMMETRIC CRYPTOGRAPHY

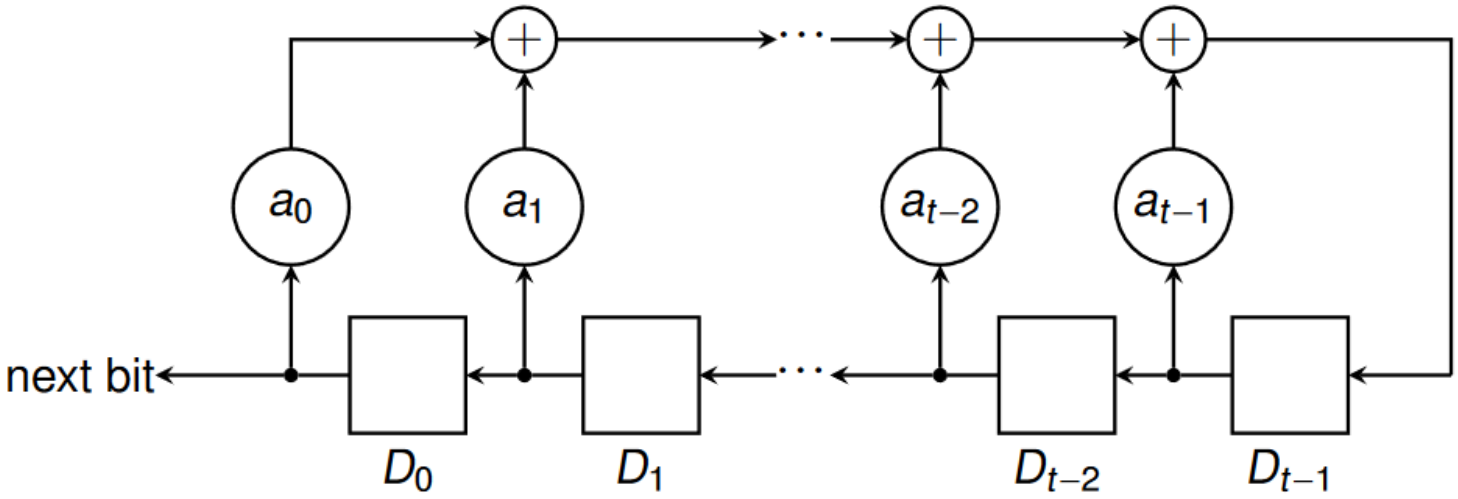


(a) Stream cipher

- Symmetric ciphers require several low-level **primitives**:
  - Pseudorandom number generators (PRNG)
  - Boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$
  - S-boxes $F : \{0,1\}^n \rightarrow \{0,1\}^n$
- Classic methods to build them: **algebraic constructions** [2]
- Alternative: **(meta)heuristics** (GA, GP, ...) [3, 9]

HOW TO BUILD PRG – LFSR

- **Linear Feedback shift Register (LFSR):** device computing a *linear recurring sequence* (LRS) $z_i = a_0 \cdot z_{i-t} \oplus a_1 \cdot z_{i-t+1} \oplus \cdots \oplus a_{t-1} \cdot z_{i-1} \qquad a_j, z_j \in \{0,1\}$



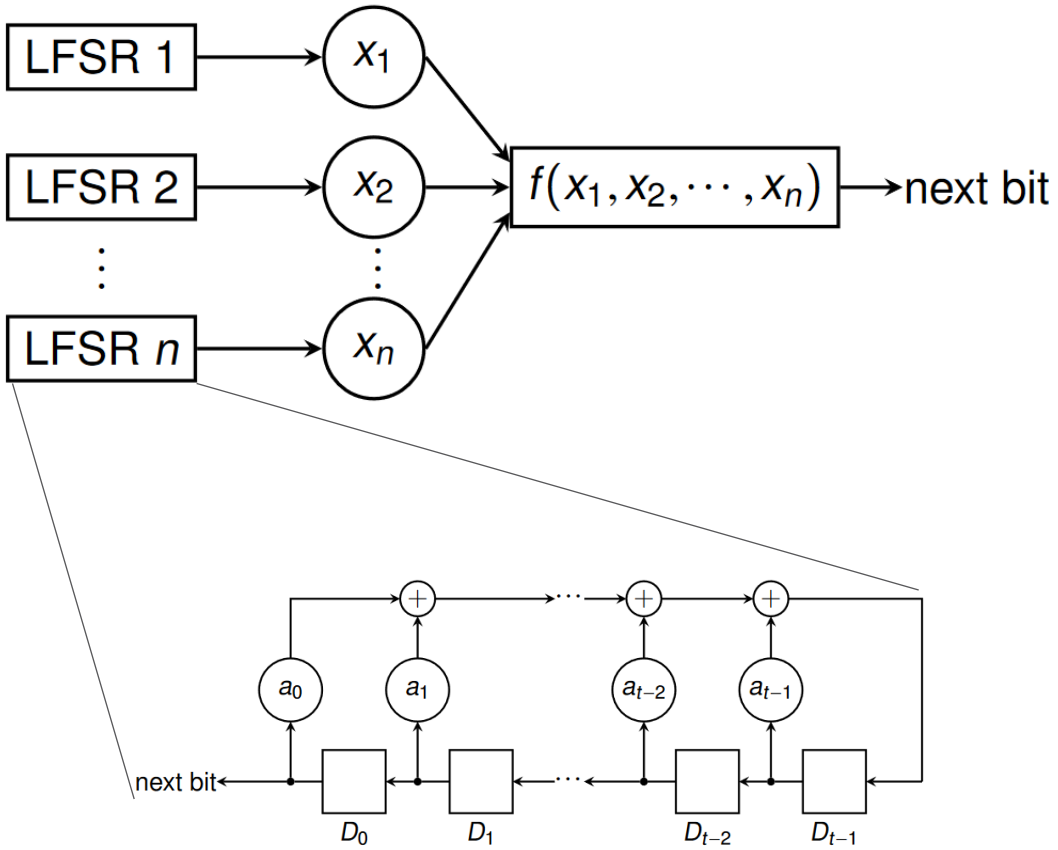
- **Problem:** very weak as a cryptographic PRG [5]

IMPROVING LFSR – COMBINER MODEL FOR PRG

- **Idea:** use n LFSRs instead of one
- LFSRs outputs *combined* using a Boolean function:

$$f : \{0,1\}^n \rightarrow \{0,1\}$$

- Security of the PRG: **cryptographic properties of** $f : \{0,1\}^n \rightarrow \{0,1\}$ [2]



BOOLEAN FUNCTIONS

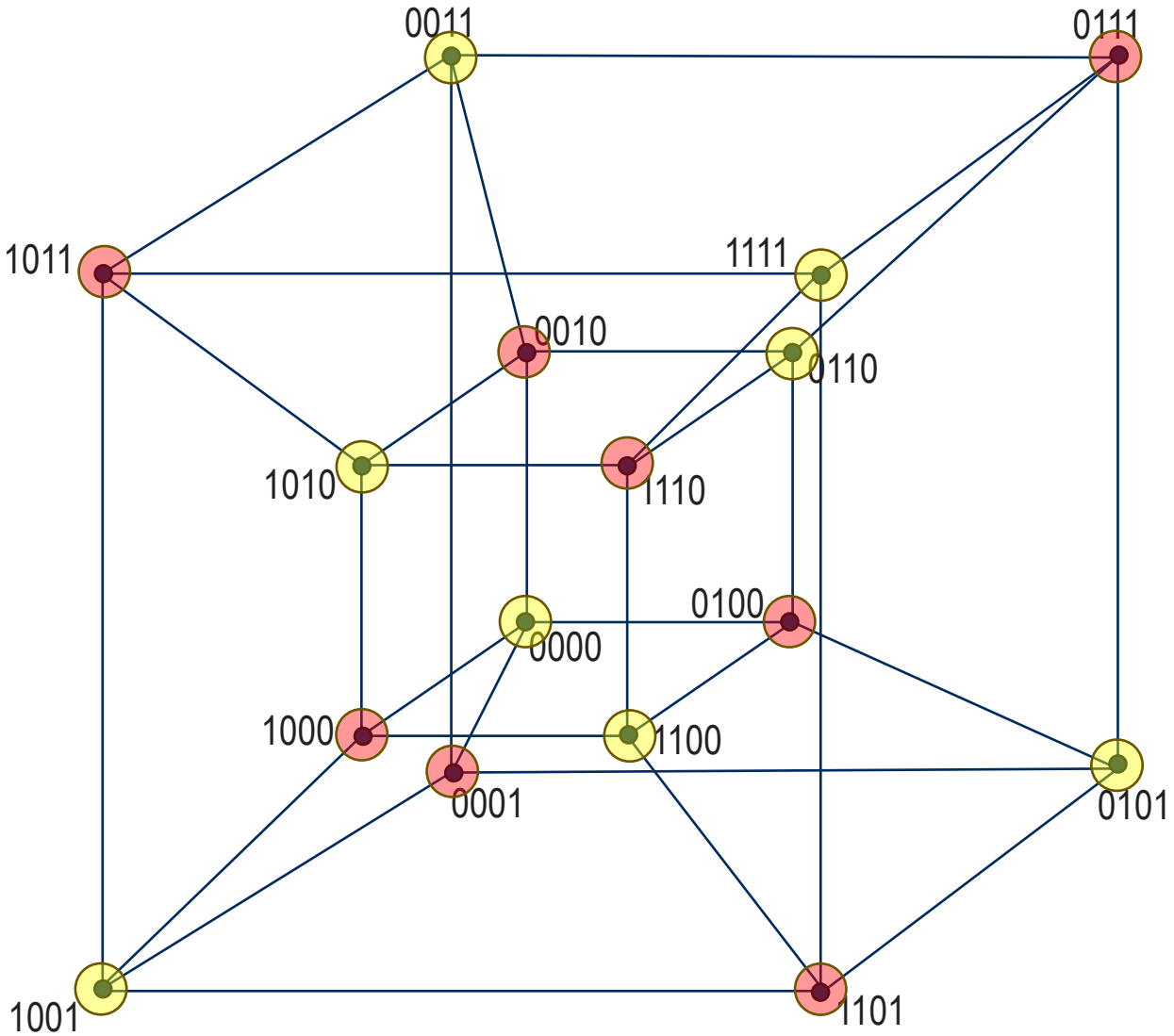
- A mapping $f : \{0,1\}^n \to \{0,1\}$ represented by a *truth table*

(x_1, x_2, x_3)	000	001	010	011	100	101	110	111
$f(x_1, x_2, x_3)$	0	1	1	0	1	0	1	0

- The function must satisfy some properties to resist specific attacks [2]:
  - **Balancedness**: equal number of 0s and 1s
  - High **Nonlinearity**: high Hamming distance from **linear** functions:

$$L_a(x) = a \cdot x = a_1 x_1 \oplus \ldots \oplus a_n x_n, \quad a_i, x_i \in \{0,1\}$$

NONLINEARITY – GEOMETRIC INTUITION



- **Example**: $n = 2$ variables (truth tables of $2^2 = 4$ bits, 16 functions in total)
- Linear functions and complements:

$a_1x_1 \oplus a_2x_2$	TT	TT $\oplus 1$
0	0000	1111
x_1	0011	1100
x_2	0101	1010
$x_1 \oplus x_2$	0110	1001

- Hamming distance from linear functions: **1**, e.g.: $f(x_1, x_2) = x_1x_2$

OPTIMIZATION PROBLEM

- Given $n \in \mathbb{N}$ **odd**, how do we fill the table so that f is highly nonlinear?

(x_1, x_2, x_3)	000	001	010	011	100	101	110	111
$f(x_1, x_2, x_3)$	?	?	?	?	?	?	?	?

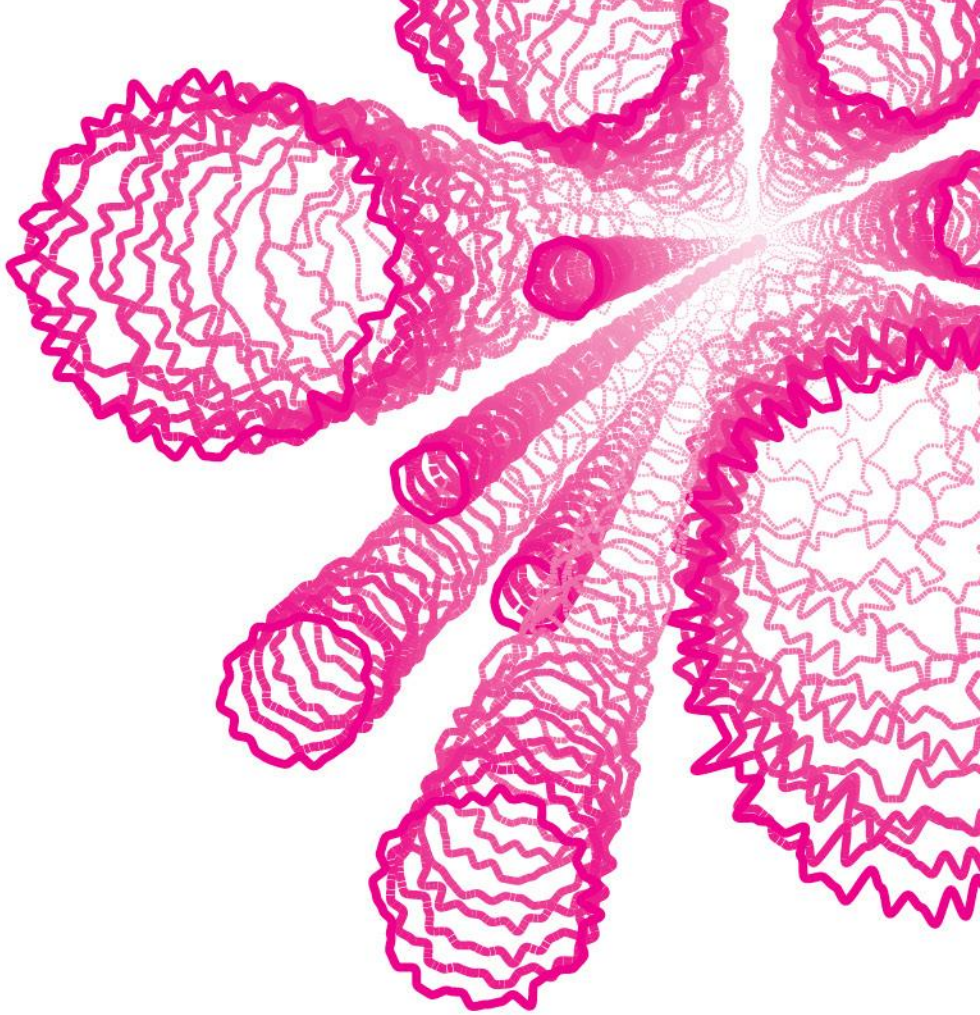
$$f^* = \text{argmax}_{f:\{0,1\}^n \to \{0,1\}}(nl_f), \qquad nl_f \leq 2^{n-1} - 2^{\frac{n}{2}-1}$$

- The truth table has size 2^n so there are 2^{2^n} combinations

n	3	4	5	6	7	8
2^{2^n}	256	65536	$4.3 \cdot 10^9$	$1.8 \cdot 10^{19}$	$3.4 \cdot 10^{38}$	$1.2 \cdot 10^{77}$

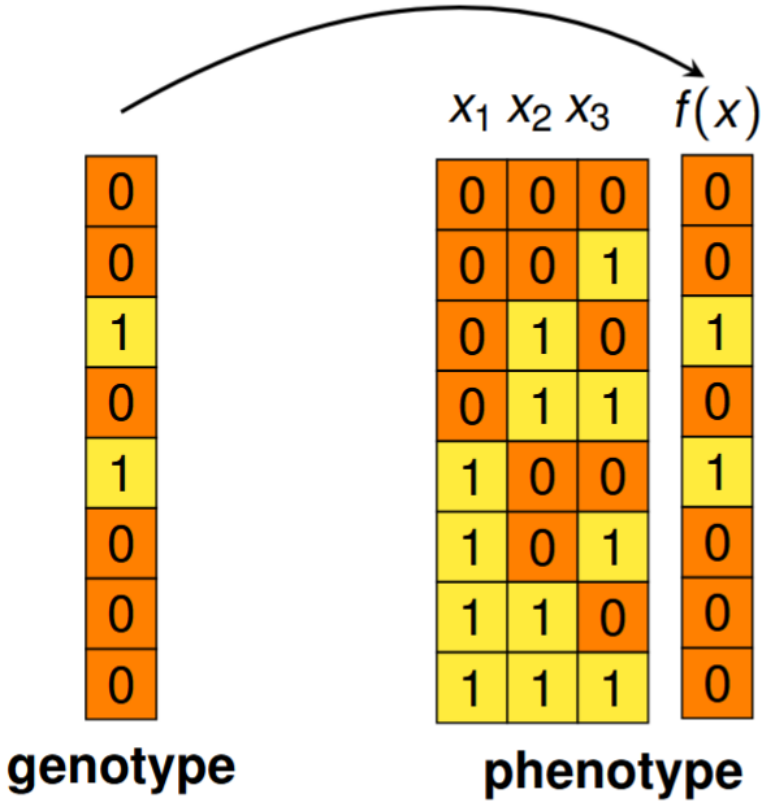
- Exhaustive search unfeasible for $n > 5$, optimum unknown for $n > 7$ odd [2]

EVOLUTIONARY ALGORITHMS SETUP



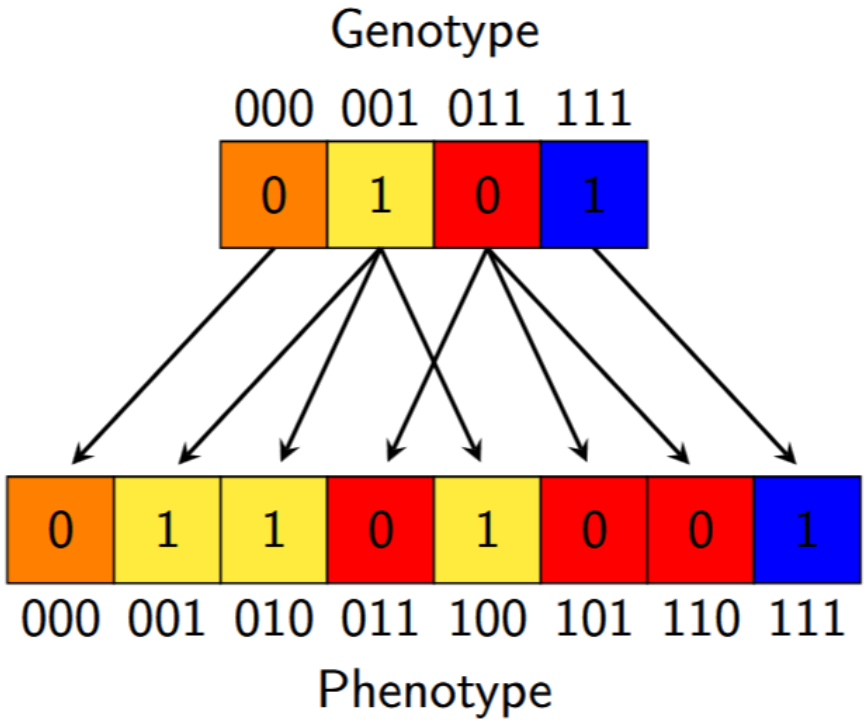
UNIVERSITY OF TWENTE.

BITSTRING ENCODING FOR GENETIC ALGORITHMS



- **GA genotype**: fixed-length bitstrings [8, 11]
- **phenotype**: truth table of f
- **Assumption:** the input vectors of $\{0,1\}^n$ are sorted lexicographically
- Usual variation operators of GA (one-point crossover, bit-flip mutation) [10, 12]

ROTATION SYMMETRIC ENCODING (RSBF)



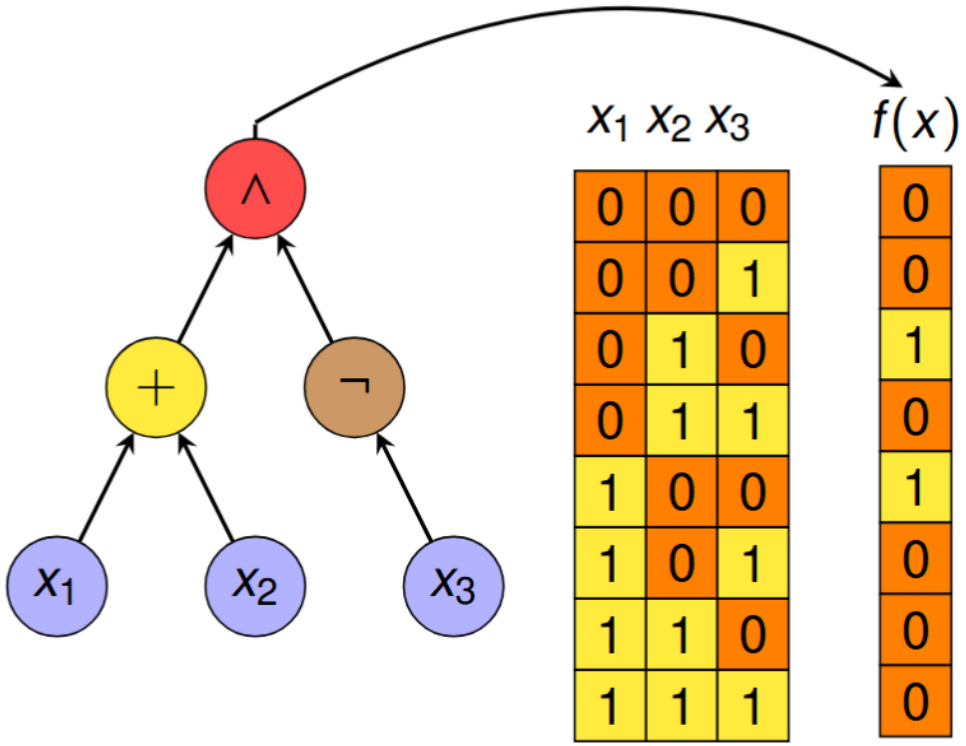
- **Rotation-symmetric BF (RSBF):** the output of f is invariant under *cyclic shifts* of the input [6]
- Greatly reduce the search space: 2^{g_n}, where

$$g_n = \frac{1}{n} \sum_{t|n} \phi(t) 2^{\frac{n}{t}}$$

- GA genotype: bitstring of length g_n

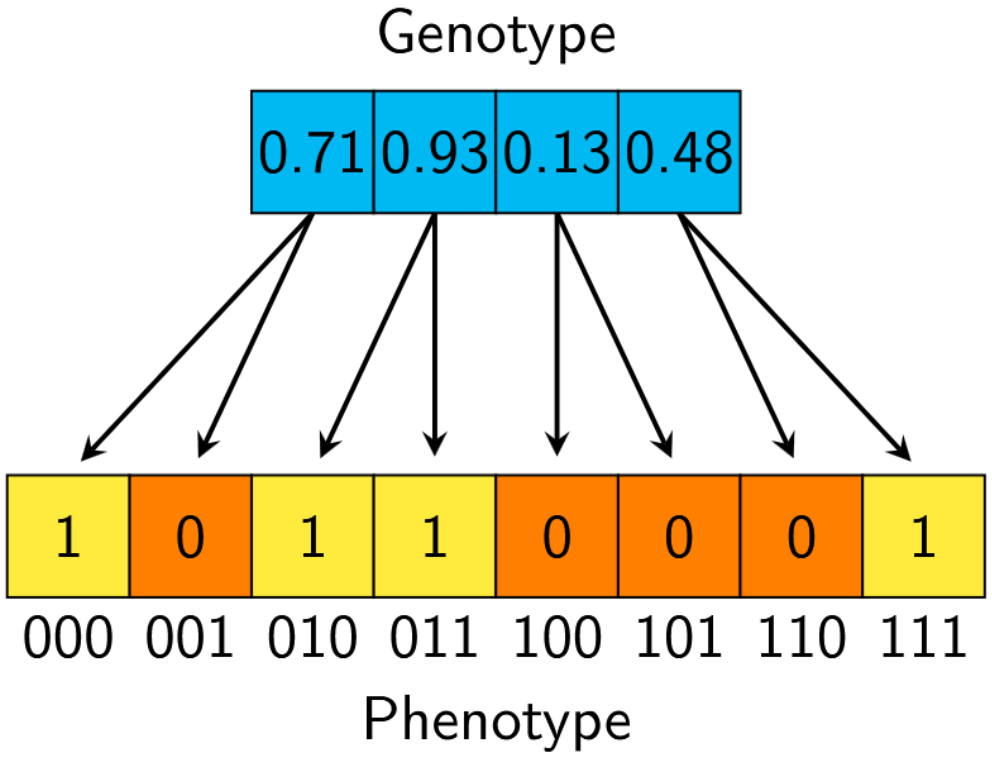
variables	7	9	11	13
g_n	20	60	188	632

SYMBOLIC ENCODING WITH GP



- **GP Genotype**: a syntactic tree
  - Leaf nodes: input variables
  - Internal nodes: operators (e.g. AND, OR, NOT, XOR, …)
- **Phenotype:** evaluate the tree for all possible assignments of the leaf nodes
- Classic GP operators (subtree crossover and mutation, …) [7, 12]

FLOATING POINT ENCODING



- **Genotype**: vector of floating point values
- **Phenotype:**
  1. decode each FP value into an integer
  2. Map the integer to a substring of bits
- Can be used with any continuous search optimization algorithms [1]

FITNESS FUNCTION

- Nonlinearity computation: through the **Walsh-Hadamard Transform** [3]:

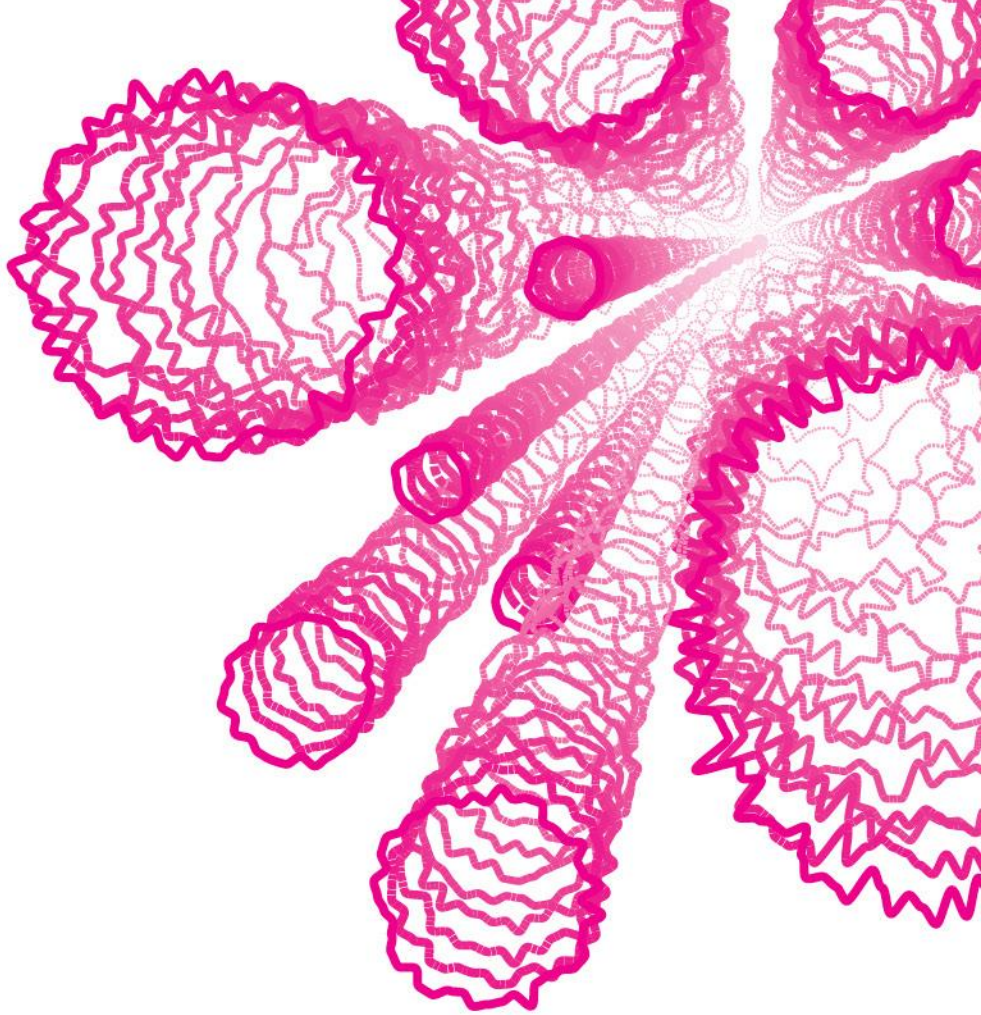
$$W_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus a \cdot x}, \qquad nl_f = 2^{n-1} - \tfrac{1}{2} \max_{a \in \mathbb{F}_2^n} |W_f(a)|$$

- **Advantage:** complexity $\mathcal{O}(n2^n)$ steps instead of $\mathcal{O}(2^{2n})$
- **Fitness Idea:** do not simply maximize nonlinearity

$$fitness = nl_f + \frac{2^n - \#max\_values}{2^n}.$$

- **Rationale:** optimize a "smoother" fitness

EXPERIMENTAL RESULTS



EXPERIMENTAL SETTING

- List of search algorithms considered in the evaluation:

Encoding	Description	Algorithm	Description
FP	Floating-point	ABC	Artificial Bee Colony
		CLONALG	Clonal Selection Algorithm
		CMAES	CMA-ES
		DE	Differential Evolution
		OPTIA	Immune Optimization Algorithm
		SST	Steady-State Tournament GA
GP	Symbolic	SST	Steady-State Tournament GP
TT	Bitstring	SST	Steady-State Tournament GA

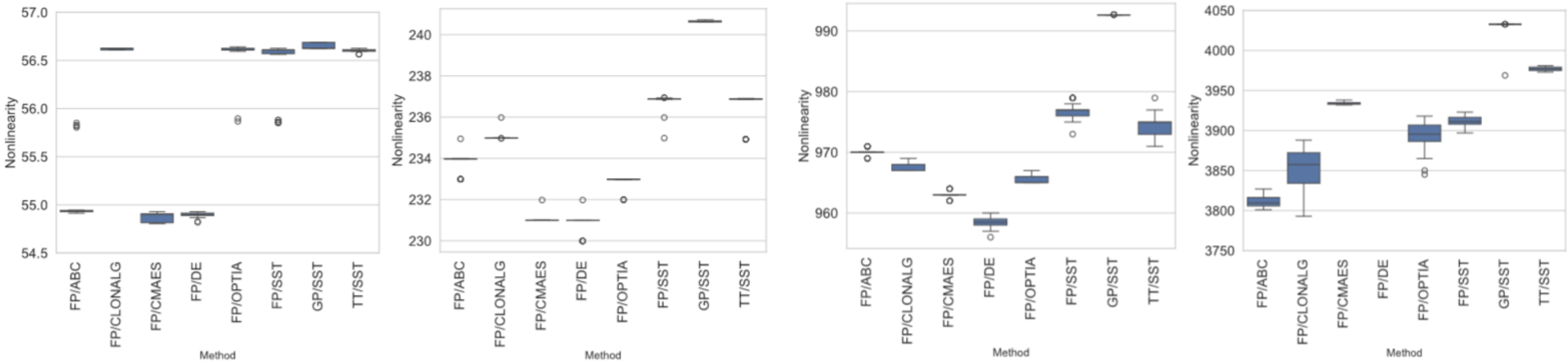
- Fitness budget: 10^6 evaluations, problem sizes from $n = 7$ to $n = 13$, 30 independent runs per experiment

RESULTS – SUMMARY STATISTICS

Enc.	Algorithm	7			9			11			13		
		max.	avg.	std.	max.	avg.	std.	max.	avg.	std.	max.	avg.	std.
FP	ABC	55.85	55.05	0.31	234.95	233.84	0.42	971.00	970.09	0.61	3827.00	3810.97	6.71
	CLONALG	56.63	56.62	0.01	235.98	235.01	0.18	969.00	967.76	0.57	3888.00	3853.40	24.37
	CMAES	54.93	54.65	0.50	231.98	231.02	0.18	964.00	963.00	0.52	3938.00	3934.23	1.41
	DE	54.93	54.90	0.03	231.98	230.79	0.48	960.00	958.50	1.01	2836.00	2701.12	58.23
	OPTIA	56.64	56.57	0.19	232.99	232.85	0.34	967.00	965.43	0.57	3918.00	3894.17	18.22
	SST	56.63	56.46	0.30	236.95	236.80	0.38	978.97	976.78	1.42	3923.00	3911.70	7.20
GP	SST	56.69	**56.64**	0.03	240.72	**240.64**	0.03	992.69	**992.63**	0.02	4032.69	**4030.52**	11.62
TT	SST	56.63	56.60	0.02	236.91	236.55	0.74	978.96	974.44	1.88	3980.99	3977.22	2.51

- **Main finding**: GP best search method on average
- Optimum reached by all methods only for size 7

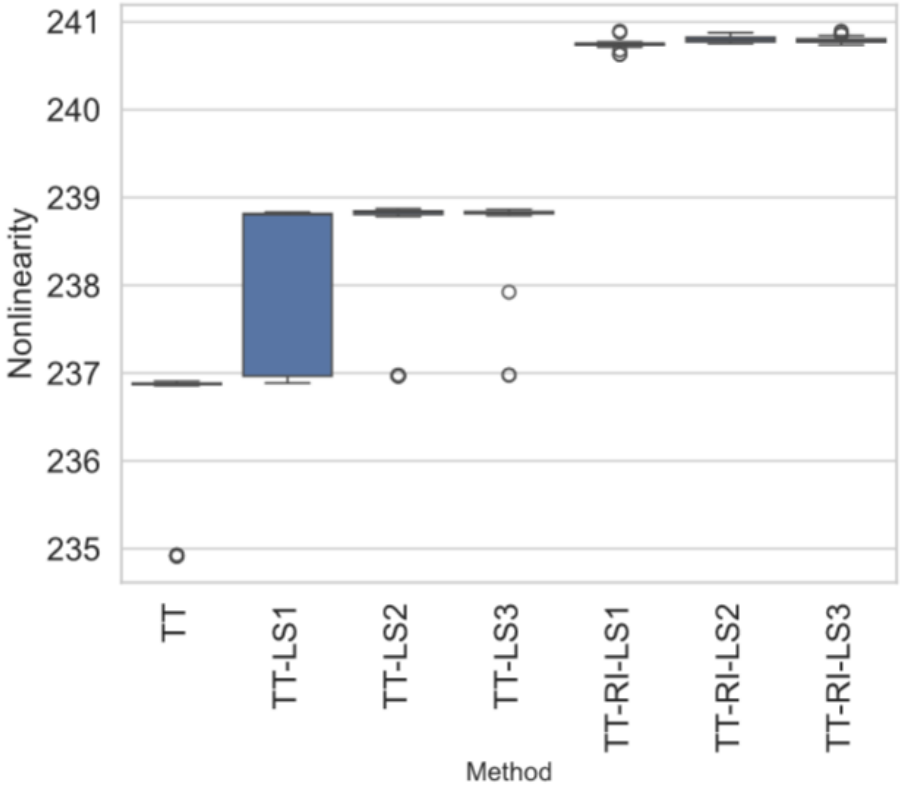
RESULTS – BEST FITNESS DISTRIBUTIONS



(a) Size 7 (b) Size 9 (c) Size 11 (d) Size 13

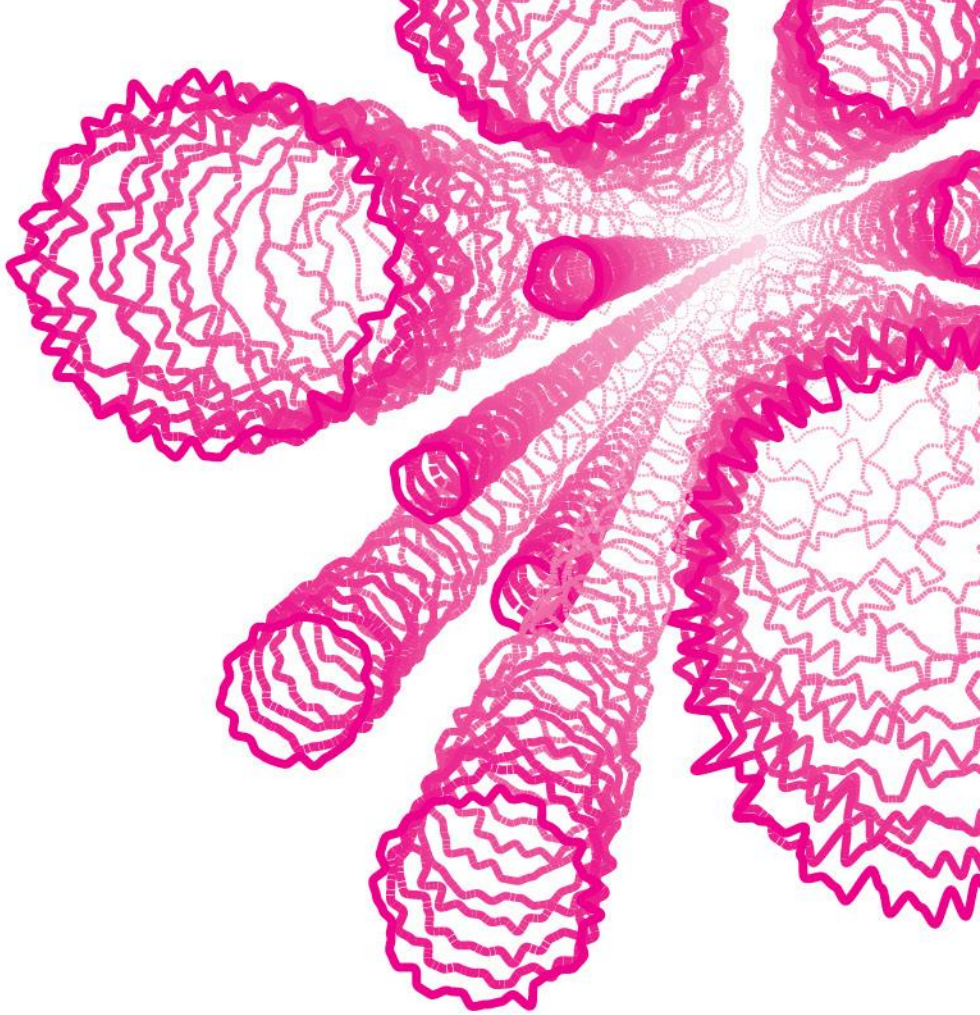
- **Main finding**: GP performs consistently better than other methods, but still falls short of ad-hoc heuristics in the literature [6]

RESULTS ON LOCAL SEARCH AND RSBF



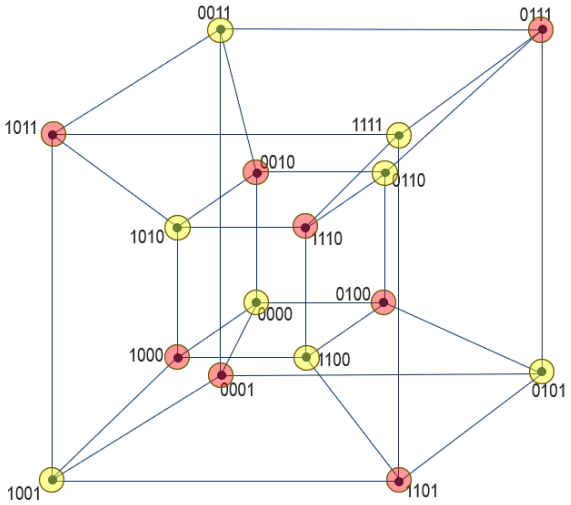
- Experimented with two improvements:
  1. Adding a **Local Search** step (LS) to the EA
  2. Restricting the space to RSBF with the bitstring encoding
- **Main result for 9 variables:** GA managed to find a function of nonlinearity 241
- This is the first time EA reach this result, matching ad-hoc heuristics [6]

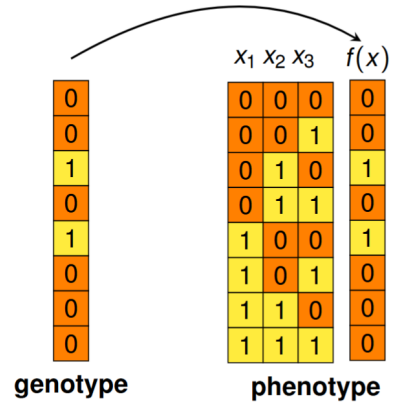
CONCLUSIONS AND INSIGHTS

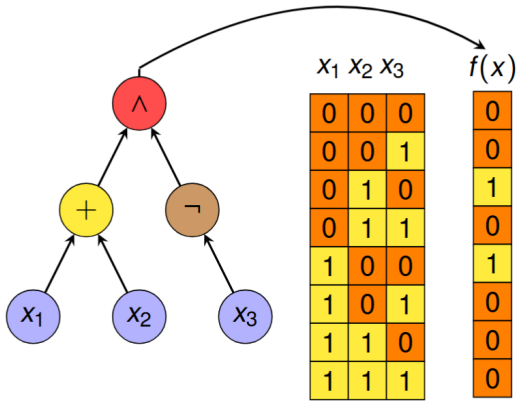


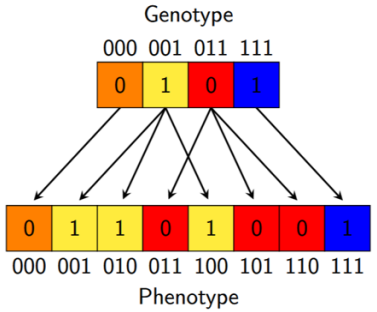
SUMMARY

- **Insights:**
  - GP is the best performing heuristic to get consistently good results, but…
  - LS is crucial to reach high nonlinearity, and GP does not cope well with LS
- **Future directions:**
  - Restrict the space of GP to RSBF
  - Explore other encodings, perform fitness landscape analysis of the problem [4]









UNIVERSITY OF TWENTE.

REFERENCES

1. C. Carlet, M. Durasevic, B. Gasperov, D. Jakobovic, L. Mariot, S. Picek: A new angle: On evolving rotation symmetric Boolean functions. In: Proc. of EvoApps 2024, pp. 287–302 (2024)
2. C. Carlet: Boolean functions for cryptography and coding theory. Cambridge University Press (2021)
3. M. Djurasevic, D. Jakobovic, L. Mariot, S. Picek: A survey of metaheuristic algorithms for the design of cryptographic Boolean functions. Cryptogr. Commun. 15(6): 1171-1197 (2023)
4. D. Jakobovic, S. Picek, M.S.R. Martins, M. Wagner: Toward more efficient heuristic construction of Boolean functions. Appl. Soft Comput. 107:107327 (2021)
5. J. Katz, Y. Lindell: Introduction to Modern Cryptography. Third Edition, CRC Press (2021)
6. S. Kavut, S. Maitra, M. D. Yucel: Search for Boolean functions with excellent profiles in the rotation symmetric class. IEEE Trans. Inf. Theory 53(5):1743–1751 (2007)
7. J.R. Koza: Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems (Vol. 34). Stanford, CA: Stanford University, Department of Computer Science (1990)
8. L. Manzoni, L. Mariot, E. Tuba: Balanced crossover operators in Genetic Algorithms. Swarm Evol. Comput. 54: 100646 (2020)
9. L. Mariot, D. Jakobovic, T. Bäck, J.C. Hernandez-Castro: Artificial Intelligence for the Design of Symmetric Cryptographic Primitives. Security and Artificial Intelligence, pp. 3-24 (2022)
10. L. Mariot, D. Jakobovic, A. Leporati, S. Picek: Hyper-bent Boolean Functions and Evolutionary Algorithms. Proc. of EuroGP 2019, pp. 262-277 (2019)
11. W. Millan, J. Clark, E. Dawson: Heuristic Design of Cryptographically Strong Balanced Boolean Functions. Proc. of EUROCRYPT 1998, pp. 489–499 (1998)
12. S. Picek, D. Jakobovic, J. F. Miller, L. Batina, M. Cupic: Cryptographic Boolean functions: One output, many design criteria. Appl. Soft Comput. 40: 635-653 (2016)