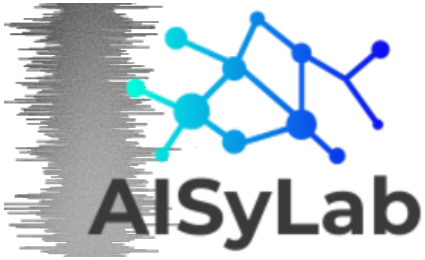


Artificial Intelligence and Security Lab
Cyber Security Research Group
Delft University of Technology

TU Delft

Investigating Reversible Cellular Automata with Evolutionary Algorithms

Luca Mariot

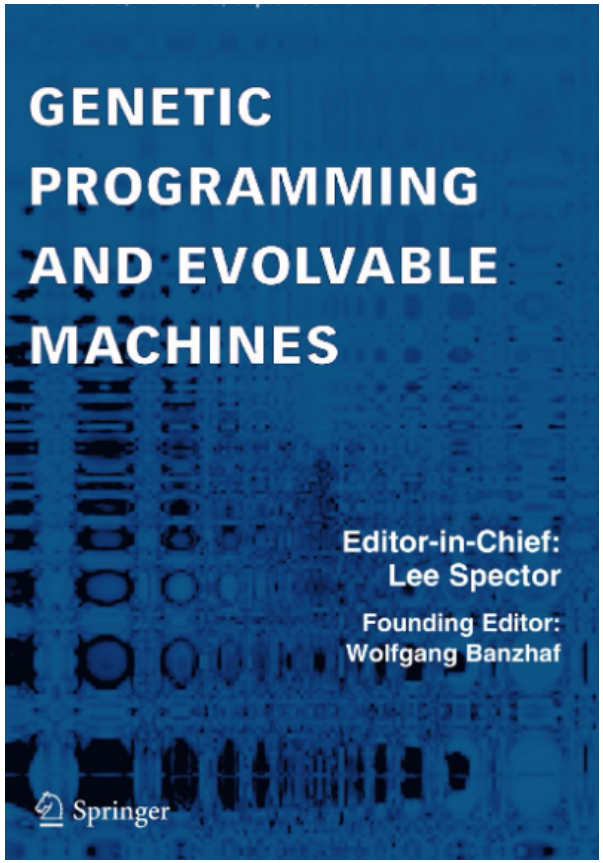
L.Mariot@tudelft.nl

Trieste – December 15, 2021

Main reference

This talk is based on the paper [MPJL21]:

L. Mariot, S. Picek, D. Jakobovic, A. Leporati: *Evolutionary algorithms for designing reversible cellular automata*. Genet. Program. Evolvable Mach. 22(4): 429-461 (2021)



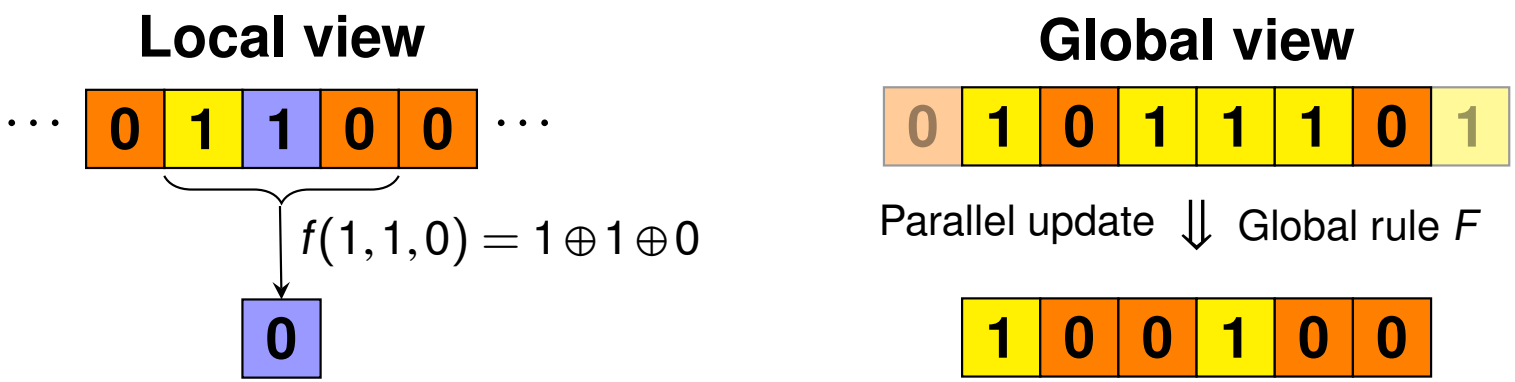
This is the extended version of [MPJL20] presented at EuroGP'20

Cellular Automata (CA)

Definition (Periodic Boolean Cellular Automata – CA)

A finite binary array of n cells, where each cell x_i updates its state by applying a *local rule* $f : \{0,1\}^d \to \{0,1\}$ to the *neighborhood* $\{x_{i-\omega}, \cdots, x_i, \cdots, x_{i-\omega+d-1}\}$ with *periodic boundary conditions*

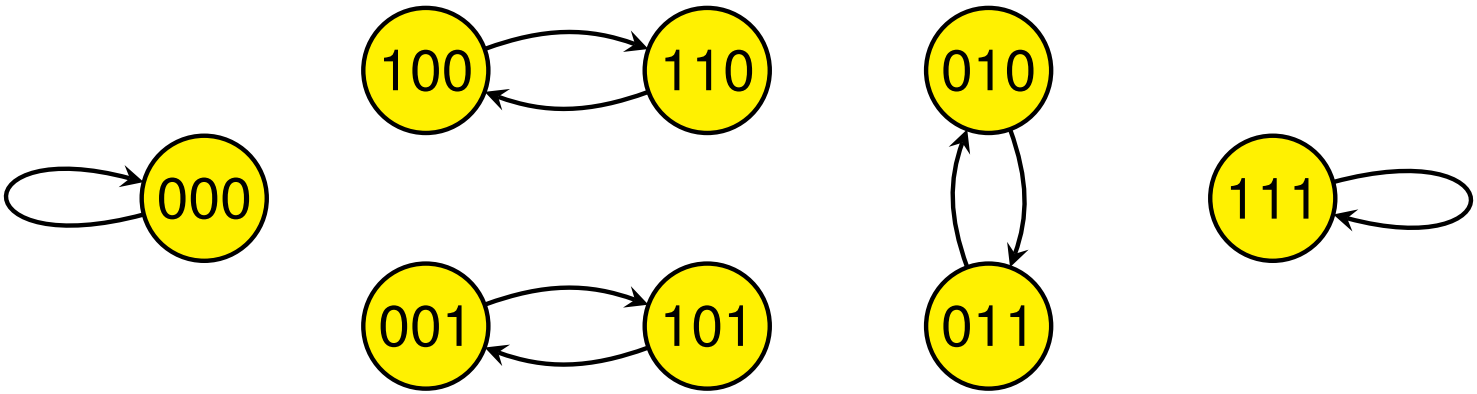
Example: $n = 6$, $d = 3$, $\omega = 1$, $f(x_{i-1}, x_i, x_{i+1}) = x_{i-1} \oplus x_i \oplus x_{i+1}$



Reversible CA

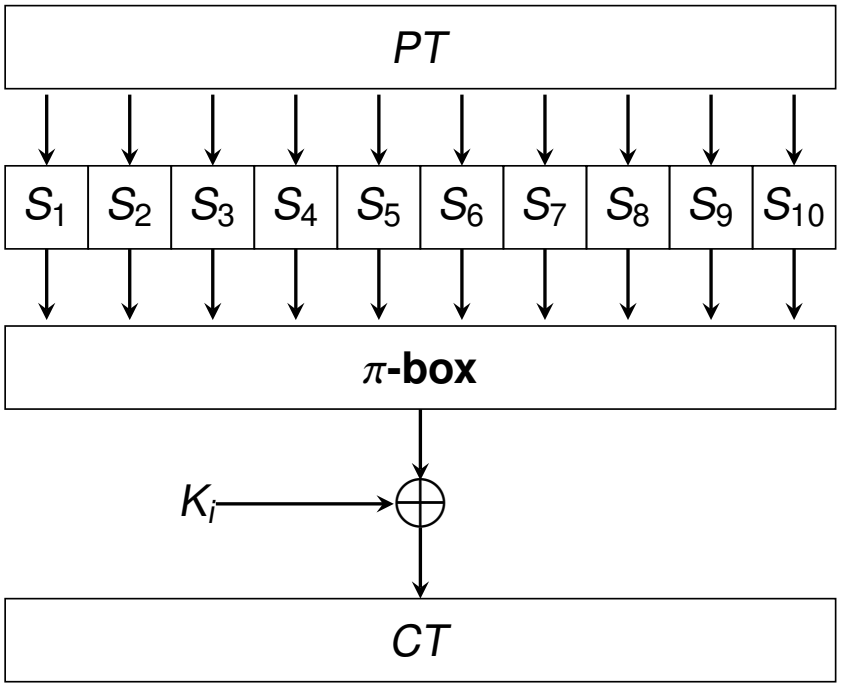
- *reversible CA* (RCA): bijective global rule $F : \{0,1\}^n \to \{0,1\}^n$ and the inverse map F^{-1} is also a CA [H69]
- Interesting for applications in *reversible computing* and *cryptography* [MPLJ19, D95]

Example: $n = 3$, $d = 3$, $\omega = 0$, $f(x_i, x_{i+1}, x_{i+2}) = x_i \oplus x_{i+1} \cdot x_{i+2} \oplus x_{i+2}$



- Local rules resulting in RCA for every size n of the array are also called *locally invertible* [D95]

RCA in Block Ciphers: Substitution-Permutation Network



- $S_i : \{0,1\}^n \rightarrow \{0,1\}^n$ are S-boxes, or vectorial Boolean functions
- The S-boxes must:
  - be bijective
  - have high nonlinearity
  - have low differential uniformity

- *Genetic Programming* used to design CA-based S-boxes, with $n = d$ and $4 \leq n \leq 8$ [MPLJ19, PMYJM17, PMLJ17]
- Other uses of CA in crypto include *pseudo-random number generators* [LM14]

Marker CA

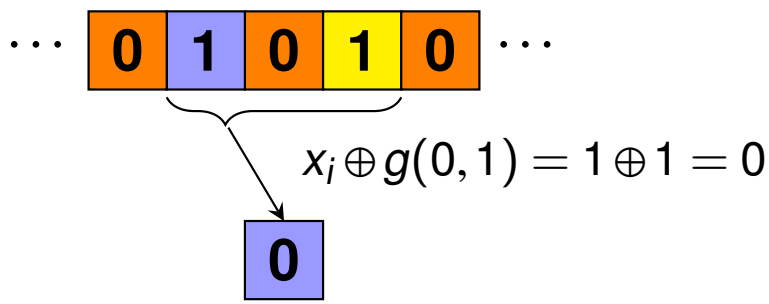
- The local rule f of marker CA is defined as follows:

$$f(x_{i-\omega}\cdots x_{i-1}x_i x_{i+1}\cdots x_{i-\omega+d-1}) = x_i \oplus g(x_{i-\omega}\cdots x_{i-1}x_{i+1}\cdots x_{i-\omega+d-1})$$

- Equivalently: the *support* of g defines the *markers* for which the central cell *flips* its state

Example: $d = 3$, $\omega = 0$, $f(x_i, x_{i+1}, x_{i+2}) = x_i \oplus x_{i+1} \cdot x_{i+2} \oplus x_{i+2}$

x_{i+1}	x_{i+2}	$g(x_{i+1}, x_{i+2})$
0	0	0
1	0	0
0	**1**	**1**
1	1	0

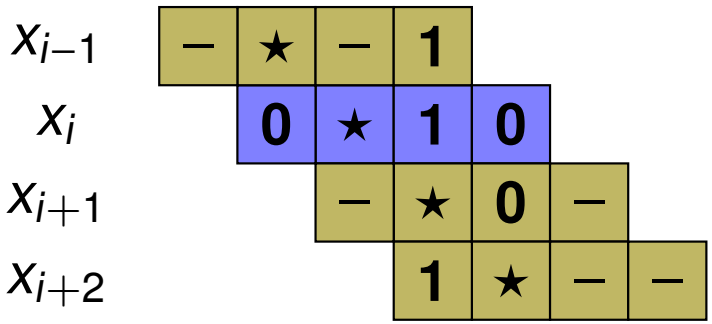


Marker: $01 \Rightarrow \star 01$ **Flipping landscape**

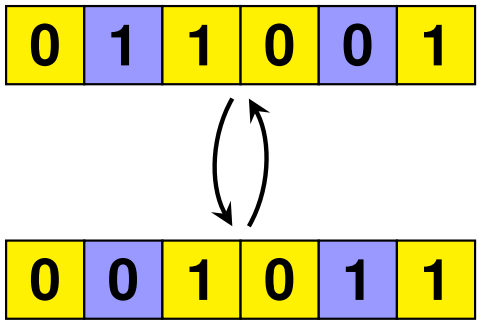
Conserved Landscape Marker CA

- *Conserved Landscape*: each cell in a flipping landscape must be in the *same* landscape after applying the CA global rule

Example: $d = 4$, $\omega = 1$, Landscape: $0 \star 10$



Landscape tabulation



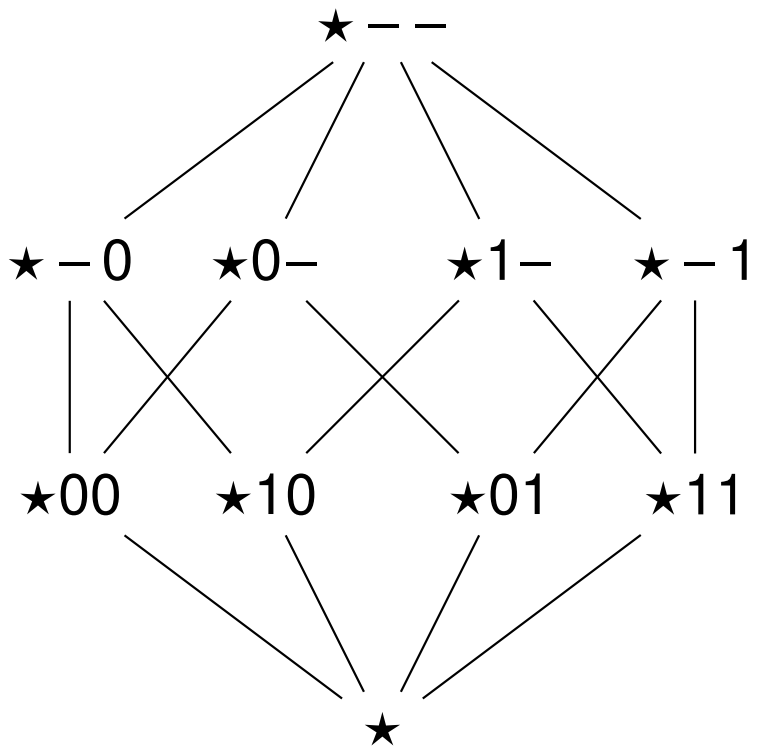
Example of orbit of period 2

- A landscape is conserved if it is *incompatible* with all its *neighborhood landscapes* [TM90]
- **Question:** How to turn the search of conserved landscape marker CA into an optimization problem?

The Poset of Landscape

Compatible landscapes induce a *partial order* $\leq_C$:

$$L \leq_C M \Leftrightarrow l_i = m_i \text{ or } l_i \in \{0,1\} \text{ and } m_i = -, \;\; 0 \leq i \leq d-1$$

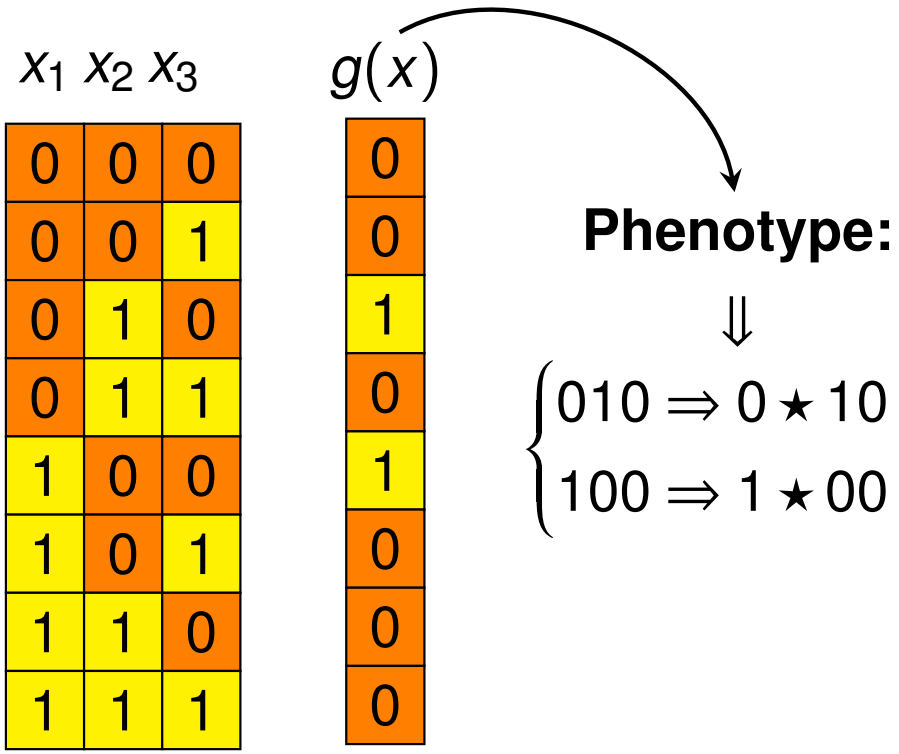


- A conserved landscape rule is an *antichain* on this poset
- Counting antichains in general finite posets is $\#\mathcal{P}$-complete! [PB83]
- Hence, we don't know the number of optimal solutions for our problem

Genotype Encoding – GA

- *Phenotype*: the set of markers in the generating function g
- *GA Genotype*: Bitstring $g(x)$ corresponding to the output column of the *truth table* of g

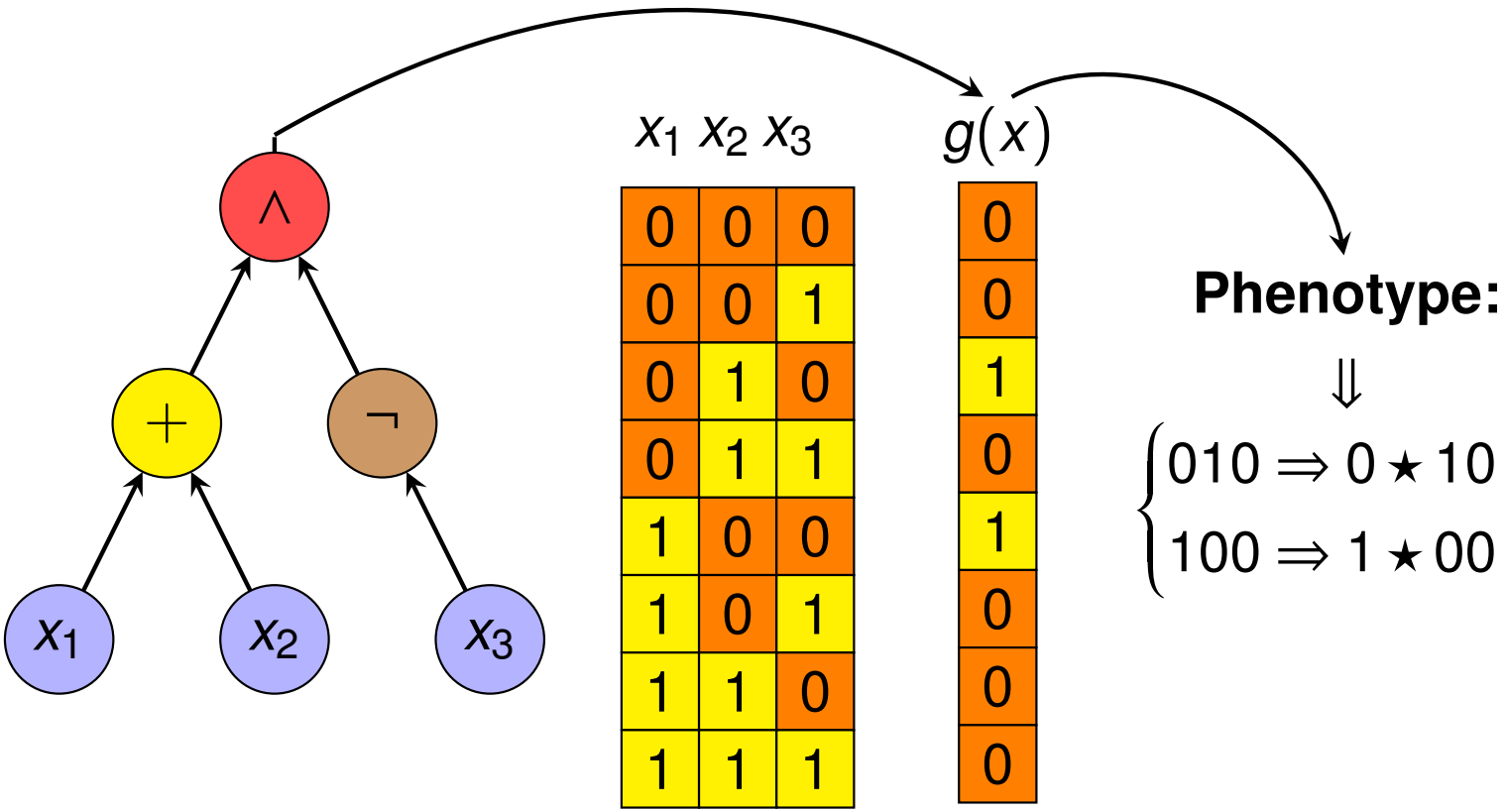
Example: $d = 4$, $\omega = 1$, $g : \{0,1\}^3 \to \{0,1\}$



Genotype Encoding – GP

- *GP Genotype*: Boolean tree
- The truth table $g(x)$ is synthesized from the tree [MPJL18]

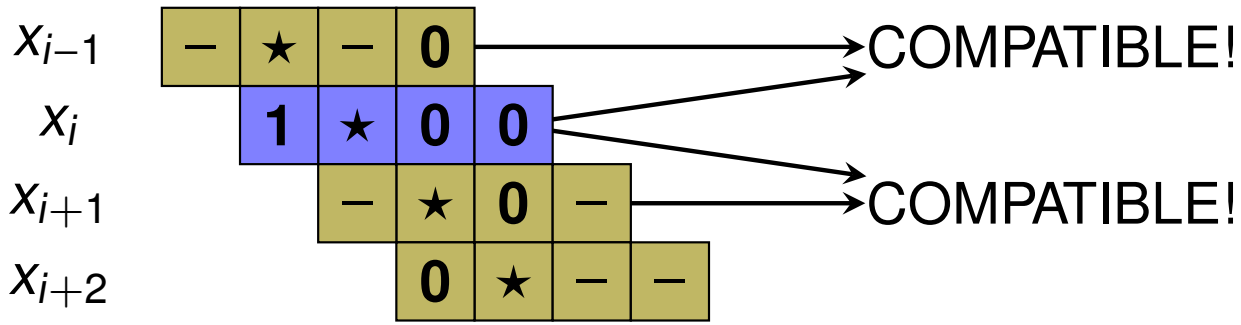
Example: $d = 4$, $\omega = 1$, $g : \{0,1\}^3 \to \{0,1\}$



First Fitness Function

- **Objective:** *minimize* the number of neighborhood landscapes that are compatible with each landscape in g

Example: $d = 4$, $\omega = 1$, Landscape: $1 \star 00$



- **Fitness function:** Loop over all landscapes in the support of g and count the compatible neighborhood landscapes

$$fit_1(g) = \sum_{i,t \in [k], j \in [d-1]_\omega} comp(M_{i,j}, L_t)$$

Second Fitness Function

- **Objective:** *maximize* the *Hamming weight* of g
- This criterion is relevant in cryptography: the higher the Hamming weight of g, the higher the nonlinearity of the CA

Example: $d = 4$, $\omega = 1$, $g : \{0,1\}^3 \rightarrow \{0,1\}$

$g(x) =$ | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

$\Downarrow$

Hamming weight: 2

- **Fitness function:** Count the number of 1s in $g(x)$

$$fit_2(g) = |supp(g(x))|$$

Preliminary Exhaustive Search up to $d = 6$

- No. of generating functions of $d-1$ variables: $\#\mathcal{P}(d) = 2^{2^{d-1}}$

d	2^{d-1}	$\#\mathcal{P}(d)$	ω	#REV	Weights	Time
4	8	256	0	0	–	~0.1s
			1	1	1	
			2	1	1	
			3	0	–	
5	16	65 536	0	0	–	~120s
			1	2	1	
			2	5	1,2	
			3	2	1	
			4	0	–	
6	32	$4.3 \cdot 10^9$	0	0	–	$\sim 2.3 \cdot 10^5$s
			1	8	1,2	
			2	23	1,2,3	
			3	23	1,2,3	
			4	8	1,2	
			5	0	–	

- The number of conserved landscape rules is *really small* wrt the number of generating functions
- The possible Hamming weights are *really low* wrt to the length

Research Questions

- **RQ1:** Given the limited number conserved landscape rules, is it difficult for GA and GP to find them?
- **RQ2:** Do there exist conserved landscapes rules of a larger diameter and with higher Hamming weight?
- **RQ3:** Is there a trade-off between the reversibility of a marker CA rule and its Hamming weight?

Experimental settings

Common Parameters:

- Problem instances: diameters $7 \leq d \leq 53$, $\omega = 3$
- Termination condition: 500 000 fitness evaluations
- Each experiment is repeated over 50 independent runs
- Selection operator: steady-state with 3-tournament operator

GA Parameters:

- Population size: 500 individuals
- Mutation probability: $p_m = 0.9$

GP Parameters:

- Boolean operators: AND, OR, NOT, AND(x_1, NOT(x_2))
- Population size: 500 individuals
- Max tree depth: $d - 1$

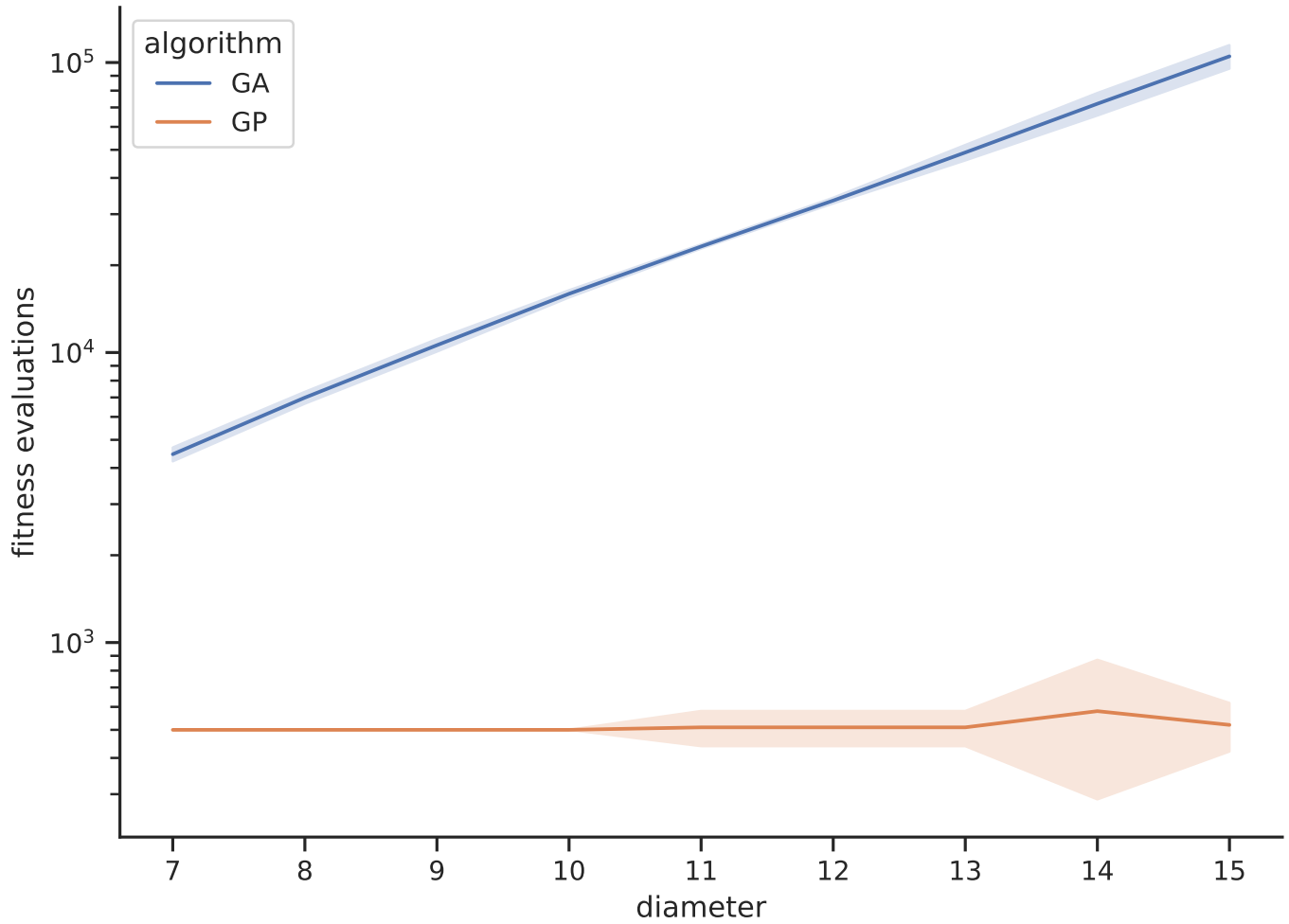
Optimization Approaches

Optimization approaches to investigate the research questions:

- *Single-objective Optimization* only of the reversibility property with GA and GP, by minimizing fit_1
- *Multi-objective Optimization* with NSGA-II, by minimizing fit_1 and maximizing the Hamming weight fit_2
- *Lexicographic Optimization* with GA GP, by first minimizing fit_1 and then maximizing fit_2 while retaining reversibility

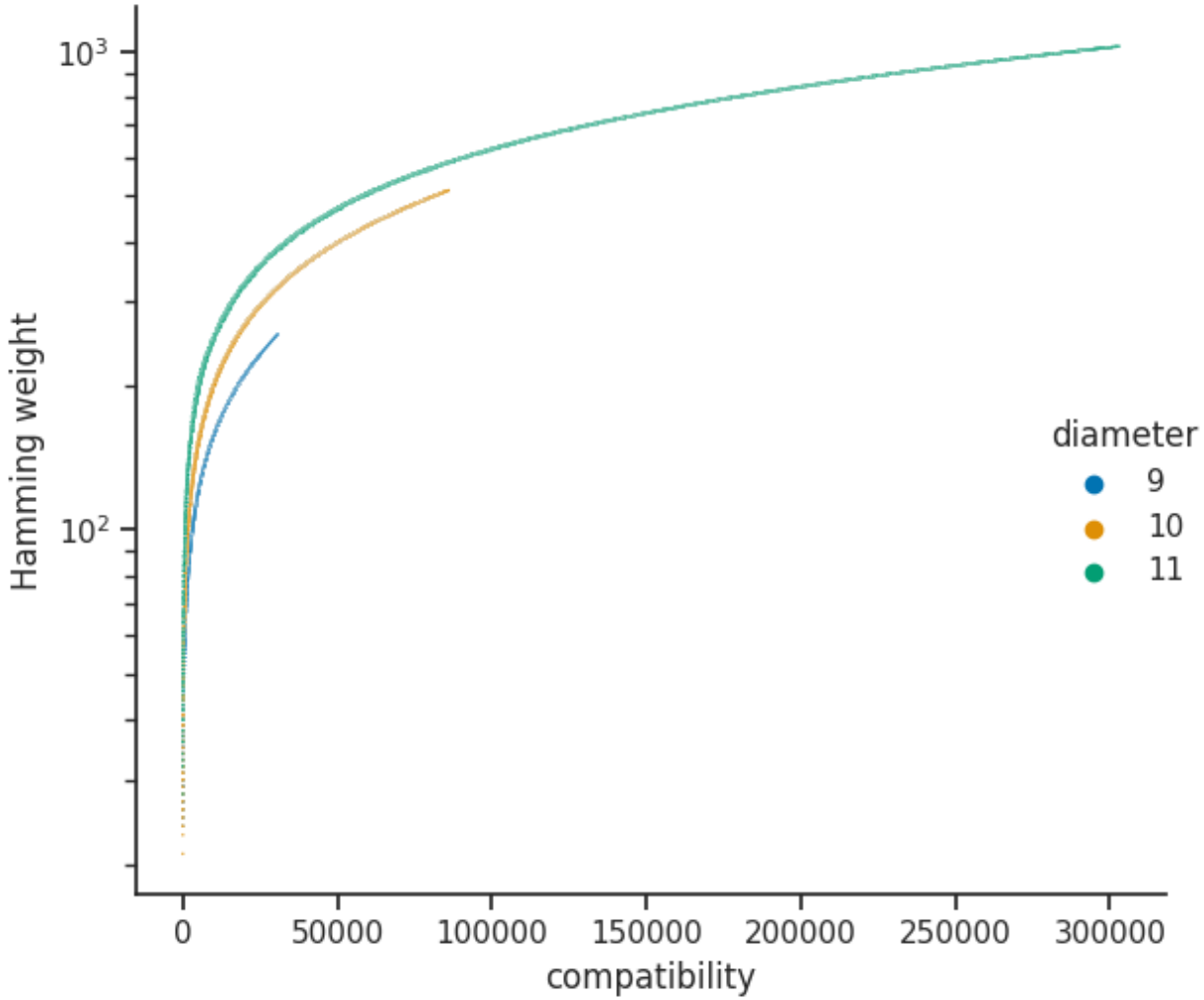
Single-Objective GA and GP

- **Main finding**: *both GA and GP converge to an optimal solution over all experimental runs, but GP is much faster*



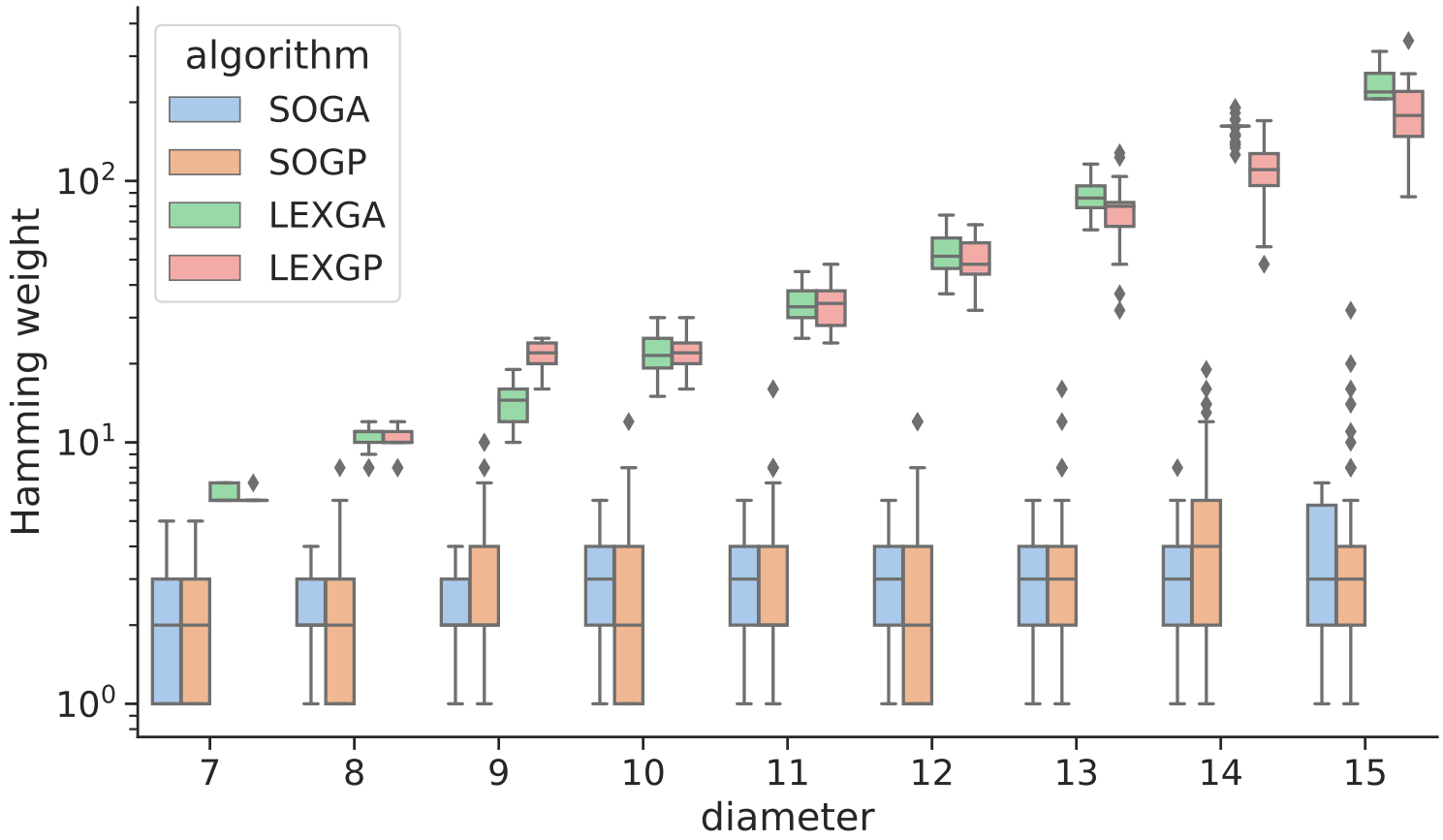
Multi-Objective Approach with NSGA-II

- **Main finding**: *The more a marker CA rule is reversible, the lower its Hamming weight must be*



Diversity Analysis with Lexicographic GA/GP

- **Main finding**: *Lexicographic GA and GP achieves the best trade-off among number of distinct optimal solutions, highest and distinct Hamming weights achieved*



Conclusions

- **RQ1:** Despite the small size of the optimal solution set, GA and GP always converge to conserved landscape rules
  - **Take away:** *no need to use EA to construct conserved landscape RCA! :-|*
- **RQ2:** Conserved landscape rules seem to be characterized by low Hamming weights with respect to their size
  - **Take away:** *conserved landscape RCA not really useful for cryptography! :-(*
- **RQ3:** The Pareto fronts suggest a clear trade-off between reversibility and Hamming weight
  - **Take away:** *EA are interesting tools for doing experimental mathematics on CA :-)*

Future Directions

Several directions open for further research:

- Investigate the performance gap between GA and GP, by performing fitness landscape analysis
- Consider marker CA rules with *partially overlapping* landscapes, which may be more interesting for cryptography
- Find a theoretical explanation for the trade-off between reversibility and Hamming weight observed on the Pareto fronts.

References

[D95] Daemen, J.: Cipher and hash function design strategies based on linear and differential cryptanalysis. PhD thesis, KU Leuven (1995)

[H69] Hedlund, G.A.: Endomorphisms and Automorphisms of the Shift Dynamical Systems. Mathematical Systems Theory 3(4): 320–375 (1969)

[LM14] Leporati, A., Mariot, L.: Cryptographic properties of bipermutive cellular automata rules. J. Cell. Autom. 9(5-6):437–475 (2014)

[MPJL21] Mariot, L., Picek, S., Jakobovic, D., Leporati A.: Evolutionary algorithms for designing reversible cellular automata. Genetic Programming and Evolvable Machines 22(4): 429-461 (2021)

[MPJL20] Mariot, L., Picek, S., Jakobovic, D., Leporati A.: An Evolutionary View on Reversible Shift-Invariant Transformations. Proceedings of EuroGP 2020, pp. 118-134 (2020)

[MPLJ19] Mariot, L., Picek, S., Leporati, A., Jakobovic, D.: Cellular automata based S-boxes. Cryptography and Communications 11(1): 41–62 (2019)

[MPJL18] Mariot, L., Picek, S., Jakobovic, D., Leporati, A.: Evolutionary Search of Binary Orthogonal Arrays. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.): PPSN 2018 (I). LNCS vol. 11101, pp. 121–133. Springer (2018)

[PB83] Scott Provan, J., Ball, M.O.: The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected. SIAM J. Comput. 12(4): 777-788 (1983)

[PMLJ17] Picek, S., Mariot, L., Leporati, A., Jakobovic, D.: Evolving S-boxes based on cellular automata with genetic programming. Proceedings of GECCO (Companion) 2017, pp. 251-252 (2017)

[PMYJM17] Picek, S., Mariot, L., Yang, B., Jakobovic, D., Mentens, N.: Design of S-boxes Defined with Cellular Automata Rules. Proceedings of Conf. Computing Frontiers 2017, pp. 409-414 (2017)

[T90] Toffoli, T., Margolus, N.H.: Invertible cellular automata: a review. Physica D 45(1-3): 229–253 (1990)