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

Deriving Smaller OA from Bigger Ones with GA

Luca Mariot

L.Mariot@tudelft.nl

WEPO 2021 – November 30, 2021

Orthogonal Arrays (OA)

- (N,k,s,t) **Orthogonal Array**: $N \times k$ matrix over $\{0,\cdots,s-1\}$ s.t. each t-uple occurs $\lambda = \frac{N}{s^t}$ times in each $N \times t$ submatrix.

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

1	0	0
0	0	0
0	1	0
0	0	1
0	1	1
1	1	1
1	0	1
1	1	0

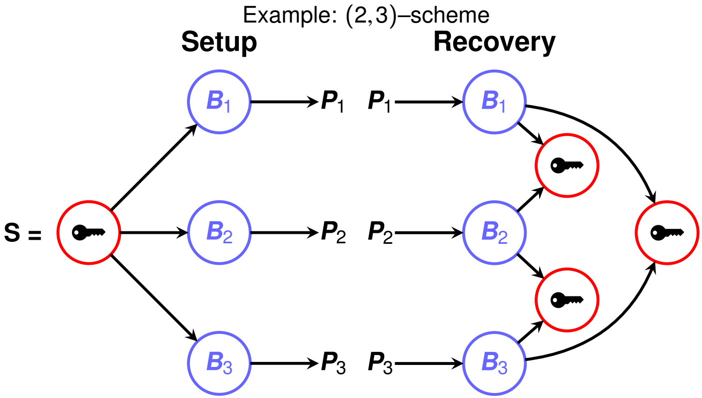
Example: OA $(8,4,2,3)$

$\Rightarrow$ Each 3-bit vector $(x_1,x_2,x_3) \in \{0,1\}^3$ appears once in the submatrix with columns 1, 3, 4

- Applications: designs of experiments, error-correcting codes, cryptography, ...

OA and Secret Sharing Schemes

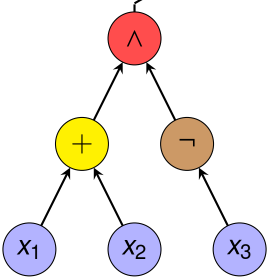
(t, n) **Threshold Scheme**: a **dealer** shares a **secret** S among n **players** so that at least t players are required to uniquely recover the secret S [B79, S79]



Remark: (t, n)–scheme $\Leftrightarrow$ OA $(N, n+1, s, t)$

Construction of OA by GA/GP [MPJL18]

- Focus on **Binary OA**: $s = 2$
- Column ⇔ truth table of a n-variable Boolean function
- For GP, the truth table is synthesized from the tree



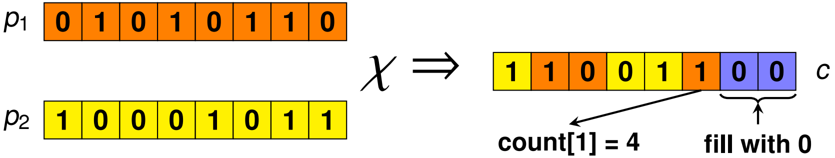
x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
0	1	1	1
1	0	0	1
1	1	0	1
1	1	0	0

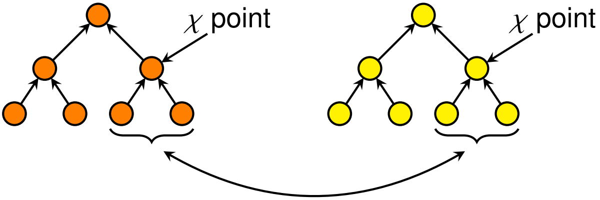
- Crossover and mutation are applied **column-wise**

Crossover Operators [MPJL18]

- **Remark**: Each column of an OA must be balanced
- **GA Crossover Idea:** Use *counters* to keep track of 0s and 1s
- Used for cryptographic Boolean functions [MCD98, ML15b]

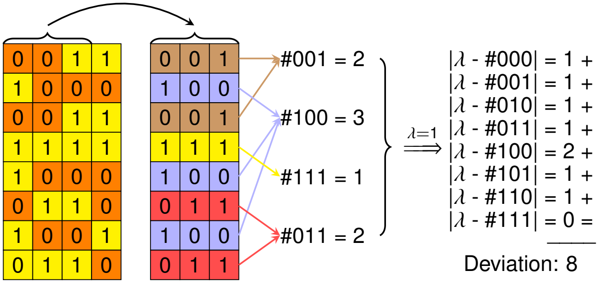


- **For GP**: Use standard subtree crossover



Fitness Function [MPJL18]

Idea: *minimize* in each $N \times t$ submatrix the number of occurrences of each t-uple deviating from λ

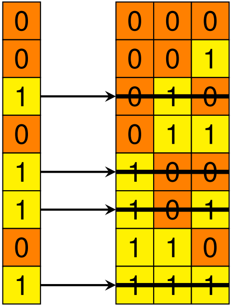


Fitness function: L^p distance between vector $(\lambda, \cdots, \lambda)$ and the vector of deviations for each submatrix

$$fit_p(A) = \sum_{S \text{ Submatrix}} \left(\sum_{x \in \{0,1\}^t} |\lambda - \#x|^p \right)^{\frac{1}{p}}$$

New problem: Removing rows from OA

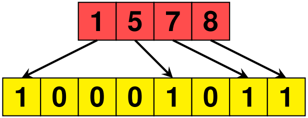
Problem statement: given an OA $(N, k, 2, t)$ with $\lambda = N/2^t$, find a smaller OA with $\lambda' < \lambda$ by removing $p = (\lambda - \lambda') \cdot 2^t$ rows



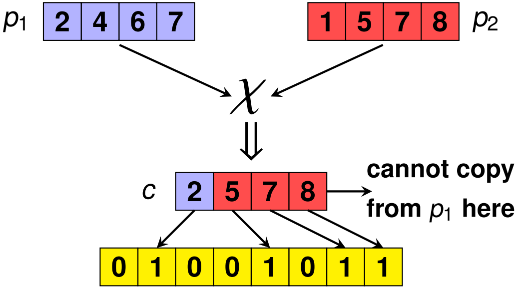
Question: How to choose the rows to remove?
Solution encoding for GA: N-bit string with p ones

Crossover Operator

Map of Ones Coding: Integer vector specifying the *positions of the* $p = (\lambda - \lambda') \cdot 2^t$ *ones* in the binary string



Idea: uniform crossover on the maps of ones, avoiding the insertion of duplicate positions in the child [MMT19, MMT20]



Experimental Setting

x_1	x_2	x_3	$x_1 \oplus x_2 \oplus x_3$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Basic test case: **zero-sum array** (ZS)

- First t cols.: all 2^t binary vectors in lexicographic order
- Col. $t+1$: XOR of the other cols.
- The whole matrix is an OA $(2^t, t+1, 2, t)$

OA/GA parameters:

- Initial OA: λ repetitions of ZS, with shuffled rows
- $t = 4$, $\lambda \in \{2,3,4\}$, $\lambda' \in \{1,2,3\}$
- Population size: 500 individuals, mutation probability: 0.2
- fitness budget: 100000 evaluations, runs: 30

Results

- **Main finding:** GA is able to converge with good success rate only on the smallest instance ($\lambda = 2$, $\lambda' = 1$)

λ, λ'	1	2	3
2	(24/30, 0.0)	–	–
3	(9/30, 7.07)	(4/30, 7.07)	–
4	(8/30, 7.07)	(0/30, 7.07)	(8/30, 7.07)

Table: Number of optimal solutions and median fitness

- Performances degrade quickly as soon as λ increases
- Best fitness follows a bi-modal distribution

Conclusions and Future Work

Recap:

- Genetic Algorithm for designing smaller Orthogonal Arrays starting from bigger ones
- The ones in the chromosome specify which rows to remove
- Experimental validation on a shuffled repetition of the zero-sum array

... Plenty of room for improvements!:

- Analyze the fitness landscape of this problem
- Systematic parameter tuning phase
- Use other crossover operators [MMT20, ML15b]
- Compare with other approaches on EA and combinatorial designs [SW92, MPJL17]
- Use other optimization methods (e.g., PSO [SUY06, ML15a])

References

[B79] Blakley, G.R.: Safeguarding cryptographic keys. In: International Workshop on Managing Requirements Knowledge, p. 313. IEEE Computer Society (1979)

[MMT20] Manzoni, L., Mariot, L., Tuba, E.: Balanced crossover operators in Genetic Algorithms. Swarm Evol. Comput. 54: 100646 (2020)

[MMT19] Manzoni, L., Mariot, L., Tuba, L.: Does constraining the search space of GA always help?: the case of balanced crossover operators. In: GECCO (Companion) 2019: 151-152. ACM (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)

[MPJL17] Mariot, L., Picek, S., Jakobovic, D., Leporati, A.: Evolutionary Algorithms for the Design of Orthogonal Latin Squares based on Cellular Automata. In: Proceedings of GECCO'17, pp. 306–313 (2017)

[ML15a] Mariot, L., Leporati. A.: Heuristic Search by Particle Swarm Optimization of Boolean Functions for Cryptographic Applications. In: GECCO 2015 (Companion): 1425-1426. ACM (2015)

[ML15b] Mariot, L., Leporati, A.: A Genetic Algorithm for Evolving Plateaued Cryptographic Boolean Functions. In: Proceedings of TPNC 2015: 33-45 (2015)

[MCD98] Millan, W., Clark, J., Dawson, E.: Heuristic Design of Cryptographically Strong Balanced Boolean Functions. EUROCRYPT 1998: pp. 489–499 (1998)

[SUY06] Saber, Z., Uddin, M.F., Youssef, A.M.: On the existence of (9, 3, 5, 240) resilient functions. IEEE Trans. Inf. Theory 52(5): 2269-2270 (2006)

[SW92] Safadi, R., Wang, R.: The use of genetic algorithms in the construction of mixed multilevel orthogonal arrays. Tech. rep., Olin Corp Cheshire CT Olin Research Center (1992)

[S79] Shamir, A.: How to share a secret. Commun. ACM 22(11):612–613 (1979)