The Influence of Local Search on GA with Balanced Representations

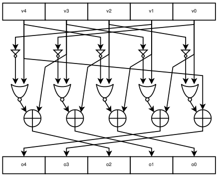
Luca Manzoni, **Luca Mariot**, Eva Tuba

luca.mariot@ru.nl

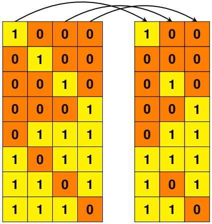
BIOMA 2022 – Maribor, November 18, 2022

Optimization with Balanced Representations

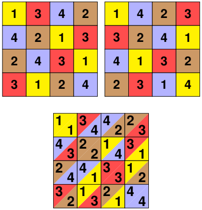
- **Setting:** feasible solutions are encoded by bitstrings composed of an equal number of 0s and 1s
- **Applications:** error-correcting codes, cryptography [M18, M19]



(a) Boolean functions

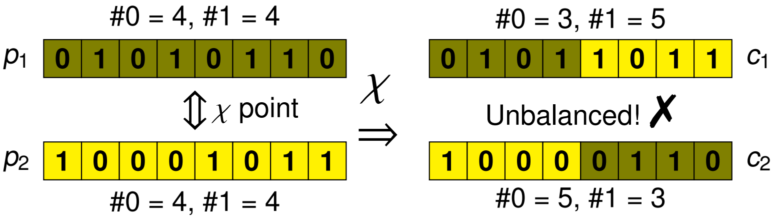


(b) Orthogonal Arrays



(c) Latin Squares

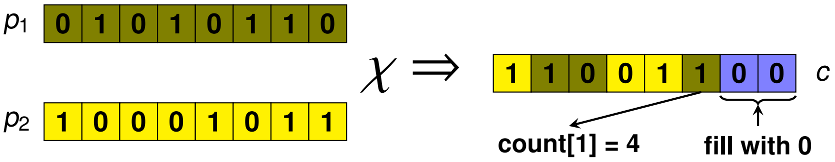
Classic Crossover on Balanced Problems



- In general, classic GA crossover operators in GA do not preserve balancedness
- **Approach:** employ balancedness-preserving operators [M20]

Counter-based crossover (CX1)

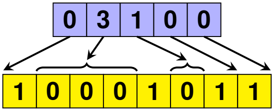
- Uniform crossover with *counters* to keep track of the multiplicities of zeros and ones [M98]
- copy the other value when the threshold is reached



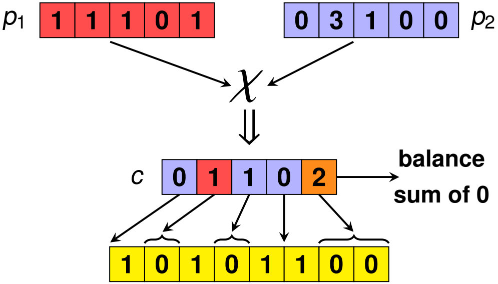
- No differences wrt order of positions to be copied [M20]

Zero-lengths Crossover (CX2)

Zero-lengths Coding: Integer vector specifying the *run lengths of zeros* between consecutive ones

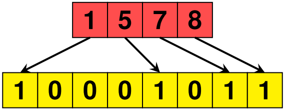


Idea: uniform crossover on the zero-lengths vectors, using an *accumulator* to track the sums of the run lengths

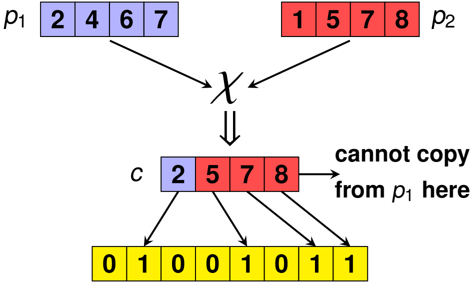


Map-of-Ones Crossover (CX3)

Map of Ones Coding: Integer vector specifying the *positions of the $N/2$ ones* in the binary string



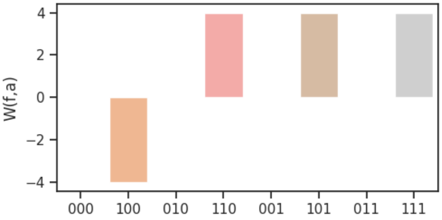
Idea: uniform crossover on the maps of ones, avoiding the insertion of duplicate positions in the child



Boolean Functions

- **Boolean function** of n variables: mapping $f : \{0,1\}^n \rightarrow \{0,1\}$
- **Walsh Transform** (WT): correlation of f with linear functions $a \cdot x = a_1 x_1 \oplus \cdots \oplus a_n x_n$

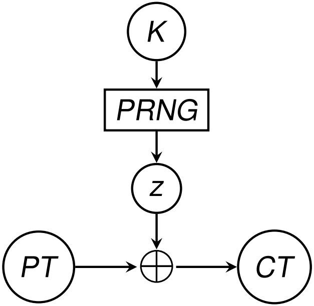
$$W_f(a) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus a \cdot x}$$



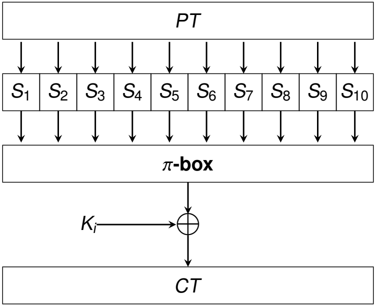
Example: $n = 3$ variables

(x_1, x_2, x_3)	$f(x)$	$W_f(a)$
000	0	0
001	1	-4
010	1	0
011	0	4
100	1	0
101	0	4
110	1	0
111	0	4

Boolean functions in symmetric crypto



(a) Stream cipher



(b) Block cipher

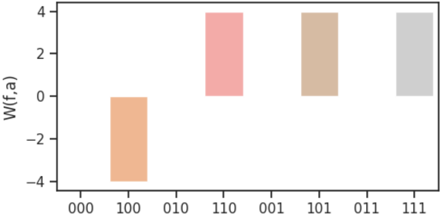
Used in the design of low-level primitives, e.g. [C21]:

- Pseudorandom number generators (PRNG)
- S-boxes $F : \{0,1\}^n \to \{0,1\}^n$, ...

Boolean Functions - Cryptographic Properties

- **Balancedness:** TT of f has the same number of 0s and 1s
- **High nonlinearity:** the nonlinearity of f is given by the WT as:

$$nl(f) = 2^{n-1} - \frac{1}{2}\max_{a\in\mathbb{F}_2^n}\left\{\left|W_f(a)\right|\right\}$$



Ex: f balanced, $nl(f) = 2^{3-1} - \frac{1}{2}\cdot 4 = 2$

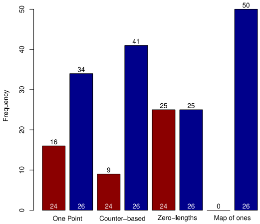
(x_1,x_2,x_3)	$f(x)$	$W_f(a)$
000	0	0
001	1	-4
010	1	0
011	0	4
100	1	0
101	0	4
110	1	0
111	0	4

- **Search space size**: 2^{2^n} (general), $\binom{2^n}{2^{n-1}}$ (balanced)

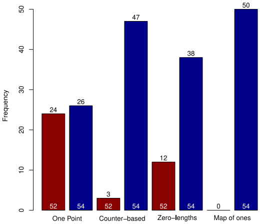
Performances of Balanced Crossover

- **Optimization objective**: max $nl(f)$, keep balancedness
- **Encoding**: 2^n-bit string $\Rightarrow$ Truth table of $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$

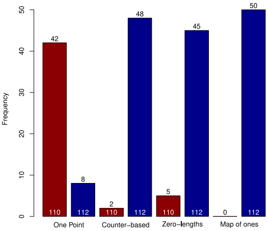
$n = 6$ ($opt = 26$)



$n = 7$ ($opt = 56$)



$n = 8$ ($opt = 116(?)$)

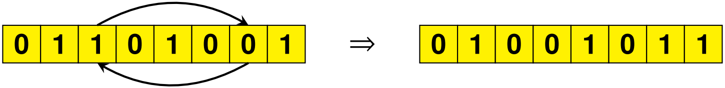


:-) Balanced crossover does give an advantage over one-point

:-(The advantage does not scale [M20, M21]

Local Search (LS) Step

- **Idea:** augment the GA with a (savvy) LS step
- **Basic move**: swap that improves nonlinearity



$f(x) = x_1 \oplus x_2 \oplus x_3$ $(nl = 0)$ $\qquad$ $f(x) = x_1 \oplus x_1 x_2 \oplus x_3$ $(nl = 2)$

- LS applied after crossover and mutation
- Efficient recomputation of the Walsh transform [M99]:

$$\Delta(a) = [(-1)^{f(y)} - (-1)^{f(z)}][(-1)^{a\cdot z} - (-1)^{a\cdot y}] \ ,$$

$$\Delta(a) \in \{-4, 0, +4\}$$

Experimental Settings

Research Hypotheses:

- **RQ1:** LS speeds up convergence to a local optimum
- **RQ2:** LS decreases diversity in the population

LS variants:

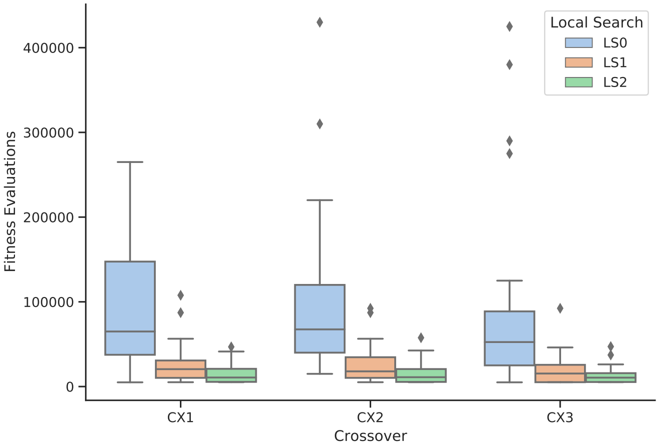
- LS0: no LS
- LS1: one step of LS
- LS2: steepest ascent

GA Parameters:

- Instances: $n = 6, 7, 8, 9$
- Fitness budget: 500 000
- Breeding: Steady-state
- Population size: 50
- Tournament size: 3
- Crossovers: CX1, CX2, CX3
- Mutation rates: 0.7
- Independent Runs: 30

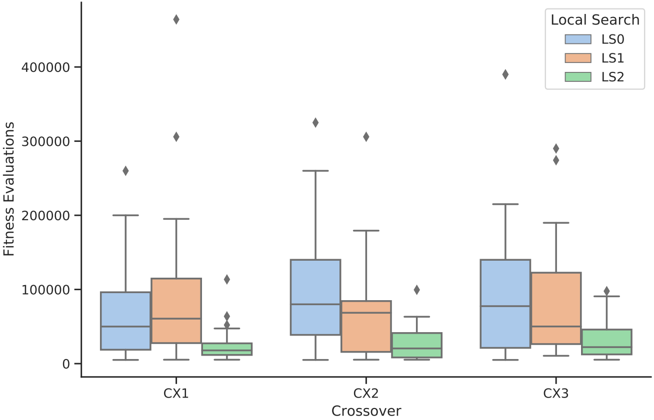
Results Convergence $n = 6$

Main Finding: LS greatly improves convergence speed



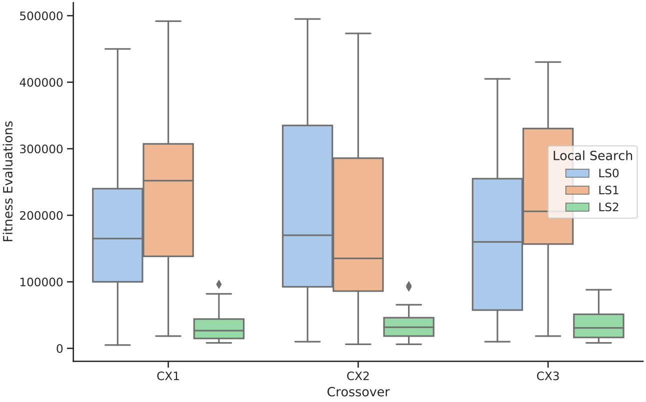
Results Convergence $n = 7$

Main Finding: Convergence speed improved by steepest ascent



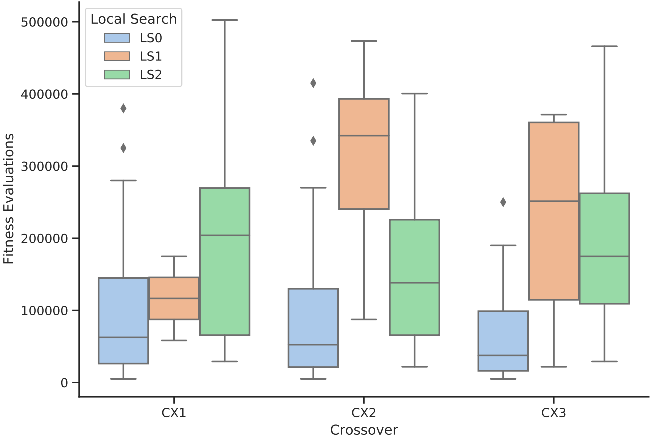
Results Convergence $n = 8$

Main Finding: No significant differences between LS0 and LS1



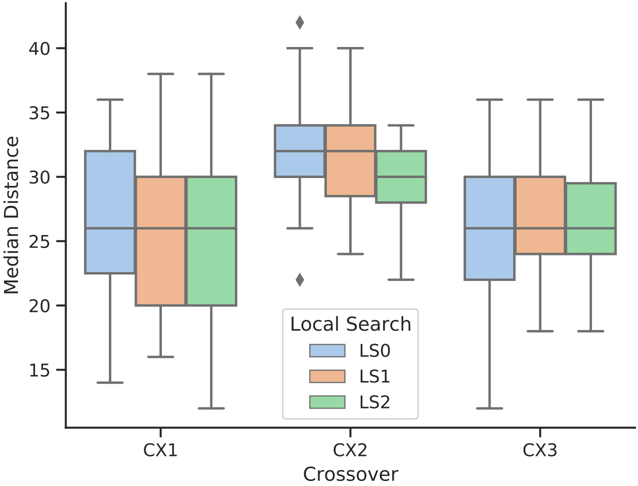
Results Convergence $n = 9$

Main Finding: LS slows convergence down (but finds better solutions)



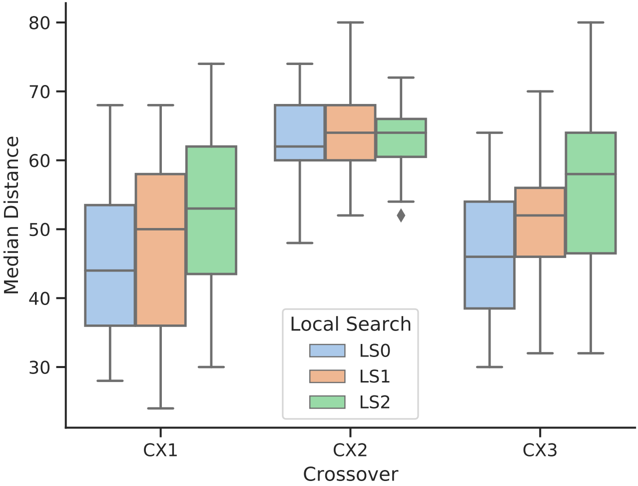
Results on Diversity $n = 6$

Main Finding: No significant differences on solutions' diversity



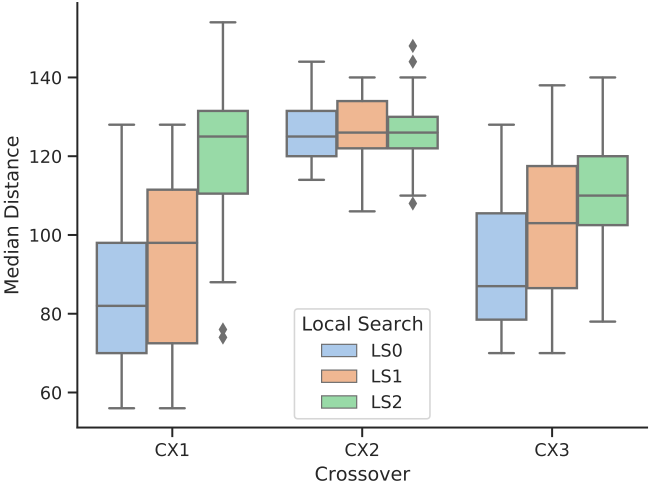
Results on Diversity $n = 7$

Main Finding: Mostly, no significant differences



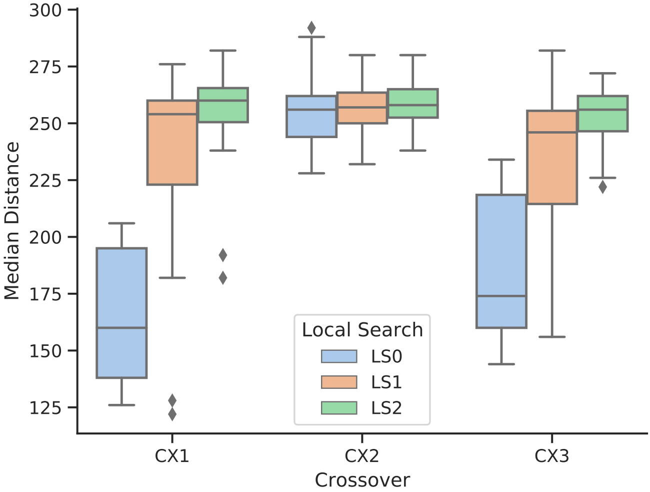
Results on Diversity $n = 8$

Main Finding: LS2 starts to *increase* diversity



Results on Diversity $n = 9$

Main Finding: LS1 and LS2 increase diversity except for CX2



Key Take Aways

Answers to our research hypotheses:

- **RH1**: as expected, LS mostly increases convergence speed
- **RH2**: surprisingly, LS has no effects or increases diversity

Possible insights:

- Improve best fitness by increasing fitness budget with LS2
- High diversity might be related to the fitness landscape shape
- Use different initialization strategies?

Conclusions and Future Works

Summing up:

- We augmented balanced GA with a LS step for the optimization of Boolean functions
- Curiously, LS makes the GA population more diverse

Future work:

- Perform Fitness Landscape Analysis to investigate the effect of different initialization strategies [J21]
- Experiments on other problems with balanced representation (orthogonal arrays [M18], Latin squares [M17], plateaued functions [M15]...)
- Compare with other approaches (e.g., GP [P16])

References

[C21] Carlet, C.: Boolean functions for cryptography and coding theory. Cambridge University Press (2021)

[J21] Jakobovic, D., Picek, S., Martins, M.S.R., Wagner, M.: Toward more efficient heuristic construction of boolean functions. Appl. Soft Comput., 107: 107327 (2021)

[M21] Manzoni, L., Mariot, L., Tuba, E.: Tip the balance: Improving exploration of balanced crossover operators by adaptive bias. In: Proceedings of CANDAR (Workshops) 2021, pp. 234–240 (2021)

[M20] Manzoni, L., Mariot, L., Tuba, E.: Balanced crossover operators in Genetic Algorithms. Swarm Evol. Comput. 54: 100646 (2020)

[M19] Mariot, L., Picek, S., Leporati, A., Jakobovic, D.: Cellular automata based S-boxes. Cryptogr. Commun. 11(1):41–62 (2019)

[M18] 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)

[17] 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)

[M15] Mariot, L., Leporati, A.: A Genetic Algorithm for Evolving Plateaued Cryptographic Boolean Functions. In: Proceedings of TPNC 2015: 33–45 (2015)

[M99] Millan, W., Clark, A.J., Dawson, E.: Boolean Function Design Using Hill Climbing Methods. In: Proceedings of ACISP 1999: 1-11 (1999)

[M98] Millan, W., Clark, J., Dawson, E.: Heuristic Design of Cryptographically Strong Balanced Boolean Functions. Proceedings of EUROCRYPT 1998, pp. 489–499 (1998)

[P16] Picek, S., Jakobovic, D., Miller, J.F., Batina, L., Cupic, M.: Cryptographic Boolean functions: One output, many design criteria. Appl. Soft Comput. 40: 635-653 (2016)