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]
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

\[
p_1 = \begin{array}{cccccccc}
0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
\end{array}
\]

\[
p_2 = \begin{array}{cccccccc}
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[
\chi \Rightarrow \begin{array}{cccccccc}
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
\end{array}
\]

count[1] = 4, fill with 0

▶ 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

![Zero-lengths vector: 0 3 1 0 0](image)

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

![Accumulator diagram](image)

- **balance sum of 0**
Map of Ones Coding: Integer vector specifying the *positions of the* \( N/2 \) *ones* in the binary string

<table>
<thead>
<tr>
<th>1</th>
<th>5</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
</table>

```
1 0 0 0 1 0 1 1
```

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

\[ p_1 \quad 2 \quad 4 \quad 6 \quad 7 \quad 1 \quad 5 \quad 7 \quad 8 \quad p_2 \]

\( \chi \)

\( c \)

\( 2 \quad 5 \quad 7 \quad 8 \)

```
0 1 0 0 1 0 1 1
```

cannot copy from \( p_1 \) here
**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

<table>
<thead>
<tr>
<th>$(x_1, x_2, x_3)$</th>
<th>$f(x)$</th>
<th>$W_f(a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>-4</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
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 \( F : \{0, 1\}^n \rightarrow \{0, 1\}^n \), ...

L. Mariot et al.  
The Influence of Local Search on GA with Balanced Representations
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:

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

Ex:

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

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

<table>
<thead>
<tr>
<th>$(x_1, x_2, x_3)$</th>
<th>$f(x)$</th>
<th>$W_f(a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>-4</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
Performances of Balanced Crossover

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

\[
\begin{align*}
n & = 6 \ (opt = 26) & n & = 7 \ (opt = 56) & n & = 8 \ (opt = 116(?) )
\end{align*}
\]

:-) 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

\[
f(x) = x_1 \oplus x_2 \oplus x_3 \quad (nl = 0)
\]

\[
f(x) = x_1 \oplus x_1 x_2 \oplus x_3 \quad (nl = 2)
\]

- 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
Main Finding: LS greatly improves convergence speed
Main Finding: Convergence speed improved by steepest ascent
Main Finding: No significant differences between LS0 and LS1
Main Finding: LS slows convergence down (but finds better solutions)
Results on Diversity $n = 6$

**Main Finding**: No significant differences on solutions’ diversity
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])


