Evolving Constructions for Balanced, Highly Nonlinear Boolean Functions with GP

Claude Carlet, Marko Djurasevic, Domagoj Jakobovic, Luca Mariot, Stjepan Picek

luca.mariot@ru.nl

GECCO 2022 – Boston, July 11, 2022
**Boolean Functions - Basic Definitions**

- \( F_2 = \{0, 1\} \), \( F_2^n \): \( n \)-dimensional vector space over \( F_2 \)
- A **Boolean function** of \( n \) variables is a mapping \( f : F_2^n \rightarrow F_2 \), most commonly represented by its **Truth Table (TT)** \( \Omega_f \)
- **Walsh Transform (WT)**: represents \( f \) as correlations with linear functions \( a \cdot x \), for \( a \in F_2^n \)

\[
W_f(a) = \sum_{x \in F_2^n} (-1)^{f(x) \oplus a \cdot x}
\]

- Example with \( n = 3 \) variables:

<table>
<thead>
<tr>
<th>((x_1, x_2, x_3))</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Omega_f)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(W_f(a))</td>
<td>0</td>
<td>-4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
To be useful in cryptography, $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ should be:

- **Balanced**: the TT of $f$ has the same number of 0s and 1s
- **Highly nonlinear**: the nonlinearity of $f$ is given by the WT as follows [C21]:

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

<table>
<thead>
<tr>
<th>$(x_1, x_2, x_3)$</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega_f$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$W_f(a)$</td>
<td>0</td>
<td>-4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

- Example: $f$ balanced, $nl(f) = 2^{3-1} - \frac{1}{2} \cdot 4 = 2$
Number of Boolean functions of $n$ variables: $2^{2n}$

<table>
<thead>
<tr>
<th>$n$</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{2n}$</td>
<td>256</td>
<td>65536</td>
<td>$4.3 \cdot 10^9$</td>
<td>$1.8 \cdot 10^{19}$</td>
<td>$3.4 \cdot 10^{38}$</td>
<td>$1.2 \cdot 10^{77}$</td>
</tr>
</tbody>
</table>

$\Rightarrow$ too huge for exhaustive search when $n > 5$!

In practice, one usually resorts to:

- **Primary Constructions**, where Boolean functions with certain properties are built from scratch [M73, D74]
- **Secondary Constructions**, where new Boolean functions are obtained from existing one [R76]
- **Heuristic optimization algorithms** (GA [M98, M15b, M20], GP [P16, M19a, M19b], PSO [M15a], ...)

The truth table $g(x)$ is synthesized from the tree [P16]

Example: $n = 3$, $f : \{0, 1\}^3 \rightarrow \{0, 1\}$

Problem: scales poorly when $n$ increases
Example of secondary construction: Rothaus’s construction [R76]

If $g, h, k$ and $g \oplus h \oplus k$ are bent (maximally nonlinear) on $\mathbb{F}_2^n$, then the following function is bent:

$$f(x_1, x_2, x) = g(x)h(x) \oplus g(x)k(x) \oplus h(x)k(x) \oplus [g(x) \oplus h(x)]x_1 \oplus [g(x) \oplus k(x)]x_2 \oplus x_1 x_2$$

where $(x_1, x_2, x) \in \mathbb{F}_2^{n+2}$ with $x_1, x_2 \in \mathbb{F}_2$, $x \in \mathbb{F}_2^n$

**Goal:** Evolve secondary constructions using GP
GP Representation

Predefined functions:

<table>
<thead>
<tr>
<th>$f_0$</th>
<th>1001</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>1010</td>
</tr>
</tbody>
</table>

Independent variables:

<table>
<thead>
<tr>
<th>$v_0$</th>
<th>0101</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>0011</td>
</tr>
</tbody>
</table>

- **Idea:** represent a secondary construction as a GP tree
- $f_0, f_1$: seed functions
- $v_0, v_1$: additional independent variables
- The GP tree yields a new function of $n + 2$ variables
- Seed functions are obtained through direct GP search

Output: 1010 1001 0101 1001
Experimental Settings

Problem-related parameters:

<table>
<thead>
<tr>
<th>Parameter description</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables</td>
<td>5, 6, 7, 8</td>
</tr>
<tr>
<td>Independent variables</td>
<td>1, 2</td>
</tr>
<tr>
<td>Number of seed functions</td>
<td>2, 4</td>
</tr>
<tr>
<td>Number of seed function groups</td>
<td>4</td>
</tr>
<tr>
<td>Seed functions type</td>
<td>balanced, bent</td>
</tr>
<tr>
<td>Type of fitness function</td>
<td>first group, sum of all groups, minimum of all groups</td>
</tr>
</tbody>
</table>

GP-related parameters:

- Population size: 500
- Mutation probability: 0.5
- Fitness budget: 500 000
- Max tree depth: 5
- Tournament size: 3
- Independent runs: 30
## Results

**Table: Results for search-based GP**

<table>
<thead>
<tr>
<th>GP search</th>
<th>n = 9</th>
<th>n = 10</th>
<th>n = 11</th>
<th>n = 12</th>
<th>n = 13</th>
<th>n = 14</th>
<th>n = 15</th>
<th>n = 16</th>
<th>n = 17</th>
<th>n = 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>min NL</td>
<td>240</td>
<td>480</td>
<td>992</td>
<td>1 953</td>
<td>3 905</td>
<td>8 001</td>
<td>16 192</td>
<td>32 512</td>
<td>65 280</td>
<td>130 561</td>
</tr>
<tr>
<td>avg NL</td>
<td>240</td>
<td>484.24</td>
<td>992</td>
<td>1 996.69</td>
<td>4 028.39</td>
<td>8 087.3</td>
<td>16 253.5</td>
<td>32 542.2</td>
<td>65 280</td>
<td>130 622</td>
</tr>
<tr>
<td>max NL</td>
<td>240</td>
<td>492</td>
<td>992</td>
<td>2 008</td>
<td>4 032</td>
<td>8 120</td>
<td>16 256</td>
<td>32 608</td>
<td>65 280</td>
<td>130 753</td>
</tr>
<tr>
<td># max</td>
<td>100%</td>
<td>2%</td>
<td>100%</td>
<td>14%</td>
<td>96%</td>
<td>1%</td>
<td>97%</td>
<td>2%</td>
<td>100%</td>
<td>2%</td>
</tr>
</tbody>
</table>

**Table: Results for secondary constructions**

<table>
<thead>
<tr>
<th>constr.</th>
<th>n = 8</th>
<th>n = 9</th>
<th>n = 10</th>
<th>n = 11</th>
<th>n = 12</th>
<th>n = 13</th>
<th>n = 14</th>
<th>n = 15</th>
<th>n = 16</th>
<th>n = 17</th>
<th>n = 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>seed NL</td>
<td>26</td>
<td>56</td>
<td>116</td>
<td>240</td>
<td>488</td>
<td>992</td>
<td>2 000</td>
<td>4 032</td>
<td>8 096</td>
<td>16 256</td>
<td>32 576</td>
</tr>
<tr>
<td>res. NL</td>
<td>116</td>
<td>240</td>
<td>488</td>
<td>992</td>
<td>2 000</td>
<td>4 032</td>
<td>8 096</td>
<td>16 256</td>
<td>32 576</td>
<td>65 280</td>
<td>130 688</td>
</tr>
</tbody>
</table>
Simplification of GP Solutions

- We used the ESPRESSO tool [R87] to minimize the best GP trees
- Performed an equivalence check among the best solutions

**Result:** many solutions turn out to be the same construction, especially when 2 seeds are used
Example of bloated GP construction:

\[
F(v_0, v_1, v) = \begin{cases} 
  f_0(v) , & \text{if } v_0 = 1 , \\
  f_1(v) \oplus v_1 , & \text{if } v_0 = 0 .
\end{cases}
\]

Main Remark: many constructions are equivalent to the well-known indirect sum construction [C21]
Conclusions and Future Works

Conclusions:
▶ GP can be used to evolve secondary constructions for Boolean functions, rather than Boolean functions directly
▶ Most of the solutions evolved by GP are equivalent, and correspond to the indirect sum construction

Future work:
▶ Test whether the indirect sum is the only construction achievable by GP under this encoding
▶ Experiment with non-independent additional variables
References


