Extending Genetic Programming with Recombinative Guidance

Hitoshi Iba and Hugo de Garis
Motivation

Recombinative Guidance for GP

Experimental results

- Performance-based Guidance
- Correlation-based Guidance for Symbolic Regression
- MDL-based Guidance for Numerical GP

Discussion

Related Works

Future Works
Motivation

- In traditional GP, recombination causes disruption of beneficial building-blocks, and mutation causes abrupt changes in the semantics. [Example](#).

- Propose a “recombinative guidance” mechanism for GP so as to realize an “adaptive recombination”.

  - [In GA](#)
  - [In GP](#)
Example

\[ a^2 + b^2 = c^2 \]
In GA

- **Schaffer (1987)**
  - Use a string feature called “punctuation”, which decides the crossover points of multi-point crossover.

  - Propose an adaptive control scheme for a mutation operator. Use a bit string to represent a mutation rate for each allele.
In GP

- **Koza (1990, ch.4. 12.2)**
  - Use a constrained crossover operator to evolve Neural Networks.

- **D’haeseleer (1994)**
  - The crossover operators attempt to preserve the context in which subtrees appeared in the parent tree.

- **Haynes (1995)**
  - Propose a strongly typed GP.

- **Whigham (1995)**
  - Use a context free grammar to control crossover and mutation operators for GP.
Recombinative Guidance for GP

- **S-values** (subtree values)
  - Denote: $SV(T)$, for a subtree $T$
  - Performance-based S-value
  - Correlation-based S-value
  - MDL-based S-value

- Use S-values to decide which subtree will be chosen.
S-value

- S-values are sorted in ascending order \((SV)\)
  - The larger S-value, the better subtree
- S-values are sorted in descending order \((\overline{SV})\)
  - The smaller S-value, the better subtree
- For subtrees \(T\) and \(T'\), \(T \preceq T'\) denoted that \(T'\) is a better building-block candidate than \(T\).
- If the S-value of \(T'\) is better than that of \(T\), then
  \[ T \preceq T' \iff \overline{SV}(T) \leq \overline{SV}(T') \]
  or
  \[ T \preceq T' \iff \underline{SV}(T) \geq \underline{SV}(T') \]
Process

- Apply a mutation operator to a subtree whose S-value is *worse*.
- Apply a crossover operator to a subtree whose S-value is *worse*, and get a subtree whose S-value is better from another parent.

**Example**
Example

- Let $W_1$ and $W_2$ be the subtrees with the worst $S$-values of $P_1$ and $P_2$.
- Let $B_1$ and $B_2$ be the subtrees with the best $S$-values of $P_1$ and $P_2$.
- A new child $C_1$ is a copy of $P_1$, in which $W_1$ is replaced by $B_2$.
- A new child $C_2$ is a copy of $P_2$, in which $W_2$ is replaced by $B_1$. 
Performance-based Guidance

- Experiment 1
  - The Lawnmower Problem
  - (Koza 1994, ch.8)
- Experiment 2
  - Artificial Ant on the San Mateo Trail
  - (Koza 1994, ch.12)
The Lawnmower Problem

- Goal: To find a program for controlling the movement of a lawnmower so that it cuts all the grass in the yard.
- Discrete 8x8 toroidal square area
- Fitness & Termination criterion
- Parameters
- S-value
- Result
Area

- **LEFT**
  Turn the orientation of the lawnmower counterclockwise by 90 degrees.

- **MOW**
  Move the lawnmower in the direction it is currently facing and mows the grass.
Fitness & Termination criterion

- **Fitness**
  - Raw fitness: the amount of grass mowed
  - Standard fitness: 64 – (Raw fitness)

- **Termination criterion:**
  Either the lawnmower has executed a total of 100 LEFT turns or 100 MOVEs.
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal Set</td>
<td>{MOW, LEFT}</td>
</tr>
<tr>
<td>Function Set</td>
<td>{PROG2, PROG3}</td>
</tr>
<tr>
<td>Population Size</td>
<td>120</td>
</tr>
<tr>
<td>Crossover Prob.</td>
<td>0.7</td>
</tr>
<tr>
<td>MutationProb.</td>
<td>0.033</td>
</tr>
</tbody>
</table>
S-value

\[ SV_1(T) = \frac{\text{raw fitness}(T)}{\text{raw fitness}(\text{ROOT})} \]

- It is not defined for the root node.
- It does not reflect the complexity of the program.

**Complexity-based S-value**

\[ SV_2(T) = \frac{\text{raw fitness}(T)}{\text{raw fitness}(\text{ROOT})} + \frac{\#N(\text{ROOT}) - \#N(T)}{\#N(\text{ROOT})} \]
Result

Std. Fitness

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>Variance</td>
</tr>
<tr>
<td>Usual Crossover</td>
<td>603.00</td>
</tr>
<tr>
<td>$SV_1$</td>
<td>655.53</td>
</tr>
<tr>
<td>$SV_2$</td>
<td>496.27</td>
</tr>
</tbody>
</table>

Generation
Artificial Ant on the San Mateo Trail

- **Goal**
  - To find a program for controlling the movement of an artificial ant so as to find all of the food lying along an irregular trail on a two-dimensional toroidal grid.

- **Area**: 13x13 grid
- **Raw fitness**: the sum of food pieces eaten

[Parameters & Result]
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal Set</td>
<td>{RIGHT, LEFT, MOVE}</td>
</tr>
<tr>
<td>Function Set</td>
<td>{IF_FOOD_AHEAD, PROGN}</td>
</tr>
<tr>
<td>Population Size</td>
<td>1200</td>
</tr>
<tr>
<td>Crossover Prob.</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation Prob.</td>
<td>0.033</td>
</tr>
</tbody>
</table>
Result

Std. Fitness

Random Crossover

Gen.
Correlation-based Guidance for Symbolic Regression

- **S-value**
- Experiment 3
  - A simple symbolic regression
- Experiment 4
  - Discovery of trigonometric identities
S-value

\[ \overline{SV_3}(T) = \frac{1}{N} \sum_{i=1}^{N} (y_i - t_i)^2 \]

\[ \overline{SV_4}(T) = |r(T)| \]

\[ r(T) = \frac{S_{yT}}{\sqrt{S_{TT} \cdot S_{yy}}} \]

\[ -1 \leq r(T) \leq 1 \]

A value of \( r(T) \) near zero indicates that the variable \( y_i \) and \( t_i \) are uncorrelated.
A simple symbolic regression

- $y = \frac{1}{2} x^2$
- **Parameters**
- **Standard fitness vs. Generation**
- **Average Number of Success Generations**
  (the number of success for 20 runs within a maximum of 200 generations)
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Fitness Cases</td>
<td>20</td>
</tr>
<tr>
<td>Terminal Set</td>
<td>{X}</td>
</tr>
<tr>
<td>Function Set</td>
<td>{+,-,\times,\div}</td>
</tr>
<tr>
<td>Population Size</td>
<td>40</td>
</tr>
<tr>
<td>Crossover Prob.</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation Prob.</td>
<td>0.033</td>
</tr>
<tr>
<td>Generation</td>
<td>200</td>
</tr>
</tbody>
</table>
Standard fitness vs. Generation

Std. Fitness

$SV_3$

Random Crossover

$SV_4$

Gen.
## Average Number of Success Generations

<table>
<thead>
<tr>
<th>Method</th>
<th># of Success</th>
<th>Average Gen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usual Recombination</td>
<td>10</td>
<td>18.88</td>
</tr>
<tr>
<td>$SV_3$</td>
<td>9</td>
<td>25.5</td>
</tr>
<tr>
<td>$SV_4$</td>
<td>14</td>
<td>5.93</td>
</tr>
</tbody>
</table>
Discovery of trigonometric identities

- $\cos^2 x = 1 - \sin^2 x$
  - 20 pairs $(x_i, y_i)$, $x_i \in [0, 2\pi]$
- **Parameters**
- **Std. Fitness vs. Gen.**
### Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># of Fitness Cases</td>
<td>20</td>
</tr>
<tr>
<td>Terminal Set</td>
<td>{x, 1.0}</td>
</tr>
<tr>
<td>Function Set</td>
<td>{+, -, \times, \div, \sin}</td>
</tr>
<tr>
<td>Population Size</td>
<td>500</td>
</tr>
<tr>
<td>Crossover Prob.</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation Prob.</td>
<td>0.033</td>
</tr>
</tbody>
</table>
Std. Fitness vs. Gen.

Graph showing the relationship between Std. Fitness and generation (Gen.) with two lines: one for Random Crossover labeled as $SV_3$ and another labeled as $SV_4$. The graph indicates a decrease in Std. Fitness over generations.
MDL-based Guidance for Numerical GP

- Advantages of STROGANOFF
- MDL fitness definition
- Chose the MDL value as the S-value for each subtree $T$ : $S_{\text{MDL}}(T) = \text{MDL}(T)$
- Predict the Mackey-Glass time series
- STROGANOFF parameters
- Results
Advantages of STROGANOFF

- Analog (polynomial) expressions complemented the digital (symbolic) semantics. The representational problem of standard GP does not arise for STROGANOFF.

- MDL-based fitness evaluation works well for tree structures in STROGANOFF, which controls GP-based tree search.

- Multiple-regressions tuned the node coefficients so as to guide GP recombination effectively.
MDL fitness definition

- MDL = 0.5N \log S_N^2 + 0.5k \log N
  - N: the number of data pairs
  - $S_N^2$: the mean square error
  - $S_N^2 = \frac{1}{N} \sum_{i=1}^{N} (\bar{y}_i - y_i)^2$
  - k: the number of parameters of the tree
The Mackey-Glass time series

\[
\frac{dx(t)}{dt} = \frac{ax(t - \tau)}{1 + x^{10}(t - \tau)} - bx(t)
\]

- \(a=0.2\), \(b=0.1\) and \(\tau=17\)
- The trajectory is chaotic and lies on an approximately 2.1 dimensional strange attractor.
### STROGANOFF parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>120</td>
</tr>
<tr>
<td>Prob. of Crossover</td>
<td>60%</td>
</tr>
<tr>
<td>Prob. of Mutation</td>
<td>3.3%</td>
</tr>
<tr>
<td>Selection Method</td>
<td>Tournament</td>
</tr>
<tr>
<td>Non-terminal Nodes</td>
<td>${a_0+a_1x_1+a_2x_2+a_3x_1x_2+a_4x_1^2+a_5x_2^2}$</td>
</tr>
<tr>
<td>Terminal Nodes</td>
<td>${x(t), x(t-6), x(t-12), x(t-18)}$</td>
</tr>
<tr>
<td>Target Variable</td>
<td>$x(t+85)$</td>
</tr>
<tr>
<td># of Training Data</td>
<td>500</td>
</tr>
<tr>
<td># of Testing Data</td>
<td>500</td>
</tr>
</tbody>
</table>

Following Hartman(1991)
Results (MDL-based)

- An Exemplar Tree
- Average Depths of Node of Best and Worst MDL values
- Prediction of Mackey-Glass Equation
An Exemplar Tree

NODE1: MSE = 0.0205, MDL = -0.0897

NODE2: MSE = 0.0207, MDL = -0.0914
  x (t-6)

NODE3: MSE = 0.0213, MDL = -0.0925
  x (t-12)

NODE4: MSE = 0.0233, MDL = -0.0920
  x (t-12)
  x (t-18)
  x (t)
Average Depths of Node of Best and Worst MDL values
Prediction of Mackey-Glass Equation
## Discussion

<table>
<thead>
<tr>
<th>Problem</th>
<th>Reference</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lawnmower</td>
<td>Koza 1994, ch.8</td>
<td>$R \leq SV_1 \leq SV_2$</td>
</tr>
<tr>
<td>Ant Trail</td>
<td>Koza 1994, ch.12</td>
<td>$R \leq SV_1 \leq SV_2$</td>
</tr>
<tr>
<td>Regression 1</td>
<td>Koza 1992, p.163</td>
<td>$SV_3 \leq R \leq SV_4$</td>
</tr>
<tr>
<td>Regression 2</td>
<td>Koza 1992, ch.10.1</td>
<td>$SV_3 \leq R \leq SV_4$</td>
</tr>
<tr>
<td>Time Series Pred.</td>
<td>Iba 1993, 1994</td>
<td>$R \leq SV_{MDL}$</td>
</tr>
</tbody>
</table>
The Success of methods

- Although there is no theoretical background for $SV_2$ definition, $SV_2$ is similar to MDL-based evaluation.
  - We may regard the first term of $SV_2$ as the inverse of the Exception_Coding_Length and the second term as the inverse of the Tree_Coding_Length.
- $SV_2$ and $SV_{MDL}$ evaluate the trade-off between performance and tree descriptions.
The Success of methods

- The MSE gives very little information as to whether a building-block will be useful.
- But, a correlation coefficient is a rather poor statistic for deciding whether an observed correlation is statistically significant.
- $SV_4$ is introduced as a way to evaluate the S-value heuristically, not as an absolute index.
Related Works

- Rosca (1994)
  - Use Block fitness function to discover a useful building block by his system AR-GP.

- Angeline (1996)

- Tackett (1995a)
  - Greedy Recombination operator.

- Teller (1996)
  - The co-evolution of intelligent recombination operators.

- Nordin (1996)
  - The introns regulate the crossover probabilities for subtrees.
Future Works

- Is it possible to design effective S-values for various problems?
- The approach can be combined with ADF.
- **Formalize S-values** in more mathematical ways.
- Eliminate the constraint of the “linear order” for the S-values.
Formalize S-values

- A characteristic of S-values: if a subtree is a good building-block, it has a good S-value. That is

\[ T \preceq T' \iff \text{Prob}(T) < \text{Prob}(T') \]

Where Prob(T) is the probability that a subtree T is part of a solution tree.