Emergent Semiotics in Genetic Programming and the Self-Adaptive Semantic Crossover

Rafael Inhasz and
rafael.inhasz@yahoo.com.br
Julio Michael Stern
jstern@ime.usp.br
Inst.of Mathematics and Statistics,
University of São Paulo, Brazil

www.ime.usp.br/∼jstern/slide/maxent10.pdf

We present SASC, Self-Adaptive Semantic Crossover, a new class of crossover operators for genetic programming.
SASC operators are designed to induce the emergence and then preserve good building-blocks, using meta-control techniques based on semantic compatibility measures.
SASC performance is tested in a case study concerning the replication of investment funds.

Key Words: Evolutionary algorithms, genetic programming, crossover operators, modularity and building blocks, emergent semiotics, semantic meta-control.

GP are meta-heuristics based on operators inspired on evolution of biological species.

Reproduction operators generate new individuals, *children*, from existing *parent(s)*.

Mutation operators act on single individuals for asexual reproduction, while *crossover* ops. act on pairs of individs. for sexual reproduction.

A mutation operation generates a random (usually small) change in the parent's code.

A crossover operation generates new children by swapping portions of their parents' codes at randomly selected *recombination points*.

Reproduction operators are random ops. However, they only introduce a limited amount of entropy in the process, making it possible for children to *inherited* many characteristics coded by their parents' genotype.

Invidivual phenotypes (genotype expression) are scored by a cost, merit or adaptation function.

A GP population evolves according to random selection processes for reproduction and death.

The entropy introduced at reproduction allows for creative innovation, while the selection processes induce learning constraints.

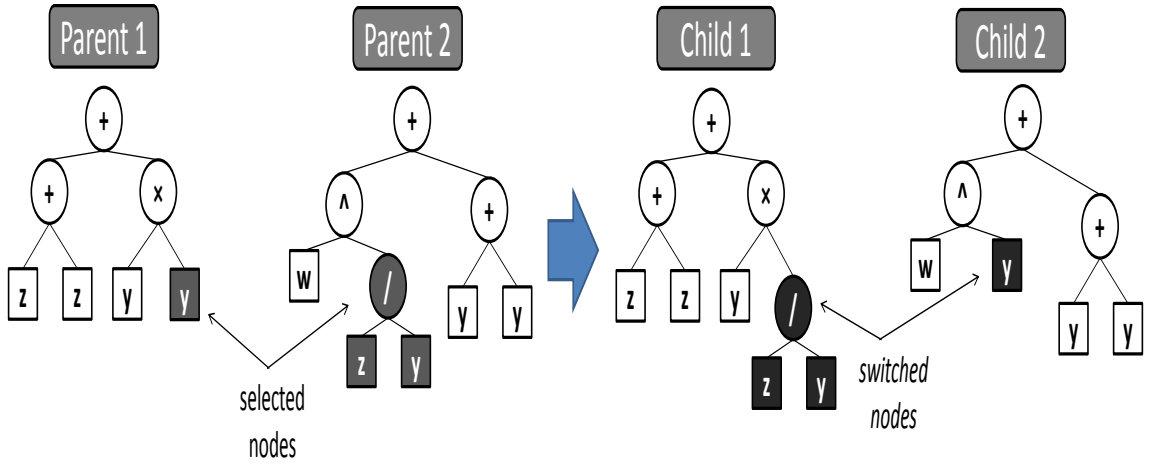
Under appropriate conditions (near) optimal individuals (likely) emerge in the population.

The *schemata theorem* shows that, under appropriate conditions, the emerging optimal solutions naturally exhibit a hierarchical modular organization. Such modules are known as *genes*, *schemata* or *building blocks*.

In light of the Schemata theorem, it is easy to understand that efficient crossover operators must be compatible with, preserve, favor, or even induce the emerging modular structure.

More efficient operators are less likely to break down existing building blocks during reproduction, an unfortunate event known in the literature as *destructive crossover*.

Figure 1: GP synthesis of functional tree for the target function, $f(w, y, z) = y^2 + w^{z/y}$, from $OP = \{+, -, \times, /, \wedge\}$, the expanded arithmetic operators. Inputs at leaves (squares), operators at internal nodes or root (circle).



Crossover w.highlighted recombination points. Parents contain the components, partial solutions or building blocks y^2 and $w^{z/y}$, not preserved by this destructive crossover.

A child inherits its root node, and hence usually most of its code, the *mother*, and a usually smaller sub-tree from its *father*.

Intuition behind modified crossover operators.

P.Angeline observed the acumulation of inert code (extraneous or junk code, introns) that nevertheless protected good building blocks.

Survivors in GP must be well adapted individuals, with good building blocks. Moreover, successful breeders must be able to give these building blocks intact to their children.

Non-uniform selection of recombination points. Large meta-control variables should mark plausible building blocks, indicating good recombination points to be used (again) in the future.

Genotype codes (exons) and meta-control variables (introns) should both co-evolve, facilitating the emergence, marking, and preservation of good building blocks.

SSAC - *Selective Self-Adaptive Crossover:*

Each node, $n(i)$, stores a meta-control variable, ρ_i, normalized to $0 \leq \rho_{min} \leq \rho_i \leq \rho_{max}$. The probability of selecting $n(i)$ for recombination is proportional to ρ_i, that is, $p_i = \rho_i / \sum_j \rho_j$.

After crossover, children's nodes carry along the meta-control variables they had at the parents, and afterwards suffer a random perturbation. For ex., the meta-control variable in node $n(i)$ can be updated as $\rho'_i = (1 + \mu_i + \sigma_i \epsilon)\rho_i$, $\epsilon \sim N(0, 1)$, drift $\mu_i \geq 0$, scale factor $\sigma_i > 0$.

SAMC - *Self-Adaptive Multi-Crossover:*

Meta-control variables interpreted as absolute probabilities, $\rho_{max} = 1$. Recombination point selected in a two steps: (1) All nodes are marked 1 w.probability ρ_i, and 0 otherwise. (2) Uniform selection from nodes marked 1.

SASC - Self-Adaptive Semantic Crossover
Incorporates information concerning the sub-trees rooted at possible recombination points. $A(i)$ at the father, and $B(j)$ at the mother. $A(i)$ and $B(j)$ are phenotypically similar if their output, computed at the available (training) records, agree within a specified tolerance.

SASC first heuristic procedure defines new meta-control variables, δ_i, at the father, A. Let $A(i)$ be the sub-tree of A rooted at $n(i)$. For each $A(i)$, the procedure searches the mother, B, for sub-trees, $B(j)$, that are similar to and also either the same size or shorter than $A(i)$. If such a short similar sub-tree is found, $\delta_i = \rho_{min}$, otherwise, $\delta_i = \rho_i$ (old meta-contr.variab.). Finally, the father's recombination point is selected with probabilities $p_i = \delta_i / \sum_j \delta_j$.

The intuition behind the first heuristic procedure is to stimulate innovation, that is, to only chose recombination points at the father that, by the crossover operation, are able to contribute with an innovative component, $A(i)$, that is not already present in the mother, B, or, at least, to contribute with a similar component that is more efficiently coded.

A second heuristic procedure selects the recombination point at the mother, $m(j)$ - root of sub-tree $B(j)$.

The idea behind this second heuristic procedure is to stimulate the crossover to exchange sub-trees, $A(i)$ and $B(j)$, with analogous meanings, compatible semantics, similar interpretations, etc. This heuristic procedure draws inspiration from biology, where analogy is defined as compatibility in function but not necessarily in structure or evolutionary origin.

Again, new meta-control variables, λ_j are defined for the nodes $m(j)$, followed by a random selection with probabilities $p_j = \lambda_i / \sum_j \lambda_j$. The formal expression used to evaluate the meta-control variables of the second heuristic is:

$$\lambda_j = w_0 + [\sum_{d=1}^{D} w_d C_k(A(i), B(j))]$$

The index d spans D semantic dimensions or factors. The positive weights, w_d, add to one, and the semantic compatibility measures, C_k, are normalized in the interval $[0, 1]$.

The functional form of the compatibility measures, $C_k(\ )$, are completely dependent on insights and interpretations for the actual problem being solved.

In the case at Figure 1, the analogy between sub-trees could be, for ex, simply the fraction of input variables they share in common.
In this case, the compatibility measure would be 1 for the blocks coding y^2 e $2y$, and 1/3 for the blocks coding y^2 and $w^{z/y}$.

After a SASC crossover, the children's nodes carry along the parents' meta-control variables, and are afterwards randomly updated with a positive drift at the recombination points and a null drifts elsewhere, $\rho'_i = (1 + \mu_i + \sigma_i \epsilon)\rho_i$.

Our implementation of SASC methods is based on ECJ, an open-source evolutionary computing system written in Java. ECJ is developed at George Mason University's ECLab Evolutionary Computation Laboratory.

ECJ also supports distributed computing, specifying the desired number of parallel threads according to the available resources offered by the hardware and operating system. This feature was especially useful for multi-population scenarios, where SASC GP had an excellent performance.

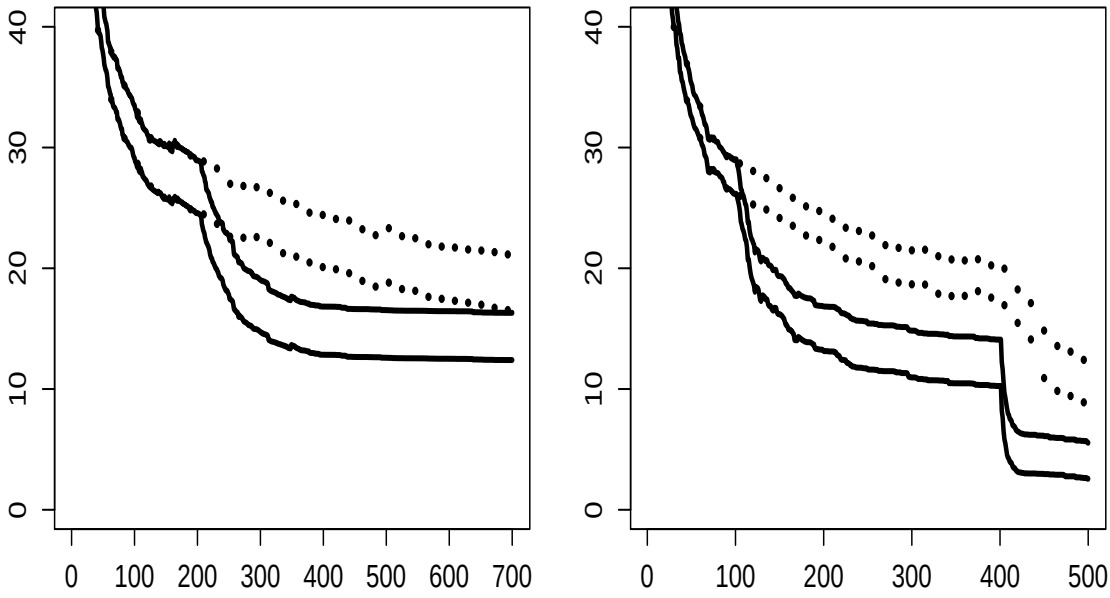
Test case problem concerning the replication of an hypothetical investment fund:
Lemon, the hypothetic fund, is based on stocks negotiated at *BM&F-Bovespa - São Paulo Securities, Commodities and Futures Exchange*.
Lemon's daily log-return, r_t, is given by the log-return average of four components, r_t^k, corresponding to key economic sectors. These are, using *BM&F-Bovespa* equity codes:
$r^1 = \min(BBDC4, PETR4, BBAS3)$,
$r^2 = \min(LAME4, LREN3, NETC4)$,
$r^3 = \max(TNLP4, TCLS4, VIVO4)$ and
$r^4 = \max(CYRE3, ALLL11, GFSA4)$.
These components represent four key economic sectors: Telecommunications, construction and transports, finance and cyclic consumption.

Portfolios of this kind are typical of correlation trade, and can be easily synthesized using readily available exotic derivatives like rainbow options, that is, calls or puts on the best or worst of several underlying assets.

An asset manager wants to synthesize a second fund, Lime, tracking fund Lemon. However, only the daily share values of fund Lemon are available, not its operational rules. The atoms for this problem are the log-returns of 63 of the most liquid stocks negotiated at *BM&F-Bovespa*, that include all the stocks used to specify fund Lemon. The primitive operators are $\{\max, \min, \text{mean}\}$, the maximum, minimum and mean value of two real numbers.

The fitness function is the mean squared error between the synthetic and the target log-returns, plus a regularization term adding, for each node, $n(i)$, a penalty $\pi(i)$. We used $\pi(i) = c_{h(i)} 2^{h(i)-1}$, where $h(i)$ is the height of node $n(i)$. The purpose of regularization term is to avoid needless complexity and overfitting in the final model. SASC's semantic compatibility function is the Boolean indicator of having at least one atom in common.

Figure 2 compares the GP results using standard and SASC crossover operators. This figure displays 95% confidence intervals for the mean square error of the best solution found at each generation over 50 independent GP runs. Cenario 1: Single population. Cenario 2: Merge of 8 sub-populations.



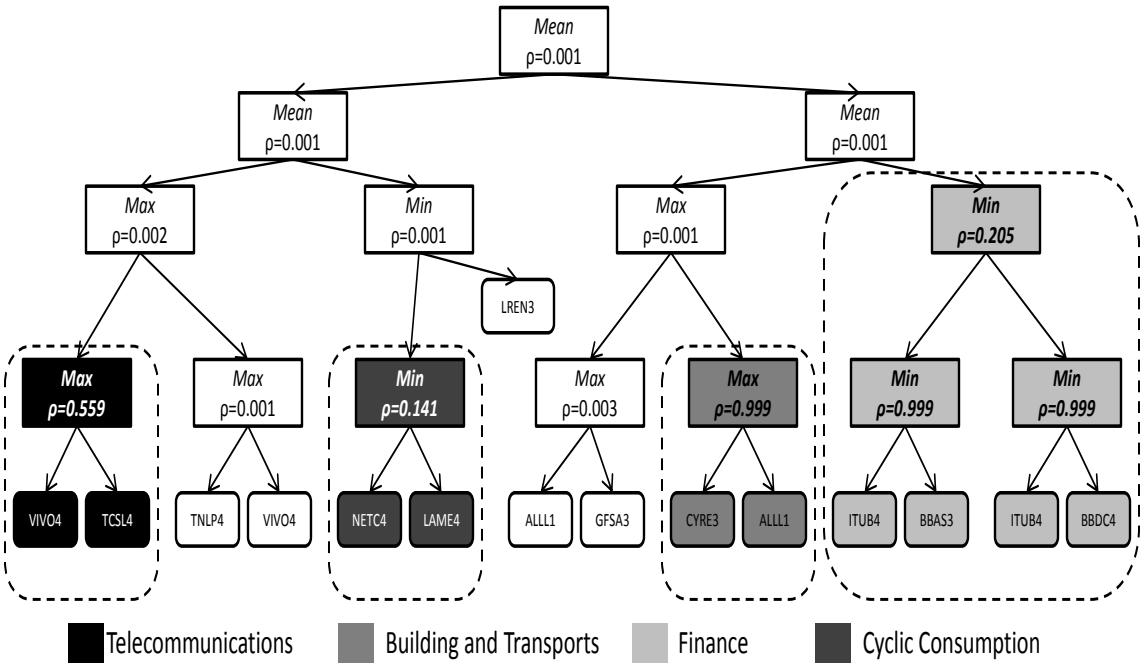


Figure 3 shows the best empirical solution found by SASC GP, highlighting the building blocks encapsulated by meta-control variables larger than a critical threshold. This solution replicates very well the target fund, and each of the highlighted building blocks corresponds to one of the key economic sectors.

Each best solution found at a batch of 50 SASC GP experiments was categorized according to the number of key economic sectors represented by a constituent building block, plus a separate category for solutions containing other (spurious) building blocks. Table 1 displays the average MSE of each category.

Category	Scenario 1	MSE	Scenario 2	MSE
One	14%	12.3	10%	8.9
Two	16%	8.1	30%	1.9
Three	8%	9.3	38%	1.4
Four	0%	-	4%	0.1
Other	62%	21.7	18%	10.2

Better adjusted functional trees have more of the four key economic sectors as a building blocks. It is remarkable how well the best solutions offered by SASC GP, synthesized only from input-output data, are able to capture the logic and semantics of the target fund.