# Geometric Semantic Grammatical Evolution

Alberto Moraglio, James McDermott and Michael O'Neill

## Abstract

Geometric Semantic Genetic Programming (GSGP) is a novel form of Genetic Programming (GP) based on a geometric theory of evolutionary algorithms that searches directly the semantic space of programs. We show how to extend this framework to Grammatical Evolution (GE). We refer to the new method as Geometric Semantic Grammatical Evolution (GSGE).

In GSGP, search operators act on the syntax of the programs but can be understood as acting directly on the underlying semantics of programs: mutation and crossover produce offspring which are, respectively, semantically close to and semantically intermediate to their parents. Specific GSGP operators for Boolean, Regression and Classification domains have been derived [1] and have a very simple form. This is possible because the mapping from genotypes to semantics in GP is simple, not complex as was widely believed before GSGP. Furthermore, the fitness landscape seen by GSGP is *always* a simple unimodal landscape, and its search is provably good on very large classes of problems [2].

GE [3] is a very successful form of GP that represents programs indirectly as integer vectors. Phenotypes are obtained through depth-first traversal of the grammar, using the genotype to select among multiple alternatives in the rules. One of the benefits of this indirect encoding is that it simplifies the application of search to different programming languages and constrained structures. A common criticism of GE is that because of the complexity of its genotype-phenotype mapping, search operators can be disruptive in terms of their syntactic and semantic effects (e.g., low locality [4]).

We introduce simple search operators for GE which are semantically geometric, i.e. perfectly well-behaved in terms of semantic effects. Given the non-trivial developmental phase of GE, it is surprising that these operators are at all possible, especially in a simple form. However, the first contribution of this research is that we have observed that the GE genotype-phenotype mapping naturally preserves (compositional) modularity: phenotypic modules (i.e., subtrees) correspond to genotypic modules (i.e., substring blocks). Together with a compositional interpretation of the geometric semantic operators, this implies the existence of a *genotypic* crossover/mutation scheme (i.e., on integer strings) equivalent to the GSGP *phenotypic* crossover/mutation scheme (i.e., on trees): that is, an implementation of geometric semantic operators for GE. We illustrate this for Boolean domains. Let us consider the simple grammar for Boolean

expressions in Fig. 1 (left).

```
(A) <expr> ::= (<expr> <biop> <expr>)  (0)
             | <uop> <expr>            (1)
             | <var>                   (2)

(B) <biop> ::= and  (0)
             | or   (1)

(C) <uop> ::= not  (0)

(D) <var> ::= x (0)
            | y (1)
            | z (2)
```

```
0:        (<expr> <biop> <expr>)
0:        (<expr> <biop> <expr>)
g(P1):    P1
0:        and
g(R):     R
1:        or
0:        (<expr> <biop> <expr>)
1:        <uop> <expr>
0:        <uop> ::= not
g(R):     R
0:        and
g(P2):    P2
```

Figure 1: Grammar (left) & derivation scheme of phenotype (right).

The geometric semantic crossover for Boolean expressions [1] is

$$O = (P1 \wedge R) \vee ((\neg R) \wedge P2)$$

where $P1$ and $P2$ are the parent Boolean expressions, $R$ is a random Boolean expression, and $O$ is the offspring Boolean expression. The corresponding geometric semantic crossover for this grammar is

$$g(O) = [0, 0, g(P1), 0, g(R), 1, 0, 1, 0, g(R), 0, g(P2)]$$

where $g(.)$ returns a genotype of its argument. The offspring $O$ has the genotype formed by substituting the genotypes of $P1$, $P2$ and $R$ in the above pattern.

Fig. 1 (right) shows that expanding the expression $g(O)$ using the grammar while considering $P1$, $R$, and $P2$ as parameter expressions we obtain the geometric semantic crossover scheme on phenotypes. A geometric semantic mutation for GE for Boolean domain can be defined analogously, as can operators for arithmetic expressions and classifiers.

We will present experimental results comparing GE and GSGE on Boolean, symbolic regression, and classifier problems.

# References


[1] Moraglio, A., Krawiec, K., Johnson, C.: Geometric semantic genetic programming. In: Proc. PPSN XII, Springer (2012) 21–31

[2] Moraglio, A., Mambrini, A., Manzoni, L.: Runtime analysis of mutation-based geometric semantic genetic programming on boolean functions. In: Proceedings of the Twelfth Workshop on Foundations of Genetic Algorithms XII. FOGA XII '13, New York, NY, USA, ACM (2013) 119–132

[3] O'Neill, M., Ryan, C.: Grammatical Evolution: Evolutionary Automatic Programming in a Arbitrary Language. Genetic programming. Kluwer Academic Publishers (2003)

[4] Rothlauf, F., Oetzel, M.: On the locality of grammatical evolution. In Collet, P., et al., eds.: EuroGP. Volume 3905 of LNCS., Budapest, Hungary, Springer (10 - 12 April 2006) 320–330