Geometric Semantic Genetic Programming with Perpendicular Crossover and Random Segment Mutation for Symbolic Regression

Qi Chen(✉), Mengjie Zhang, and Bing Xue

School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{Qi.Chen,Mengjie.Zhang,Bing.Xue}@ecs.vuw.ac.nz

Abstract. Geometric semantic operators have been a rising topic in genetic programming (GP). For the sake of a more effective evolutionary process, various geometric search operators have been developed to utilise the knowledge acquired from inspecting the behaviours of GP individuals. While the current exact geometric operators lead to overgrown offsprings in GP, existing approximate geometric operators never consider the theoretical framework of geometric semantic GP explicitly. This work proposes two new geometric search operators, i.e. perpendicular crossover and random segment mutation, to fulfil precise semantic requirements for symbolic regression under the theoretical framework of geometric semantic GP. The two operators approximate the target semantics gradually and effectively. The results show that the new geometric operators bring a notable benefit to both the learning performance and the generalisation ability of GP. In addition, they also have significant advantages over Random Desired Operator, which is a state-of-the-art geometric semantic operator.

Keywords: Genetic programming · Symbolic regression · Geometric semantic operators

1 Introduction

Semantic Genetic Programming (SGP) [8], which is a recently developed variant of Genetic Programming (GP) [5], makes use of semantic-aware search operators to produce offsprings that are highly correlated with their parents in behaviour. The definition of semantics in GP is different from domain to domain. In symbolic regression, the semantics of a GP program refers to a vector, the elements of which are the corresponding outputs of the program for the given samples [8]. The semantics of GP individuals form the semantic space. The fitness function (i.e. a kind of distance measure) spanning over the semantic space has a conic shape, thus search in such a unimodal space should be easy in principle [6]. However, it is not the case in practice, since the semantic space is not the space being searched in SGP. One particular category of SGP, Geometric Semantic GP

Y. Shi et al. (Eds.): SEAL 2017, LNCS 10593, pp. 422–434, 2017.
https://doi.org/10.1007/978-3-319-68759-9_35

(GSGP) [7], opens a new direction to utilise the semantics of GP individuals. GSGP implements the exact geometric semantic operators, which generate offsprings that lie either on the segment of the parents in the semantic space or in the interval bound defined by one parent. In this way, GSGP makes the desired semantics as the main driving force of the evolutionary process.

Research on GSGP mainly focuses on two aspects: *the theoretical framework* which poses the geometric requirement to the offspring, and *the implementation algorithm of* the geometric operators. Previous work [9] has figured out that the implementation algorithm in GSGP [7], which is a linear combination of parents, typically leads to over-grown offsprings, while the theoretical framework of it is the principle of many successful applications [2,10]. The over-grown offsprings are expensive to execute in both memory and time, which makes GSGP difficult to use in practice. Recent research has proposed variants of geometric operators to overcome this limitation. Random Desired Operator (RDO) [11] is one of the state-of-the-art geometric operators. Although these approximate geometric operators can eliminate over-grown individuals to certain extent, they never follow the theoretical framework of GSGP to further improve the effectiveness of the semantic operators. This work aims to fulfil this gap to some extent.

The overall goal of this work is to develop new geometric semantic operators for crossover and mutation to fulfil new semantic requirements under the theoretical framework of GSGP. The new desired semantics for offspring individuals will be much more precise than that in GSGP, and more diverse than that in RDO (which only consider the target semantics). The newly desired semantics are expected to guide the evolutionary search in a more effective way. A comparison between the proposed geometric operators and RDO will be conducted to investigate the effect of the new operators. Specifically, three research objectives emerge in this work:

- whether the proposed geometric operators can improve the learning performance of RDO on the training data,
- whether the proposed geometric operators can generalise better than RDO on new/unseen test data, and
- whether the new geometric operators can evolve programs with a smaller program size, and accordingly being more understandable.

2 Related Work

2.1 Geometric Semantic Genetic Programming

The definition of the theoretical framework of GSGP is as follows [7]:

Definition 1. *Given two parent individuals P_1 and P_2, geometric semantic crossover generates offspring $O_j (j \in 1, 2)$ having semantics $S(O_j)$ on the segment between the semantics of their parents, i.e., $\|S(P_1), S(P_2)\| = \|S(P_1), S(O_j)\| + \|S(O_j), S(P_2)\|$.*

Definition 2. *Given a parent P, r-geometric semantic mutation produces offspring O in a ball of radius r centered in P i.e., $\|S(P), S(O)\| \leq r$.*

Moraglio et al. [7] proposed an algorithm to implement the exact geometric operators. This implementation is a linear combination of the parent(s) and one/two randomly created programs. The proposal of the exact geometric semantic operators opens a new direction in SGP, since geometric semantic operators aim to search directly in the semantic space. However, the excessive growth in the size of the offspring produced by these operators is an obstacle to the application of GSGP, since it leads to an expensive computational cost and decreases the interpretability of the evolved models. In addition, the exact geometric crossover is criticised to contribute little to improving the generalisation ability of GP [4].

2.2 RDO and Semantic Backpropagation

Variants of GSGP have been proposed to overcome the limitation of GSGP. Pawlak et al. [11] proposed a semantic operator named Random Desired Operator (RDO), which generates offspring individuals with the target semantics as their desired semantics. The rationale behind RDO is that achieving a subtarget (part of the whole target) is much easier than the whole target. Based on a randomly selected node of a GP individual, the desired semantics of the GP individual is split into two parts, which are the semantics of the fixed suffix and the desired semantics of the selected node. Then RDO only needs to approximate the semantics of the selected node. To this end, a semantic backpropagation (SB) algorithm was proposed [11]. The SB algorithm obtains the desire semantics of the selected node by backpropagating through a chain of nodes from the root to the node. Then a new subtree with the desired semantics is obtained from a predefined semantic library, and replaces the original subtree. Although the SB algorithm provides a sensible way to obtain the subtarget semantics, RDO can potentially lead to a limitation of quickly losing the semantic diversity, which might be caused by the unique desired semantics for all the offspring. Furthermore, RDO has a greedy nature in chasing a lower training error, which might make GP suffer from overfitting and can not generalise well on the test data.

3 The Proposed Geometric Semantic Operators

This work proposes two new geometric operators named *perpendicular crossover* and *random segment mutation* to fulfil new semantic requirements for the offspring individuals. The new requirements are either more effective in approximating the target semantics or easier to achieve than the originally desired semantics in both GSGP and RDO. The semantics of the children individuals rely on their parent(s) in GSGP, while RDO only considers the target semantics. The new geometric operators utilise both types of semantics. For presentation convenience, GP with the two new geometric operators is named *NGSGP*.

3.1 Perpendicular Crossover

To have a more precisely semantic control and approximate the target semantics more effectively than the exact geometric semantic crossover operator, we propose a new semantic crossover operator called—*perpendicular crossover*. Given two parent individuals, perpendicular crossover (PC) generates offsprings having two geometric properties in semantic space. The first property is that the semantics of offsprings (i.e. represented as a point in semantic space) need to stand on the line defined by the semantics of their parents. The second one is that the line defined by the target semantics and the offspring point should be perpendicular to the given line of their parents. The first three columns in Fig. 1 illustrate the three possible positions of offspring $\boldsymbol{O}$. Suppose the target semantic is $\boldsymbol{T}$, and the semantics of the two parents are $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. As shown in the figure, the three points define a triangle. α refers to the angle between the relative semantics of $\boldsymbol{P}_2$ and $\boldsymbol{T}$ to $\boldsymbol{P}_1$, while β is its counterpart to $\boldsymbol{P}_2$. The angle α and β are defined as follows:

$$\alpha = \arccos\left(\frac{(\boldsymbol{T}-\boldsymbol{P}_1)\cdot(\boldsymbol{P}_2-\boldsymbol{P}_1)}{\|\boldsymbol{T}-\boldsymbol{P}_1\|\cdot\|\boldsymbol{P}_2-\boldsymbol{P}_1\|}\right) \quad \beta = \arccos\left(\frac{(\boldsymbol{T}-\boldsymbol{P}_2)\cdot(\boldsymbol{P}_1-\boldsymbol{P}_2)}{\|\boldsymbol{T}-\boldsymbol{P}_2\|\cdot\|\boldsymbol{P}_1-\boldsymbol{P}_2\|}\right) \tag{1}$$

where $(\boldsymbol{T}-\boldsymbol{P}_1)\cdot(\boldsymbol{P}_2-\boldsymbol{P}_1)=\sum_{i=1}^{n}(t_i-p_{1i})\cdot(p_{2i}-p_{1i})$, $\|\boldsymbol{T}-\boldsymbol{P}\|=\sqrt{\sum_{i=1}^{n}(t_i-p_i)^2}$ and $\|\boldsymbol{P}_2-\boldsymbol{P}_1\|=\sqrt{\sum_{i=1}^{n}(p_{1i}-p_{2i})^2}$. p_{1i}, p_{2i} and t_i are the values of $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{T}$ in the ith dimension, respectively. The semantics of the offspring $\boldsymbol{O}$ is corresponding to the base of the perpendicular dropped from $\boldsymbol{T}$ to the relative semantics $\boldsymbol{P}_2-\boldsymbol{P}_1$. In the first case (as shown in Fig. 1(a)), when α and β are both smaller than 90°, the offspring $\boldsymbol{O}$ (represented by the green point in the figure) stands on the segment of $\boldsymbol{P}_2-\boldsymbol{P}_1$. In the other two cases, where either α or β is larger than 90°, the offspring stands along the segments on the $\boldsymbol{P}_1$ or $\boldsymbol{P}_2$ side. Now, a key step is to obtain point $\boldsymbol{O}$ on the line $\boldsymbol{P}_1\boldsymbol{P}_2$.

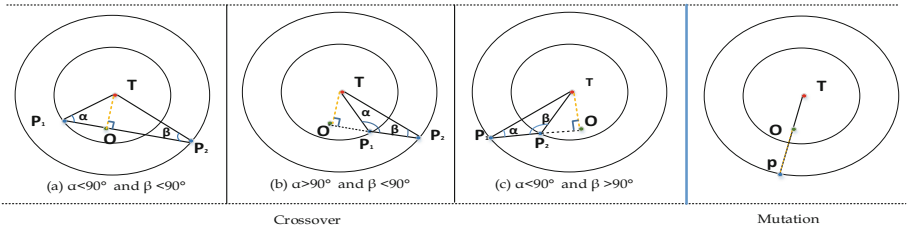

Fig. 1. PC and RSM (Color figure online)

In the scenario of obtaining a point on the line defined by two given points, the parametric equation, which is the most versatile equation to define a line in an n dimensional space, is a good choice to express a line. Then once the distance from $\boldsymbol{O}$ to one of the given points is calculated, it is easy to obtain $\boldsymbol{O}$. Specifically,

suppose L is the line given by two points $\boldsymbol{P}$ and $\boldsymbol{Q}$ in an n dimensional space. A particular point in line L is given in Eq. (2).

$$\boldsymbol{O} = \boldsymbol{P} + k \cdot (\boldsymbol{Q} - \boldsymbol{P}) \tag{2}$$

where $\boldsymbol{Q} - \boldsymbol{P}$ gives the direction of L, the elements of which are defined as $\{q_1 - p_1, q_2 - p_2, \ldots, q_n - p_n\}$. $k = \|\boldsymbol{O} - \boldsymbol{P}\|/\|\boldsymbol{Q} - \boldsymbol{P}\|$ is a real number parameter, which stands for the relative distance between $\boldsymbol{O}$ and $\boldsymbol{P}$. When $0 < k < 1$, $\boldsymbol{O}$ is a point on the segment between $\boldsymbol{P}$ and $\boldsymbol{Q}$. Further, if $k < 0$, $\boldsymbol{O}$ is outside the segment on the $\boldsymbol{P}$ side, while if $k > 1$, $\boldsymbol{O}$ is outside on the $\boldsymbol{Q}$ side. The procedure of obtaining the semantics of $\boldsymbol{O}$ is showing in Algorithm 1.

Algorithm 1. Obtaining the Desired Semantics in Perpendicular Crossover

Input : Target semantics $\boldsymbol{T}$, and the semantics of the two parents $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$
Output: The desired semantics of the offspring $\boldsymbol{O}$
Obtain the $\boldsymbol{P}_2 - \boldsymbol{P}_1$, $\|\boldsymbol{P}_2 - \boldsymbol{P}_1\|$, $\|\boldsymbol{T} - \boldsymbol{P}_1\|$ and $\|\boldsymbol{T} - \boldsymbol{P}_2\|$;
Calculate the angle α and β according to Equation (1);
if $\alpha < 90$ *and* $\beta < 90$ **then**
  $\|\boldsymbol{O} - \boldsymbol{P}_1\| = \|\boldsymbol{T} - \boldsymbol{P}_1\| \cdot cos(\alpha)$;
  $\boldsymbol{O} = \boldsymbol{P}_1 + \|\boldsymbol{O} - \boldsymbol{P}_1\|/\|\boldsymbol{P}_2 - \boldsymbol{P}_1\| \cdot (\boldsymbol{P}_2 - \boldsymbol{P}_1)$;
else
  if $\alpha > 90$ **then**
    $\|\boldsymbol{P}_1 - \boldsymbol{O}\| = \|\boldsymbol{T} - \boldsymbol{P}_1\| \cdot cos(180 - \alpha)$;
    $\boldsymbol{O} = \boldsymbol{P}_1 - \|\boldsymbol{P}_1 - \boldsymbol{O}\|/\|\boldsymbol{P}_2 - \boldsymbol{P}_1\| \cdot (\boldsymbol{P}_2 - \boldsymbol{P}_1)$;
  else
    if $\beta > 90$ **then**
      $\|\boldsymbol{O} - \boldsymbol{P}_2\| = \|\boldsymbol{T} - \boldsymbol{P}_2\| \cdot cos(180 - \beta)$;
      $\boldsymbol{O} = \boldsymbol{P}_2 + \|\boldsymbol{O} - \boldsymbol{P}_2\|/\|\boldsymbol{P}_2 - \boldsymbol{P}_1\| \cdot (\boldsymbol{P}_2 - \boldsymbol{P}_1)$;

3.2 Random Segment Mutation

RDO treats the target semantics as the only desired semantics for all the offspring. To utilise the target semantics in a better way and make the desired semantics to be more achievable, we propose the *random segment mutation* (RSM). RSM makes a small but important change to RDO, which is to utilises the target semantics in an implicit way. The rationale for this change is twofold. First RSM aims to maintain the semantic diversity of the population by assigning vary desired semantics to offspring. Rapid loss of semantic diversity has been considered as a major cause for the premature convergence of GP. Thus, maintaining a high semantic diversity is important for GP to escape from local optima. Secondly, RSM intends to approximate the target gradually. When tackling real-world data containing noise, the property of less greedy to the target semantics is expected to help GP to avoid the issue of overfitting.

In RSM, the desired semantics of the offspring individual stands on the segment of the parent and the target point in the semantic space. As shown in the last column in Fig. 1, given a parent $\boldsymbol{P}$, RSM firstly needs to find the segment between the target semantics $\boldsymbol{T}$ and $\boldsymbol{P}$. Then a random point is obtained along

this segment, which is treated as the desired semantics of the offspring $\boldsymbol{O}$. Then $\boldsymbol{O}$ is obtained according to $\boldsymbol{O} = \boldsymbol{P} + k \cdot (\boldsymbol{T} - \boldsymbol{P})$, $k \in (0, 1)$.

3.3 Fulfilling the Semantic Requirements

Both PC and RSM rely on semantic backpropagation to fulfil the semantic requirements of the offspring individuals.

Figure 2 gives an example of semantic backpropagation. Given a randomly selected node in a GP individual, semantic backpropagation decomposes the individual into two parts: the prefix expressed by the subtree rooted at the selected node and the suffix corresponding to the rest of the individual. Accordingly, the desired semantics of the GP individual is achieved by the combination of the fixed semantics of the suffix and a new semantics of the prefix. This new semantics is the semantic difference between the target and the fixed suffix. Specifically, to calculate the desired semantics of the prefix, semantic backpropagation starts from the desired semantics of the whole individual (like D_1 shown in Fig. 2), then backpropagates through a chain of nodes from the root to the selected node, i.e., in order to obtain D_3, which is the desired semantics of the selected subtree, the semantics of its parent node, which is referred to D_2, is needed beforehand. The same requirement is propagated to the root node $D1$. For the given example, the backpropagation chain is from D_1 to D_2, then to D_3. In addition, the inverted function operator is applied, which is the invert of the original operators. For example, in Fig. 2, the output of the parent node ("+") of the selected node ("2") is equal to $S_2 + D_3$, then accordingly $D_3 = D_2 - S_2$, thus "−" is the inverted operator of "+", and vice versa.

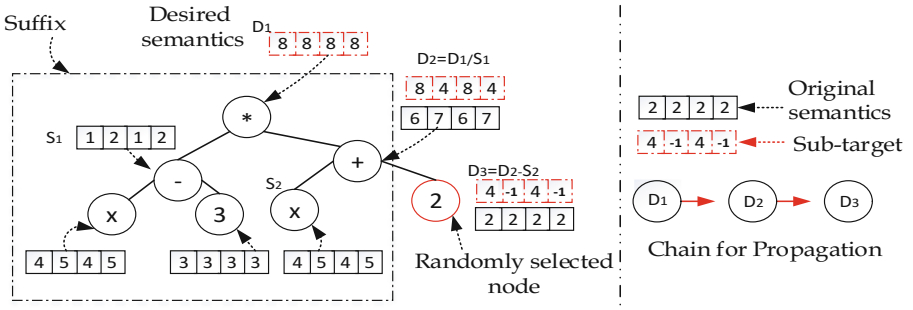

Fig. 2. One example for semantic backpropagation.

After obtaining the desired semantics, the prefix will be replaced by a new subtree from a predefined semantic library. Specifically, an exhaustive search will perform on the library to find a subtree with the desired semantics. If a subtree with the exact desired semantics does not exist, the search will return a subtree with the closest Euclidean distance to the desired semantics among the

library. The diversity of the semantic library is important for the success of the exhaustive search, thus the semantic library is updated and filled with subtrees taken from individuals at every generation. These subtrees have unique semantics. When two subtrees have the same semantics, the small one is preferred.

Compared with GSGP and RDO, the proposed NGSGP algorithm has the following properties. First NGSGP utilises both the target semantics and the semantics of the parent(s) to have a precise control over the semantics of the offspring individuals. Meanwhile, in contrast with driven by a uniform target semantics, the precise control has been conducted in a way of fulfilling various semantic requirements, which are more achievable and maintain more diverse semantics of the GP population. Second instead of relying on a linear combination of parent(s) and randomly generated trees, NGSGP fulfils the semantic requirement by semantic backpropagation and selects appropriate subtrees from a semantic library. With these significant improvements, NGSGP is expected to not only achieve a good learning ability but also generalise well.

4 Experiment Design

To investigate the effectiveness of NGSGP, it is compared with RDO. In addition, to have a more precise investigation, the performance of GP with only one of the two operators, i.e. GP with only PC (denoted as PC) and GP with only RSM (denoted as RSM), are also examined. Standard GP is used as a baseline for comparison. All the GP methods are implemented under the GP framework provided by Distributed Evolutionary Algorithms in Python (DEAP) [3]. The five GP methods are tested on six commonly used synthetic datasets, which are taken from previous work on GSGP [6,11]. The target functions and sampling strategies are shown in Table 1. In addition, we are also interested in testing the proposed method on real-world datasets [1,12] shown in Table 2, which have not been widely used in GSGP.

Parameter settings for all the GP methods are summarised in Table 3. Most of these parameters are common settings in GP [5,7]. In the four GSGP methods (RDO, RSM, PC and NGSGP), the Euclidean metric is adopted to measure the distance between two points. For an easy comparison, at every generation, the root mean square error (RMSE) of the best-of-generation model on the training

Table 1. Synthetic datasets. (The training samples are drawn using regular intervals from the interval range, while the test samples are drawn randomly within the same interval. Each of them have 20 samples. The interval range is $[-1; 1]$ (except for Nguyen-7, which has a range of $[0; 2]$).)

Problem	Function	Problem	Function
Septic	$x^7 - 2x^6 + x^5 - x^4 + x^3 - 2x^2 + x$	Nonic	$\sum_{i=1} 9x^i$
Nguyen-7	$log(x+1) + log(x^2+1)$	R1	$(x+1)^3/(x^2 - x + 1)$
R2	$(x^5 - 3x^3 + 1)/(x^2 + 1)$	R3	$(x^6 + x^5)/(x^4 + x^3 + x^2 + x + 1)$

Table 2. Real-world problems

Name	# Features	#Total instances	#Training instances	#Test instances
LD50	626	234	163	71
DLBCL	7399	240	160	80

Table 3. Parameter settings for GP

Parameter	Value	Parameter	Value
Population size	256	Generations	100
Crossover rate	0.9	Mutation rate	0.1
Elitism	1	Maximum tree depth	17
Initialisation	Ramped half-and-half	Initialisation depth	2–6
Function set	$+$, $-$, $*$, %protected, *exp*, *log*, *sin*, *cos*	Fitness function	Euclidean distance, RMSE

set and its corresponding test error are recorded. 100 independent runs has been conducted for each method on each dataset.

5 Results and Discussions

This section compares and discusses the experiment results obtained by the five GP methods. The comparison will be presented in terms of training performance, generalisation ability and the size of the evolved programs by the GP methods. The non-parametric statistical significance test—Wilcoxon test, which has a significance level of 0.05, is conducted to compare the training RMSEs and test RMSEs of the 100 best-of-run models.

5.1 Overall Results

The mean and standard deviation of RMSEs achieved by the best-of-run programs on the training sets and the corresponding results on the test sets are shown in Tables 4 and 5, respectively. The best (i.e. smallest) values among the five methods are marked in bold.

As shown on Table 4, the four GSGP methods all have much smaller training RMSEs than standard GP on most of the datasets. On seven of the eight training sets (except for LD50), NGSGP obtains smaller mean RMSEs than RDO. On LD50, NGSGP has a higher training error than RDO. The difference between NGSGP and RDO on the eight datasets are all significant. RSM and PC both have better training performance than RDO on five of the eight problems.

Table 5 shows clearly that the four GSGP methods can generalise better than standard GP on all the synthetic benchmarks. The advantages are all significant. However, it is not the case on the two real-world datasets. On LD50 and DLBCL,

Table 4. Training errors of the 100 best-of-run programs

Problem	GP	RDO	RSM	PC	NGSGP
	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
Septic	0.07 ± 0.04	0.01 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	**2.1E−3 ± 1.7E−3**
Nonic	0.06 ± 0.03	0.01 ± 0.02	4E−3 ± 2.3E−3	0.01 ± 0.01	**2.9E−3 ± 1.8E−3**
Nguyen7	0.01 ± 0.01	2.1E−3 ± 3E−3	3.6E−4 ± 3.1E−4	9.5E−4 ± 1.2E−3	**3.3E**−4 ± **3.3E**−4
R1	0.05 ± 0.03	0.01 ± 0.01	3.2E−3 ± 2.8E−3	3.6E−3 ± 3.2E−3	**2.6E−3 ± 2.3E−3**
R2	0.05 ± 0.03	3E−3 ± 3.1E−3	1.4E−3 ± 8.9E−4	1.5E−3 ± 1.3E−3	**1.3E−3 ± 8.8E−4**
R3	0.01 ± 3.9E−3	2.1E−3 ± 0.01	**8E−4 ± 6.1E−4**	1.9E−3 ± 4.2E−3	**8.8E−4 ± 1.4E−3**
LD50	1950.94 ± 67.66	**1692.2 ± 317.1**	1888.59 ± 108.26	1952.63 ± 51.68	1844.52 ± 88.5
DLBCL	0.65 ± 0.02	0.63 ± 0.07	0.65 ± 0.04	0.62 ± 0.03	**0.57 ± 0.08**

Table 5. Corresponding test errors of the 100 best-of-run programs

Problem	GP	RDO	RSM	PC	NGSGP
	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
Septic	0.07 ± 0.05	0.03 ± 0.01	0.05 ± 0.07	0.04 ± 0.07	**0.02 ± 0.03**
Nonic	0.06 ± 0.04	0.02 ± 0.02	0.06 ± 0.18	0.02 ± 0.02	**0.01 ± 0.02**
Nguyen7	0.01 ± 0.01	0.01 ± 0.02	**6.8E−4 ± 5.4E−4**	3.2E−3 ± 0.01	0.01 ± 0.03
R1	0.06 ± 0.04	0.01 ± 0.02	0.01 ± 0.02	0.01 ± 0.01	**3.2E−4 ± 7E−3**
R2	0.05 ± 0.03	0.02 ± 0.05	0.01 ± 0.02	0.03 ± 0.08	**0.01 ± 0.03**
R3	0.01 ± 0.01	0.02 ± 0.05	0.24 ± 1.43	4.8E−3 ± 0.01	**4.4E−3 ± 0.01**
LD50	2007.5 ± 67.1	4354.9 ± 9236.7	1996.8 ± 79.4	2020.7 ± 83.3	**1987.88 ± 87.5**
DLBCL	0.7 ± 0.04	0.71 ± 0.04	0.7 ± 0.04	0.69 ± 0.05	**0.62 ± 0.07**

while RDO generalises much worse than standard GP, GP equipped with the two new operators (solely or together) can generalise better than standard GP. Particularly on LD50, where all the GSGP methods can have much better training performance than standard GP, only RSM and NGSGP can generalise slightly better than standard GP. On DLBCL, while RDO still can not generalise better than standard GP, the other three methods can generalise significantly better than standard GP. On these two real-world datasets, PC, RSM and NGSGP all have significantly better generalisation performance than RDO. The ranking of the generalisation performance of the five GP methods on the synthetic datasets is consistent with that on the training set, which indicates the geometric semantic methods are resistant to overfitting on the comparatively simple problems. On the real-world datasets, which might contain noise, RDO is prone to overfitting and NGSGP has a much better generalisation ability than RDO.

5.2 Behavior Analysis—Training/Evolutionary Process

The evolutionary plots on the training sets, which are drawn using the median of training errors of the 100 best-of-generation individuals, are shown in Fig. 3. On five of the seven training sets, except for Septic and R3, NGSGP outperforms

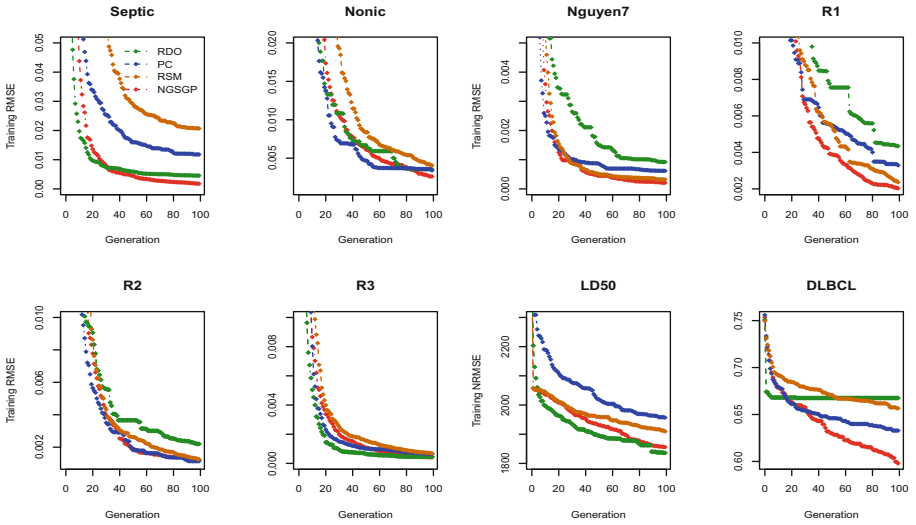

Fig. 3. Training/Evolutionary process on the training sets.

RDO in an early stage and this advantage becomes larger over generations. This might be due to the less greedy nature of NGSGP for achieving the target semantics. NGSGP is driven by vary target semantics, which are a kind of intermediate target under the theoretical requirement of GSGP. These intermediate targets are different from individual to individual, which can help maintain the semantic diversity of the population better than utilising only one target. Higher semantic diversity will lead to a better exploration ability of GP and has a positive effect on enhancing the effectiveness of the evolutionary search. Furthermore, these intermediate targets are generally closer to the parents than the target semantics. Searching from a semantic library, which consists of semantic uniquely subtrees from the population, the semantics of these intermediate targets are easier to match. Following these reliable and achievable intermediate targets, NGSGP is able to get even closer to the target semantics than RDO gradually and smoothly.

Comparing PC and RSM with RDO, it can be observed that both new geometric operators generally have a positive effect on enhancing the training performance of RDO to some extent. Combining the new geometric crossover, which is relatively stable, with the new geometric mutation operator, which brings randomness, can lead to even better learning performance.

5.3 Analysis of Generalisation Behaviour

The generalisation performance of the evolved models on unseen data is a key metric to measure the effectiveness of the proposed GP method. Figure 4 shows the evolutionary plots on the generalisation performance. The overall pattern on the test sets is similar to that on the training sets. Clearly, NGSGP is the winner

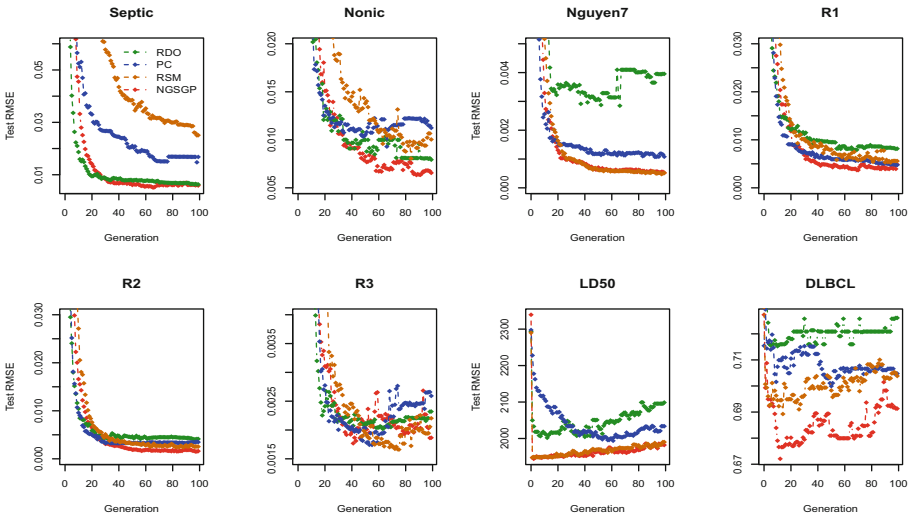

Fig. 4. Results on the test sets against the evolution.

on the generalisation performance among the four geometric semantic methods. On seven of the eight benchmarks (except for Septic), NGSGP generalises significantly better than RDO. On Septic, no significant difference can be found between the two methods. On Nonic, R1 and R2, where RDO generalises well, NGSGP can generalise even better than RDO. On the other four datasets, where RDO overfits the training data, NGSGP can either overfit less (i.e. on R3, LD50 and DLBCL) or resistant to overfitting (on Nguyen7). On these datasets, before the 40th generation on LD50 and the 20th generation on DLBCL, both method do not overfit, NGSGP still has much smaller generalisation error than RDO. On LD50, RDO achieves the best training performance but the worst generalisation performance, which indicates it overfits to the training set the severest among the four methods.

Compared PC and RSM with RDO, on five of the eight benchmarks, except for Septic, Nonic and R3, both PC and RSM have better generalisation performance than RDO. Clearly, the advantage of NGSGP over RDO on the generalisation performance is brought by both the perpendicular crossover and random segment mutation. Compared with RDO, which uses the target semantics in an explicit way, the two semantic operators in NGSGP intend to approximate the target gradually. When tackling the real-world data containing noise, the property of NGSGP, i.e. being less greedy, to the target semantics will surely lead it to less overfit the training set, thus generalise better.

5.4 Comparison on the Program Size

The average and minimum sizes of the best-of-run individuals are presented in Table 6. Not surprisingly, the modes evolved by GSGP methods have a much

Table 6. Program size of the evolved models

Problem	GP	RDO	NGSGP	Problem	GP	RDO	NGSGP
	Mean(Min)				Mean(Min)		
Septic	215.04 (111)	1180.32 (39)	702.16 (239)	Nonic	184.12 (59)	1050.28 (31)	1016.4 (471)
Nguyen7	115.8 (51)	2152.63 (15)	1111.0 (571)	R1	168.92 (67)	1715.9 (29)	1670.16 (325)
R2	144.32 (47)	1261.88 (39)	1008.16 (151)	R3	200.76 (63)	1141.43 (17)	1085.0 (485)
LD50	179.76 (39)	947.2 (23)	78.04 (7)	DLBCL	60.2 (3)	205.56 (3)	46.0 (5)

larger size than standard GP on most of the datasets, which is 5–15 times larger. However, it is interesting to see that, on the two real-world datasets, NGSGP has the smallest average program size, which is even smaller than standard GP. These smaller models generally have better interpretability, which is a desired property in many real-world problems. Comparing the program size of RDO and NGSGP, it is clear that on all the eight benchmarks, NGSGP has a smaller average program size but much higher minimum size than RDO. This indicates that the programs evolved by the different runs in NGSGP are more stable than RDO in size, and NGSGP is less likely to generate oversize programs.

6 Conclusions and Future Work

This work aimed to address the limitations of GSGP and RDO, which are either a loosely or a uniformly semantic control over the offspring individuals and can potentially lead to premature convergence and overfitting in GP. The goal has been achieved by developing two new geometric semantic operators named perpendicular crossover and random segment mutation to conduct a precise semantic control under the theoretical requirement of GSGP. The behavioural analysis of the evolved models releases that NGSGP, which employs the two operators, outperforms RDO at an early stage of learning process and this advantage increases along with generations. More importantly, NGSGP has a much better generalisation ability than RDO shown by obtaining lower testing errors and eliminating/releasing overfitting. In addition, NGSGP also evolves more compact models than RDO.

For future work, we are interested in speeding up the new operators, as the overall computational cost of the new geometric operators is much higher than the canonical form of GP operators. The major expense is in searching the semantic library and evaluating the large individuals. Thus, we would like to improve the algorithm on semantic library search and introduce bloat free mechanism to the new operators in the near future. In addition, this work only compares with one kind of geometric operators, a more comprehensive comparison between the new operators and other variants of geometric operators on improving the performance of GP will be performed in the future.

References

1. Archetti, F., Lanzeni, S., Messina, E., Vanneschi, L.: Genetic programming for computational pharmacokinetics in drug discovery and development. Genetic Program. Evol. Mach. **8**(4), 413–432 (2007)
2. Chen, Q., Xue, B., Mei, Y., Zhang, M.: Geometric semantic crossover with an angle-aware mating scheme in genetic programming for symbolic regression. In: McDermott, J., Castelli, M., Sekanina, L., Haasdijk, E., García-Sánchez, P. (eds.) EuroGP 2017. LNCS, vol. 10196, pp. 229–245. Springer, Cham (2017). doi:10.1007/978-3-319-55696-3_15
3. Fortin, F.A., Rainville, F.M.D., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: evolutionary algorithms made easy. J. Mach. Learn. Res. **13**, 2171–2175 (2012)
4. Gonçalves, I., Silva, S., Fonseca, C.M.: On the generalization ability of geometric semantic genetic programming. In: Machado, P., Heywood, M.I., McDermott, J., Castelli, M., García-Sánchez, P., Burelli, P., Risi, S., Sim, K. (eds.) EuroGP 2015. LNCS, vol. 9025, pp. 41–52. Springer, Cham (2015). doi:10.1007/978-3-319-16501-1_4
5. Koza, J.R.: Genetic Programming: on the Programming of Computers by Means of Natural Selection, vol. 1. MIT Press, Cambridge (1992)
6. Krawiec, K., O'Reilly, U.M.: Behavioral programming: a broader and more detailed take on semantic GP. In: Proceedings of the 16th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO), pp. 935–942 (2014)
7. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric semantic genetic programming. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012. LNCS, vol. 7491, pp. 21–31. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32937-1_3
8. Nguyen, Q.U., Nguyen, X.H., O'Neill, M.: Semantic aware crossover for genetic programming: the case for real-valued function regression. In: Vanneschi, L., Gustafson, S., Moraglio, A., De Falco, I., Ebner, M. (eds.) EuroGP 2009. LNCS, vol. 5481, pp. 292–302. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01181-8_25
9. Pawlak, T.P.: Geometric semantic genetic programming is overkill. In: Heywood, M.I., McDermott, J., Castelli, M., Costa, E., Sim, K. (eds.) EuroGP 2016. LNCS, vol. 9594, pp. 246–260. Springer, Cham (2016). doi:10.1007/978-3-319-30668-1_16
10. Pawlak, T.P., Krawiec, K.: Guarantees of progress for geometric semantic genetic programming. In: Semantic Methods in Genetic Programming, Ljubljana, Slovenia, vol. 13 (2014)
11. Pawlak, T.P., Wieloch, B., Krawiec, K.: Semantic backpropagation for designing search operators in genetic programming. IEEE Trans. Evol. Comput. **19**(3), 326–340 (2015)
12. Rosenwald, A., Wright, G., Chan, W.C., Connors, J.M., Campo, E., Fisher, R.I., Gascoyne, R.D., Muller-Hermelink, H.K., Smeland, E.B., Giltnane, J.M., et al.: The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. N. Engl. J. Med. **346**(25), 1937–1947 (2002)