Geometric Semantic Crossover with an Angle-aware Mating Scheme in Genetic Programming for Symbolic Regression

Qi Chen, Bing Xue, Yi Mei, and Mengjie Zhang

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

Abstract. Recent research shows that incorporating semantic knowledge into the genetic programming (GP) evolutionary process can improve its performance. This work proposes an angle-aware mating scheme for geometric semantic crossover in GP for symbolic regression. The angle-awareness guides the crossover operating on parents which have a large angle between their relative semantics to the target semantics. The proposed idea of angle-awareness has been incorporated into one state-of-the-art geometric crossover, the locally geometric semantic crossover. The experimental results show that, compared with locally geometric semantic crossover and the regular GP crossover, the locally geometric crossover with angle-awareness not only has a significantly better learning performance but also has a notable generalisation gain on unseen test data. Further analysis has been conducted to see the difference between the angle distribution of crossovers with and without angle-awareness, which confirms that the angle-awareness changes the original distribution of angles by decreasing the number of parents with zero degree while increasing their counterparts with large angles, leading to better performance.

1 Introduction

In recent years, semantic genetic programming (GP) [11, 18], which incorporates the semantic knowledge in the evolutionary process to improve the efficacy of search, attracts increasing attention and becomes a hot research topic in GP [6]. One popular form of semantic methods, geometric semantic GP (GSGP), has been proposed recently [12]. GSGP searches directly in the semantic space of GP individuals. The geometric crossover and mutation operators generate offspring that lies within the bounds defined by the semantics of the parent(s) in the semantic space. The fitness landscape that these geometric operators explore has a conic shape, which contains no local optimal and is easier to search. In previous research, GSGP presents a notable learning gain over standard GP [19, 17]. For the generalisation improvement, GSGP shows some positive effect. However, while the geometric mutation is remarked to be critical in bringing the generalisation benefit, the geometric crossover is criticised to have a weak effect on promoting generalisation for some regression tasks [5]. One possible reason is that of the target output on the test set is beyond the scope of the convex combination of the parents for crossover [13] in the test semantic space. Another possible reason is that crossover might operate on similar parents standing in a compact volume of the semantic space, which leads to generating offspring having duplicate semantics with their parents. In this case, the population has difficulty to converge to the target output, no matter the target semantic is in or out of the covered range. Thus, the offspring produced by the geometric crossover is difficult to generalise well. Therefore, in this work, we are interested in improving the geometric crossover by addressing this issue.

The overall goal of this work is to propose a new angle-aware mating scheme to select for geometric semantic crossover to improve the *generalisation* of GP for symbolic regression. An important property of the geometric semantic crossover operator is that it generates offspring that stands in the segment defined by the two parent points in the semantic space. Therefore, the quality of the offspring is highly dependent on the positions of the two parents in the semantic space. However, such impact of the parents on the effectiveness of geometric semantic crossover has been overlooked. In this paper, we propose a new mating scheme to geometric crossover to make it operats on parents that are not only good at fitness but also have large *angle* in terms of their relative positions to the target point in the semantic space. Our goal is to study the effect of the newly proposed mating scheme to geometric crossover operator. Specific research objectives are as follows:

– to investigate whether the geometric crossover with angle-awareness can improve the learning performance of GSGP,
– to study whether the geometric crossover with angle-awareness can improve the generalisation ability of GSGP, and
– to investigate how the geometric crossover with angle-awareness influences the computational cost and the program size of the models evolved by GSGP.

2 Background

This section introduces geometric semantic GP in brief and reviews some state-of-the-art related work on geometric crossovers.

2.1 Geometric Semantic GP

Before introducing geometric semantic GP, a formal concept of individual semantics in GP needs to be given. A widely used definition of semantics in regression domain is as follows: the semantics of a GP individual is a vector, the elements of which are the outputs produced by the individual corresponding to the given instances. Accordingly, the semantics of an individual can be interpreted as a point in a n dimension semantic space, where n is the number of elements in the vector [9, 11].

Geometric semantic GP is a relatively new branch in semantic GP. It searches directly in the semantic space, which is a notable difference from the other non-direct semantic methods, such as [2] and [16]. Searching in the semantic space is accomplished by its exact geometric semantic crossover and mutation. The definition of the geometric semantic crossover (GSX) is given below [12]:

Definition 1. *Geometric Semantic Crossover: Given two parent individuals p_1 and p_2, a geometric semantic crossover is an operator that generates offspring $p'_i(i \in (1, 2))$ having semantics $s(p'_i)$ in the segment between the semantics of their parents, i.e., $\|s(p_1), s(p_2)\| = \|s(p_1), s(p'_i)\| + \|s(p'_i), s(p_2)\|$*

Another important concept related to geometric crossover is the convex hull. It is a concept from geometry, which is the set of all convex combinations of a set of points. In geometric semantic GP, the convex hull can be viewed as the widest volume that the offspring generated by geometric crossover can cover.

Various geometric crossover operators [12, 9] have been developed to satisfy the semantic constraint in Definition 1 in different ways. Locally geometric semantic crossover [9] (LGX) is a typical one with good performance.

2.2 Locally Geometric Semantic Crossover

Krawiec et al. [8] develop the locally geometric semantic crossover(LGX), which attempts to produce offspring that satisfies the semantic constraint in Definition (1) at the subtree level. A library L consisting of a set of small size trees needs to be generated before applying the LGX. Trees in the library L have a maximum depth limitation M, and generally, each tree has unique semantics. Then given two parents p_1 and p_2, LGX tries to find their homologous subtree, which is the largest structurally common subtree of the two parents. Two corresponding crossover points are selected within the homologous subtree. Then the two subtrees p_{c1} and p_{c2} that root in these two crossover points are replaced by a tree p_r selected from L. p_r is randomly selected from a number of K programs which are the closest neighbour to the semantics of midpoint of p_{c1} and p_{c2}, i.e., $S(p_r) \approx \frac{(S(p_{c1})+S(p_{c2}))}{2}$, where $S(p)$ represents the semantics of p. The advantage of LGX is that it can satisfy the semantic constraint by retrieving small subtrees in the library but without bringing exponential growth in the size of the offspring. The application shows that LGX brings notable improvement to the performance of GP [9].

2.3 Related Work

GSX performs a convex combination of the two parents. It generates offspring that lies in the segment defined by the parent points in the semantic space. Consequently, under Euclidean metric, the offspring can not be worse than the worse parent.

Moraglio et al. [12] develop the exact geometric crossover which is a transformations on the solution space that can satisfy the semantic constraint at the level of whole tree, i.e., $P_{xo} = P_1 \cdot F_r + P_2 \cdot (1 - F_r)$ where P_i are parents for crossover, F_r is a random real functions that outputs values in the range of $[0, 1]$. Despite the potential success of exact geometric crossover, it is criticised by leading the size of offspring to an exponential growth. Vanneschi et al. [17] propose an implementation of geometric operators to overcome the drawback of the unmanageable size of offspring. They aim to obtain the semantic of the offspring without generating the new generation in structure. The new implementation makes GSX can be applied to real-world high-dimensional problems. However, the evolved models are still hard to show and interpret.

During the evolutionary process if the target output is outside the convex hull, then surely GSX is impossible to find the optimal solution. Oliveira et al. [13] proposed a dispersion operator for GSX to address this issue. They proposed a geometric dispersion operator to move individuals to less dense areas around the target output in the semantic space. By spreading the population, the new operator increases the probability that the convex hull of the population will cover the target. Significant improvement is achieved on the learning and generalisation performance on most of the examined datasets.

However, even if the convex hull of the population covers the target, GSX may still fail and the population may still converge to a small volume far away from the target if the parents of GSX are not properly selected. It is known that due to the convexity of the squared Euclidean distance metric, the offspring cannot be worse than both parents. However, at the level of the whole population, there is still a high probability that this progress does not have much effect on guiding the population toward the target output in the semantic space, especially when a large proportion of crossovers perform on very similar parents in the semantic space. In this work, we propose a new mating scheme to geometric semantic crossover to prevent this trend and promote the exploration ability of the GSX.

3 Angle-aware Geometric Semantic Crossover (AGSX)

In this work, tree based GP is employed, and we propose a new angle-aware mating scheme for Geometric Semantic Crossover (AGSX). This section describes the main idea, the detailed process, the characteristics of AGSX, and the fitness function of the GP algorithm.

3.1 Main Idea

How the crossover points spread in the semantic space is critical to the performance of GSGP. A better convergence to the target point can be achieved if the convex combinations cover a larger volume when the convex hull is given. AGSX should be applied to the parents that the *target output* is around the intermediate region of their semantics. Given that the semantics of the generated offspring tend to lie in the segment of the semantics of the parents as well, AGSX is expected to generate offspring that is close to the target output. To promote the convex combinations to cover a larger volume, the two parents should have a larger distance in the semantics space.

The semantic distance between the parents can be used here, but it often leads to a quick loss of semantic diversity in the population and then results in a premature solution. Therefore, we utilise the angle between the relative semantics of the parents to the target output to measure their distance in the semantic space. Specifically, suppose the target output is $\boldsymbol{T}$, and the semantics of the two parents are $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, the angle α between the relative semantics of the two parents to the target output is defined as follows:

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

where $(\boldsymbol{S}_1 - \boldsymbol{T}) \cdot (\boldsymbol{S}_2 - \boldsymbol{T}) = \sum_{i=1}^{n}(s_{1i} - t_i) \cdot (s_{2i} - t_i)$ and $\|\boldsymbol{S} - \boldsymbol{T}\| = \sqrt{\sum_{i=1}^{n}(s_i - t_i)^2}$. i stands for the ith dimension in the $n-$dimensional semantic space. s_{1i}, s_{2i}, and t_i are the values of S_1, S_2 and $\boldsymbol{T}$ in the ith dimension, respectively.

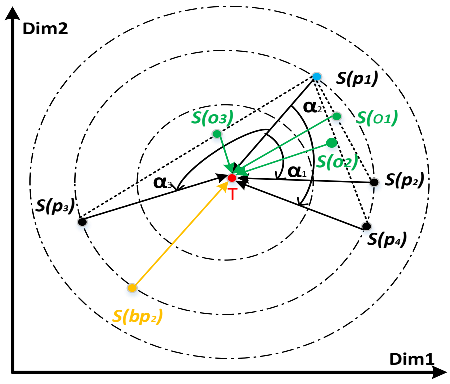

Fig. 1. AGSX in two Dimension Euclidean Semantic Space.

Fig.1 illustrates the mechanism of AGSX in a two-dimensional Euclidean space, which can be scaled to any n-dimensional space. Each point represents the semantics of one individual in GP. As shown in the figure, there are four individuals p_1, p_2, p_3 and p_4, which can be selected as the parents of AGSX. Assume p_1 (in blue colour) has been selected as one parent and the mate, i.e. the other parent, needs to be selected from p_2, p_3 and p_4 to perform AGSX. α_1, α_2, and α_3 show the angles in the three pairs of parents, i.e. $\langle p_1, p_2\rangle$, $\langle p_1, p_4\rangle$ and $\langle p_1, p_3\rangle$, respectively. The three green points, $S(o_1)$, $S(o_2)$, and $S(o_3)$, show the three corresponding offspring of the three pairs of parents, and the green lines indicates their distances to the target point. It can be seen from the figure that the pair of parents $\langle p_1, p_3\rangle$ has a larger angle, i.e. α_3, and the generated offspring $S(o_3)$ is closer to the target output. In the ideal case where the yellow point $S(bp_2)$ is the second parent, the generated offspring is very likely to be the target point. In other words, if the parents have a larger angle between their relative semantics to the target output, the generated offspring tends to be closer

Algorithm 1: Pseudo-code of AGSX

Input : $WaitingSet[i_1, i_2, ..., i_m]$ consists of m individuals on which will perform crossover. T is the target semantics point.
Output: The generated offspring

1 **while** *$WaitingSet$ is not empty* **do**
2     $p_1 =$ is the first individual in $WaitingSet$;
3   remove $p_1 =$ from $WaitingSet$;
4     $maxangle = 0$; /* *i.e. the maximum angle that has been found* */
5   *top* is an empty list;
6   **for** *each individual p in $WaitingSet$* **do**
7   calculate the *angle* between the relative semantics of $S(p_1)$, $S(p)$ to T according to Equation (1);
8   **if** *angle is equal to 180, i.e. p is the optimal mate for p_1* **then**
9   *top*=*p*;
10   **else**
11   **if** *angle is larger than maxangle* **then**
12                 $maxangle = angle$;
13   *top*=*p*;
14   **else**
15   **if** *angle is equal to the maxangle* **then**
16   add p to *top*;
17   randomly select an individual, p_2, from *top*;
18   perform geometric crossover on p_1 and p_2;
19   remove p_1 and p_2 from $WaitingSet$.

to the target output. Therefore, we need to select parents with a large angle in their relative semantics to the target output.

To achieve this, we develop a new mating scheme to select parents with a large angle in their relative semantics to the target output. First, a list of candidate parents called the $WaitingSet$ is generated by repetitively applying a selection operator (e.g. tournament selection) to the current population. The size of $WaitingSet$ is determined by the population size N and the crossover rate R_X, i.e. $|waitingset| = N \cdot R_X$. Then, the parents for each AGSX operation are selected from $WaitingSet$ without replacement so that the angles between the relative semantics of the selected parents can be maximised. The detailed process of AGSX is given in Section 3.2.

3.2 The AGSX Process

The pseudo-code of AGSX is shown in Algorithm 1. The procedure of finding the mate having the largest relative angle for a given parent p_1 is shown in Lines $3 - 18$. The angles are calculated according to Equation (1), as shown in Line 6.

3.3 Main Characteristics of AGSX

Compared with GSX, AGSX has three major advantages. Firstly, AGSX employs an angle-aware scheme, which is flexible and independent of the crossover process itself and can be applied to any form of the geometric semantic operator. Secondly, AGSX operates on distinct individuals in the semantic space. This way, the generated offspring are less likely to be identical with their parents in the semantic space. That is, AGSX can reduce semantic duplicates. Thirdly, by operating on parents with large angles between their relative semantics to the target output, AGSX is more likely to generate offspring that are closer to the target output.

3.4 Fitness Function of the algorithm

The Minkowski metric $L_k(X,Y) = \sqrt[k]{\sum_{i=1}^{n} |x_i - y_i|^k}$, which calculates the distance between two points, is used to evaluate the performance of individuals. Typically, two kinds of Minkowski distance between the individual and the target could be used. They are Manhattan distance (L_1 by setting $k = 1$ in $L_k(X,Y)$) and Euclidean distance (L_2). According to previous research [1], Euclidean distance is a good choice and is used in this work. The definition is as follows:

$$D(X,T) = \sqrt{\sum_{i=1}^{n} |x_i - t_i|^2} \tag{2}$$

where X is the semantics of the individual and T is the target semantics.

4 Experiments Setup

To investigate the effect of AGSX in improving the performance of GP, a GP method implements the angle-awareness into one recent approximate geometric crossover, the locally geometric semantic crossover has been proposed and named GPALGX. A comparison between GPALGX and GP with locally geometric semantic crossover (GPLGX) has been conducted. We have a brief introduction of LGX in Section 2.2. For more details of the GPLGX, readers are referred to [9]. Standard GP is used as a baseline for comparison as well. All the compared methods are implemented under the GP framework provided by Distributed Evolutionary Algorithms in Python (DEAP)[4].

4.1 Benchmark Problems

Six commonly used symbolic regression problems are used to examine the performance of the three GP methods. The details of the target functions and the sampling strategy of the training data and the test data are shown in Table 1. The first two problems are the recommended benchmarks in [10]. The middle three are used in [14]. The last one is from [3] which is a modified version of the commonly used Quartic function. These six datasets are used since they have been widely used in recent research on geometric semantic GP [14, 15]. The notation *rnd[a,b]* denotes that the variable is randomly sampled from the interval *[a, b]*, while the notation *mesh([start:step:stop])* defines the set is sampled using regular intervals. Since we are more interested in the generalisation ability of the proposed crossover operator, the test points are drawn from ranges which are slightly wider than that of the training points.

Table 1. Target Functions and Sampling Strategies.

Benchmark	Target Function	Training	Test
Keijzer1	$0.3xsin(2\pi x)$	20 ponits	1000 ponits
Koza2	$(x^5 - 2x^3 + X)$	x=mesh((-1:0.1:1])	x=Rnd[-1.1,1.1]
Nonic	$\sum_{i=1}^{9} x^i$	20 ponits	1000 ponits
R1	$(x+1)^3/(x^2 - x + 1)$	x=mesh((-2:0.2:2])	x=Rnd[-2.2,2.2]
R2	$(x^5 - 3x^3 + 1)/(x^2 + 1)$		
Mod_quartic	$4x^4 + 3x^3 + 2x^2 + x$		

Table 2. Parameter Settings

Parameter	Values	Parameter	Values
Population Size	512	Generations	100
Crossover Rate	0.9	Reproduction Rate	0.1
#Elitism	10	Maximum Tree Depth	17
Initialisation	Ramped-Half&Half	Initial Depth	range(2,6)
Maximum tree depth in Library-M	3	Neighbourhood Number-K	8
Function Set	$+, -, *,$ protected %, log, sin, cos, exp		
Fitness function	Root Mean Squared Error (RMSE) in standard GP Euclidean distance in GPLGX and GPALGX		

4.2 Parameter Settings

The parameter settings can be found in Table 2. For standard GP, the fitness function is different from that of GPLGX and GPALGX. Since the primary interest of this work is the comparison of the generalisation ability of the various crossover operators, all the three GP methods only have crossover operators. No mutation operator has taken apart. The values of the two key parameters M and K in implementing LGX, which represent for the maximum depth of the small size tree in the library and the number of the closest neighbouring trees respectively, are following the recommendation in [9].

Overall, the three GP methods are examined on six benchmarks. Each method has 100 independent runs performed on each benchmark problem.

5 Results and Discussions

The experiment results of GP, GPLGX and GPALGX are presented and discussed in this section. The results will be presented in terms of comparisons of RMSEs of the 100 best models on the training sets and their corresponding test RMSEs. The fitness values of models in GPLGX and GPALGX are calculated using Euclidean distance. However, for comparison purpose, the Root Mean Squared Error (RMSE) of models are also recorded. The major comparison is presented between GPLGX and GPALGX. Thus, we also compare the angle distribution of GPLGX and GPALGX. The computational time and program size are also discussed. The non-parametric Wilcoxon test is used to evaluate the statistical significance of the difference on the RMSEs on both the training sets and the test sets. The significance level is set to be 0.05.

5.1 Overall Results

The results on the six benchmarks are shown in Fig.2, which displays the distribution of RMSEs of the 100 best-of-the-run individuals on the training sets and

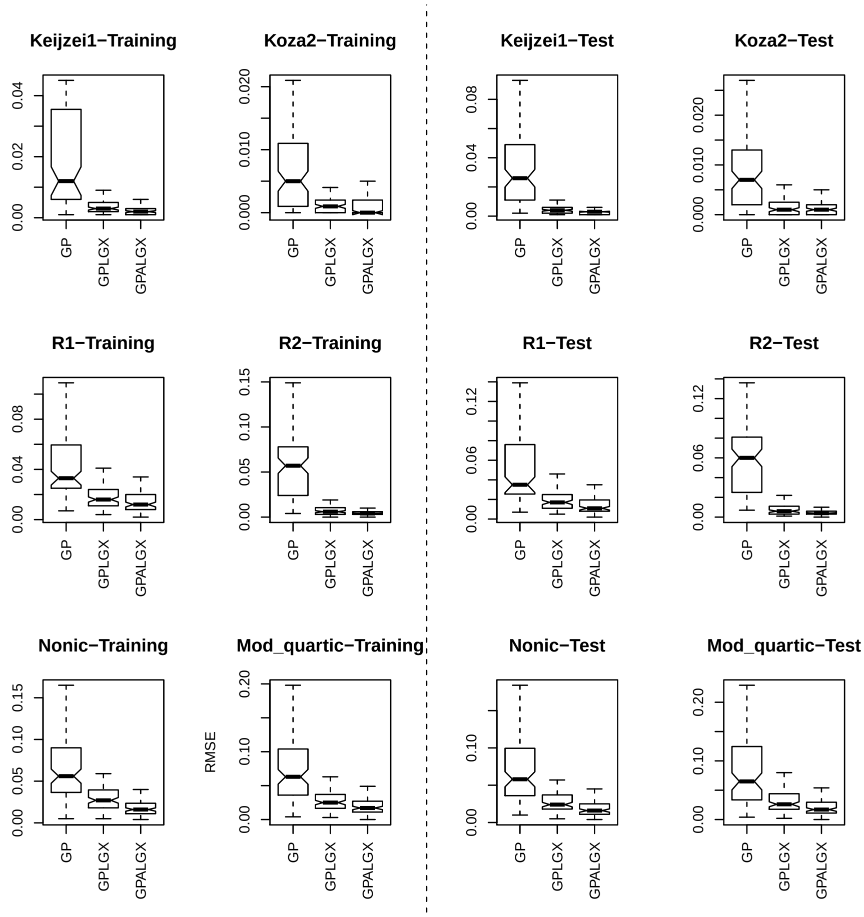

Fig. 2. Distribution of Training RMSE and the Corresponding Test RMSE of the 100 best-of-run individuals.

the test sets. As it shows, on all the six benchmarks, GPALGX has the best training performance among the three GP methods. For every benchmark, GPALGX has a better training performance than GPLGX and GP, by the smaller median value of the 100 best training errors and the much shorter boxplot. This indicates the training performance of GPALGX is superior to the other two methods in a notable and stable way. The results of statistical significance test confirm that the advantage of GPALGX over GPLGX and GP are all significant on the six training sets.

The overall pattern on the test sets is the same as the training set, which is GPALGX achieves the best generalisation performance on all the benchmarks. On each benchmark, the pattern in the distribution of the 100 test errors is also the same as that on the training set. GPALGX has the shortest boxplot which indicates the more consist generalisation error among the 100 runs. GPLGX has a larger distribution than GPALGX, which is still much shorter than standard GP. A significant difference can be found on the six benchmarks between

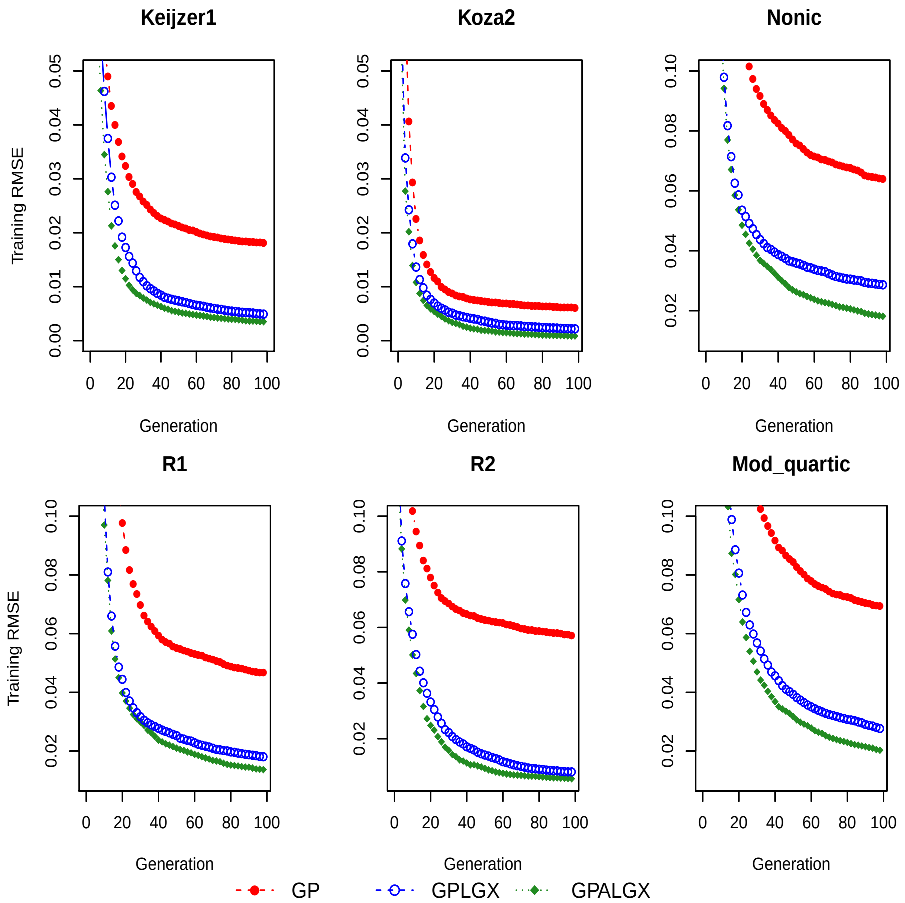

Fig. 3. Evolutionary plot on the training set.

GPALGX, GPLGX and GP, i.e. GPALGX generalises significantly better than GPLGX, while the two geometric methods are significantly superior to GP. The generalisation advance of LGX and ALGX over standard crossover is consistent with the previous research on LGX. In [9], the generalisation gain of LGX has been investigated and confirmed. This generalisation gain has been justified to own to the library generating process which helps reduce the semantic duplicates. The further generalisation gain of ALGX over LGX might lie in the fact that the angle-awareness helps extend the segment connecting each pair of parents for crossover, thus can reduce the semantic duplicates more intensively, and enhance the exploration ability of LGX to find better generalised solutions.

5.2 Analysis on the Learning Performance

The evolutionary plots on the training sets are provided in Fig.3. To analysis the effect of ALGX on improving the learning performance of GP. These evolutionary plots are drawn using the mean RMSEs of the best-of-generation individuals over the 100 runs.

As expected, GP with ALGX achieves the best learning performance. It is superior to the other two GP methods from the early stage of the evolutionary process, which is generally within the first ten generations. The advances of the two geometric GP methods over standard GP on the learning performance confirms that searching in the geometric space is generally much easier, since the semantic space is unimodal and has no local optimal. The comparison between the two geometric GP methods indicates ALGX is able to generate offspring which is much closer to the target point in the semantic space from the very beginning of the searching process. On all the six benchmarks, GPALGX not only has significantly smaller training RMSEs but also has higher average fitness gain from generation to generation. On $Koza_2$ and R_2, the two geometric GP methods can find models which are excellent approximations (the RMSE of which is smaller than 0.001), and GPALGX converges to the target semantics much faster than GPLGX. This might be because ALGX performs crossover on individuals having larger angles than GPLX, thus produces offspring closer to the target in the semantic space in an effective way. In this way, it will increase the exploitation ability of LGX and find the target more quickly. For the other four benchmarks, although none of the two geometric GP methods finds the optimal solution, on three of them, the increasingly larger difference between the two methods along with the increase of generations indicates the improvement that ALGX brings is increasing over generations. One of the possible reasons is that, over generations, compared with LGX, ALGX will perform on individuals having smaller relative semantic distance with target output in larger angle pairs, which will generate even better offspring.

5.3 Analysis of the Evolution of Generalisation Performance

Compared with the training performance, we are more interested in the generalisation performance of GP with ALGX. Therefore, further analysis on the generalisation ability of GPALGX and a more comprehensive comparison between the generalisation of the three methods is carried out. In Fig.4, the evolutionary plots on the test sets are reported along generations for each benchmark on the three GP methods. These plots are based on the mean RMSEs of the corresponding test errors obtained by the best-of-generation models over 100 runs. (On each generation, the test performance of the best-of-generation model obtained from the training data has been recorded, but the test sets never take apart in the evolutionary training process)

The evolution plots confirm that GPALGX has a better generalisation gain than the other two methods on all the test sets of the considered benchmarks, which is notable. On all the six benchmarks, GPALGX can generalise quite well, while its two counterparts suffer from the overfitting problems on some datasets. On the six problems, GP overfits the training sets. The test RMSEs increase after decreasing over a small number of generations at the beginning. Also, GP generally has a very fluctuate mean RMSE on most test sets. It indicates that

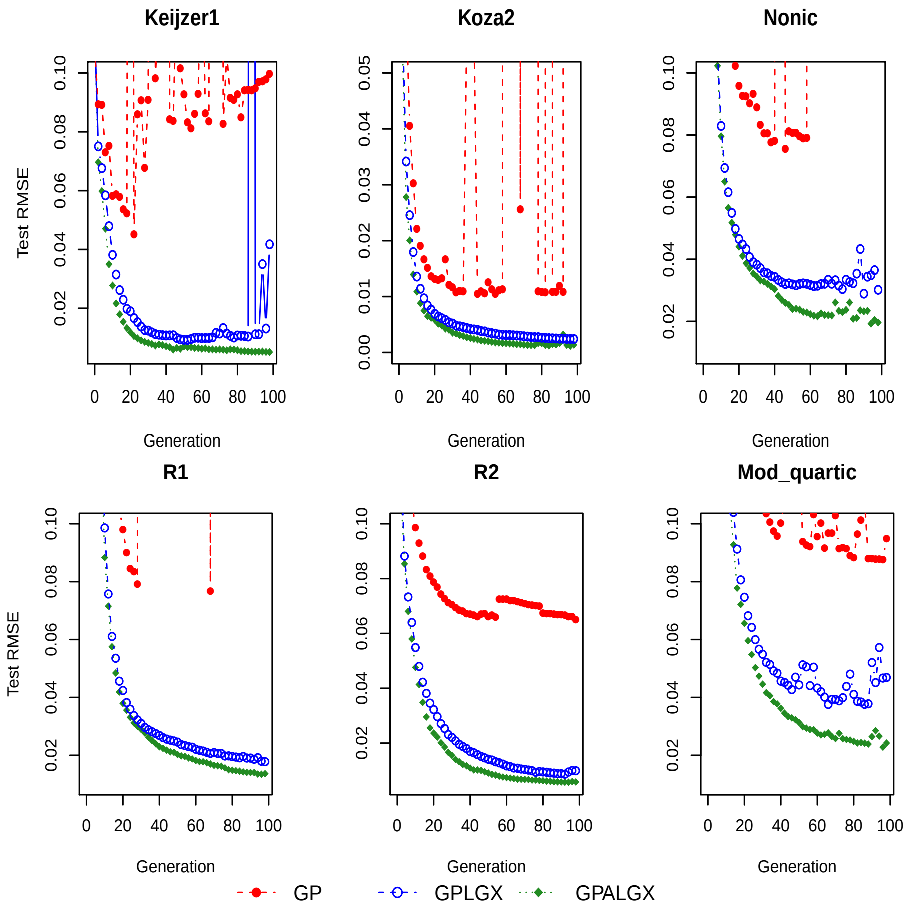

Fig. 4. Evolutionary plot on the test set.

training the models on a small number of points (20 points), while testing the models on a larger number of points (1000 points) distributed over a slightly larger region is difficult for GP. GPLGX can generalise much better than GP but still encounters overfitting problems on three benchmarks, i.e., on Keijzer1, Nonic and Mod_quartic. On these three datasets, GPLGX has an increasing RMSEs on the last ten generations. On other three datasets, GPLGX generalises well. Overall, GPALGX generalises better than GPLGX and GP, shown as obtaining lower generalisation errors and having a smaller difference with its training errors.

The excellent generalisation ability of geometric crossover can be explained by the fact that the geometric properties of this operator are independent of the data to which the individual is exposed. Specifically, the offspring produced by LGX and ALGX lie (approximately) in the segments of parents also hold in the semantic space of the test data. Since this property holds for every set of data, no matter where the test data distributes in, the fitness of the offspring can never be worse than the worse parent. In the population level, this property can not

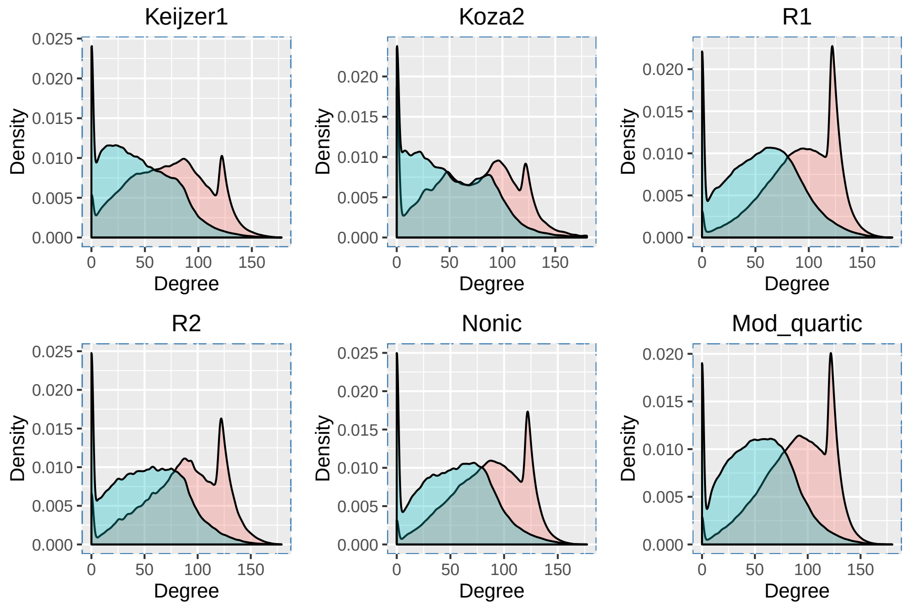

Fig. 5. Distribution of Angles of the Parents for Crossover.

guarantee to improve the test error on every generation for every benchmark (in fact, we can find on the last several generations, LGX has an increasing test error on three benchmarks), but during the process it surely has a high probability of generalisation gain on the test set and only a few times of getting worse generalisation over generations. That is why LGX has the ability to control overfitting and generalise better than the regular crossover.

This interpretation has a direct relationship on why ALGX is less likely to overfitting and generalises better than LGX on the test sets. In other words, ALGX puts more effect on selecting parents which consequently limits the probability of having not good enough parents to crossover, so it can lead to a large number of offspring with better generalisation at the population level. AlGX shares the same benefit with LGX, which is the geometric property leading to offspring never worse than parents on the test set. More importantly, the angle-awareness in ALGX makes the large angle between the parents also holds in the test semantic space. This leads to a higher probability to have a good process on the test data at the population level. The details of the angle distribution will be discussed in the next subsection.

5.4 Analysis of the Angles

To investigate and confirm the influence of ALGX to the distribution of angles of the parents, the angles between each pair of parents which performs crossover have been recorded in both GPLGX and GPALGX. In Fig.5, the density plots show the distribution of the angles in the two GP methods. The green one is for

GPLGX, and the one in orange colour is for GPALGX. The density plots are based on around $2,250,000 (\approx 225 * 100 * 100)$ values of angles in each method. While the x-axis represents the degree of angles, the y-axis is the percentage of the corresponding degree in the $2,250,000$ recorded values.

From Fig.5 we can see that the distribution of angles of parents in GPALGX is different from GPLGX in two aspects. On the one hand, it has a much smaller number of angles which are zero degrees. While in GPLGX, the peak of the distribution is at the zero degrees on all the six datasets, in GPALGX, the angle-awareness can stop the pairs of individuals with zero degrees from performing crossover. The direct consequence of this trend is the elimination of semantic duplicates, and the higher possibility of generating better offspring.

On the other hand, GPALGX has a larger number of larger angles. Most of its angles are over 90 degrees. The peak of the distribution is all around 120 degrees on the six datasets, specifically on the last four datasets. At the first several generations, the larger angles with similar (or the same) vectors will lead to better offspring, which is represented by a shorter vector. At the last several generations, larger angles along with the shorter vectors will lead to a population of even better offspring. This can explain why the distance between the training error and test error of GPLGX and GPALGX increases over generations on most of the benchmarks.

Table 3. Computational Time and Program Size.

Benchmarks	Method	Time(in second)	Program size (Node)	Significant Test
		Mean±Std	Mean±Std	(on program size)
Keijzer1	GPLGX	523±83.8	90.52±28.72	=
	GPALGX	1400±317	87.74±23.85	
Koza2	GPLGX	560±105	72.66±29.93	+
	GPALGX	1330±232	93.82±24.18	
R1	GPLGX	523±84.5	88.82±27.07	=
	GPALGX	1250±253	92.18±31.71	
R2	GPLGX	524±83.9	89.62±27.41	=
	GPALGX	1250±218	83.8±28.6	
Nonic	GPLGX	571±112	84.5±32.62	+
	GPALGX	1250±212	101.5±39.33	
Mod_quartic	GPLGX	554±105	99.98±38.25	=
	GPALGX	1420±369	105.38±37.29	

5.5 Comparison on Computational Time and Program Size

The comparison between the computational cost and program size of the evolved models have been performed between the two geometric methods. Table 3 shows the computational time in terms of the average training time for one GP run in each benchmark. The average program size represented by the number of nodes in the best_of_run models in each benchmark is also provided. The statistical significance results on the program size are also listed in the table. While "-" means the program size of the evolved model in GPALGX is significantly smaller than GPLGX, "+" indicates the significant larger program size of GPALGX. "=" represents no significant difference can be found.

As shown in Table 3, on all the six benchmarks, the average computational time for one run in GPALGX is much higher than GPLGX, which is generally around two times as that of GPLGX. This is not surprising since GPALGX needs more effort to identify the most suitable pairs of parents during the crossover process. The longer computational time can be decreased by reducing the population size in GPALGX. Moreover, the computational time for each GP run in both methods is short, which is hundreds to two thousand second. Thus, the additional computational cost of GPALGX is affordable.

In term of the program size, on four benchmarks, i.e., Keijzer1, R1, R2, and Mod_quartic, the two methods have a similar program size and no significant difference has been found. On the other two datasets, ALGX produces offspring which are significantly larger than LGX. However, it is interesting to note that these much more complex models in term of program size still can generalise better than its simpler counterparts on the two test sets, while the simpler model of GPLGX slightly overfits on the Nonic problem.

6 Conclusions and Future work

This work proposes an angle-aware mating scheme to select parents for geometric semantic crossover, which employs the angle between the relative semantics of the parents to the target output to choose parents. The proposed ALGX performs on parents having a large angle so that the segment connecting the parents is close to the target output. Thus, ALGX can generate offspring that have better performance. To investigate and confirm the efficiency of the proposed ALGX, we run GP employed ALGX on six widely used symbolic regression benchmark problems and compare its performance with GPLGX and GP. The experimental results confirm that GPALGX has not only better training performance but also significantly better generalisation ability than GPLGX and GP on all the examined benchmarks.

Despite the improvement ALGX brings on performance, it generally is computational more expensive than GPLGX. In the future, we aim to improve the angle detecting process. Instead of using the deterministic method to calculate the angle between two individuals iteratively, we can introduce some heuristic search methods to find the best parent pairs to reduce the computational cost. We also would like to explore a further application of ALGX, for example, to introduce the angle-awareness to other forms of geometric crossover, such as the exact geometric semantic crossover [12] and Approximate geometric crossover [7], to investigate their effectiveness. In addition, this work involves solely crossover and no mutation. The effect of angle-awareness to mutation and using both crossover and mutation are also interesting topics to work on.

References

1. Albinati, J., Pappa, G.L., Otero, F.E., Oliveira, L.O.V.: The effect of distinct geometric semantic crossover operators in regression problems. In: European Conference on Genetic Programming. pp. 3–15. Springer (2015)
2. Beadle, L., Johnson, C.G.: Semantically driven crossover in genetic programming. In: IEEE Congress on Evolutionary Computation. pp. 111–116 (2008)

3. Burks, A.R., Punch, W.F.: An efficient structural diversity technique for genetic programming. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 991–998. ACM (2015)
4. Fortin, F.A., Rainville, F.M.D., Gardner, M.A., Parizeau, M., Gagné, C.: Deap: Evolutionary algorithms made easy. Journal of Machine Learning Research 13(Jul), 2171–2175 (2012)
5. Gonçalves, I., Silva, S., Fonseca, C.M.: On the generalization ability of geometric semantic genetic programming. In: European Conference on Genetic Programming. pp. 41–52. Springer (2015)
6. Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection, vol. 1. MIT press (1992)
7. Krawiec, K., Lichocki, P.: Approximating geometric crossover in semantic space. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. pp. 987–994. ACM (2009)
8. Krawiec, K., Pawlak, T.: Locally geometric semantic crossover. In: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation. pp. 1487–1488. ACM (2012)
9. Krawiec, K., Pawlak, T.: Locally geometric semantic crossover: a study on the roles of semantics and homology in recombination operators. Genetic Programming and Evolvable Machines 14(1), 31–63 (2013)
10. McDermott, J., White, D.R., Luke, S., Manzoni, L., Castelli, M., Vanneschi, L., Jaskowski, W., Krawiec, K., Harper, R., De Jong, K., et al.: Genetic programming needs better benchmarks. In: Proceedings of the 14th annual conference on Genetic and evolutionary computation. pp. 791–798. ACM (2012)
11. McPhee, N.F., Ohs, B., Hutchison, T.: Semantic building blocks in genetic programming. In: European Conference on Genetic Programming. pp. 134–145. Springer (2008)
12. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric semantic genetic programming. In: International Conference on Parallel Problem Solving from Nature. pp. 21–31. Springer (2012)
13. Oliveira, L.O.V., Otero, F.E., Pappa, G.L.: A dispersion operator for geometric semantic genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 773–780
14. Pawlak, T.P., Wieloch, B., Krawiec, K.: Review and comparative analysis of geometric semantic crossovers. Genetic Programming and Evolvable Machines 16(3), 351–386 (2015)
15. Szubert, M., Kodali, A., Ganguly, S., Das, K., Bongard, J.C.: Reducing antagonism between behavioral diversity and fitness in semantic genetic programming. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference. pp. 797–804. ACM (2016)
16. Uy, N.Q., Hien, N.T., Hoai, N.X., ONeill, M.: Improving the generalisation ability of genetic programming with semantic similarity based crossover. In: European Conference on Genetic Programming. pp. 184–195. Springer (2010)
17. Vanneschi, L., Castelli, M., Manzoni, L., Silva, S.: A new implementation of geometric semantic gp and its application to problems in pharmacokinetics. In: European Conference on Genetic Programming. pp. 205–216. Springer (2013)
18. Vanneschi, L., Castelli, M., Silva, S.: A survey of semantic methods in genetic programming. Genetic Programming and Evolvable Machines 15(2), 195–214 (2014)
19. Vanneschi, L., Silva, S., Castelli, M., Manzoni, L.: Geometric semantic genetic programming for real life applications. In: Genetic Programming Theory and Practice XI, pp. 191–209. Springer (2014)