

Semantic Genetic Programming Operators Based on Projections in the Phenotype Space

Mario Graff, Eric Sadit Tellez, Elio Villaseñor, Sabino Miranda-Jiménez

INFOTEC - Centro de Investigación e Innovación en
Tecnologías de la Información y Comunicación,
Cátedras CONACyT,
México

{mario.graff,eric.tellez,
elio.villasenor,sabino.miranda}@infotec.com.mx

Abstract. In the Genetic Programming (GP) community there has been a great interest in developing semantic genetic operators. These type of operators use information of the phenotype to create offspring. The most recent approaches of semantic GP include the GP framework based on the alignment of error space, the geometric semantic genetic operators, and backpropagation genetic operators. Our contribution proposes two semantic operators based on projections in the phenotype space. The proposed operators have the characteristic, by construction, that the offspring's fitness is as at least as good as the fitness of the best parent; using as fitness the euclidean distance. The semantic operators proposed increment the learning capabilities of GP. These operators are compared against a traditional GP and Geometric Semantic GP in the Human oral bioavailability regression problem and 13 classification problems. The results show that a GP system with our novel semantic operators has the best performance in the training phase in all the problems tested.

Keywords: semantic crossover, symbolic regression, geometric semantic genetic programming.

1 Introduction

Genetic Programming (GP) is an evolutionary algorithm that has received a lot of attention lately due to its success in solving hard real-world problems [11]. One of the most promising ideas to improve the performance of GP is to develop semantic genetic operators. The difference between semantic operators and traditional genetic operators is that the former uses the information of the phenotype to generate offspring, and, the later generates an offspring using only the syntax of the individual.

There has been a number of different proposals that fit in the field of semantic operators and systems. Recently, Stefano Ruberto *et al.* [15] introduced the concept of error vector and error space. Briefly, the idea is that the optimal

solution can be constructed from two individuals that are aligned in the vector space. Consequently, the objective of the search procedure can be changed from finding the closest individual to the desired behavior (i.e., the origin in the error space) to finding aligned individuals. This novel type of GP has shown success in solving two complex real-life applications in drug discovery, namely, human oral availability and median lethal dose.

Semantic genetic operators gather the information from the phenotype space by either sampling it or using some properties of the space. Among the operators that sample the space, it is found the work done by Blickle *et al.* [1], who proposed to select as crossing points only those nodes that have an impact on the fitness function. The work done by Nguyen *et al.* [8,16] produces offspring that are semantically different from its parents; this difference is measured by evaluating the individuals in a set of random inputs.

Backpropagation [13] can be used in subtree crossover, subtree mutation, and point mutation to propagate the error —i.e., the information of the phenotype— to the crossover or mutation point. The information propagated can be used to guide the genetic operator, that is, it can be used to either generate a tree that reduces the error propagated or to select, from a second parent, the point that reduces the most the propagated error. Backpropagation was used by Pawlak *et al* [9] to find a subtree that analytically reduces the error produced in the crossover point. Graff *et al.* (see [4,5]) used backpropagation to compute the partial derivative error in the crossover and mutation point. Then this information was used to either select the best crossover point in the second parent or, in the case of point mutation, to select the best function from the function set.

The semantic operators most related to this contribution are the Geometric Semantic Crossover and Mutation proposed by Moraglio *et al.* [7] and the novel implementation of them developed by Vanneschi *et al.* [18] that allows the algorithm to be executed with traditional GP parameters. For a recent review in semantic operators we referred the interest reader to [19].

This contribution presents a novel semantic crossover and mutation, PrXO and PrMut, respectively. These operators have as their principal feature that the fitness (using as fitness the euclidean distance) of the offspring is at least as good as the fitness of the best parent. The offspring produced by PrXO and PrMut is the orthogonal projection of the target behaviour in the parents' plane in the phenotype space. That is, traditionally, the phenotype space is $\mathbb{R}^{|\mathbb{T}|}$ (see [10]), where \mathbb{T} is the training set. Therefore, each individual in the population is a point in $\mathbb{R}^{|\mathbb{T}|}$ and the objective is to find a target behaviour $\mathbf{t} \in \mathbb{R}^{|\mathbb{T}|}$. Under these circumstance the offspring produced by PrXO and PrMut is the orthogonal projection of \mathbf{t} in the plane generated by the linear combinations of the parents. As the result show the use of PrXO and PrMut in a steady state GP system, namely hereafter PrGP, outperform, in the training set, a steady state Geometric Semantic GP and a steady state GP in two classes of problems: Human oral bioavailability problem and 13 classification problems. This is a clear indication of the learning capabilities of PrGP; however, more work is needed to

identify the overfitting.

The rest of the paper is organized as follows. Section 2 presents the semantic genetic operators based on projections in the phenotype space. The problems and the parameters setting used to test the novel semantic operators are described in Section 3. Section 4 presents our results and compared the operators against the Semantic Geometric GP and traditional GP. Finally, the conclusions and some possible directions for future work are given in Section 5.

2 Projection Semantic Genetic Operators

PrGP is a supervised learning algorithm that learns the instances of a training set \mathbb{T} formed by $n \in \mathbb{N}$ pairs of inputs and outputs, i.e., $\mathbb{T} = \{(x_i, y_i) | i = 1 \dots n\}$ where x_i represents the i -th input, and y_i is the associated output. The objective is to find a function f such that $\forall (x,y) \in \mathbb{T} f(x) = y$ and that could be evaluated in any element x of the input space.

In general, it is not possible to find a function f that learns perfectly \mathbb{T} , consequently, one tries to find a function f that minimize an error function e.g. sum of squared errors $\sum_{(x,y) \in \mathbb{T}} (y - f(x))^2$.

Let us consider a fixed order in \mathbb{T} to define $\mathbf{t} = (y_1, \dots, y_n) \in \mathbb{R}^n$, namely the target vector, which contains all the outputs in \mathbb{T} . Let $s(p, x)$ be a function that evaluates the individual p on input x . Using the order in \mathbb{T} , it is possible to define $\mathbf{p} = (s(p, x_1), \dots, s(p, x_n))$ that contains the evaluation of individual p in all the inputs x of the training set. In this scenario the fitness (using as fitness function the sum of squared error) of individual p can be computed as $\|\mathbf{t} - \mathbf{p}\|$ which is the euclidean norm.

A source of inspiration for PrXO and PrMut is the geometric semantic genetic operators proposed by Moraglio *et al.* [7] which are defined as follows:

Geometric Semantic Crossover Let p_1 and p_2 be the first and second parent the offspring produce by these parent is $o = p_1 r + p_2(1 - r)$, where r is a random function or a constant in the range $[0, 1]$. The output of individual o at input x is computed as $s(o, x) = s(p_1, x)s(r, x) + s(p_2, x)(1 - s(r, x))$.

Geometric Semantic Mutation Let p_1 be the individual to be mutated and r_1 and r_2 two random functions, then the offspring produced is $o = p_1 + m(r_1 - r_2)$ where m is the mutation step. Vanneschi *et al.* [18] proposed a variant of this operation, the difference is that r_1 and r_2 are normalized using a sigmoid.

Let us assume that r in the geometric semantic crossover is a constant, then the offspring is just a linear combination of the parents. This combination lies in the line segment intersecting the parents.

This characteristic influenced the development of PrXO and PrMut. That is, it is reasonable to investigate that whether there is a better linear combination between the two parents, and, effects that this modifications has in the converge of the algorithm.

Let us rewrite the geometric crossover and mutation with the constraint that r is a constant. The geometric crossover is computed as $o = \alpha p_1 + \beta p_2$ where $\alpha = r$ and $\beta = 1 - \alpha$; and the geometric mutation is $o = \alpha p_1 + \beta(r_1 - r_2)$ where $\alpha = 1$ and $\beta = m$.

Using this notation it is evident that the geometric operators constraints the values of α and β . Operators PrXO and PrMut removes these restrictions but are not geometric as defined by Moraglio *et al.* [7].

PrXO Let p_1 and p_2 be the first and second parent, then the offspring o is computed as $o = \alpha p_1 + \beta p_2$ where α and β are calculated solving the following equation $A(\alpha, \beta)' = \mathbf{t}'$ where $A = (\mathbf{p}'_1, \mathbf{p}'_2)$, $\mathbf{p}_i = (s(p_i, x_1), \dots, s(p_i, x_n))$ is the evaluation of parent i in all the inputs, and \mathbf{t} is the target vector.

PrMut Let p_1 be the parent to be mutated then the offspring is $o = \alpha p_1 + \beta(r_1 - r_2)$ where r_1 and r_2 are two random individuals and α and β are obtained by solving $A(\alpha, \beta)' = \mathbf{t}'$ where $A = (\mathbf{p}'_1, (\mathbf{r}_1 \text{ and } \mathbf{r}_2)')$.

By construction the offspring o generated by PrXO and PrMut is the projection of \mathbf{t} on the plane produced by the linear combination of \mathbf{p}_1 and \mathbf{p}_2 in the case of crossover; or \mathbf{p}_1 and $(\mathbf{r}_1 - \mathbf{r}_2)$ otherwise. Given that \mathbf{o} is the projection of \mathbf{t} then it is the closest point to \mathbf{t} in the plane, consequently if the fitness function is the euclidean distance then the offspring has at least the fitness of the best parent. In the implementation it was decided to discard any individual that has an equal fitness to its parents.

3 Problems and Parameters Settings

PrXO and PrMut are tested on two classes of problems, symbolic regression and classification. The symbolic regression problem is the *Human oral bioavailability*¹ problem previously used in [18]. The problem is to find a function that accurately predicts the percentage of the initial orally submitted drug dose that effectively reaches the system blood circulation after passing through the liver. It consists of 241 inputs, namely molecular descriptor that describe the drug and 359 instances.

The classification problems used are described in Table 1.² These problems have been used traditionally as benchmarks for classification methods, these present different characteristics in terms of number input features, size of the training set, and test set, among other.

PrXO and PrMut are compared on the two previous classes of problem against the geometric semantic crossover and mutation; and the traditional

¹ This dataset is available at <http://gpbenchmarks.org>

² These datasets are available in <http://theoval.cmp.uea.ac.uk/matlab/benchmarks/>

Table 1. Data sets used in the comparison.

Data Set	Input Features	Training set instances	Test set instances
Banana	2	400	4900
Titanic	3	150	2051
Thyroid	5	140	75
Diabetes	8	468	300
Breast-Cancer	9	200	77
Flare-Solar	9	666	400
Heart	13	170	100
Ringnorm	20	400	7000
Twonorm	20	400	7000
German	20	700	300
Image	20	1300	1010
Waveform	21	400	4600
Splice	60	1000	2175

subtree crossover and mutation. In order to make the comparison as fair as possible, each the genetic operators were used in a steady state evolution with tournament selection (tournament size 2). The system with PrXO and PrMut is denominated PrGP (the source code is available for download at <https://...>) the system with the geometric genetic operators are identified as GSGP, and, the system with the traditional genetic operators is labeled GP. The rest of the parameters used in all the systems are shown on Table 2.

Table 2. Parameters used for PrGP, GSGP and GP on the symbolic regression problem (Human oral bioavailability) and the classification problems

Parameter	Symbolic Regression	Classification
Mutation depth	random $\in [1, 5]$	
Selection	Tournament of size 2	
Population Size	100	1000
Number of Generations	500	50
Function Set (\mathcal{F})	{+, -, ×, /}	{+, -, ×, /, · , exp, √, sin, cos, sigmoid, if, max, min, ln, (·) ² , argmax}
Crossover rate	50%	90%
Mutation rate	50%	10 %
Max length (only on GP)	1024	$\min(\frac{\lfloor \tau \rfloor}{2}, 256)$

As can be seen, the parameters used in GP are standard having been used previously e.g., [12] and [6]. Nonetheless, there are some parameters that deserve an explanation. The difference in the crossover rate and population size between the regression and classification problems is because GSGP obtained better performance in the bioavailability using these parameter and also these

parameters have been used previously by Vanneschi *et al.* [18]. It was decided to use a different function set \mathcal{F} for each class of problem based on the performance exhibit by GP on classification problem (see [17]) and that results presented in [18] with the bioavailability problem.

The function set \mathcal{F} used in the classification problems is formed by arithmetic functions, transcendental functions and max, min, if and argmax. These four later functions are implemented using arithmetic operators and exp as can be seen in Equations (1)-(4). The if function is a sort of conditional function that selects y or z depending on whether the value of x is 0 or 1, respectively. The argmax returns the index of the subtree that has the highest value.

$$\max(x, y) = \frac{x - y}{1 + e^{-100(x-y)}} + y \tag{1}$$

$$\min(x, y) = \frac{y - x}{1 + e^{-100(x-y)}} + x \tag{2}$$

$$\text{if}(x, y, z) = \frac{y - z}{1 + e^{-100x}} + z \tag{3}$$

$$\text{argmax}(x) = \sum_i \frac{e^{\beta x_i}}{\sum_j e^{\beta x_j}} i \tag{4}$$

Regarding the length of the individuals it is important to note that PrPG and GSGP do not have a maximum length. The maximum length impose to GP is 1024 for the bioavailability problem and for the classification problems it was used the function suggested in [17]. The maximum length was set as $\min(\frac{\lfloor T \rfloor}{2}, 256)$. This value was inspired by the degrees of freedom in a function whose parameters can be linearly identified. That is, in order to identify k parameters it is needed at least $k + 1$ points. Roughly, in a expression with n nodes at least $\frac{n}{2}$ of these nodes are operands and the other half are variables, so assuming that each variable has a coefficient to be identified one needs at least $\frac{n}{2}$ examples.

The last consideration is the procedure used to classify. Each classification problem was treated as a symbolic regression. In order to obtain a label from a continous value, the output of the individual was rounded and the output was limited to be in the range $[0, 1]$ given that all the datasets have only two classes. In addition to this, following the ideas presented on [17], an ensemble of k classifiers is used. That is, each system is initialized k times each of them is executed with different seeds, and for each of them the best individual is kept in a set. Then, the best individuals are used to predict the test set. The class of each object corresponds to the one that receives the major number of votes. Finnaly, the number of examples are balanced to have exactly the same instances for each class. This was performed removing the necessary examples in the training set until all the classes have the same number. In addition to this, the features are normalized to have zero mean and one standard deviation.

4 Results

Figure 1 presents the performance in terms of the root mean square error (RMSE) of the different systems, namely PrGP, GSGP, and GP on the bioavailability problem. This problem contains 359 instances from these instances 30 different training and test set were created using the following procedure. The training set was created by selecting 252 instances and the rest of the instances composed the test set. This process was repeated 30 times.

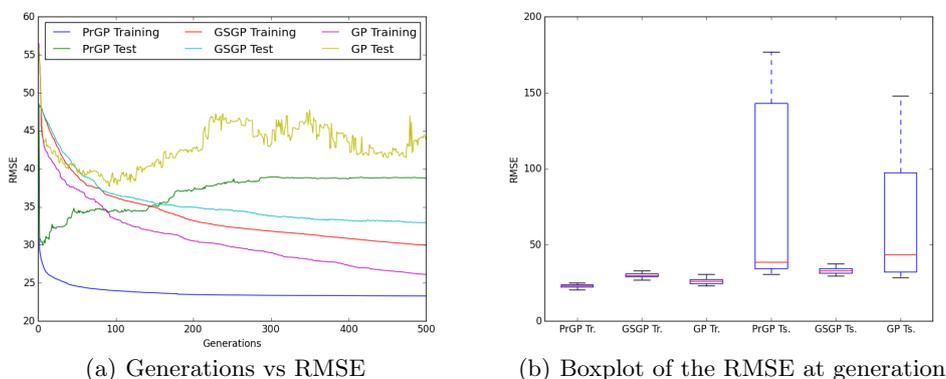


Fig. 1. Performance in terms of Root Mean Square Error (RMSE) in Human oral bioavailability problem on the training set (Tr.) and test set (Ts.). The boxplot presents the performance at generation 500.

Figure 1 (a) depicts the median of the performance (RMSE) through the generations of the different systems. In the training set it is observed that PrGP obtained the best performance reaching it at around generation 200. A particular characteristic of PrGP is that it learns faster than the other algorithms. GP obtained the second best performance and GSGP was the system with worst performance. The story in the test set is different, there GSGP is the system that got the best performance at the end of the run. Nonetheless, PrGP had the best performance before generation 150. This behaviour is an indication that PrGP presents overfitting, consequently, a procedure to prevent it is necessary.

Figure 1 (b) presents boxplots of the performance in the training and test set for the different systems. From left to right, the first three boxplot presents the performance in the training set and the last three correspond to the test set. The boxplots are created with the performance at generation 500. It is observed that PrGP had the best performance in the training set; and GSGP had the best performance in the test set. It depicts that PrGP and GP are unstable in the test set and some actions must be taken in order to prevent this behaviour.

It is important to note that GSGP in this problem does not present overfitting. This features has been noted by Vanneschi *et al.* [18] and recently by Goncalver *et al* [3]. The source of this feature is the modification performed by Vanneschi *et al.* [18] to original formulation of the geometric semantic mutation.

The classification problems consist in 13 different datasets all of these problems have two classes. The performance in all case is the balance error rate (BER). It was decided to split the datasets in those that contained less that 20 features and those with 20 or more features. Figure 2 presents the median performance through the generations in the training and test set. In all the cases it is observed that PrGP obtained the best performance in the training set and in addition to this its convergence rate is higher than the other systems. In the test set, it can be seen that PrGP obtained the best performance in the first generations and then it starts presenting overfitting. The only dataset where PrGP was the second best in the first generations is breast cancer.

Figure 3 presents the performance of the systems in the rest of the classification problems. A similar behaviour is presented on the training set, the exception is in the twonorm dataset, there GP had the best performance in the training set at the end of the run. Nonetheless, PrGP had a higher convergence rate reaching its best performance in the first generations. In the test set PrGP presented overfitting in almost all the datasets the exception is the image dataset. Furthermore, in image, waveform and splice is where PrGP outperforms the other systems in the test set. These dataset are the ones that have the largest number of dimensions and instances in the training set.

In order to complement the information presented on Figures 2 and 3, Table 3 presents the average performance and its standard deviation of PrGP, GSGP and GP on the training set. From the table, it can be seen that PrGP had the best performance in all the datasets. This is an indication that PrXO and PrMut increase the learning capabilities of genetic programming.

Table 3. Performance in terms of the balance error rate on the training set. The best performance is in boldface

Dataset	PrGP	GSGP	GP
Banana	6.98 ± 1.44	13.35 ± 2.41	9.47 ± 1.53
Titanic	26.04 ± 3.85	26.06 ± 3.77	26.17 ± 4.26
Thyroid	0.00 ± 0.00	1.03 ± 0.90	0.09 ± 0.32
Diabetes	6.99 ± 1.30	18.37 ± 1.53	19.63 ± 1.75
Breast-Cancer	4.38 ± 1.46	16.02 ± 2.54	20.52 ± 3.14
Flare-Solar	29.05 ± 1.09	30.47 ± 1.06	30.31 ± 1.03
Heart	0.44 ± 0.49	7.30 ± 1.68	8.20 ± 1.95
Ringnorm	0.18 ± 0.24	2.96 ± 0.83	0.94 ± 0.55
Twonorm	0.11 ± 0.17	2.25 ± 0.67	0.26 ± 0.26
German	10.64 ± 1.20	21.51 ± 1.64	25.80 ± 1.71
Image	4.12 ± 0.41	9.27 ± 0.88	6.64 ± 1.23
Waveform	0.57 ± 0.41	8.77 ± 1.51	5.66 ± 1.46
Splice	2.78 ± 0.52	7.95 ± 0.77	9.63 ± 1.46

Semantic Genetic Programming Operators Based on Projections in the Phenotype Space

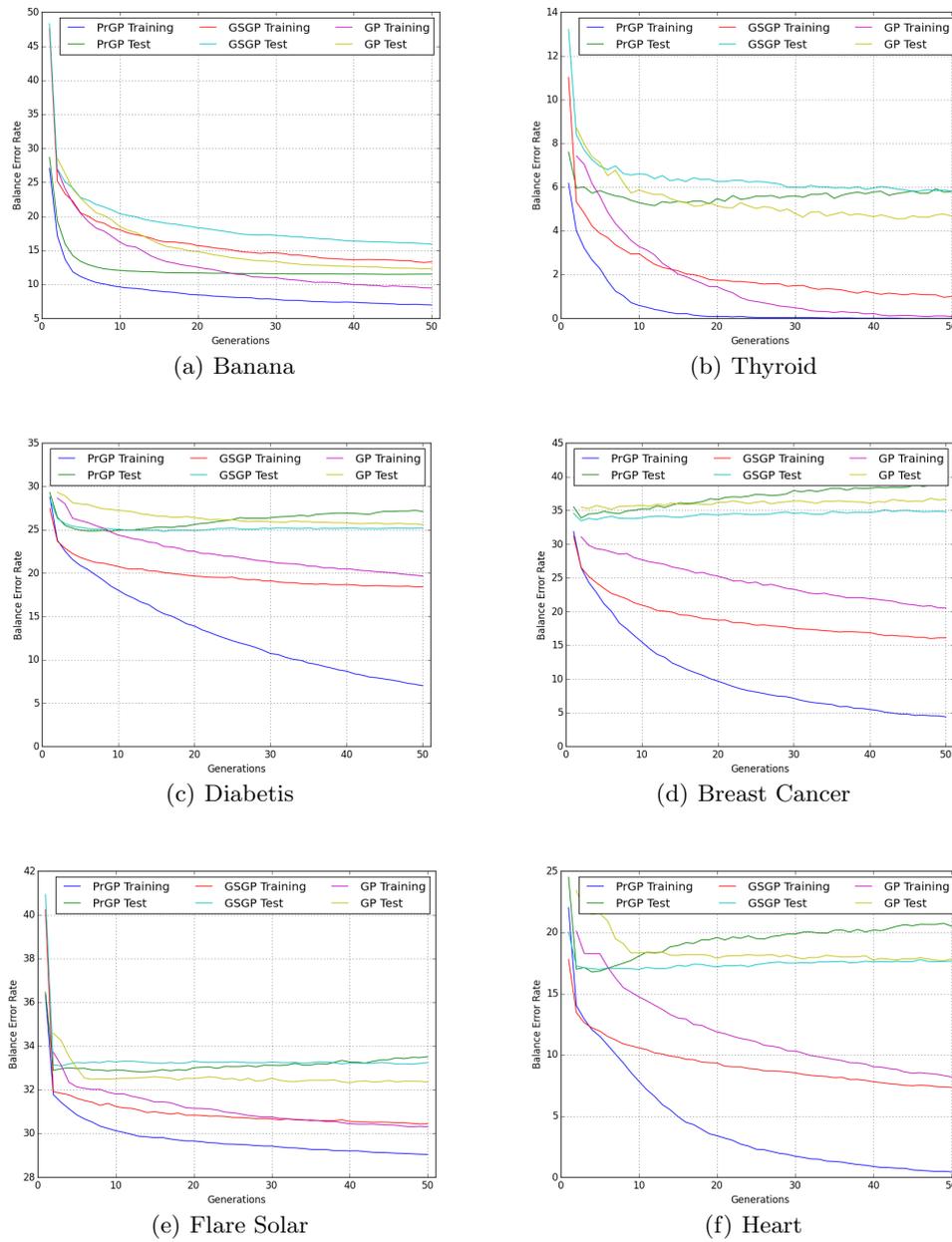


Fig. 2. Performance in terms of the balance error rate

So far, we have only compared the performance of PrGP against other genetic

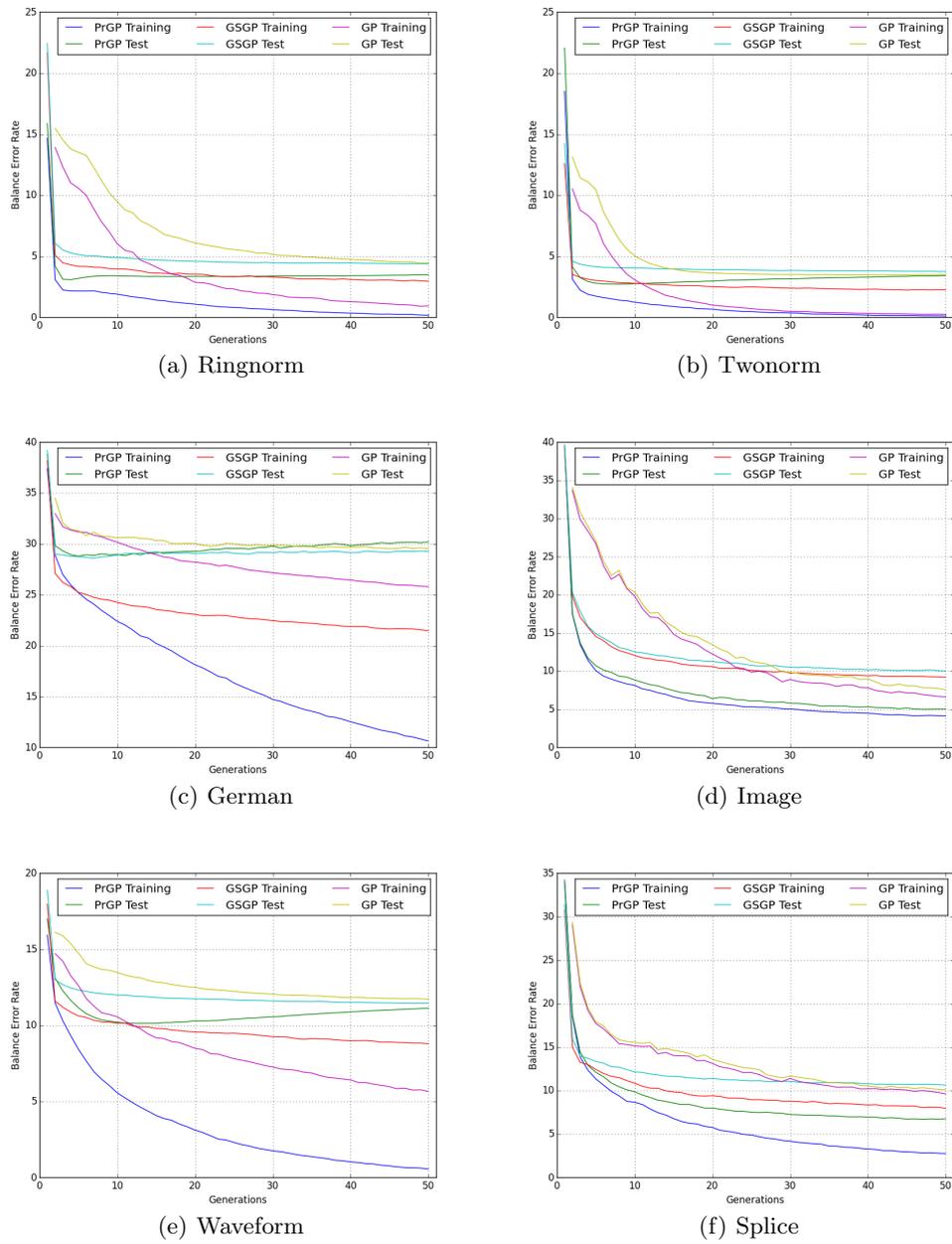


Fig. 3. Performance in terms of the balance error rate

programming systems. In the case of classification, it was decided to compare them against state of the art classifiers. The first classifier is an optimized version of support vector machine (SVM) [14] and the second is the Particle Swarm Model Selection (PSMS) [2]. SVM is one of the most used methods in pattern classification due to its proved effectiveness. PSMS uses particle swarm optimization to search for a classification model that maximizes an estimate of classification performance; where a classification model can also include preprocessing, and feature selection techniques. PSMS explores the space of all models that can be built by using a wide variety of methods and returns a very effective classification model. It is important to note that the performance of SVM and PSMS were taken from [14].

Table 4 presents the performance in the test set of PrGP, GSGP, GP, SVM and PSMS. It is observed that SVM had the best performance in four datasets, in second place there is a tie between PrGP and PSMS both having the best performance in three datasets. GP had the best performance in two datasets and GSGP had it in one dataset. Looking at the performance in the datasets with 20 dimensions or more, PrGP had the best performance in three out of 6. In Ringnorm dataset, PrGP presented overfitting it would be more competitive if a stopping criteria would have been used such as kfold-validation. On the other hand, PrGP presented on Image dataset underfitting, this can be corroborated on Figure 3 (f), there it is observed that the performance in the training set is closely followed by the performance on the test set and both are still decreasing at the time the evolution stop.

Table 4. Performance (BER) in the test set of the different genetic programming systems (PrGP, GSGP, and GP) and two state of the art classifiers (SVM and PSMS). The best performance is in boldface.

Dataset	PrGP	GSGP	GP	SVM	PSMS
Banana	11.50 ± 0.53	15.96 ± 1.99	12.30 ± 0.95	46.11 ± 3.64	10.81 ± 0.64
Titanic	30.76 ± 2.44	30.56 ± 2.62	30.44 ± 2.12	22.51 ± 0.16	22.81 ± 1.10
Thyroid	5.80 ± 3.13	5.78 ± 3.23	4.68 ± 2.54	11.60 ± 3.61	4.80 ± 2.82
Diabetes	27.11 ± 2.41	25.21 ± 2.34	25.57 ± 1.95	23.17 ± 1.69	27.73 ± 1.95
Breast-Cancer	38.78 ± 4.74	34.90 ± 5.04	36.61 ± 5.23	29.87 ± 3.77	31.95 ± 3.93
Flare-Solar	33.52 ± 1.73	33.27 ± 1.64	32.35 ± 1.65	32.73 ± 1.63	32.80 ± 1.50
Heart	20.50 ± 3.53	17.69 ± 3.47	17.84 ± 4.07	17.90 ± 2.85	24.90 ± 10.73
Ringnorm	3.49 ± 0.31	4.39 ± 0.43	4.45 ± 0.67	24.75 ± 0.51	2.37 ± 2.20
Twonorm	3.43 ± 0.26	3.77 ± 0.40	3.51 ± 0.44	3.57 ± 0.59	7.82 ± 14.88
German	30.21 ± 2.67	29.25 ± 2.51	29.54 ± 2.44	23.60 ± 2.22	25.80 ± 3.98
Image	5.03 ± 0.61	10.05 ± 0.82	7.57 ± 1.21	15.37 ± 1.01	3.90 ± 0.83
Waveform	11.13 ± 0.60	11.44 ± 0.67	11.73 ± 0.81	13.45 ± 0.63	12.08 ± 1.23
Splice	6.78 ± 0.42	10.64 ± 0.65	10.10 ± 1.70	16.37 ± 0.85	12.78 ± 1.92

5 Conclusions

In this contribution we have presented a novel semantic genetic operators, namely PrXO and PrMut. These operators are based on projections in the phenotype space and have as their more prominent characteristic that the fitness of the offspring is at least as good as the fitness of the best parent. These operators have been tested in two classes of problems: symbolic regression and classification problems. The results show that PrGP (GP using PrXO and PrMut) had the best performance in the training set in all the cases tested. Furthermore, it is the system that presented the highest convergence rate which makes possible to reduce the number of generations and still obtained the same performance.

Regarding the generalization ability of PrGP, it was noted that it is needed a procedure to identify the time when PrGP start to overfit. Such a procedure could be to incorporate a kfold validation, or any other appropriate technique. We leave this research avenue for future work. Nonetheless, it is observed that PrGP had the best performance in three out of six of the classification problems that have at least 20 dimensions. This might be a niche of opportunity for PrGP; however, more research is needed to validate this assertion.

References

1. Blicke, T., Thiele, L.: Genetic programming and redundancy. *choice* 1000, 2 (1994)
2. Escalante, H.J., Montes, M., Sucar, L.E.: Particle swarm model selection. *The Journal of Machine Learning Research* 10, 405–440 (2009)
3. Goncalves, 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., Garcia-Sanchez, P., Burelli, P., Risi, S., Sim, K. (eds.) 18th European Conference on Genetic Programming. LNCS, vol. 9025, pp. 41–52. Springer, Copenhagen (8–10 Apr 2015), forthcoming
4. Graff, M., Flores, J.J., Bejar, J.O.: Genetic Programming: Semantic point mutation operator based on the partial derivative error. In: 2014 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC). pp. 1–6 (Nov 2014)
5. Graff, M., Graff-Guerrero, A., Cerda-Jacobo, J.: Semantic crossover based on the partial derivative error. In: Nicolau, M., Krawiec, K., Heywood, M.I., Castelli, M., Garcia-Sanchez, P., Merelo, J.J., Santos, V.M.R., Sim, K. (eds.) 17th European Conference on Genetic Programming. LNCS, vol. 8599, pp. 37–47. Springer, Granada, Spain (23–25 Apr 2014)
6. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Natural Selection*. MIT Press, Cambridge, MA, USA (1992)
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.) *Parallel Problem Solving from Nature - PPSN XII*, pp. 21–31. No. 7491 in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg (Jan 2012)
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., Falco, I.D., Ebner, M. (eds.) *Genetic Programming*,

- pp. 292–302. No. 5481 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2009)
9. Pawlak, T., Wieloch, B., Krawiec, K.: Semantic Backpropagation for Designing Search Operators in Genetic Programming. *IEEE Transactions on Evolutionary Computation Early Access Online* (2014)
 10. Poli, R., Graff, M., McPhee, N.F.: Free lunches for function and program induction. In: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms. pp. 183–194. FOGA '09, ACM, Orlando, Florida, USA (2009), 00011 ACM ID: 1527148
 11. Poli, R., Langdon, W.B., McPhee, N.F.: *A Field Guide to Genetic Programming*. Lulu Enterprises, UK Ltd (Mar 2008)
 12. Poli, R., Langdon, W.B., McPhee, N.F.: *A field guide to genetic programming*. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk> (2008), <http://www.gp-field-guide.org.uk>, (With contributions by J. R. Koza)
 13. Rojas, R.: *Neutral Networks: A Systematic Introduction*. Springer (1996)
 14. Rosales-Pérez, A., Escalante, H.J., Gonzalez, J.A., García, C.A.R.: Bias and variance optimization for SVMs model selection. In: FLAIRS Conference. pp. 136–141. Association for the Advancement of Artificial Intelligence, Florida, USA (2013), <http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/download/5890/6055>
 15. Ruberto, S., Vanneschi, L., Castelli, M., Silva, S.: ESAGP – A semantic GP framework based on alignment in the error space. In: Nicolau, M., Krawiec, K., Heywood, M.I., Castelli, M., Garci-Sanchez, P., Merelo, J.J., Santos, V.M.R., Sim, K. (eds.) *17th European Conference on Genetic Programming*. LNCS, vol. 8599, pp. 150–161. Springer, Granada, Spain (23–25 Apr 2014)
 16. Uy, N.Q., Hoai, N.X., O’Neill, M., McKay, R.I., Galvan-Lopez, E.: Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines* 12(2), 91–119 (Jul 2010), <http://www.springerlink.com/content/48411662081364h6/>
 17. Valencia-Ramirez, J.M., Raya, J.A., Cedeno, J.R., Suarez, R.R., Escalante, H.J., Graff, M.: Comparison between Genetic Programming and full model selection on classification problems. In: 2014 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC). pp. 1–6 (Nov 2014)
 18. Vanneschi, L., Castelli, M., Manzoni, L., Silva, S.: A new implementation of geometric semantic GP and its application to problems in pharmacokinetics. In: Krawiec, K., Moraglio, A., Hu, T., Etaner-Uyar, A.i., Hu, B. (eds.) *Genetic Programming*, pp. 205–216. No. 7831 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2013)
 19. Vanneschi, L., Castelli, M., Silva, S.: A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines* 15(2), 195–214 (Jun 2014), <http://link.springer.com/article/10.1007/s10710-013-9210-0>