On the Use of Semantics in Multi-objective Genetic Programming

Edgar Galván-López[1(✉)], Efrén Mezura-Montes[2], Ouassim Ait ElHara[3], and Marc Schoenauer[3]

[1] School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland
edgar.galvan@scss.tcd.ie

[2] Universidad Veracruzana, Xalapa, Veracruz, Mexico
emezura@uv.mx

[3] TAO, INRIA and LRI, CNRS & U. Paris-Sud,
Université Paris-Saclay, Orsay, France
{ouassim.ait_elhara,marc.schoenauer}@inria.fr

Abstract. Research on semantics in Genetic Programming (GP) has increased dramatically over the last number of years. Results in this area clearly indicate that its use in GP can considerably increase GP performance. Motivated by these results, this paper investigates for the first time the use of Semantics in Muti-objective GP within the well-known NSGA-II algorithm. To this end, we propose two forms of incorporating semantics into a MOGP system. Results on challenging (highly) unbalanced binary classification tasks indicate that the adoption of semantics in MOGP is beneficial, in particular when a semantic distance is incorporated into the core of NSGA-II.

1 Introduction

Genetic Programming (GP) [9] has been successfully used in a range of different challenging problems (see Koza's article on human competitive results for a comprehensive review [10]). Despite its proven success, it also suffers from some limitations and researchers have been interested in making GP more robust by studying various elements of the search process, and also by e.g., considering other GP forms [7].

One of these elements that has relatively recently attracted the attention of researchers is the study of semantics in GP, resulting in a dramatic increase in the number of related publications (e.g., [2,8,11,12]).

Semantics is a broad concept that has been studied in different fields making it hard to give a precise definition of the concept. Moreover, the way semantics has been adopted in canonical GP varies significantly e.g., Beadle and Johnson [2] used reduced ordered binary decision trees on Boolean problems

Research conducted during Galván's stay at TAO, INRIA and LRI, CNRS & U. Paris-Sud, Université Paris-Saclay, France.

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 353–363, 2016.
DOI: 10.1007/978-3-319-45823-6_33

to study semantics, whereas Uy's work on semantics has focused on repeatedly applying crossover to encourage semantic difference between parents and offspring (see [15] for a summary of works carried out in semantics).

This work uses a popular version of semantics GP, as originally proposed in [12], and used in recent works from the first author [8,13], in which the *semantics* of a (sub)tree is defined as the vector of output values computed by this (sub)tree for each set of input values in turn (a.k.a. each fitness case in most cases). Several semantic-based approaches have been proposed for GP which take semantics into account when e.g., choosing and modifying subtrees, such as the one that has been demonstrated beneficial in [13] and it is adopted in this work too.

To the best of our knowledge, however, there is no scientific study on the adoption of semantics in Evolutionary Multi-objective Optimisation at large [5], and in Multi-objective GP in particular and this paper intends to start filling this important research area.

The goal of this paper is to incorporate semantics into a Multi-objective GP paradigm by using the well-known NSGA-II. To this end, we adopted two different forms of incorporating semantics into NSGA-II: (a) one based on a relatively simple, efficient and straightforward semantic-based single-objective GP approach, and (b) one based on the adoption of a semantic distance into the core of the NSGA-II algorithm.

This paper is organised as follows. In Sect. 2, we introduce our proposed approaches. Section 3 provides details on the experimental setup used. The results presented in this paper are discussed in Sect. 4, and finally, conclusions and future work are drawn in Sect. 5.

2 Semantics in Multi-objective Genetic Programming

In this work, following [12], the *semantics* of a GP tree describes the behaviour of the tree when various values are given to the input variables. Two trees can be syntactically very different while behaving identically. What matters, as far as solving the problem at hand is concerned, is in fact the behaviour of the tree, i.e., its response to given inputs. These arguments support the use of semantics adopted here and at least partly explain the benefits of using semantics in GP as reported in [8,13].

In the case of a fitness based on the computation of several fitness cases, the semantics of a GP individual is a vector of size the number of fitness cases, one value for each fitness case. For instance, in the case of the problems used in this work (unbalanced data sets introduced in Sect. 3), the semantics of a GP tree is the vector of real-valued output by the tree for each of the examples in the e.g., training data set. In this work, the semantic distance between two trees is the number of outputs that are different between their semantics. Commonly, when computing the semantic distance, two outputs are considered different if their absolute difference is greater than a given threshold [8,13]. In this work, we set the threshold at 0.5.

2.1 Evolutionary Multi-objective Optimisation

Multi-objective optimisation (MO) is concerned with the simultaneous optimisation of several objectives. When these are in conflict, no single solution exists, and trade-offs between the objectives must be sought. The optimal trade-offs are the solutions for which no objective can be further improved without degrading another objective. This idea is captured in the Pareto dominance relation: a point x in the search space is said to *Pareto-dominate* another point y if x is at least as good as y on all objectives and strictly better on at least one objective.

The set of optimal trade-off solutions of a MO problem can then be defined as the set of points of the search space that are not dominated by any other point, and is called the *Pareto set* of the problem at hand. The goal of Pareto MO is to identify the Pareto set, or a good approximation of it. The *Pareto front* is the image of the Pareto set in the objective space.

Evolutionary multi-objective optimisation (EMO) [5] is based on the following: by replacing the single-objective selection steps, based on the comparison of fitness values, by some Pareto-based comparison, one turns a single-objective evolutionary optimisation algorithm into a multi-objective evolutionary optimisation algorithm, but because Pareto dominance is not a total order, some additional criterion must be used so as to allow the comparison of any pair of points of the search space.

In NSGA-II [6], the Pareto-based comparison uses the non-dominated sorting procedure: all non-dominated individuals in the population are assigned Rank 1 and removed from the population, the remaining non-dominated individuals are assigned Rank 2, and so on. The secondary criterion is the *crowding distance* that promotes diversity among the individuals having the same Pareto rank: in objective space, for each objective, the individuals in the population are ordered, and the partial crowding distance for each of them is the difference in fitness between its two immediate neighbours. The crowding distance is the sum over all objectives of these partial crowding distances. Intuitively, it can be seen as the Manhattan distance between the extremal vertices of the largest hypercube containing the point at hand and no other point of the population. Selecting points with the largest crowding distance amounts to favour the low-density regions of the objective space, thus favouring behavioural diversity.

The NSGA-II proceeds as follows. From a given population of size N, N offspring are created using standard variation operators (crossover and mutation). Parents and offspring are merged, and the resulting population, of size $2N$, is ordered using non-dominated sorting, and crowding distance as secondary criterion. The best N individuals according to this ranking are selected to survive at the next generation.

Because the underlying idea within NSGA-II is to favour behavioural diversity, but only considering the fitness as a whole, it can be hoped that introducing semantics in NSGA-II can only enforce this idea.

2.2 Incorporating Semantics in MOPG

In this work, we investigate two ways of incorporating semantics into a MOGP system (recall we use NSGA-II). One natural form to do so is to use semantics as commonly adopted in canonical GP (e.g., semantically-based crossover [13]). In our study, we adopted the semantics in the selection tournament mechanism [8] due to its simplicity and efficiency. Briefly, the idea is to create offspring that are semantically different from their parents when tournament selection is applied: the first parent is selected as usual and the second parent is selected if it is semantically different and fitter than the already selected parent, if this is not satisfied for any individual in the pool, one is chosen at random. We call this NSGA-II Semantics in Selection (SiS).

The second proposed way to add semantics to NSGA-II is to replace the crowding distance (see above) with a semantic-based indicator called Semantic-based Crowding Distance (SCD). This is computed the following way: a *pivot* is chosen, being the individual from the first Pareto front (Rank 1) that is the furthest away from the other individuals of this front using the crowding distance. For each point, its semantic distance with the pivot is computed. Similarly to the crowding distance, the SCD is computed as the average of the semantic distance differences with its closest neighbours in each direction. The higher values of this SCD are favored during the selection step of NSGA-II. This allows us to have a set of individuals that are spread in the semantic space, therefore, promoting semantic diversity, the same way NSGA-II promotes diversity ('spreadness') in the objective space. It is worth pointing out that this approach also works when there is only one front. This variant of NSGA-II will be called Distance-based Semantics (DBS) in the following.

3 MOGP Configuration and Experimental Design

To study the effects of semantics in MOGP, we used challenging binary (highly) unbalanced classification problems taken from the literature [1]. These problems are of different nature and complexity, e.g., they have from a few features up to dozens of them, these features include binary, integer, and real-valued features. Table 1 gives the details for all datasets. These have been used 'as is' (i.e., we did not try to balance the classes out). For each dataset, half of the data (with the same class balance than in the whole dataset) was used as a training set and the rest as a test set. All reported results are on the latter.

The terminal and function sets used in these experiments were the same than in [3]. The terminals are the problem features. The function set consists of the conditional IF function and the typical four standard arithmetic operators: $\mathcal{F} = \{if, +, -, *, /\}$, where the latter operator is the protected division, which returns the numerator if the denominator is zero. The IF function takes three arguments: if the first one is negative, the second argument is returned, otherwise the last argument is returned. These functions are used to build a classifier (e.g., mathematical expression) that returns a single value for a given input (data example to be classified). This number is mapped onto a set of class labels using

Table 1. Binary unbalanced classification data sets used in our research. Table adapted from [3].

Data set	Classes	Number of examples			Imb.	Features	
	Positive/Negative (Brief description)	Total	Positive	Negative	Ratio	No	Type
Ion	Good/bad (ionsphere radar signal)	351	126 (35.8 %)	225 (64.2 %)	1:3	34	Real
Spect	Abnormal/normal (cardiac tom. scan)	267	55 (20.6 %)	212 (79.4 %)	1:4	22	Binary
Yeast_1	mit/other (protein sequence)	1482	244 (16.5 %)	1238 (83.5 %)	1:6	8	Real
Yeast_2	me3/other (protein sequence)	1482	163 (10.9 %)	1319 (89.1 %)	1:9	8	Real

Table 2. Confusion matrix.

	Predicted positive	Predicted negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

zero as the class threshold. In our studies, an example is assigned to the minority class if the output of the classifier is greater or equal to zero. It is assigned to the majority class, otherwise.

The common way to measure the fitness of a classifier for classification tasks is the overall classification accuracy: for binary classification, the four possible cases are shown in Table 2. Assuming the minority class is the positive class, the accuracy is given by $Acc = \frac{TP+TN}{TP+TN+FP+FN}$. The drawback of using Acc alone is that it rapidly biases the evolutionary search towards the majority class [3]. A better approach is to treat each objective (class) 'separately' using a multi-objective approach: Two objectives are considered, the true positive rate $TPR = \frac{TP}{TP+FN}$, and the true negative rate $TNR = \frac{TN}{TN+FP}$, that measure the distinct accuracy for the minority and majority class, respectively.

The experiments were conducted using a steady state approach with tournament selection (of size 2 for NSGA-II and NSGA-II DBS, and of size 7 for NSGA-II SiS to encourage semantic diversity). Initialisation and sub-tree mutation used the ramped half-and-half method (initial and final depth set at 1 and 5, respectively). To control bloat, a maximum depth of 8 was specified (root is at depth 0), or a maximum number of 800 nodes was used. Crossover and mutation rates were set at 60 % and 40 %, respectively. To obtain meaningful results, we performed 50 independent runs for each of the MOGP approaches for each of the problems used in this work.

Table 3. Average (± standard deviation) hypervolume, where the reference point is (0,0), of evolved Pareto-approximated fronts, Pareto optimal (PO) front for the three MOGP used in this work: NSGA-II, NSGA-II SiS and NSGA-II DBS, over 50 runs.

Methods	Hypervolume	Ion	Spect	Yeast$_1$	Yeast$_2$
NSGA-II	Average	0.842 ± 0.070	0.542 ± 0.024	0.822 ± 0.041	0.944 ± 0.021
	PO Front	0.948	0.637	0.875	**0.978**
NSGA-II SiS	Average	0.858 ± 0.063	0.542 ± 0.020	0.827 ± 0.035	0.939 ± 0.048
	PO Front	0.960	0.642	**0.876**	0.977
NSGA-II DBS	Average	**0.856 ± 0.051**	0.548 ± 0.026	0.827 ± 0.015	**0.948 ± 0.011**
	PO Front	**0.977**	**0.664**	0.873	0.977

4 Results and Discussion

4.1 Front Hypervolume

As a measure of performance, in order to compare the different approaches, we use the hypervolume [4] of the evolved Pareto approximations. For bi-objectives problems, the hypervolume of a set of points in objective space (using reference point $(0, 0)$) is easily computed as the sum of the areas of all trapezoids fitted under each point. Such measure was chosen as being the only known Pareto-compliant indicator to-date [16]: the larger the hypervolume, the better the performance. We also computed the Pareto-optimal (PO) front with respect to all 50 runs, i.e., the set of non-dominated solutions after merging all 50 Pareto-approximated fronts.

Table 3 reports, for each problem, both the average hypervolume over 50 runs, and the hypervolume of the PO. In this table, the best hypervolumes are highlighted in boldface. Furthermore, the statistical significance for the results on the average hypervolume was computed using Wilcoxon Test at 90 % level of significance, independently comparing each of the semantic-based approaches (NSGA-II SiS, NSGA-II DBS) against NSGA-II.

According to these results, in three out of the four problems, both semantic-based MOGP approaches achieve a higher hypervolume of the PO front compared to the NSGA-II. Moreover, the NSGA-II DBS is statistically better (indicated in boldface) than the NSGA-II on two classification problems, but not statistically different on the other two problems. On the other hand, NSGA-II SiS is not statistically different on any of the problems compared to NSGA-II. This suggests that the adoption of semantics into a MOGP approach should be in one of the pillars of the MO approach.

4.2 Evolved Solutions and Pareto-Optimal Front

Let us now focus on the coverage of the objective space achieved by the semantic variants of NSGA-II: Fig. 1 displays together on the same plot, for each problem (top to bottom), and for NSGA-II and NSGA-II DBS (left and center respectively), the 50 Pareto front approximations obtained in the 50 independent runs

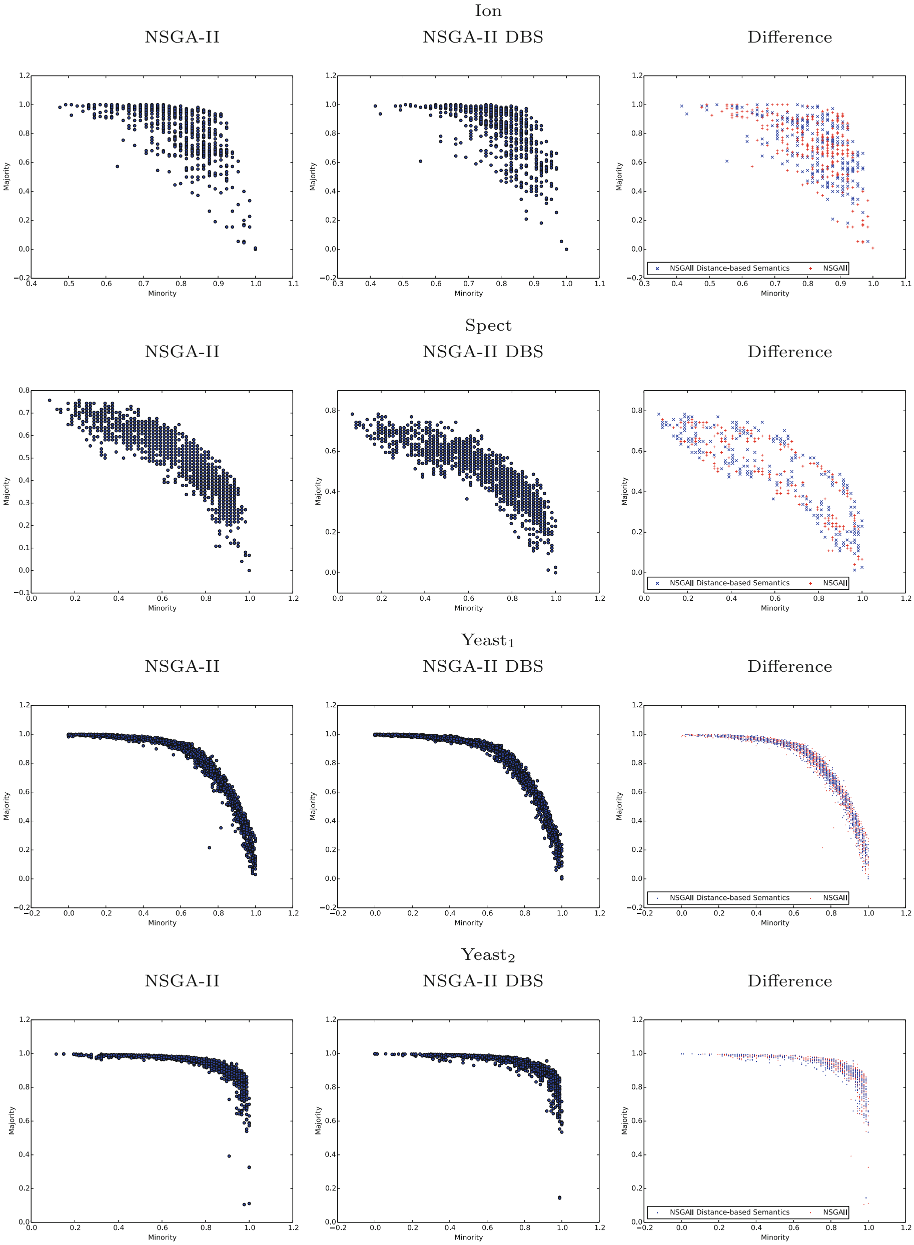

Fig. 1. Accuracy of all evolved solutions over 50 runs using the canonical NSGA-II and the NSGA-II DBS, shown in the left-hand side and centre of the figure, respectively. Plots in the the right-hand side of the figure show the evolved solutions that were exclusively found by either NSGA-II (indicated by a red plus '+' symbol) or NSGA-II DBS (indicated by a blue cross 'x' symbol). For clarity purposes, we reduced the size of the marker symbols in problems with denser areas (i.e., Yeast problems).

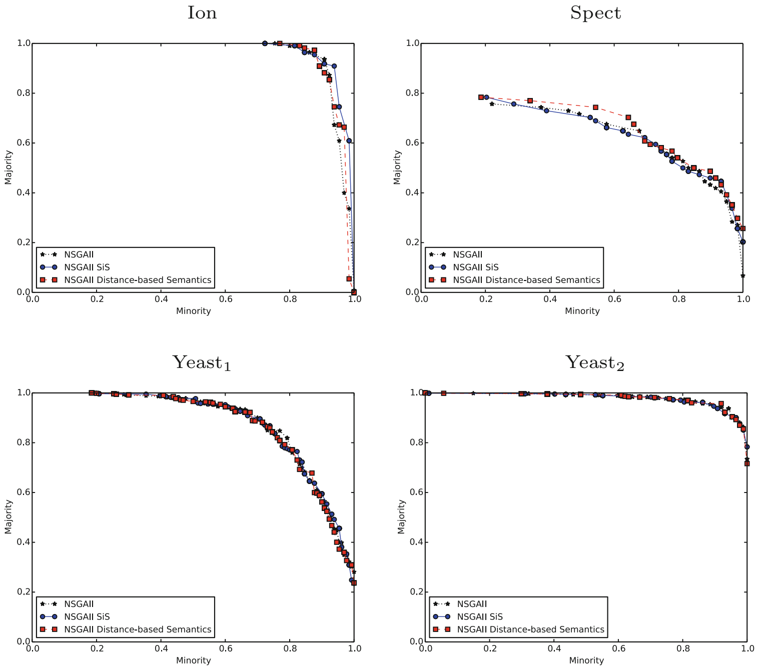

Fig. 2. Pareto-Optimal fronts each of the four problems for NSGA-II (black star symbols), NSGA-II SiS (blue circle symbols) and NSGA-II DBS (red square symbols). (Color figure online)

(NSGA-II SiS is omitted due to space constraints). For some problems (e.g., Ion), it is relatively easy to see that NSGA-II DBS has a better coverage of the objective space. A better look at the difference between NSGA-II and NSGA-II DBS is proposed on the right-hand side of the figure: only the points found by one of both algorithms are plotted, a red plus '+' symbol for NSGA-II, a blue cross 'x' symbol for NSGA-II DBS. More blue cross 'x' symbols are visible on top or right of the objective space, where the true Pareto front lies and explains why DBS has a better performance on the Ion data set.

Figure 2 shows, for each problem, the Pareto-Optimal fronts (POs) for each of the MOGP approaches used in this work. In accordance to the results reported in Table 3, little difference is observed among the three methods on the Yeast$_2$ problem, while NSGA-II SiS dominates on the Yeast$_1$ problem; and both semantic variants dominate parts of the front for Ion, while NSGA-II DBS is a clear winner for Spect.

4.3 Bloat

Bloat (dramatic increase of tree sizes as evolution proceeds) has always been an issue in GP, and should be monitored carefully when designing new GP variants.

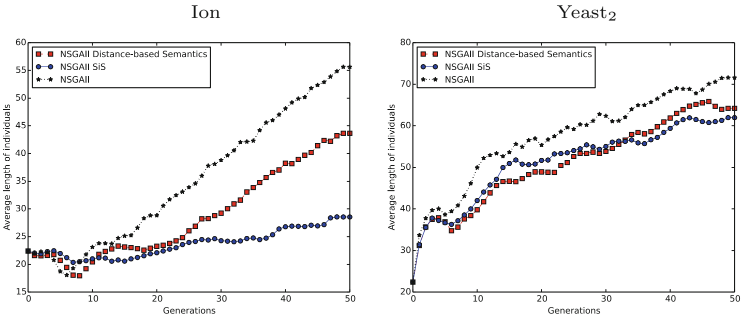

Fig. 3. Average length of evolved solutions vs generations, over 50 independent runs, for the Ion and Yeast$_2$ problems, for NSGA-II (black star symbols), NSGA-II SiS (blue circle symbols) and NSGA-II DBS (red square symbols). (Color figure online)

Contradictory results regarding bloat have been reported for semantic-based GP: semantics seems to prevent bloat in [8], while it exacerbates it in [13]. To shed some light on this issue here, Fig. 3 shows, for the Ion and Yeast$_2$ problems, the average length of evolved trees during evolution.

It is clear that the semantic-based approaches tend to produce slightly shorter programs compared to canonical NSGA-II on the Yeast$_2$ problem – and similar tendency was observed for the Yeast$_1$ and Spect problems (not shown here due to space constraints). Surprisingly, NSGA-II SiS is indeed able to produce much shorter programs compared to both other methods for the Ion problem. This is aligned to the results reported in [8], that indicates that SiS is capable of producing shorter programs compared to e.g., the well-known semantic-based crossover [13]. From this, we believe that researchers tend to report mixed results on bloat because its appearance is dependant on both: problem and approach used, and so, no general conclusions can be drawn on this.

5 Conclusions and Future Work

In Genetic Programming, semantics is commonly defined as the behaviour of syntactically correct programs. In canonical GP, semantics is represented by the output vector of the tree for different known inputs and the similarity between the semantics of two trees gives a much smoother idea of the similarity between the trees than either the syntactic description of the trees or their raw fitness.

This work proposed two ways to add semantics to Multi-Objective GP, more precisely NSGA-II for GP. The first one, Semantics in Selection (SiS), was adapted from canonical GP to NSGA-II. The second approach, named Distance-based Semantics (DBS), consists in using a semantic distance in lieu of the crowding distance at the heart of NSGA-II.

We have learned that semantic-based NSGA-II GP behaves better than plain NSGA-II GP on some well-known unbalanced binary classification problems. We also learned that NSGA-II DBS outperforms NSGA-II SiS. We believe that the reason behind this is because the concept of semantic distance is used into the very core of NSGA-II. There are multiple research areas that we will consider in the near future. An in-depth analysis is required to confirm, and understand why DBS outperforms SiS. Given the encouraging results, it is worth studying the effects of semantics in other parts of a MOGP algorithm (e.g., ranking system). It is also necessary to study the adoption of semantics and its impact in other well-known MO approaches.

Acknowledgements. EGL's research is funded by the Irish Research Council and co-funded by Marie Curie Actions. EGL would like to thank the TAO group at INRIA Saclay France for hosting him during the outgoing phase of the fellowship. The authors would like to thank the anonymous reviewers for their helpful comments.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007)
2. Beadle, L., Johnson, C.: Semantically driven crossover in genetic programming. In: 2008 IEEE Congress on Evolutionary Computation CEC 2008. IEEE World Congress on Computational Intelligence, pp. 111–116, June 2008
3. Bhowan, U., Johnston, M., Zhang, M., Yao, X.: Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Trans. Evol. Comput. **17**(3), 368–386 (2013)
4. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-objective Problems. Genetic and Evolutionary Computation. Springer, Secaucus (2006)
5. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, New York (2001)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**, 182–197 (2002)
7. Galván-López, E.: Efficient graph-based genetic programming representation with multiple outputs. Intl. J. Autom. Comput. **5**(1), 81–89 (2008)
8. Galván-López, E., Cody-Kenny, B., Trujillo, L., Kattan, A.: Using semantics in the selection mechanism in genetic programming: a simple method for promoting semantic diversity. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2972–2979, June 2013
9. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge (1992)
10. Koza, J.R.: Human-competitive results produced by genetic programming. Genet. Prog. Evol. Mach. **11**(3–4), 251–284 (2010)
11. Krawiec, K., Pawlak, T.: Locally geometric semantic crossover: a study on the roles of semantics and homology in recombination operators. Genet. Prog. Evol. Mach. **14**, 31–63 (2013)
12. McPhee, N.F., Ohs, B., Hutchison, T.: Semantic building blocks in genetic programming. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 134–145. Springer, Heidelberg (2008)
13. Uy, N.Q., Hoai, N.X., O'Neill, M., McKay, R.I., Galván-López, E.: On the roles of semantic locality of crossover in genetic programming. Inf. Sci. **235**, 195–213 (2013). Data-Based Control, Decision, Scheduling and Fault Diagnostics

14. Uy, N.Q., Hoai, N.X., ONeill, M., McKay, R., Phong, D.N.: On the roles of semantic locality of crossover in genetic programming: application to real-valued symbolic regression. Genet. Prog. Evol. Mach. **12**(2), 91–119 (2011)
15. Vanneschi, L., Castelli, M., Silva, S.: A survey of semantic methods in genetic programming. Genet. Prog. Evol. Mach. **15**(2), 195–214 (2014)
16. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 862–876. Springer, Heidelberg (2007)

Semantic Forward Propagation for Symbolic Regression

Marcin Szubert[1](✉), Anuradha Kodali[2,3], Sangram Ganguly[3,4], Kamalika Das[2,3], and Josh C. Bongard[1]

[1] University of Vermont, Burlington, VT 05405, USA
Marcin.Szubert@uvm.edu
[2] University of California, Santa Cruz, CA 95064, USA
[3] NASA Ames Research Center, Moffett Field, CA 94035, USA
[4] Bay Area Environmental Research Institute, Petaluma, CA 94952, USA

Abstract. In recent years, a number of methods have been proposed that attempt to improve the performance of genetic programming by exploiting information about program semantics. One of the most important developments in this area is *semantic backpropagation*. The key idea of this method is to decompose a program into two parts—a subprogram and a context—and calculate the *desired* semantics of the subprogram that would make the entire program correct, assuming that the context remains unchanged. In this paper we introduce Forward Propagation Mutation, a novel operator that relies on the opposite assumption—instead of preserving the context, it retains the subprogram and attempts to place it in the semantically right context. We empirically compare the performance of semantic backpropagation and forward propagation operators on a set of symbolic regression benchmarks. The experimental results demonstrate that semantic forward propagation produces smaller programs that achieve significantly higher generalization performance.

Keywords: Genetic programming · Program semantics · Semantic backpropagation · Problem decomposition · Symbolic regression

1 Introduction

Standard tree-based genetic programming (GP) searches the space of programs using traditional operators of subtree-swapping crossover and subtree-replacing mutation [4]. These operators are designed to be generic and produce syntactically correct offspring regardless of the problem domain. However, their actual effects on the behavior of the program, and thus its fitness, are generally hard to predict. For this reason, many alternative search operators have been recently proposed that take into account the influence of syntactic modifications on program semantics [1,10,11,13].

Semantic backpropagation [12,15] is arguably one of the most powerful techniques employed by such semantic-aware GP operators. The two operators based

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 364–374, 2016.
DOI: 10.1007/978-3-319-45823-6_34