On the Use of Semantics in Multi-objective Genetic Programming

Edgar Galván-López, Efrén Mezura-Montes, Ouassim Ait Elhara, Marc Schoenauer

To cite this version:

Edgar Galván-López, Efrén Mezura-Montes, Ouassim Ait Elhara, Marc Schoenauer. On the Use of Semantics in Multi-objective Genetic Programming. 14th International Conference Parallel Problem Solving from Nature – PPSN XIV, Sep 2016, Edinburgh, United Kingdom. pp.353 - 363, 10.1007/978-3-319-45823-6_33 . hal-01387632

HAL Id: hal-01387632
https://hal.inria.fr/hal-01387632

Submitted on 25 Oct 2016

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, Ireland
`edgar.galvan@scss.tcd.ie`
[2] Universidad Veracruzana, Xalapa, Veracruz, México
`emezura@uv.mx`
[3] TAO, INRIA and LRI, CNRS & U. Paris-Sud, Université Paris-Saclay, 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, 13]).

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 have been adopted in canonical GP varies significantly e.g., Beadle and Johnson [2] used reduced ordered binary decision trees on Boolean problems to study semantics, whereas Uy's work on semantics has focused on repeatedly applying crossover to encourage semantic difference between parents and offspring (see [16] for a summary of works carried out in semantics).

This work uses a popular version of semantics GP, as originally proposed in [13], and used in recent works from the first author [8, 14], 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 [14] and it is adopted too in this work.

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

2 Semantics in Multi-Objective Genetic Programming

As indicated in Section 1, following [13], the *semantics* of a GP tree describes the behavior of the tree when various values are given to the input variables. 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 Section 3), the semantics of a GP individual is the vector of real-valued outputs of the tree for each of the total examples in the 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, 14]. In this work, we set the threshold at 0.5. It is clear that two trees can be syntactically very different while behaving identically. What matters, as far as solving the problem at hand, is in fact the behavior of a tree: its response to given input values. These arguments support the use of semantics, and at least, partly explain the benefits of using semantics in GP as reported in [8, 14].

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 optimization algorithm, but because Pareto dominance is not a total order, some additional criterion must be added to Pareto dominance 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 excluded from this comparison, 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 immediate neighbours. The crowding distance is the sum of these partial crowding distances for all objectives. 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 explained above. The best N according to this ranking are selected to survive at next generation.

Because the underlying idea within NSGA-II is to favour behavioral 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 investigated 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 [14]). 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 specialised literature [1]. These problems are of different nature and complexity (e.g., from a few number of features up to dozens of them, and these include real, binary and integer-value features, from relatively low to highly unbalanced data). Table 1 shows the sizes of both majority and minority classes for each of these problems along with details about the data used in our study. We used the data as 'it is' (e.g., we did not try to balance the classes out). We used half of each problem to train our MOGP systems and the other half to test it.

To generate classifiers via MOGP, we defined our terminal and function in the same manner as in [3]. The terminal set consists of example features. The function set consists of a conditional IF function and the typical four standard arithmetic operators, and so, the function set is defined by $F = \{if, +, -, *, /\}$, where the latter operator is protected division, which returns the numerator if the denominator is zero. The IF function takes three arguments: if the first 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 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 form to measure the fitness in classification is the overall classification accuracy: using the four outcomes for binary classification shown in

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 object	Predicted non-object
Actual object	True Positive (TP)	False Negative (FN)
Actual non-object	False Positive (FP)	True Negative (TN)

Table 2 and assuming the minority class is the positive class, we have that the accuracy is given by $Acc = \frac{TP+TN}{TP+TN+FP+FN}$. The drawback with Acc is that it can bias the evolutionary search towards the majority class [3]. As mentioned also in [3], a better approach is to treat each objective (class) 'separately' using MOGP. To this end, we use in the fitness function of our MOGP the true positive rate ($TPR = \frac{TP}{TP+FN}$) and the true negative rate ($TNR = \frac{TN}{TN+FP}$) to measure the accuracy for the minority and majority class, respectively.

The experiments were conducted using a steady state approach with tournament selection (Size 2 for NSGA-II and NSGA-II DBS and Size 7 for NSGA-II SiS to encourage semantic difference via this operator). We used the traditional 90/10 crossover operator and subtree mutation. To initialise our population and to apply mutation, we 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 (where root was considered of depth 0) or a maximum number of 800 nodes was used. Crossover and mutation rates were 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.

4 Results and Discussion

4.1 Front Hyperarea

We computed the hyperarea or hypervolume [4] of the evolved Pareto-approximated fronts as a measure to determine which MOGP paradigm is better on the used tasks. The hyperarea is calculated by taking the sum of the areas of individual trapezoids fitted under each front solution in objective-space. Thus, the higher the value, the better the performance. Such measure was chosen because its popularity in the area as well as its theoretical foundations [12].

Table 3 reports the average hyperarea computed over all runs along with its standard deviation (±). We also computed the Pareto-optimal (PO) front with respect to all MOGP runs, i.e., the PO front is the set of non-dominated solutions from all Pareto-approximated fronts evolved from 50 independent runs.

Table 3. Average (± standard deviation) hyperarea, 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	*Hyperarea*	*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

In Table 3, the better hyperarea (average and PO) is highlighted in boldface for each of the four problems used in this work. We computed the statistical significance test on the average hyperarea using the common Wilcoxon Test technique at 90% level of significance independently comparing each of the semantic-based approaches (NSGA-II SiS, NSGA-II DBS) against NSGA-II.

According to this table, in three out of the four problems, both semantic-based MOGP approaches achieve a higher hyperarea 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 and not statistically different on the remaining problems. Furthermore, NSGA-II SiS is not statistically different on any of the problems compared to NSGA-II. This appears to suggest 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

We now turn our attention on the accuracy achieved by all evolved solutions over 50 runs for each of the problems used in this work. Due to space constraints and for clarity purposes, we only show in Figure 1 accuracy values evolved by the NSGA-II and the NSGA-II DBS. In some problems (e.g., Ion), it is relatively easy to see how the NSGA-II DBS, shown in the centre of the figure, is able to find more unique values. To better interpret the results, we also show in the right-hand of the figure, the difference between those values exclusively found either by NSGA-II or NSGA-II DBS. That is, for the NSGA-II, we plotted those values that were not found by the NSGA-II DBS, and *vice versa*. NSGA-II DBS (indicated by a blue cross 'x' symbol) is able to find more values close to the frontier side of all evolved solutions on Ion, Spect and Yeast$_1$ problems compared to NSGA-II (indicated by a red plus '+' symbol).

Figure 2 shows the Pareto fronts for each of the MOGP approaches used in each of the problems employed in this work. As it can be seen, little difference is observed among the three methods on the Yeast problems, which is in accordance to the results reported in Table 3. The advantage of the semantic-based MOGP approaches can be observed in the rest of the problems.

4.3 Bloat

There are a mixture of reports that indicate that either semantics prevent bloat (e.g., dramatic increase of programs' length without observing a corresponding improvement in fitness) e.g., [8] or exacerbates it e.g., [15]. To shed some light on this, we also plotted the average length of programs evolved by each of the MOGP approaches on only two problems (Ion and Yeast$_2$) due to space constrains. This is shown in Figure 3.

As it can be seen from this figure, 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 the same tendency was observed for Yeast$_1$ and Spect problems (not shown in the figure). Surprisingly, NSGA-II SiS is indeed able to produce much shorter programs compared to the other two methods on the Ion problem. This is aligned to the results reported in [8], where the authors pointed out that SiS is capable of producing shorter programs compared to, for example, the well-known semantically-based crossover [14].

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 meaning (behaviour) of syntactically correct programs. In canonical GP, it is said that two individuals are semantically different if their output vectors are different, they are considered semantically similar, otherwise.

In this work we adopted two forms of semantics in Multi-Objective GP. One was the adoption of Semantics in Selection (SiS) adapted from canonical GP to NSGA-II. The other approach, named Distance-based Semantics (DBS), consisted in using a semantic distance into the core of the NSGA-II mechanism.

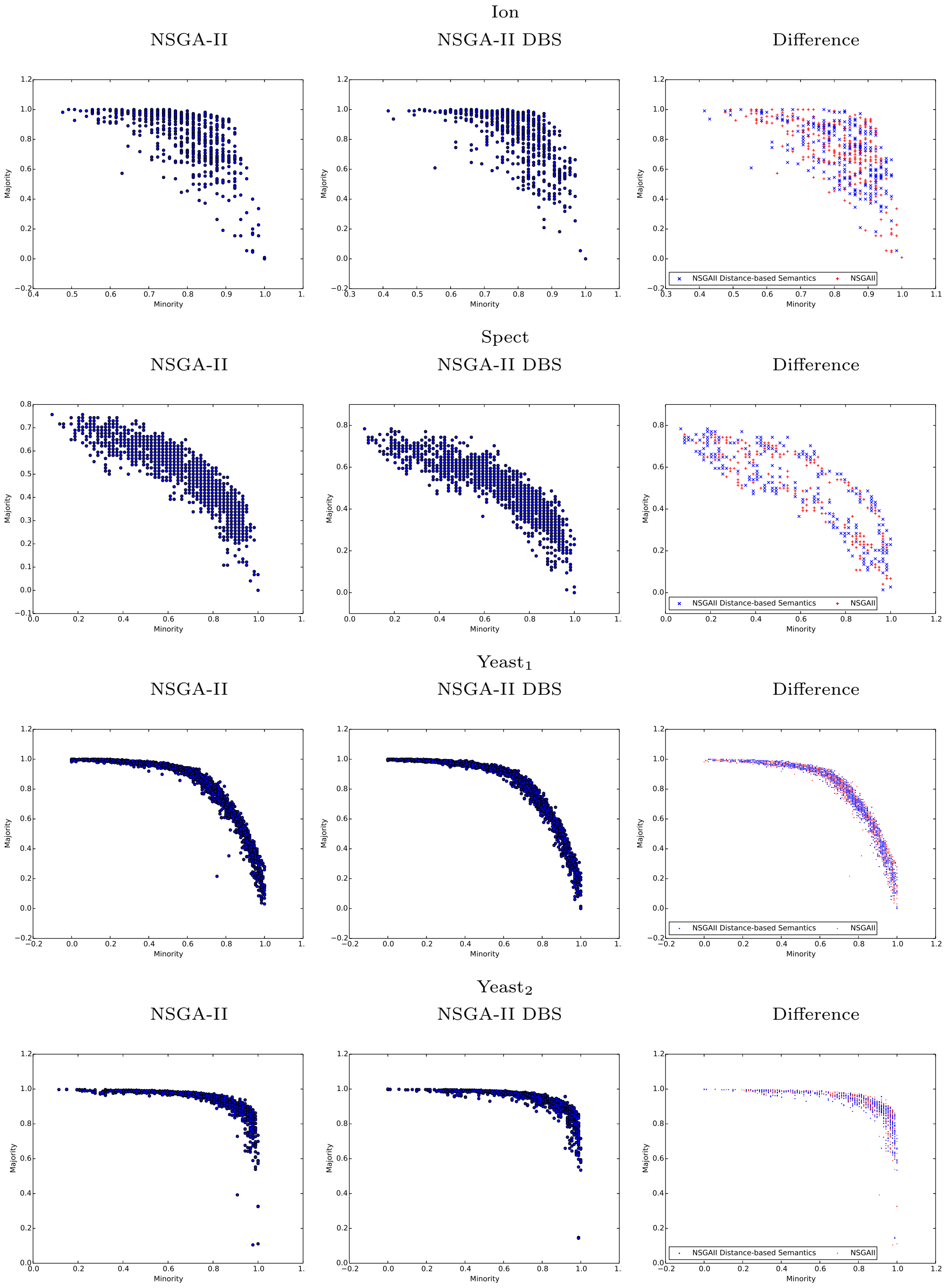
We have learned that semantic-based NSGA-II behaves better than NSGA-II on the unbalanced binary classification problems used in this work. We also learned that the NSGA-II DBS behaves better compared to NSGA-II SiS. We believe that the reason behind this is because the concept of semantic distance is used into the very core of the NSGA-II. There are multiple research areas that we will consider in the near future. An in-depth analysis is required to know why DBS is better than SiS. Given the encouraging results, it is worth considering to study 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 by Marie Curie Actions. The authors would like to thank the reviewers.

References

1. D. N. A. Asuncion. UCI machine learning repository, 2007.

2. L. Beadle and C. Johnson. Semantically driven crossover in genetic programming. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 111 –116, june 2008.
3. U. Bhowan, M. Johnston, M. Zhang, and X. Yao. Evolving diverse ensembles using genetic programming for classification with unbalanced data. *IEEE Transactions on Evolutionary Computation*, 17(3):368–386, June 2013.
4. C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
5. K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.
6. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
7. E. Galván-López. Efficient graph-based genetic programming representation with multiple outputs. *International Journal of Automation and Computing*, 5(1):81–89, 2008.
8. E. Galván-López, B. Cody-Kenny, L. Trujillo, and A. Kattan. Using semantics in the selection mechanism in genetic programming: A simple method for promoting semantic diversity. In *2013 IEEE Congress on Evolutionary Computation*, pages 2972–2979, June 2013.
9. J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, Massachusetts, 1992.
10. J. R. Koza. Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, 11(3-4):251–284, Sept. 2010.
11. K. Krawiec and T. Pawlak. Locally geometric semantic crossover: a study on the roles of semantics and homology in recombination operators. *Genetic Programming and Evolvable Machines*, 14:31–63, 2013.
12. P. K. Lehre, F. Neumann, J. E. Rowe, X. Yao, A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theoretical foundations of evolutionary computation hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75 – 103, 2012.
13. N. F. McPhee, B. Ohs, and T. Hutchison. Semantic building blocks in genetic programming. In *Proceedings of the 11th European conference on Genetic programming*, EuroGP'08, pages 134–145, Berlin, Heidelberg, 2008. Springer-Verlag.
14. N. Q. Uy, N. X. Hoai, M. O'Neill, R. I. McKay, and E. Galván-López. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, 2011.
15. N. Q. Uy, N. X. Hoai, M. ONeill, R. McKay, and D. N. Phong. On the roles of semantic locality of crossover in genetic programming. *Information Sciences*, 235:195 – 213, 2013. Data-based Control, Decision, Scheduling and Fault Diagnostics.
16. L. Vanneschi, M. Castelli, and S. Silva. A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines*, 15(2):195–214, June 2014.



Ion

NSGA-II NSGA-II DBS Difference

Spect

NSGA-II NSGA-II DBS Difference

Yeast$_1$

NSGA-II NSGA-II DBS Difference

Yeast$_2$

NSGA-II NSGA-II DBS Difference

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).

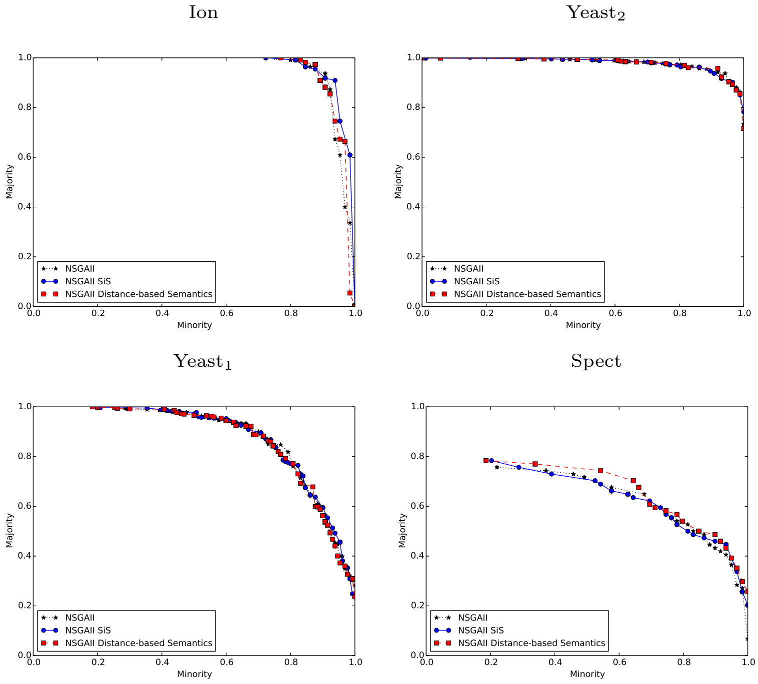


Fig. 2. Performance on the classification task in each of the four problems used in this work achieved by NSGA-II (indicated with a black star symbol), NSGA-II SiS (indicated with a blue circle symbol) and NSGA-II Distance-based Semantics (indicated with a red square symbol).

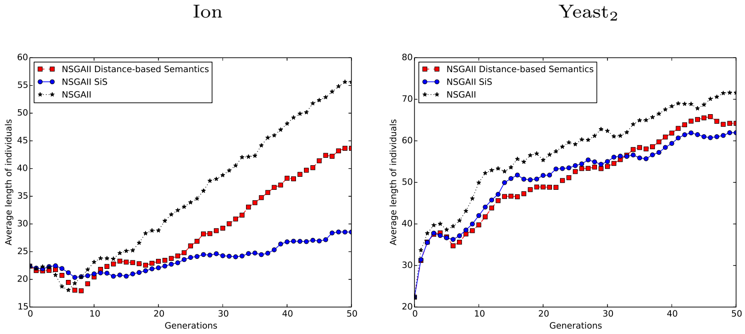


Fig. 3. Average length of evolved solutions, indicated in the x-axis, over 50 generations, indicated in the y-axis, over 50 independent runs using NSGA-II (indicated with a black star symbol), NSGA-II SiS (indicated with a blue circle symbol) and NSGA-II Distance-based Semantics (indicated with a red square symbol) for the Ion and Yeast$_2$ problems.