

Using Semantics in the Selection Mechanism in Genetic Programming: a Simple Method for Promoting Semantic Diversity

¹Edgar Galván-López, ¹Brendan Cody-Kenny, ²Leonardo Trujillo and ³Ahmed Kattan

¹Distributed Systems Group, School of Computer Science and Statistics, Trinity College Dublin

²Doctorado en Ciencias de la Ingeniería, Instituto Tecnológico de Tijuana, Mexico

³AI Real-World Applications Lab, Computer Science Department, UQU, Saudi Arabia

{edgar.galvan, codykenb}@scss.tcd.ie; leonardo.trujillo@tectijuana.edu.mx; ajkattan@uqu.edu.sa

Abstract—Research on semantics in Genetic Programming (GP) has increased over the last number of years. Results in this area clearly indicate that its use in GP considerably increases performance. Many of these semantic-based approaches rely on a trial-and-error method that attempts to find offspring that are semantically different from their parents over a number of trials using the crossover operator (crossover-semantics based - CSB). This, in consequence, has a major drawback: these methods could evaluate thousands of nodes, resulting in paying a high computational cost, while attempting to improve performance by promoting semantic diversity. In this work, we propose a simple and computationally inexpensive method, named *semantics in selection*, that eliminates the computational cost observed in CSB approaches. We tested this approach in 14 GP problems, including continuous- and discrete-valued fitness functions, and compared it against a traditional GP and a CSB approach. Our results are equivalent, and in some cases, superior than those found by the CSB approach, without the necessity of using a “brute force” mechanism.

I. INTRODUCTION

Genetic Programming (GP) [12] has been successfully used in a wide range of different challenging problems (see Koza’s article on human competitive results for a comprehensive review [13]). Despite its proven success, it also suffers from some limitations and researchers have been interested in making GP more robust, or reliable, by studying various elements of the search process (e.g., neutrality [4], [8], [9], [21], locality [5], [6], [7], special representations [3]).

One of these elements that has recently attracted the attention of researchers is the study of semantics, resulting in a dramatic increase in the number of related publications (e.g., [1], [10], [11], [14], [15], [16], [20], [23]).

Semantics is a broad concept that has been studied in different fields (e.g., natural language, psychology), making it hard to give a precise definition of the concept. Thus, in this work we adopted the popular use of semantics in GP from recent related works [10], [20], [22], [23], where researchers have used it as the difference of the raw outputs of two programs¹.

Research in this area has clearly demonstrated that the study and application of semantics in the GP process enhances its performance [10], [11], [20], [22], [23]. These studies have relied on the use of semantics at the crossover

operator in an attempt to find, over a period of trials, offspring that are semantically different from their parents. Whereas these type of methods have proven to have a superior performance than a traditional GP in terms of finding problem solutions and started shedding the importance of semantics in GP, they also suffer from one particular limitation: these are computationally expensive as a result of their trial-and-error approach [16].

The main goal of this paper is to explore the possibility of using semantics in canonical GP without the necessity of evaluating, potentially, thousands of nodes while at the same time maintaining a similar performance compared to methods based on trial-and-error. More specifically, we propose a simple and computationally inexpensive method of using semantics in the selection process, where one parent is selected by considering its fitness, while the selection of the second parent considers fitness and semantic dissimilarity w.r.t. the first selected parent. This eliminates the need of using a “brute force” mechanism to find children that are semantically different from their parents, and so, the computational cost of this new semantic-based approach, denominated *semantics in selection* (SiS), remains the same compared to a traditional GP system.

This paper is organised as follows. In Section II, we present previous work carried out in the area of semantics in GP. In Section III, we introduce our proposed approach. Section IV provides details on the experimental setup used. The results presented in this paper are discussed in Section V, and finally, conclusions and future work are drawn in Section VI.

II. RELATED WORK

McPhee et al. [15] analysed the impact of subtree crossover in terms of semantic building blocks by proposing two forms of approaches: semantics of subtrees and semantics of context. Within the context of Boolean problems, the authors were able to show the importance of diversity in GP semantics. That is, McPhee et al. pointed out how the 90%-10% crossover operator used in GP (i.e., 90%-10% internal-external node selection policy) leads to a high proportion of crossover events that do not have any useful impact in the semantic space of GP, leading to a lack of increase in performance, measured in terms of finding fitter individuals over generations.

¹In these works, semantics was originally referred as the “meaning” of programs. However, as we discuss in Section III, it might be better to re-define it as *functionality*.

Beadle and Johnson [1] proposed a crossover operator, called Semantically Driven Crossover (SDC), that promotes semantic diversity during search. More specifically, they used reduced ordered binary decision diagrams (ROBDD) on Boolean problems (i.e., Multiplexer and the even-5-parity problem) to check for semantic similarity between parents and offspring. Beadle and Johnson showed a significant improvement, in terms of increased fitness, when using SDC. Moreover, they also showed that by using ROBDD on these particular problems, the SDC operator was able to considerably reduce bloat.

Uy et al. [20] proposed four different forms of applying semantic crossover operators on real-valued scenarios (e.g., symbolic regression problems). To this end, the authors measured the semantic equivalence of two given expressions by measuring them against a random set of points sampled from the domain. If the resulting outputs of these two expressions were close to each other, subject to a threshold value called semantic sensitivity, these expressions were regarded as semantically equivalent. In their first two scenarios, Uy et al. focused their attention on the semantics of subtrees. More specifically, for Scenario I, the authors tried to encourage semantic diversity by executing, for a number of trials, crossover if two subtrees were semantically equivalent. Scenario II explored the opposite idea of Scenario I. For the last two scenarios, the authors focused their attention on full trees. That is, for Scenario III, Uy et al. checked if offspring and parents were semantically equivalent. If so, the parents were transmitted into the following generation and the offspring were discarded. The authors explored the opposite idea of Scenario III in Scenario IV (children semantically different from their parents). They showed, for a number of symbolic regression problems, that Scenario I produced better results compared to the other tree scenarios proposed by them.

Semantics has also been studied by Jackson [10], calling it phenotypic diversity. In his work, the author measured the semantics of programs based on their output. For this purpose the authors used problems from different domains (e.g., Boolean, symbolic regression, and maze-like problems). For the Boolean problems (e.g., even- n -parity problems), the author measured semantic difference between two programs based on their difference in the corresponding bits of their output strings. For the symbolic regression problem, Jackson used an approach similar to Uy's approach [20] described above. For the maze problems (e.g., artificial ant), the author kept record on the path history. The author's approach is also similar to Uy's approach based on the use of a maximum number of trials, around 20, to trying to promote semantic diversity. Jackson showed how semantic diversity promotes a better search, in terms of finding solutions more frequently compared to a traditional GP.

More recently, Moraglio et al. [16] proposed Geometric Semantic GP, where the main idea was to use it directly in the space of the underlying semantics of the potential solutions (programs). That is, the authors considered properties of se-

mantic spaces for different metrics and provided insights for designing semantically-based geometric crossover operators. They tested their approach in a variety of problems, showing how semantically different programs, produced by means of crossover, yield better results compared to standard GP, agreeing with the results obtained by other semantic-based approaches. It is also interesting to notice that the authors also reported how their approach produced bigger programs, contradicting the results found by Uy et al. [20].

Recently, Krawiec and Pawlak [14] explore the concept of Geometric Semantic GP with the key concept being that it would be ideal to produce offspring that were the semantic median of the parent programs. This means that it would be beneficial if offspring had an equal blend, or mixture, of the parents semantics. They speculate that finding offspring that meet a median measure of semantics between parents, would increase the chances that the offspring would have a higher fitness than both parents. Their approach tries to regulate the crossover effect. Where syntactic crossover may cause a huge change in semantics (or none at all), their approach seeks to provide a more uniform semantic change when offspring are created. As the creation of such offspring is hard, the authors provide preliminary evidence for this claim by inspecting a more practical measure of semantic median which is localised in homologous regions of the parent programs.

A. Final Comments on Semantics in GP

From the previous summary, it is clear that there is no a single approach to incorporate semantics in GP. For instance, the work carried out by Beadle and Johnson [1] is completely different from the work proposed by Uy et al. [20].

However, what is interesting to observe is how authors consistently report an improvement in performance in GP search (measured in terms of finding a solution more frequently) when semantics is explicitly considered.

Inspired by these approaches and their encouraging results reported in [10], [11], [20], [22], [23] and briefly summarised previously, this work continues using the same form of semantics (e.g., [10], [22]), reinforced in the following section in the context of the problems used, and proposes a simple idea to overcome the computational expensive limitation observed by these CSB trial-and-error approaches. It is worth mentioning, however, that recent works have started shedding some light on this but with other form of semantics [16] and what are referred to as behavior-based approaches [17], [18], [19].

III. SEMANTICS IN SELECTION

As can be seen from the previous section, semantics has mainly been explored in GP using crossover as the main genetic operator (and few works have also explored its use using mutation, e.g., [2]) reporting outstanding results in both continuous [10] and discrete-valued fitness functions [20]. One potential limitation on these approaches (e.g., [10], [11], [20], [22], [23]) is the fact that the authors reported the use of a maximum number of attempts, when applying crossover,

Algorithm 1 Semantics in Tournament Selection

```
1: procedure SELECTING INDIVIDUALS
2:    $parent_1 \leftarrow TournamentSelection$       ▷ Apply
   tournament selection as usual
3:    $parent_2 \leftarrow TournamentSemantics(parent_1)$ 
4: end procedure
5: procedure TOURNAMENTSEMANTICS( $parent_1$ )
6:    $semantics_{parent_1} \leftarrow Pop.getSemantics(parent_1)$ 
7:    $best \leftarrow nextInt(populationSize)$ 
8:    $fbest \leftarrow fitness[best]$ 
9:    $count \leftarrow 0$ 
10:  while  $count < tournamentSize$  do
11:     $comp \leftarrow nextInt(populationSize)$       ▷ select
    randomly a competitor from the population
12:     $semantics_{comp} \leftarrow Pop.getSemantics(comp)$ 
13:    if  $semantics_{parent_1} \langle \rangle semantics_{comp}$  then
14:      if  $fitness[comp] > fbest$  then
15:         $fbest \leftarrow fitness[comp]$ 
16:         $best \leftarrow comp$ 
17:      end if
18:    end if
19:     $count \leftarrow count + 1$ 
20:  end while
21:  return best
22: end procedure
```

to find children that are semantically different from their parents. As a consequence of this trial-and-error approach, the GP system could, potentially, evaluate many more nodes compared to a traditional GP.

In this work, we make an effort to overcome this limitation by considering semantics during the selection process without the need of using a maximum number of trials. The approach is tested in both continuous and discrete-valued fitness cases by using well-known GP benchmark problems (i.e., Artificial Ant, Even- n -Parity and Symbolic Regression problems introduced in Section IV).

Before explaining the approach, it is important to indicate how we measure semantics, which is based on previous works reported by [10], [20], where the authors defined semantics as the meaning of syntactically correct programs. We believe that it is better to define semantics as the *functionality* of programs (raw outputs). The main reason is because in these works, the authors measured semantics diversity by the difference of the outputs of two GP individuals when their instructions are executed. This is explained next for each of the problems used in this work.

For the case of the Artificial Ant problem, we keep track of the individual's semantics by recording the movements produced by the execution of the program. Thus, every time the ant moves to a different square, we record where the ant is facing (i.e., north, east, south, west) in a vector. For this particular problem, we regard two individuals to be semantically different if their output vectors are different, they are semantically similar otherwise.

For the Even- n -Parity problems ($n = \{3, 4, 5\}$), the semantics of an individual is measured in terms of the output it produces. More specifically, we keep a record of the result that each fitness case produces in a vector of size 2^n . So, we regard two individuals to be semantically different if their output vectors are different, they are considered semantically similar otherwise.

Finally, for the Symbolic Regression problems, we again keep track of the semantics of an individual in terms of the output it produces, as in the Even- n Parity problems. The main difference is that in this continuous-valued fitness function problem we also use a threshold value, ($\alpha = 0.01$), to indicate if two individuals are semantically different. That is, in a vector of size f_c , where f_c is the number of fitness cases used, we check whether the absolute differences between corresponding outputs lie within α . Thus, we regard two individuals to be semantically different if for each corresponding value contained in the vector the difference is greater than the threshold value α , they are considered semantically similar otherwise.

As indicated before, researchers have reported an improvement in the performance of a GP system by encouraging semantic diversity (e.g., parent and offspring being semantically different). In this work, instead of promoting it at the crossover level, where the application of this operator is repeated until the offspring is semantically different than their parents or until a maximum number of trials (e.g., $n_{max} = 20$) are executed, whatever occurs first [10], [11], [20], [22], [23] we encourage semantic diversity at the selection operator, in this case using tournament selection.

That is, we select the first parent in the typical way: we define a pool of t_{size} individuals, and for a maximisation problem, the one with the highest fitness is chosen to be used in the crossover operator. The selection of the second parent is chosen by considering both: fitness and the semantic difference from the first selected parent. More specifically, the second parent is chosen from a pool of t_{size} individuals that is semantically different, as explained in the previous paragraphs, from the first parent and that has the highest fitness value. Algorithm 1 describes this idea in detail. For problems where the goal is to minimise, the method works the same, with the difference that the individuals with the lowest fitness is selected.

The motivation behind this idea, is that, by having two parents that are not only fit but also semantically different, their offspring obtained via crossover, could increase the probability of producing semantically different individuals without the necessity of using a trial-and-error approach, eliminating the number of unnecessary evaluated nodes incurred by applying crossover n_{max} of times. We further discuss this in Section V.

IV. EXPERIMENTAL SETUP

For our analysis, we have used 14 GP benchmark problems: the Artificial Ant Problem [12], the Even- n -Parity ($n = \{3, 4, 5\}$) problem (problems that require the combination of several XOR functions, and are difficult if no

TABLE I
SUMMARY OF PARAMETERS.

Parameter	Value
Population Size	126, 250, 500
Generations	200, 100, 50
Type of Crossover	Any node (used in each of the 3 approaches)
Crossover Rate	100
Mutation	None
Selection	Tournament (size = 7)
Initialisation Method	Ramped half-and-half
Initialisation Depths:	
Initial Depth	5
Final Depth	7
Maximum Length	9000 nodes
Maximum Final Depth	9
Maximum Trials for CSB	20
Independent Runs	100

bias favorable to their induction is added in any part of the algorithm), and Real-Valued Symbolic Regression problems (with 10 different target functions, as indicated in the left-most column of Table II).

The first problem, the Artificial Ant Problem [12, pp. 147–155], consists of finding a program that can successfully navigate an artificial ant along a path of 89 pellets of food on a 32 x 32 toroidal grid. When the ant encounters a food pellet, its (raw) fitness increases by one, to a maximum of 89. The problem is in itself challenging for many reasons. The ant must eat all the food pellets (normally in 600 steps) scattered along a twisted track that has single, double and triple gaps along it. The terminal set used for this problem is $T = \{Move, Right, Left\}$. The function set is $F = \{IfFoodAhead, P2, P3\}$.

The second, third and fourth problems are Boolean Even- n -Parity problems ($n = \{3, 4, 5\}$) where the goal is to evolve a function that returns true if an even number of the inputs evaluate to true, and false otherwise. The maximum fitness for this type of problem is 2^n . The terminal set is the set of inputs. The function set is $F = \{AND, OR, NOT\}$.

The rest of the problems are real-valued symbolic regression problems. The goal of this type of problem is to find a program whose output is equal to the values of functions. Thus, the fitness of an individual in the population reflects how close the output of an individual comes to the target (F_1, \dots, F_{10}) (see the left-most column of Table II). It is common to define the fitness as the sum of absolute errors measured at different values of the independent variable x , in this case in the range $[-1.0, 1.0]$. In this study we have measured the errors for $x, y \in \{-1.0, -0.9, -0.8 \dots 0.8, 0.9, 1.0\}$. We have defined an arbitrary threshold of 0.01 to indicate that an individual with a fitness less than the threshold is regarded as a correct solution, i.e. a “hit”. The function set is $F = \{+, -, *, /, Sin, Cos, Exp, LOG\}$, where $/$ is protected division. We used the same threshold, 0.01, to indicate whether two individuals are semantically different of similar

($\alpha = 0.01$), as explained in Section III.

To evaluate our proposed approach, *semantics in selection* (SiS), and for comparison purposes, we implemented two other methods: a traditional GP system and a crossover-semantics based approach, referred as GP and CSB, respectively. The CSB tries to promote semantic difference at the crossover operator with a maximum number of trials, as described in Section III (see [20] for details).

The experiments were conducted using a steady state approach with tournament selection and the traditional crossover operator for each of the three approaches used. The rest of the parameters used are shown in Table I. To obtain meaningful results, we performed extensive empirical experimentation (100 * 42 * 3 runs in total)².

V. RESULTS AND ANALYSIS

A. Performance Comparison

Let us start by analysing the performance, measured in terms of percentage of runs that found a solution, for the first combination of Population Size = 126 and Generations = 200 for each of the three approaches used, shown in the second, third and fourth column of Table II, for each of the 14 problems used in this study. For the first problem, Artificial Ant, there is very little to say because all three approaches, GP, CSB and SiS, behave equally bad. That is, none of them was able to find a solution. For the Even-3-Parity Problem, there is no difference in performance between the approaches, because all the approaches were able to find the solution all the time (100% success rate). For the Even-4-Parity Problem, the situation is clearer. In this problem, the semantic-based approaches are much better compared to the traditional GP system: 19, 60, 58 success rate for GP, CSB and SiS, respectively. For the Even-5-Parity problem, all approaches have a poor performance, with the difference being that SiS is able to find a solution, although very few times, compared to GP and CSB where none of them was able to solve the problem. For the last type of problems, Symbolic Regression, shown in the last 10 rows of Table II, where for some functions (i.e., F_1, F_2, F_3, F_4, F_6), the semantic-based approaches shown superior performance over the GP approach. It is clear how semantics consistently improves performance compared to GP without semantics. For other Symbolic Regression problems (e.g., F_5, F_9, F_{10}) the situation is less clear, because the performance of all three approaches is more or less similar. Finally, it is interesting to see, how for almost all the problems used, both methods based on semantics take longer (number of generations are indicated within parenthesis in Table II) to find a solution compared to traditional GP (except for the Even-4 parity problem).

Let us now turn our attention to the second configuration of Population Size = 250 and Generations = 100 for each of

²100 independent runs, 42 (i.e., three different combination of population sizes and number of generations that results, more or less, in the same number of evaluations, and 14 different problems,) and three approaches (GP, CSB and SiS).

TABLE II

SUCCESS RATE OVER 100 INDEPENDENT RUNS AND THE AVERAGE NUMBER OF GENERATIONS WHEN THE PROBLEM WAS SOLVED INDICATED WITHIN PARENTHESIS, USING THREE DIFFERENT COMBINATIONS OF POPULATION SIZES AND NUMBER OF GENERATIONS, FOR EACH OF THE THREE APPROACHES USED: GP, CROSSOVER-SEMANTICS BASED (CSB) AND SEMANTICS IN SELECTION (SiS). HIGHEST SUCCESS RATES ARE HIGHLIGHTED IN BOLDFACE.

	Population Size = 126 Generations = 200			Population Size = 250 Generations = 100			Population Size = 500 Generations = 50		
	GP	CSB	SiS	GP	CSB	SiS	GP	CSB	SiS
Artificial Ant	0 -	0 -	0 -	0 -	1 (36)	0 -	2 (15.5)	1 (3)	0 -
Even-3-Parity	100 (9.29)	100 (5.04)	100 (5.85)	95 (15.64)	99 (6.53)	99 (11.15)	100 (5.93)	100 (4.28)	100 (4.49)
Even-4-Parity	19 (85.47)	60 (76.03)	58 (92.34)	11 (55.18)	48 (43.94)	49 (56.86)	32 (26.5)	70 (22.76)	74 (24.61)
Even-5-Parity	0 -	2 (113.5)	1 (174)	0 -	0 -	3 (90.67)	0 -	0 -	3 (33)
$F_1 = x^3 + x^2 + x$	28 (3.19)	27 (14.26)	50 (19.46)	22 (6.18)	46 (7.11)	46 (10.65)	64 (3.69)	71 (3.86)	88 (4.47)
$F_2 = x^4 + x^3 + x^2 + x$	17 (10.53)	31 (18.48)	16 (17.12)	14 (7.14)	29 (8.72)	16 (17.81)	16 (5.5)	31 (5.74)	44 (7.98)
$F_3 = x^6 + x^5 + x^4 + x^3 + x^2 + x$	10 (4.8)	23 (19.22)	23 (36.65)	8 (15.88)	16 (13.06)	14 (14.93)	15 (7.4)	20 (6.6)	27 (8.41)
$F_4 = \sin(x^2)\cos(x) - 1$	0 -	10 (19.9)	4 (46)	2 (59)	4 (16.5)	2 (27)	2 (4.5)	7 (7.57)	10 (17.82)
$F_5 = \sin(x) + \sin(x + x^2)$	2 (26.5)	0 -	1 (30)	1 (12)	0 -	4 (69.5)	0 -	1 (14)	7 (19.71)
$F_6 = \log(x + 1) + \log(x^2 + 1)$	12 (47.92)	17 (46.71)	24 (48.62)	14 (14)	18 (13.28)	24 (18.17)	22 (6)	30 (12.80)	22 (14.59)
$F_7 = \text{sqrt}(x)$	0 -	2 (8.5)	1 (5)	0 -	2 (9)	3 (14.67)	3 (3.67)	1 (7)	4 (16.75)
$F_8 = \sin(x) + \sin(y^2)$	0 -	1 (6)	6 (59.83)	4 (4.25)	6 (29.5)	4 (13)	0 -	0 -	0 -
$F_9 = 2\sin(x)\cos(y)$	0 -	3 (69.67)	3 (7.33)	2 (51)	3 (33)	2 (10)	5 (28)	7 (11.71)	6 (23)
$F_{10} = x^y$	2 (32)	1 (10)	0 -	2 (12.67)	0 -	1 (64)	2 (13)	4 (6.5)	4 (20)

the three approaches used in this study, shown in the fifth, sixth and seventh column of Table II. As before, there is very little to say for the Artificial Ant problem because the three approaches behave equally bad. For the Even- n -Parity problem there are some significant differences between the GP approach and the approaches based on semantics. In particular, for the Even-4-Parity problem where the results found by the semantic-based approaches (i.e., CSB and SiS) are much better, around four times better, compared to GP. A similar trend is observed in some Symbolic Regression functions (e.g., F_1, F_2, F_3, F_6), although the difference in performance is not as impressive as in the case of the Even-4-Parity problem, where the performance increased is, in average, the double compared to the GP approach. For other functions (e.g., F_5, F_{10}) the three approaches behave equally bad as very few runs were able to find a solution for these problems.

For the last combination of population size and number of generations, 500 and 50, respectively, the same trend is observed for the three approaches used and the 14 benchmark problems used in this study.

That is, the semantic-based approaches are consistently better compared to the GP approach (e.g., Even-4-Parity, F_1, F_2, F_3, F_4) regardless of the combination used for the

population size and number of generations.

B. Crossover-Semantics Based vs. Semantics in Selection

From the results reported in Table II, it is clear how both semantic-based approaches outperformed the performance of a GP system, where in some cases, these approaches were four times better than the latter approach.

The performance shown by the CSB approach (see Table II) agrees with the results previously reported by Jackson [10] and Uy et al. [20] where the authors reported excellent results when using semantics at the crossover level for trying to find semantically different children (using a maximum number of attempts). As a consequence of the latter, the GP system could evaluate dozens of thousands of nodes resulting in a highly computationally expensive process.

The main benefit of our approach (SiS) is that it does not suffer from the flaw of needing to, potentially, evaluate thousands of nodes (we discuss this in the following paragraphs). Moreover, our proposed approach is equivalent, and in some cases superior, in performance compared to the CSB approach, as discussed above.

Now, let us focus our attention on the number of evaluated nodes by CSB and SiS, shown in Figure 1 (notice that due

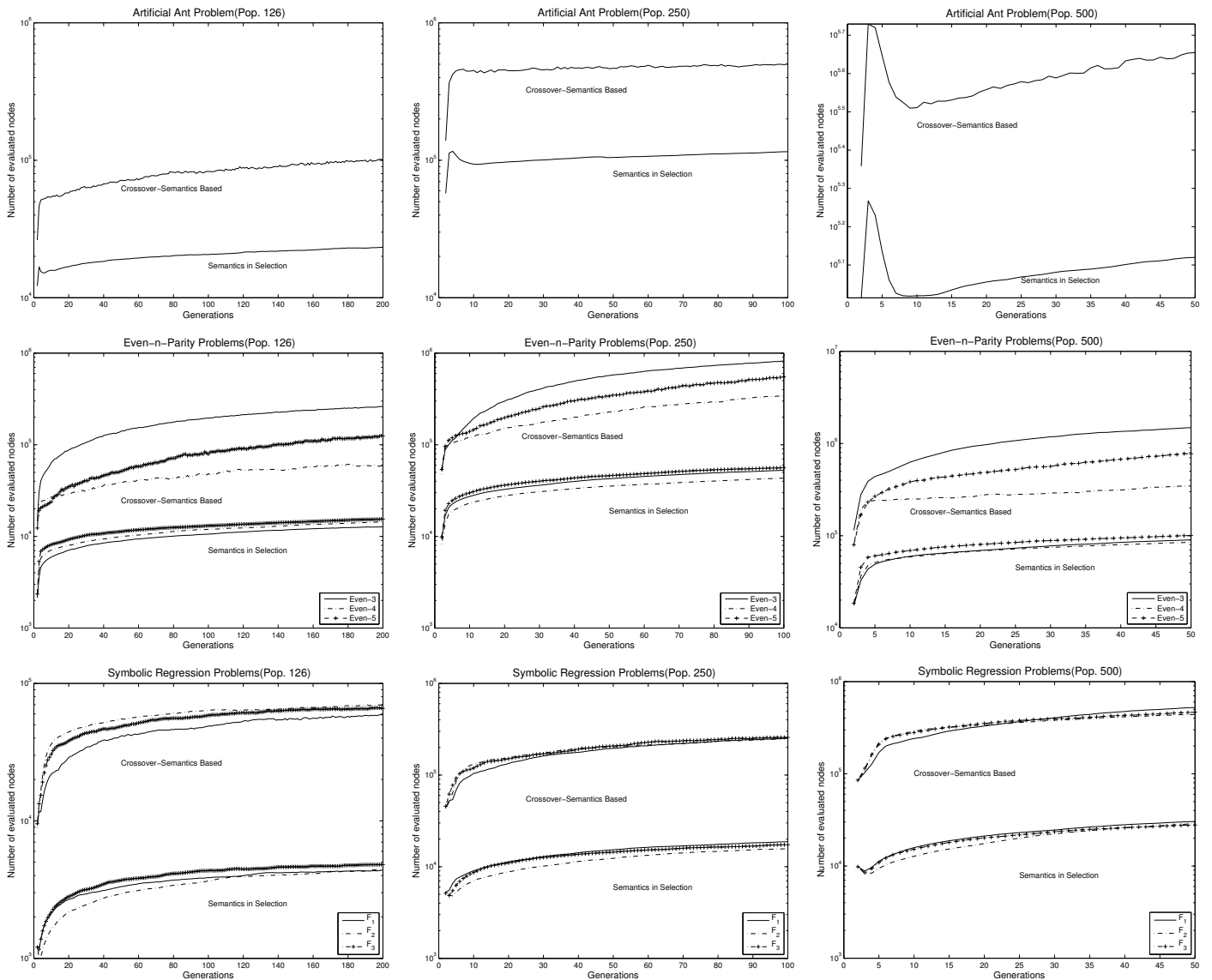


Fig. 1. Number of evaluated nodes for the two semantic-based approaches: crossover-semantics based and semantics in selection, reported for the Artificial Ant Problem (top row), Even- n -Parity Problems (middle row) and for functions F_1, F_2, F_3 , for each of the three different combinations of population sizes (PopSize) and number of generations (Gen) used: PopSize = 126, Gen = 200 (left-most side); PopSize = 250, Gen = 100 (centre), and PopSize = 500 and Gen = 50 (right-most side).

to space restrictions and for clarity purposes, we plotted the evaluated nodes for the Artificial Ant, Even- n -Parity and the first three Symbolic Regression problems). From the plots shown in Figure 1, it is clear that our proposed approach (SiS) evaluates a much lower number of nodes compared to the other semantic approach (crossover-based). This is to be expected since, as we have discussed in Section III, the latter approach executes an exhaustive search via crossover, with a maximum number of trials, to find children that are semantically different from their parents. It is also interesting to notice how there seems to be a positive correlation between the size of the population and the number of nodes evaluated by the crossover-semantics based approach.

One might think that the large number of evaluated nodes

in the crossover-based approach could be the result of bloat (growing of an individual at a rapid pace) rather than performing an extensive search using the crossover operator. Thus, to show that this is not the case, we have measured the average number of trials needed by the CSB approach. This is shown in Figure 2. These plots confirm our previous findings: the results of evaluating dozens of thousand of nodes in the referred approach is the result of trying to find, via the crossover operator, offspring that are semantically different from their parents over a number of trials (in this study set at 20, as indicated in Table I).

The number of trials used by the CSB approach vary according to the problem. For instance for the Artificial Ant problem (shown at the top of Figure 2), around three attempts

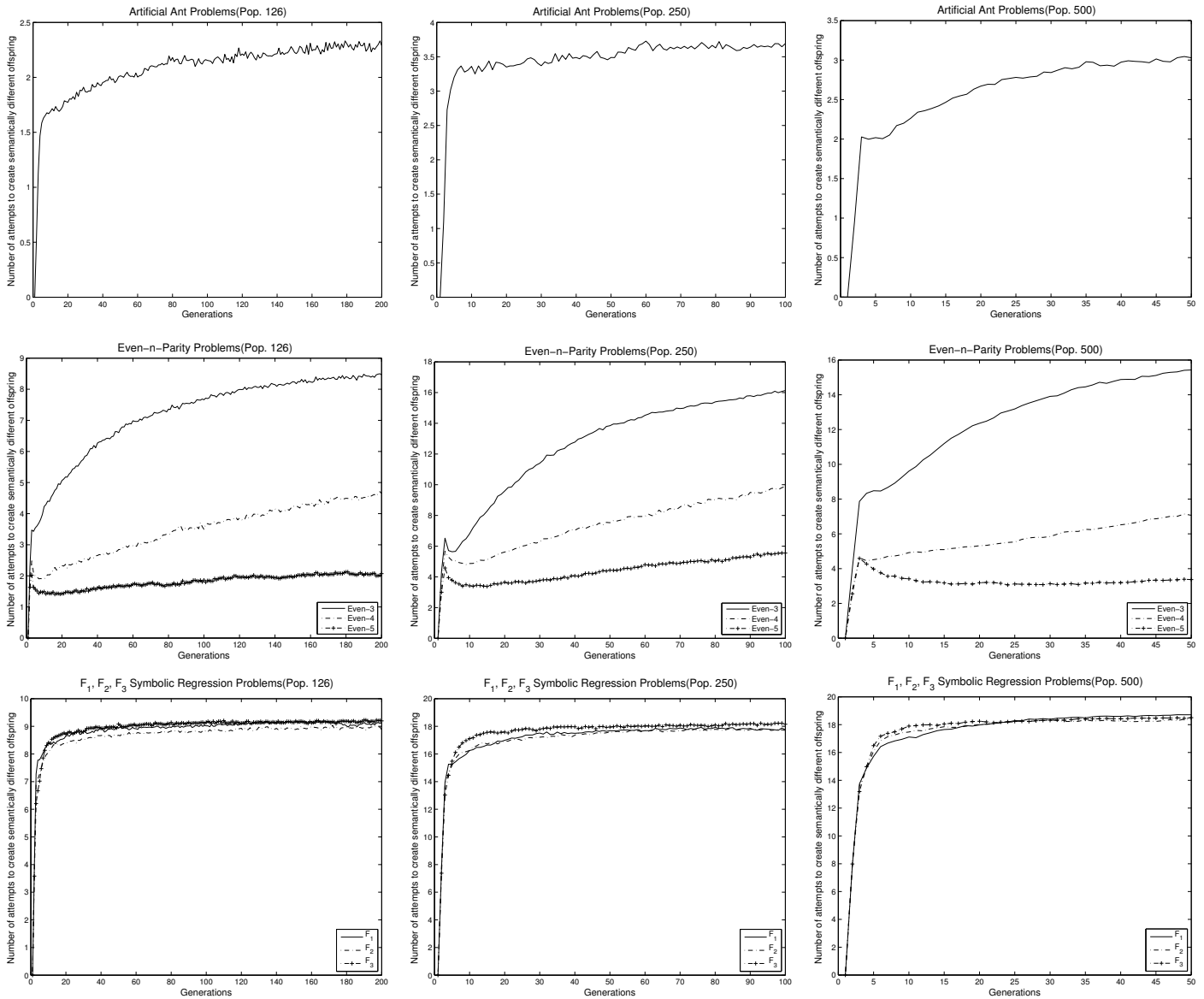


Fig. 2. Number of average trials for the crossover-semantic based approach required to find semantically different offspring compared to their parents, reported for the Artificial Ant Problem (top row), Even- n -Parity Problems (middle row) and for functions F_1, F_2, F_3 (bottom row), for each of the three different combinations of population sizes (PopSize) and number of generations (Gen) used: PopSize = 126, Gen = 200 (left-most side); PopSize = 250, Gen = 100 (centre), and PopSize = 500 and Gen = 50 (right-most side).

are necessary, regardless of the size of the population size and number of generations defined.

For the Even- n -Parity problem, this number varies. When $n = 3$ it requires a higher number of trials to find children that are semantically different from their parents and it decreases as n increases. This is to be expected since there are more chances to find children that are semantically different from their parents the larger the number of fitness cases used, because as explained in Section III, two individuals are regarded semantically different if their output vectors are different. What is also interesting to point out in this type of problems is that, the number of trials increases for the Even-4 and Even-5 problems as the population size increases

too, indicating that by having a bigger population size, does not necessarily imply that it will be easier to find children semantically different from their parents.

The same trend is observed for the Symbolic Regression problems (for clarity purposes, we again only plotted F_1, F_2, F_3 , see bottom of Figure 2). That is, the number of trials increases as the population size increases too, until a certain limit, though. For example, when using 126 and 250 individuals, the number of trials is almost double, 9 and 18, respectively. This increase is not observed when using 250 and 500 individuals, where the number of trials remain more or less the same when using these two population sizes (around 18 trials).

VI. CONCLUSIONS AND FUTURE WORK

Over the last years, GP systems that explicitly consider semantics in their mechanism, have demonstrated to have a superior performance compared to a traditional GP approach. In this work, we have presented a simple and computationally inexpensive approach to use semantics in GP, called semantics in selection, that eliminates the necessity of, potentially, evaluating dozens of thousands of nodes during evolution compared to semantic-based approaches that try to promote semantic diversity at the crossover operator using an expensive trial-and error approach (e.g., [10], [11], [20], [23]), referred in this work as crossover-semantics based approach.

Thus, by using semantics in selection we guarantee that the computational effort of the GP system remains the same. To test the efficiency of this approach, we used 14 GP benchmark problems, including both continuous- and discrete-valued fitness functions, and compare the results using a traditional GP and a crossover-semantics based approach.

The semantics in selection approach proposed in this paper has shown promising results, in many cases achieving superior results compared to the crossover-semantics based approach. We will extend and refine our approach to explore any further benefits. For instance, the comparison semantic difference used with Boolean problems and the Ant problem can be adjusted, by using well-define metrics (e.g., Hamming distance). As for the Symbolic Regression problems, a threshold could be dynamically adjusted depending on the progress of the GP search.

ACKNOWLEDGMENTS

The first author wishes to acknowledge Science Foundation Ireland (SFI) under the Principal Investigator research program 10/IN.1/I2980 “Self-organizing Architectures for Autonomic Management of Smart Cities” and by SFI grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie) for their support to conduct this research. The first and third author wish to acknowledge funding for this work provided by CONACYT (Mexico) Basic Science Research Project No. 178323. Finally, the third author acknowledges DGEST (Mexico) Research Project No. TIJ-ING-2012-110.

REFERENCES

- [1] 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.
- [2] L. Beadle and C. Johnson. Semantically driven mutation in genetic programming. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 1336–1342, May 2009.
- [3] E. Galván-López. Efficient graph-based genetic programming representation with multiple outputs. *International Journal of Automation and Computing*, 5(1):81–89, 2008.
- [4] E. Galván-López, S. Dignum, and R. Poli. The effects of constant neutrality on performance and problem hardness in gp. In *Proceedings of the 11th European conference on Genetic programming*, EuroGP'08, pages 312–324, Berlin, Heidelberg, 2008. Springer-Verlag.
- [5] E. Galván-López, J. McDermott, M. O'Neill, and A. Brabazon. Defining locality in genetic programming to predict performance. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2010.

- [6] E. Galván-López, J. McDermott, M. O'Neill, and A. Brabazon. Towards an understanding of locality in genetic programming. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 901–908, NY, USA, 2010. ACM.
- [7] E. Galván-López, J. McDermott, M. O'Neill, and A. Brabazon. Defining locality as a problem difficulty measure in genetic programming. *Genetic Programming and Evolvable Machines*, 12(4):365–401, 2011.
- [8] E. Galván-López and R. Poli. An empirical investigation of how and why neutrality affects evolutionary search. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 1149–1156, New York, NY, USA, 2006. ACM.
- [9] E. Galván-López and R. Poli. Some steps towards understanding how neutrality affects evolutionary search. In T. Runarsson, H.-G. Beyer, E. Burke, J. Merelo-Guervos, L. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 778–787. Springer Berlin Heidelberg, 2006.
- [10] D. Jackson. Phenotypic diversity in initial genetic programming populations. In A. I. Esparcia-Alcázar, A. Ekárt, S. Silva, S. Dignum, and A. S. Etnaner-Uyar, editors, *EuroGP*, volume 6021 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2010.
- [11] D. Jackson. Promoting phenotypic diversity in genetic programming. In R. Schaefer, C. Cotta, J. Kolodziej, and G. Rudolph, editors, *PPSN (2)*, volume 6239 of *Lecture Notes in Computer Science*, pages 472–481. Springer, 2010.
- [12] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, Massachusetts, 1992.
- [13] J. R. Koza. Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, 11(3-4):251–284, Sept. 2010.
- [14] 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.
- [15] 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.
- [16] A. Moraglio, K. Krawiec, and C. G. Johnson. Geometric semantic genetic programming. In C. A. C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *PPSN (1)*, volume 7491 of *LNCS*, pages 21–31. Springer, 2012.
- [17] E. Naredo, Y. Martínez, L. Trujillo, and E. Galván-López. Searching for novel regression functions. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2013, Cancun, Mexico, 20-23 June 2013*. IEEE, 2013.
- [18] E. Naredo and Trujillo. Searching for novel clustering programs. In *Proceeding from the Genetic and Evolutionary Computation Conference, GECCO 2013*. ACM, 2013.
- [19] E. Naredo, L. Trujillo, and Y. Martínez. Searching for novel classifiers. In *Proceedings from the 16th European Conference on Genetic Programming, EuroGP 2013*, volume 7831 of *LNCS*, pages 145–156. Springer-Verlag, 2013.
- [20] Q. U. Nguyen, X. H. Nguyen, and M. O'Neill. Semantic aware crossover for genetic programming: The case for real-valued function regression. In *Proceedings of the 12th European Conference on Genetic Programming, EuroGP '09*, pages 292–302, Berlin, Heidelberg, 2009. Springer-Verlag.
- [21] R. Poli and E. Galván-López. The effects of constant and bitwise neutrality on problem hardness, fitness distance correlation and phenotypic mutation rates. *IEEE Trans. Evolutionary Computation*, 16(2):279–300, 2012.
- [22] 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.
- [23] N. Q. Uy, M. O'Neill, N. X. Hoai, B. McKay, and E. Galván. Semantic similarity based crossover in gp: The case for real-valued function regression. In P. Collet, N. Monmarché, P. Legrand, M. Schoenauer, and E. Lutton, editors, *Artificial Evolution*, volume 5975 of *Lecture Notes in Computer Science*, pages 170–181. Springer, 2009.