

# Semantics in Multi-objective Genetic Programming

Edgar Galván<sup>a,\*</sup>, Leonardo Trujillo<sup>b</sup>, Fergal Stapleton<sup>a</sup>

<sup>a</sup> *Naturally Inspired Computation Research Group, Department of Computer Science, Hamilton Institute, National University of Ireland Maynooth, Lero, Ireland*

<sup>b</sup> *Tecnológico Nacional de México/IT de Tijuana, Tijuana, BC, Mexico*

## ARTICLE INFO

### Article history:

Received 28 April 2021

Received in revised form 1 October 2021

Accepted 2 November 2021

Available online 26 November 2021

### Keywords:

Multi-objective Genetic Programming

Semantics

Diversity

NSGA-II

SPEA2

## ABSTRACT

Semantics has become a key topic of research in Genetic Programming (GP). Semantics refers to the outputs (behaviour) of a GP individual when this is run on a dataset. The majority of works that focus on semantic diversity in single-objective GP indicates that it is highly beneficial in evolutionary search. Surprisingly, there is minuscule research conducted in semantics in Multi-objective GP (MOGP). In this work we make a leap beyond our understanding of semantics in MOGP and propose SDO: Semantic-based Distance as an additional criterion. This naturally encourages semantic diversity in MOGP. To do so, we find a pivot in the less dense region of the first Pareto front (most promising front). This is then used to compute a distance between the pivot and every individual in the population. The resulting distance is then used as an additional criterion to be optimised to favour semantic diversity. We also use two other semantic-based methods as baselines, called Semantic Similarity-based Crossover and Semantic-based Crowding Distance. Furthermore, we also use the Non-dominated Sorting Genetic Algorithm II and the Strength Pareto Evolutionary Algorithm 2 for comparison too. We use highly unbalanced binary classification problems and consistently show how our proposed SDO approach produces more non-dominated solutions and better diversity, leading to better statistically significant results, using the hypervolume results as evaluation measure, compared to the rest of the other four methods.

© 2021 Elsevier B.V. All rights reserved.

## 1. Introduction

Genetic Programming [1], one of the four canonical Evolutionary Algorithms paradigms, was popularised by Koza in the early 1990s. Over the years, researchers have been interested in making GP more amenable to evolutionary search. A key element that has been proven to make GP more robust is semantics. The latter has become a key topic of research in GP. Semantics can be seen as the behaviour of a GP program. This behaviour is the output of a GP program when executed on a set of fitness cases.

The number of scientific publications in GP semantics has increased significantly thanks to promising results found by the research community. We discuss in Section 2 some relevant works of semantics in GP. Interestingly, the vast majority of these work have concentrated on Single-objective GP (SOGP), with minuscule progress in Multi-objective GP (MOGP), with the exception of [2–5]. Thus, this scientific work extends significantly this line of research and uses three forms of semantics in a MOGP setting. Each of these are compared independently against

two well-established Evolutionary Multi-objective Optimisation (EMO) approaches: the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [6] and the Strength Pareto Evolutionary Algorithm (SPEA2) [7]. The semantic-based MOGP approaches used in this study are:

**Semantic Similarity-based Crossover. (SSC).** This is motivated by the SOGP approach presented in [8]. This approach was one of the early methods in SOGP semantics where the authors were able to promote it in continuous search spaces. We extended this well-known method in MOGP.

**Semantic-based Crowding Distance. (SCD).** Here, the main idea is to replace the crowding distance, commonly used in EMO algorithms, by a semantic-based distance, originally studied in the first author's MOGP works [3,4].

**Semantic-based Distance as an additional criterion. (SDO).** This approach draws from SCD and uses the resulting semantic distance as another component to optimise by an EMO algorithm, briefly studied in [5].

\* Corresponding and leading author.

E-mail addresses: [edgar.galvan@mu.ie](mailto:edgar.galvan@mu.ie) (E. Galván), [leonardo.trujillo@tectijuana.edu.mx](mailto:leonardo.trujillo@tectijuana.edu.mx) (L. Trujillo), [fergal.stapleton.2020@mumail.ie](mailto:fergal.stapleton.2020@mumail.ie) (F. Stapleton).

Using these three semantic-based methods allow us to show the following:

**Firstly**, by using SSC, we show how the semantic distance computed in the crossover operator and used to successfully promote semantic diversity in single-objective GP does not have the same positive impact in MOGP.

**Secondly**, by using SCD, inspired by the crowding distance commonly used in EMO, we show that a semantic distance can be naturally computed between every GP tree in the population and a “pivot”. The latter is an individual in the sparsest region of the first Pareto front. SCD then moves away from SSC, which tries to promote semantic diversity by forcing diversity to emerge by using crossover repeatedly, as proposed in [8].

**Finally**, we build from our understanding drawn from SSC and SCD in semantics and propose a robust mechanism for the emergence of semantic diversity in MOGP. Particularly, we use the semantic distance value as an additional indicator to evolve the population. This naturally promotes semantic diversity in MOGP leading to better, statistically significant results based on the average hyper-volume of the evolved Pareto approximations with respect to the other four methods (two semantic-based methods and two EMO methods) in a range of highly imbalanced datasets.

### 1.1. Main contributions of this scientific study

In our previous work [5], we carried out an initial limited study on semantics in Multi-objective Genetic Programming (MOGP). Specifically, we initially proposed and used three semantic-based methods, named Semantic Similarity-based Crossover (SSC), Semantic-based Distance as an additional criterion (SDO) and Pivot Similarity Semantic-based Distance as an additional criterion (PSDO).

The main conclusions from our initial investigation is that the use of a semantic-based distance value as computed in either SDO or PSDO to be used as another objective to be optimised in an EMO setting is robust enough to outperform the results yield by the well-known NSGA-II and SPEA2 approaches. Furthermore, in our initial research, we found out that the distance computed from a pivot, which is the furthest point in the search space, to every individual in the population and used as an additional criterion to be optimised in a EMO setting tends to improve the performance of our semantic-based approaches. Moreover, we were able to fine-tune how this distance can be computed to significantly improve the evolutionary search. This is attained using the SDO approach, which is used again in this work. It is, however, worth saying that these conclusions were drawn from an initial limited study including a restricted statistical analysis impeding drawing general conclusions, limited results as well as a lack of explanation that help us to clearly indicate why SDO yields better results compared to their respective canonical methods as well as the other two semantic-based approaches. Furthermore, in our initial study [5], we omitted to discuss the limitations of SDO.

In this work, we have addressed all these issues. More specifically, the main contributions of this scientific study are as follows:

- We consistently show how Semantic Similarity-based Crossover (SSC) used in single-objective GP and widely reported to be beneficial in GP does not have the same positive impact in a multi-objective GP (MOGP) setting.

- From this, we show how a semantic-based distance approach can enhance the evolutionary search in MOGP. To this end we use two semantic-based approaches: Semantic-based Crowding Distance (SCD) and a Semantic-base Distance as an additional criterion (SDO).
- We demonstrate how SDO yields better results against all the approaches used in this work, including the semantic-based methods and canonical EMO approaches.
- Another major contribution of this scientific study is to include detailed results using two well-established EMO approaches NSGA-II and SPEA2. By doing so, as opposed to the limited results reported in [5], we are now in a position to draw sound conclusions by carrying out a systematic statistical analysis, explained in detail in Section 6.
- Another important contribution in this work is that we are able to explain why the semantic-based technique employed in SDO tends to improve evolutionary search. We do so by extensively analysing the behaviour of the SDO in terms of number of unique solutions, duplicate frequency of solutions over generations, etc.
- Finally, another major contribution of this work is the discussion of the limitations of this work by using a Multi-objective Evolutionary Algorithm Based on Decomposition [9].

This work is organised and presented as follows. Relevant studies to this work are presented in Section 2. The fundamental background in semantics and in MOPG is discussed in Section 3. Section 4 presents the MOGP semantic methods proposed and used in this work. The setup of experiments is presented in Section 5. Section 6 presents in detail the results yield by all the MOGP semantic approaches (SSC, SDO and SCD) and by the EMO methods (NSGA-II and SPEA2). It also offers an explanation as to why SDO finds better results compared to all the other algorithms. Section 7 discusses the limitations of SDO in Multi-Objective Evolutionary Algorithms based on Decomposition. In Section 8, we draw some conclusions.

## 2. Relevant work

### 2.1. Semantics

Semantics has become a key topic of research in GP and multiple definitions have been proposed. Semantics can be seen as the *behaviour* (recorded outputs over a dataset) of a GP program. We give a formal definition of semantics in Section 3. Research in semantics in GP has grown substantially in the last decade as a consequence of the research community reporting better results when semantics has been promoted in evolutionary search as compared to those GP approaches that do not promote it explicitly. The focus of these studies range from dealing with direct semantic methods, such as the use of geometric operators [10], to the analysis of indirect semantic methods [3,8]. Next, we discuss some relevant works in this area.

The analysis of McPhee et al. [11] laid the foundations for indirect semantic works. Their research focused on analysing the semantics of program subtrees and contexts (a context being the remaining portion of a program after a subtree has been removed) for Boolean problems. A key outcome from this research demonstrated that the commonly used ratio 90–10 crossover produced a high proportion of individuals which were semantically equivalent. In other words, the majority of crossover events do not result in an effective search of the semantic space and consequently limits the potential performance benefits of performing this operation.

To overcome this, Beadle and Johnson [12] proposed an operator that would help promote semantic diversity dubbed Semantically Driven Crossover (SDC). To verify whether parents and corresponding offspring are equivalent or not, the authors used reduced ordered binary decision diagrams to this end. In their studies, the authors applied crossover multiple times when both parents and offspring were semantically equivalent. Later, Beadle and Johnson [13] also explored a similar technique for the mutation operator, called Semantically Driven Mutation (SDM). This verifies the semantic equivalence between a parent and a offspring, where the latter is generated by replacing a subtree with another randomly generated subtree. This equivalence analysis is carried out by reducing the offspring to a canonical form which then can be easily compared to its parent, that has also been reduced. They reported that both techniques increased semantic diversity and led to improved evolutionary search.

A drawback of these methods is that they use discrete fitness-value cases, hence limiting their applicability to continuous search spaces. Uy et al. [14] overcame this limitation by using a motivational approach which involves employing ingenious semantic crossover operators in continuous search spaces. The semantics are approximated by evaluating a predefined sample of points from a given problem domain. The semantic equivalence therefore of two expressions (two trees or two subtrees) can be calculated from the absolute difference between the outputs of these expressions. If the difference of these two expressions fall within the bounds of a predefined threshold (a parameter referred to as Semantic Sensitivity) then these expressions can be deemed to be semantically equivalent. Uy et al. proposed four different scenarios for implementing this form of semantics, the first two of which dealt with the semantics of sub-trees. Scenario I sought to promote semantic diversity by checking the semantic equivalence of subtrees used in a crossover operation and if they were deemed to be equivalent, the parents were retained. Otherwise, the crossover was applied again, but now using two different randomly selected crossover nodes. In Scenario II, the subtrees are picked if these are equivalent to each other. The later two scenarios consider semantics by taking into consideration the full program trees. Scenario III checks the semantics of the parents in relation to their offspring. That is, if the child and parent trees are found to be semantically equivalent the offspring are discarded and the parent is retained into the next generation. Scenario IV is the same as Scenario III but in this case the condition is reversed, in other words the offspring are retained into the next generation if the semantic equivalence criteria is met. Scenario I was shown to produce better results on symbolic regression problems when compared with the other three methods.

There is no guarantee that a semantically equivalent offspring will be found immediately after performing the crossover operation and an extension of the above method introduced a trial and error mechanism to perform the crossover operation multiple times until a suitable candidate is found or until some predefined iteration is reached [8]. However, a significant downside is that the Uy et al. method can be computationally high-intensive. To address this drawback, Galván et al. put forward a cost-effective approach by trying to promote semantic diversity through the tournament selection operator. The first parent is selected as usual. The second parent is selected by considering: semantic dissimilarity and fitness, in that order. If there is no individual that is dissimilar to the first parent, then the second parent is selected as commonly done in tournament selection. The semantic dissimilarity is obtained in the same manner as done by Uy et al. [14] and described above in Scenario I. This cost-effective approach performs similar to Uy et al. method with the benefit of removing the expensive trial and error mechanism.

Forstenlechner et al. [15] investigated the use of semantics for program synthesis, a different domain from those discussed before (Symbolic Regression and Boolean problems). A variety of different data types are used in this approach as opposed to just a single data type. The semantics of two GP subtrees are stored in a pair of vectors. In their work, the authors used two conditions for checking semantic similarity (1) 'Partial change' which denotes a vector whose semantics have stayed the same for at least one entry but have changed for another entry; and (2) 'Any change' which denotes a vector where any entry is different. Two subtrees undergo crossover if a partial change occurs. If this is not present, then the 'any change' is checked. The authors reported better results when adopting these two conditions in four out of the eight problems used in their studies. An analysis of semantic methods have also been applied to local search methods in respect to indirect semantics. For example, Dou and Rockett [16] tested a mixture of various GP and local search variants, including three subtree selection methods and four replacement strategies. The authors found that a semantic-based local search following a steady-state or generational GP performed significantly better compared to baseline GP methods with statistically smaller tree sizes.

Direct semantic approaches have also received attention in the GP community thanks to the improvement on search performance compared to a standard GP system. The driving motivation by incorporating direct semantics is that previous indirect methods were deemed to be wasteful [8,14]. To tackle this issue, Moraglio et al. [10] used his previous theoretical results [17] to allow modifications on the genotype of GP trees to correlate to geometric operators. This had the consequence of inheriting their properties. This results in having a cone landscape by construction, providing to the evolutionary process an "easier" search direction. One potential limitation of this approach, however, is that it allows for the presence of neutrality. This can be beneficial or detrimental depending on the features of the problem at hand – see, for example, [18–21] where Galván and Poli give an in-depth explanation on the effects of neutrality in different type of problems. One more limitation in the approach proposed in [10] is that this modification to GP trees tends to produce larger individuals. To deal with the latter, Vanneschi et al. [22] proposed a cache implementation of Moraglio's approach. To this end, the authors store the semantics of GP trees in a table making the process efficient indeed. However, one limitation in Vanneschi et al. approach is that the reconstruction of GP individuals is cumbersome and difficult to obtain in some cases. This is a drawback specially for applications where the expression is required. Uy et al. [23] took a different approach compared to Vanneschi et al. [22] to deal with the size of individuals. Uy et al. proposed subtree semantic geometric crossover (SSGX) operator that allows them to control the size of individuals. However, their approach has some limitations too such as determining the right values for multiple parameters that are needed to apply SSGX in a GP system. Moreover, it is also based on an expensive trial-and-error mechanism (a maximum of 20 trials is set in their work to search for suitable subtrees).

In [5], we proposed semantic-based methods to integrate them into a MOGP system. In this preliminary study, we were able to promote semantics naturally in a MOGP framework by using a pivot taken from the best Pareto front to compute the semantic distance between this and every individual in the population. We found that this distance and a variant of it tend to produce better results compared to its canonical evolutionary multi-objective variant, in this case, against the well-known NSGA-II [6] and SPEA2 [7] algorithms. However, the findings reported in [5] were limited: a lack of explanation as to why SDO tends to yield better results compared to the other methods used in the initial study,

a lack of an in-depth statistical analysis and of a discussion on the limitation of the SDO approach. In contrast, the current work addresses all these issues, as we should see in the next sections. In particular, we incorporated in detail the results yielded by the SPEA2 algorithm as well as the semantic-based variants using this EMO approach, we explain in detail why SDO works by using different elements such as the frequency of duplicated individuals in the population over generations. We also carried out a detailed statistical analysis of the results. Finally, we also incorporate our main proposal within a Multi-objective Evolutionary Algorithm based on Decomposition [9] to demonstrate the limitations of the proposal.

## 2.2. Multi-objective genetic programming

The objective of a multi-objective optimisation problem is to discover candidate solutions based on the simultaneous consideration of multiple, potentially conflicting, objectives. In the context of GP there are a number of ways that this can be achieved. Broadly speaking there are two primary methods. One is to incorporate multiple objectives into a single fitness function. Another method is to consider the objectives separately making use of the *Pareto dominance* relationship between candidate solutions [24–26]. Since the aim of Evolutionary Multi-objective Optimisation (EMO) is to discover the best balance of solutions amongst the objectives of an evolutionary run, Pareto dominance allows for an intuitive approach to handle multiple (conflicting) objectives. The details of EMO will be explained further in Section 3.2. EMO is one of the most popular and active research area in EAs, with many applications and often achieves impressive results [24–26]. Next, we discuss just a few works that have been adopted EMO within GP systems.

Bleuler et al. [27] proposed an EMO approach to naturally control bloat on even- $n$ -parity problems. The authors defined two objectives to be optimised simultaneously within a MOGP framework; the first objective being the fitness of a program and the second objective being the size of the program tree. This method was compared against other well-known techniques for controlling bloat, such as using an aggregated single-objective function that includes both fitness and parsimony pressure (both Constant and Adaptive Parsimony Pressure were tested) and also via a two stage optimisation process. This approach was not only novel at the time, but also demonstrated how to successfully control for GP bloat, leading to solutions which could be evaluated faster when compared with the other methods.

Related to this study is the work carried out by Bhowan et al. [28], where the authors used a MOGP to find high accuracy on binary unbalanced classes achieving good results compared to well-established machine learning classification methods. Galván et al. also used MOGP for the same type of problems [2–4]. In the same vein, Zhao showed how MOGP can be successfully employed to define partial preferences on the objectives by carefully inserting this bias into the fitness function [29]. The motivating reason to incorporate this embedding is that classification errors are often cost-sensitive in real life scenarios, where the benefits of a correct prediction on one class may significantly outweigh the correct prediction on another. For example, it is more costly to approve a bad loan than denying a good loan.

Shao et al. [30] demonstrated how images can be classified using domain-adaptive global features that are automatically generated. Similar to how Bleuler et al. [27] implemented their optimisation process, Shao et al. use the length of the individuals as an objective to be optimised in their MOGP framework, where the second objective is the classification error rate. Shao et al. reported that their MOGP approach consistently yielded better performance when compared against fourteen other methods, including two neural network-based approaches.

## 3. Background

This section defines some of the basic concepts relevant to this work, namely semantics, MO and EMO algorithms.

### 3.1. Semantics

We use a well-established definition of semantics originally defined in [31]. For a supervised learning task in general, and for GP in specific, the problem is normally specified as a set of input–output pairs, also known as fitness cases, of the form  $T = \{(in_i, o_i)\}$ , where  $in_i \in I$  are the inputs and  $o_i \in O$  are the desired outputs, and  $i = \{1, \dots, l\}$ , where  $l$  is the number of the fitness cases. The semantics  $s(p)$  of a program  $p$  is defined as the vector of output values computed by this program based on the inputs given in the set of fitness cases of a problem. This is formally defined as,

$$s(p) = [p(in_1), p(in_2), \dots, p(in_l)] .$$

### 3.2. Multi-objective optimisation

The goal in Multi-objective optimisation (MO) is to simultaneously optimise two or more objective functions. When multiple objective functions are considered, often these will be in conflict, so the focus then is to search for a set of trade-off solutions as a global optimum becomes unattainable. A natural form to solve this problem is to use the Pareto dominance relation: a solution  $x_1$  in the search space is said to *Pareto-dominate* solution  $x_2$  if  $x_1$  is at least as good as  $x_2$  for all objectives and strictly better for at least one of the objectives.

In this work, the objectives are to be maximised. The class of problems used in this work, defined in Section 5, is binary classification problems where the goal is to maximise the classification accuracy of two conflicting objectives. The *Pareto dominance* concept is defined as in Eq. (1):

$$S_i \succeq S_j \iff \forall_m [(S_i)_m \geq (S_j)_m] \wedge \exists k [(S_i)_k > (S_j)_k] \quad (1)$$

where  $(S_i)_m$  denotes the value of solution  $S_i$  in the  $m$ th objective. Conversely, solutions are considered to be *non-dominated* if no other solution in the population dominates them. The set of optimal trade-off solutions of a MO problem at hand is referred to as the *Pareto optimal set*. Thus, the objective of an EMO algorithm is to find (a good approximation) of this set. The *Pareto optimal front* is the objective space representation of the Pareto optimal set.

Pareto-dominance has been used in different ways to handle such criterion to bias the search in an EMO algorithm. Some of the most widely-known are dominance rank [6] and dominance count [27]. Dominance rank is based on the number of solutions that dominate another within a population, this means that a lower value is desirable. On the other hand, dominance count is the number of solutions that a particular solution dominates, in this case a higher value is preferable. There are two popular EMO algorithms that include the aforementioned ways to use Pareto-dominance and they are adopted in this study: the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [6] that uses dominance rank, and the Strength Pareto Evolutionary Algorithm (SPEA2) [27] that uses dominance rank and dominance count.

In NSGA-II, dominance rank is used as a fitness value for solution  $S_i$ . This is expressed as

$$NSGA-II(S_i) = \{\#j, j \in Pop | S_j \succeq S_i\} . \quad (2)$$

SEA2, on the other hand, uses and dominance count and dominance rank when computing the fitness value of an individual.

First, a *strength* value  $D$  is given to every candidate solution in the population. This is formally defined by,

$$D(S_i) = |\{j | j \in \text{Pop} \wedge S_j \geq S_i\}|. \quad (3)$$

$D$  then determines the number of solutions that a particular solution ( $S_i$ ) dominates. To determine the fitness value of a candidate solution  $i$  in SPEA2, we then use the strengths of all the solutions that dominate solution  $i$ . This is expressed as,

$$\text{SPEA2}(S_i) = \sum_{j \in \text{Pop}, S_j \geq S_i} D(S_j). \quad (4)$$

### 3.2.1. Diversity preservation in multi-objective optimisation through the use of the crowding distance

A total order is not provided by Pareto dominance, such that another criterion is required to effectively compare different points in the search space, which is necessary to perform selection and survival during evolution. One approach is to use a *crowding distance* measure, such that regions that are sparsely populated are preferable compared to denser regions. The crowding distance is normally used in objective space to differentiate between individuals having the same Pareto rank, preferring those that are in the less populated regions. Algorithm 1 shows how this crowding distance is computed, adapted from [6].

**Algorithm 1** Crowding distance. Adapted from [6].

```

1: procedure CROWDING-DISTANCE-ASSIGNMENT( $I$ )
2:    $l \leftarrow |I|$  ▷ number of solutions in  $I$ 
3:   for each  $i$  do
4:      $I[i]_{\text{distance}} \leftarrow 0$  ▷ initialise distance
5:   end for
6:   for each objective  $m$  do
7:      $I \leftarrow \text{sort}(I, m)$  ▷ sort using each objective value
8:      $I[1]_{\text{distance}} \leftarrow \infty$  ▷ so that boundary points are
always selected
9:     for  $i \leftarrow 2$  to  $(l - 1)$  do ▷ for all other points
10:       $I[i]_{\text{distance}} \leftarrow I[i]_{\text{distance}} +$ 
 $(I[i+1].m - I[i-1].m) / (f_m^{\text{max}} - f_m^{\text{min}})$ 
11:    end for
12:  end for
13: end procedure

```

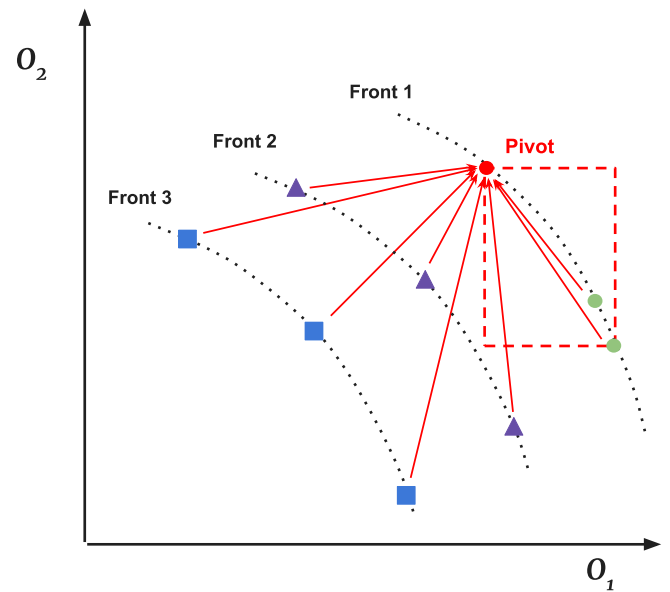
### 3.2.2. MOGP algorithm

The MOGP framework employed in this work is based on NSGA-II, which is described next. This is the framework that we use for all our experiments and we only change the way fitness of individuals is computed by using either NSGA-II or SPEA2, formally defined in Eqs. (2) or (3), respectively.

Both populations (parents and offspring) are joint at every generation. The best individuals from this overall population are copied into the archive population, which contains the same number of individuals as the original population. The archive population then serves as the parent population for the next generation. The archive population provides elitism preserving the set of non-dominance solutions throughout the evolutionary process.

## 4. Semantic-based MOGP methods

Next, we present the semantic-based approaches employed in this work that are incorporated into the baseline MOGP algorithms, namely NSGA-II and SPEA-2.



**Fig. 1.** Semantic-based Distance as an Additional Criterion. First, we get non-dominated solutions using either dominance rank (see Eq. (2)) or the dominance rank and dominance count to determine the strength of a given solution (see Eq. (4)) and store these in  $R_r$ . Second, once the non-dominated fronts have been found, we proceed to find a pivot from the first non-dominated front. To do so, we use the crowding distance defined in Section 3.2.1. This pivot is in the sparsest region of the front (the dotted red rectangle depicts this idea). Third, we compute the semantic distance (either using Eq. (5) or Eq. (6)), from the pivot to each individual in  $R_r$ . Fourth, the distance values are used as an additional criterion for the EMO to optimise, along with conflicting objectives,  $O_1$  and  $O_2$ , namely the TPR and the TNR for unbalanced binary classification problems used in this work.

### 4.1. Semantic similarity-based crossover MOGP

The first approach that we consider to incorporate semantics within MOGP is the Semantic Similarity-based Crossover (SSC), originally proposed by Uy et al. [8] for single-objective genetic programming.

SSC requires a semantic distance. The distance is computed as the average of the absolute difference of values for every  $in \in I$  (or a partial set of inputs) between parents and offspring. When the distance value falls within a specific range, defined by one or two bounds, an offspring is generated by way of crossover. Given that this may be difficult to meet, the original approach encourages diversity by repeating crossover with a maximum of 20 attempts. If the condition is not satisfied, then crossover is executed as usual.

SSC was a notable contribution to GP since it showed that it was feasible to promote semantic diversity in continuous search spaces, leading to several subsequent studies [2,32–35]. In this work we implement SSC as proposed in [8], but for the first time the method is evaluated as part of a MOGP framework, using both NSGA-II and SPEA2 as the baseline algorithms. Our results, discussed in Section 6, show that unlike for single-objective GP, MOGP SSC did not lead to notable performance improvements.

### 4.2. Semantic-based Crowding Distance

In Semantic-based Crowding Distance (SCD), the key element is to define an individual (pivot) that can be used to compute a semantic distance between this and some elements from the population. Algorithm 2 shows how the SCD MOGP works. It creates a new population  $P_{t+1}$  using the non-dominated sorted solutions from the parent population  $P_t$  and the offspring population  $Q_t$ , which are merged into population  $R_t$  (Lines 2–3). This continues until the size of  $P_{t+1}$  is equal to the size of  $P_t$  (Lines 4–12). It may

be the case that a particular front does not fit entirely into the new population and a second criterion is necessary to complete  $P_{t+1}$ . If this is the case, we proceed to store ( $F_r$ ), the remaining of those individuals that have not been used to complete the population (Lines 13–24). We then use the semantic distance as a criterion to select those points from  $F_r$  to complete  $P_{t+1}$ . To do so, we proceed to find a pivot  $v$  which is the furthest point from the first front using the crowding distance, explained in Section 3.2.1 (Lines 15–16). The dotted red rectangle in Fig. 1 exemplifies how the pivot (red dot) is chosen from the first front.

The semantic distance between every point in  $F_r$  and the pivot is computed (Line 17). Thus, only one pivot is necessary to compute this distance. Formally, this distance is computed as in Eq. (5):

$$d(p_j, v) = \sum_{i=1}^l 1 \text{ if } \text{LBSS} \leq |p(in_i) - v(in_i)| \leq \text{UBSS} \quad (5)$$

where  $p_j$  is an individual in  $R_t$ ,  $l$  is the number of fitness cases, and LBSS and UBSS are the lower bound and upper bound for the semantic similarity values, respectively. These last two values are used to promote semantic diversity within a range, as reported beneficial in [8].

There is also a number of studies that have concluded that only one bound is necessary to promote semantic diversity [2,3,23]. We can compute the semantic distance between the pivot  $v$  and every individual in  $R_t$  using the following distance formally described in Eq. (6),

$$d(p_j, v) = \sum_{i=1}^l 1 \text{ if } |p(in_i) - v(in_i)| > \text{UBSS} \quad (6)$$

We then use the semantic distance values on the stored  $F_r$  (Line 18) and select the individuals that are in sparse regions of the search space, until we complete  $P_{t+1}$  (Lines 20–23).

---

#### Algorithm 2 Semantic-based Crowding Distance

---

```

1: procedure FORMING NEW POPULATION
2:    $R_t \leftarrow P_t \cup Q_t$ 
3:    $F \leftarrow \text{non\_dominated\_sort}(R_t)$ 
4:    $P_{t+1} \leftarrow 0$ ;  $i \leftarrow 1$ ;
5:   while  $|P_{t+1}| \leq |P|$  do
6:     if  $|P_{t+1}| + |F_i| < |P|$  then
7:        $P_{t+1} \leftarrow P_{t+1} \cup F_i$ 
8:        $i \leftarrow i + 1$ 
9:     else
10:      break
11:    end if
12:  end while
13:  if  $|P_{t+1}| < |P|$  then
14:     $F_r \leftarrow F \cap \{F_1 \dots F_i\}$ 
15:     $CD_1 \leftarrow \text{crowding\_distance}(F_1)$ 
16:     $\text{pivot} \leftarrow \text{furthest\_point}(CD_1)$ 
17:     $SV \leftarrow \text{compute\_semantics}(F_r, \text{pivot})$ 
18:     $F_r \leftarrow \text{crowding\_distance}(SV(F_r))$ 
19:     $j \leftarrow 0$ 
20:    for  $j + |P_{t+1}| < |P|$  do
21:       $P_{t+1} \leftarrow P_{t+1} \cup F_{r[j]}$ 
22:       $j \leftarrow j + 1$ 
23:    end for
24:  end if
25: end procedure

```

---



---

#### Algorithm 3 Semantic-based Distance as an Additional Criterion

---

```

1: procedure FORMING NEW POPULATION
2:    $R_t \leftarrow P_t \cup Q_t$ 
3:    $F \leftarrow \text{non\_dominated\_sort}(R_t)$ 
4:    $CD_1 \leftarrow \text{crowding\_distance}(F_1)$ 
5:    $\text{pivot} \leftarrow \text{furthest\_point}(CD_1)$ 
6:    $R_t \leftarrow \text{compute\_semantics}(R_t, \text{pivot})$ 
7:    $F \leftarrow \text{sort}(R_t)$ 
8:    $P_{t+1} \leftarrow 0$ ;  $j \leftarrow 0$ 
9:   for  $j < |P|$  do
10:     $P_{t+1} \leftarrow P_{t+1} \cup F[j]$ 
11:     $j \leftarrow j + 1$ 
12:  end for
13: end procedure

```

---

#### 4.3. Semantic-based distance as an additional criterion

We further expand SCD by now using the resulting semantic distance values as another indicator to select solutions by the MO process. We refer to this approach as Semantic-based Distance as an additional criterion (SDO). As with SSC, we continue using the MOGP framework described above, for SDO. This approach is described in detail in Algorithm 3.

First, we merge the parent population  $P_t$  and the offspring population  $Q_t$  into  $R_t$ . We then get the non-dominated sorted solutions (Lines 2 – 3). To compute the semantic distance for each point contained in  $R_t$  we proceed as follows. We first compute the crowding distance from the first front and select the point that is the furthest away. We use this as pivot  $v$  to then compute the semantic distance between  $v$  and each individual contained in  $R_t$  (Lines 4 – 5). We then use the resulting distance value, using either Eq. (5) (two bound values are required) or Eq. (6) (one bound value is necessary), as another criterion to be computed (Line 6). Next, we sort  $R_t$  (Line 7) and form the new population  $P_{t+1}$  (Lines 8 – 12). In case the size of  $P_{t+1} \cup F[j]$  is greater than  $|P|$ , we proceed to take only the individuals needed to complete  $P_{t+1}$ . Fig. 1 depicts this idea.

#### 5. Experimental setup

The use of benchmark problems has allowed the research community to test, validate and explain a plethora of evolutionary algorithms. In this work, we also adopt well-known, robust and tested benchmark problems used in other studies [5,28,36] that will allow us to (i) test the algorithms used in this work, (ii) to use well-defined metrics that allow us to compare one method against another one, (iii) to allow us to explain why one particular method behaves better than others, (iv) to draw sound conclusions on the results reported in the following sections. Thus, for this study, the impact of semantics in MOGP are analysed using several unbalanced binary classification problems, taken from the UCI Machine Learning repository [37]. Table 1, adapted from [28], gives the details of all datasets used in this work. These binary classification problems have various degrees of class imbalance, from 1:3 to 1:130, for the Ion and the Abal<sub>2</sub> dataset, respectively. The 50%–50% training/test sets also range from being well-represented (Abal<sub>2</sub> has around 2,100 instances) to sparsely represented (Spect has around 133 instances, where around 27 are from the minority class). Moreover, these datasets range between low dimensionality (Yeast<sub>1</sub> has 8 features) to high dimensionality (Ion has 34 features). Finally, our datasets include binary and real-valued features. Thus, these datasets represent class imbalance problems of various degrees of difficulty, size, dimensionality and types of features reasonably well. Moreover,

**Table 1**  
Summary of the binary unbalanced classification datasets.

Dataset	Positive/Negative Class Brief description	Number of examples Total/Positive(%) / Negative(%)	Ratio	Features No.(Type)
Ion	Good/Bad: Ionosphere radar signals	351/126(35.8%)/225(64.2%)	1:3	34(Real)
Spect	Abnormal/Normal Tomography scan	267/55(20.6%)/212(79.4%)	1:4	22(Binary)
Yeast <sub>1</sub>	MIT/Other Protein sequences	1482/244(16.5%)/1238(83.5%)	1:6	8(Real)
Yeast <sub>2</sub>	ME3/Other Protein sequences	1482/163(10.9%)/1319(89.1%)	1:9	8(Real)
Abal <sub>1</sub>	9/18 Biology of abalone	731/42(5.7%)/689(94.2%)	1:17	8(Real)
Abal <sub>2</sub>	9/Other Biology of abalone	4177/32(0.77%)/4145(99.2%)	1:130	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)

we carefully choose these benchmark problems so that our evaluations on the three semantic-based methods (SSC, SCD and SDO) and the two EMO approaches (NSGA-II and SPEA2) are not-problem dependent.

The terminal and function sets used in this work are as follows. The terminals are the problem features. The function set contains the most common arithmetic operators, namely  $\mathcal{F} = \{+, -, *, /\}$ , where the division operator is protected by returning the numerator when the denominator has a value of zero. The models evolved by GP map each input pattern in a dataset to a single output value. When the output of a GP model is greater than, or equal to, zero the pattern is labelled as part of the minority class, and it is labelled as a majority class pattern, otherwise.

The common way to measure fitness in a classification task is to use 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  $Acc = \frac{TP+TN}{TP+TN+FP+FN}$ , where  $TP$  are the true positives,  $TN$  are the true negatives,  $FP$  are the false positives and  $FN$  are the false negatives. The drawback of using  $Acc$  alone is that it rapidly biases the evolutionary search towards the majority class [28]. A better approach is to treat each class 'separately' using a MO approach. Two objectives considered are thus the true positive rate, given by  $TPR = \frac{TP}{TP+FN}$ , and the true negative rate given by  $TNR = \frac{TN}{TN+FP}$ . They measure the distinct accuracy for the minority (TPR) and majority class (TNR), respectively.

The experiments were conducted using a generational approach. Tree size was controlled by using a maximum length or a maximum final depth, whatever happens first. When a child tree exceeds any of these, the offspring is generated again until these conditions are satisfied. The parameters used are shown in Table 3. These include the use of different bounds, defined as Upper Bound Semantic Similarity (UBSS) = {0.25, 0.50, 0.75, 1.0} values and Lower Bound Semantic Similarity (LBSS) = {0.001, 0.01, 0.1} values, to compute the semantic distances defined in Eqs. (5) and (6). This results in conducting a thorough analysis: for each of the semantic-based approaches and for each of the datasets used, we have 16.<sup>1</sup> independent results, each being the result of 50 independent runs. To obtain meaningful results, we carried

<sup>1</sup> 4 values for UBSS and 4 cases for LBSS: 3 values and 1 case where LBSS is not defined.

**Table 3**  
Summary of parameters.

Parameter	Value
Population size	500
Generations	50
Type of crossover	90% internal nodes, 10% leaves
Crossover rate	0.60
Type of mutation	Subtree
Mutation Rate	0.40
Selection	Tournament (size = 7)
Initialisation method	Ramped half-and-half
Initialisation depths:	
Initial depth	1 (Root = 0)
Final depth	5
Maximum size	800 nodes
Maximum final depth	8
Independent runs	50
Semantic bounds	UBSS = {0.25, 0.5, 0.75, 1.0} LBSS = {0.001, 0.01, 0.1}

out an extensive empirical experimentation (29,400 independent runs in total).<sup>2</sup>

The results reported in the following section are based on the testing dataset.

## 6. Results and analysis

### 6.1. Diversity

Diversity in GP can be quantified in many different ways, in this work we focus on phenotypic diversity, based on the number of unique fitness values [38,39]. This measure of diversity is particularly relevant in a MO problem, since the spread of solutions in the Pareto front is a desired feature of the optimisation process. Specifically, we quantify diversity by computing the number of unique solutions produced in objective space, considering the TPR and the TNR (see Section 5) of each solution found by either a canonical EMO approach (NSGA-II or SPEA2) vs. a semantic-based approach (SDO, SSC or SCD) for each of the six datasets used in this work (see Table 1). A unique solution is defined as a solution that was obtained by a particular approach (for instance NSGA-II) but not obtained by another approach (for instance SDO). Consequently, a non-unique solution is one that was obtained by two approaches.

We know from Section 5 that we used 16 different configurations of values to compute the semantic distance for each of the semantic-based approaches in each of the datasets used. We summarised these by computing the average and standard

<sup>2</sup> 50 independent runs, 6 datasets, 3 semantic-based MOGP approaches (SSC, SCD, SDO), 16 different combination of values for UBSS and LBSS, 2 canonical EMO methods (NSGA-II, SPEA2).

**Table 4**

Number of unique solutions (average and standard deviation) found by NSGA-II vs. SDO, SSC or SCD.

Data	NSGA-II					
Set	Canonical	SDO	Canonical	SSC	Canonical	SCD
Ion	162.5 ± 20.1	751.1 ± 187.9	269.7 ± 11.2	252.8 ± 20.5	270.0 ± 25.8	245.4 ± 27.5
Spect	81.1 ± 13.9	331.7 ± 56.7	151.8 ± 9.5	155.8 ± 14.3	156.7 ± 9.1	159.5 ± 24.0
Yeast <sub>1</sub>	798.5 ± 33.3	1991.5 ± 149.0	1166.5 ± 26.1	1050.4 ± 30.6	1168.7 ± 19.4	1058.0 ± 44.5
Yeast <sub>2</sub>	235.9 ± 15.53	855.1 ± 138.5	418.8 ± 13.3	414.9 ± 16.9	418.0 ± 9.9	398.1 ± 16.7
Abal <sub>1</sub>	128.6 ± 13.2	485.8 ± 55.3	192.6 ± 5.9	205.1 ± 9.1	193.4 ± 6.0	204.5 ± 6.2
Abal <sub>2</sub>	183.5 ± 21.3	1034.7 ± 284.8	248.9 ± 2.4	187.9 ± 11.8	244.4 ± 4.2	215.7 ± 13.2

**Table 5**

Number of unique solutions (average and standard deviation) found by SPEA2 vs. SDO, SSC or SCD.

Data	SPEA2					
Set	Canonical	SDO	Canonical	SSC	Canonical	SCD
Ion	147.8 ± 15.7	775.5 ± 161.9	247.2 ± 9.1	82.6 ± 14.0	242.1 ± 6.6	273.9 ± 18.3
Spect	95.5 ± 18.1	334.1 ± 45.2	169.8 ± 16.0	168.8 ± 16.5	183.5 ± 12.5	147.8 ± 13.4
Yeast <sub>1</sub>	749.1 ± 40.7	2076.6 ± 189.5	1144.0 ± 22.8	1067.5 ± 36.9	1131.9 ± 35.6	1087.6 ± 44.4
Yeast <sub>2</sub>	253.1 ± 21.4	886.3 ± 153.1	439.1875 ± 14.3	399.8125 ± 10.1	430.25 ± 8.6	411.5 ± 26.
Abal <sub>1</sub>	149.3 ± 12.7	498.1 ± 70.9	213.7 ± 6.6	198.8 ± 7.7	213.7 ± 8.1	205.0 ± 11.6
Abal <sub>2</sub>	166.1 ± 13.9	1120.4 ± 233.3	218.8 ± 2.5	198.1 ± 14.7	216.6 ± 3.8	230.5 ± 14.3

deviation of these 16 values and compared it against a canonical MO approach. Table 4 reports the number of unique solutions found by either NSGA-II or SDO, NSGA-II or SSC and NSGA-II or SCD. As can be seen from this table, our proposed SDO approach, which treats semantic distance values as an additional criterion, third column from left to right of Table 4, produces more unique solutions compared to canonical NSGA-II for any of the datasets used. For example, the lowest ratio of SDO vs. NSGA-II is  $\approx 2.4$  on the Yeast<sub>1</sub> dataset. That is, there are 2.4 more solutions produced by SDO compared to NSGA-II ( $1991.5 \pm 149.0$  vs.  $798.5 \pm 33.3$ ). At the opposite end, we have that the highest ratio of SDO vs. NSGA-II is  $\approx 5.6$  on the Abal<sub>2</sub> dataset ( $183.5 \pm 21.3$  vs.  $1034.7 \pm 284.8$ ). When we then turn our attention to the other two semantic-based approaches, we can see that there is no real advantage for SSC or SCD over NSGA-II, in terms of producing more solutions on the two conflicting classes.

Table 5 also reports the number of unique solutions found by either SPEA2 vs. any of the semantic-based approaches used in this work. Similarly to what has been reported in Table 4, as well as discussed before, we can observe that our proposed SDO produces significantly more solutions compared to canonical SPEA2. For example, the lowest ratio of SDO vs. SPEA2 is  $\approx 2.7$  on the Yeast<sub>2</sub> dataset ( $253.1 \pm 21.4$  vs.  $886.3 \pm 153.1$ ) and the highest ratio of SDO vs. SPEA2 is  $\approx 6.7$  on the Abal<sub>2</sub> dataset ( $166.1 \pm 13.9$  vs.  $1120.4 \pm 233.3$ ). Similarly to what can be seen in Table 4, the other two semantic-based approaches (SSC and SCD) do not show a significant improvement when compared to SPEA2.

From this analysis, it is clear that our proposed SDO produces more unique solutions compared to any of the canonical MO approaches used in this work, regardless of the dataset used. However, we cannot say whether these unique solutions found by our approach are better or worse compared to those obtained by NSGA-II or SPEA2. To address this, we provide further analysis of the approximated Pareto fronts in Section 6.3. Figs. 2 and 3 show the evolved solutions that were exclusively found by (i) either NSGA-II or NSGA-II SDO, and (ii) either SPEA2 or SPEA2 SDO, respectively, for all the datasets used in this work. It is clear to see from Figs. 2 and 3, that the unique evolved solutions that were found by SDO (setting UBSS = 0.5 to be used in Eq. (6)), represented by green hollow square symbols, are more numerous and more spread than those produced by the canonical EMO approaches: NSGA-II (Fig. 2) or SPEA2 (Fig. 3), represented by black hollow circles. This is particularly clear for the Ion, Spect, Yeast<sub>2</sub>, Abal<sub>1</sub> and Abal<sub>2</sub> datasets. The situation is less clear for the Yeast<sub>1</sub> dataset.

**Table 6**Average ( $\pm$  standard deviation) hypervolume of evolved Pareto-approximated fronts and PO fronts for NSGA-II and SPEA2 over 50 independent runs. No significant differences were found in any single dataset.

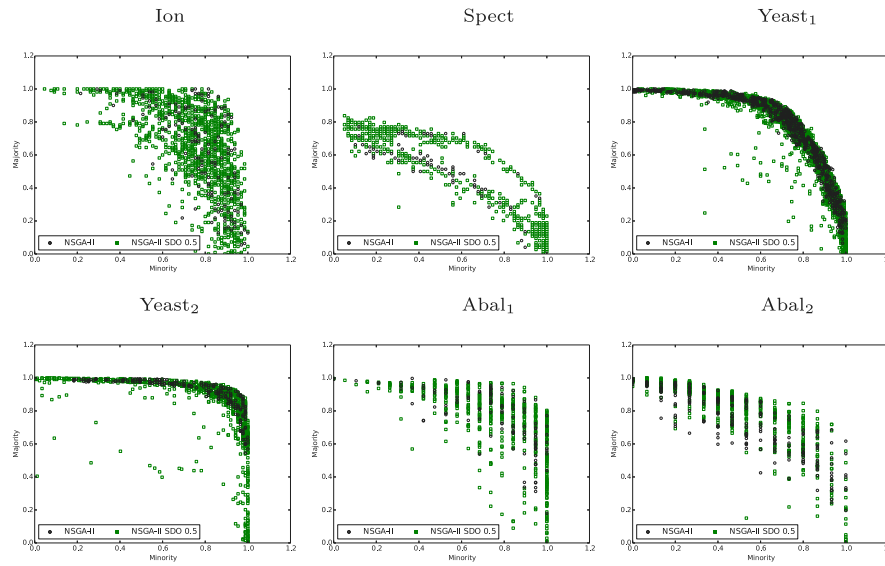
Dataset	NSGA-II		SPEA2	
	Hypervolume		Hypervolume	
	Average	PO front	Average	PO front
Ion	0.766 ± 0.114	0.938	0.786 ± 0.094	0.948
Spect	0.534 ± 0.024	0.647	0.544 ± 0.032	0.659
Yeast <sub>1</sub>	0.838 ± 0.011	0.876	0.838 ± 0.008	0.877
Yeast <sub>2</sub>	0.950 ± 0.009	0.976	0.946 ± 0.015	0.978
Abal <sub>1</sub>	0.847 ± 0.058	0.961	0.832 ± 0.078	0.960
Abal <sub>2</sub>	0.576 ± 0.122	0.842	0.544 ± 0.147	0.834

## 6.2. Hypervolume comparison

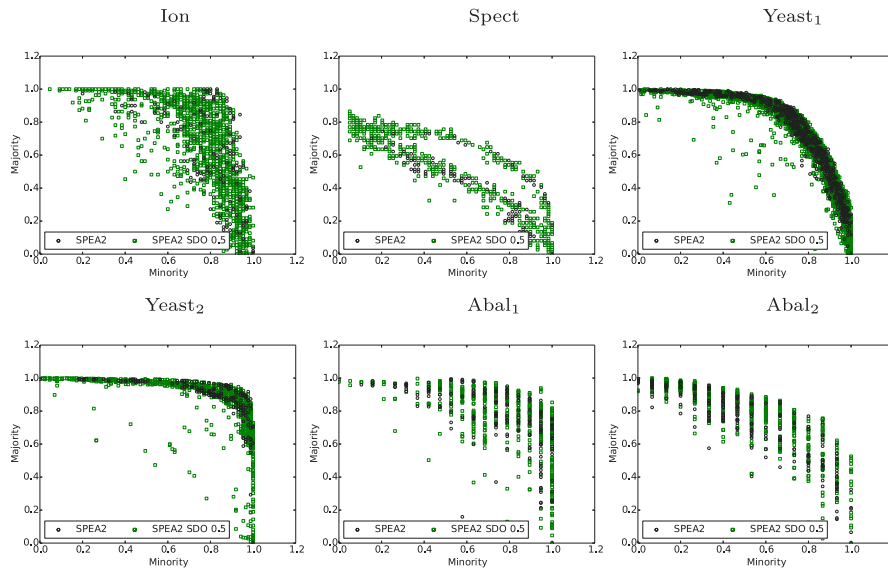
To further compare the three semantic-based methods and the two EMO methods employed in our study, we calculate the hypervolume [40] of the approximated Pareto fronts, which is a commonly used performance metric for MO techniques. For bi-objective problems such as the unbalanced binary classification problems employed in this work, the hypervolume is computed as the sum of all the trapezoidal areas fitted under each point in objective space, where the set of points represent the candidate solutions. Moreover, the hypervolume explicitly captures the underlying goals of the MOGP, to concurrently maximise the accuracies of both the majority class and the minority class. Additionally, the accumulated Pareto-optimal (PO) front, which is the set of non-dominated solutions after merging the Pareto-approximated fronts for all the independent runs, set at 50 in this study, is also computed.

Tables 6–8 show the average hypervolume across 50 runs and the hypervolume of the accumulated PO front of these runs for each of the problems shown in Table 1. In Table 6, we show these hypervolume values for the two EMO approaches used in this work: NSGA-II and SPEA2. In the same manner, Tables 7 and 8 show these hypervolume values yield by the three semantic-based methods using NSGA-II and SPEA2, respectively. Furthermore, for Tables 7 and 8, these hypervolume values are given for each of the 16 different configurations of each semantic methods (based on the upper and lower bounds). Bold text in the tables indicates an average performance that is superior than the average performance of the baseline method (NSGA-II or SPEA2). The results suggest that, on average, our proposed method SDO (upper





**Fig. 2.** Evolved solutions that were exclusively found by either NSGA-II, represented by black hollow circle symbols, and its SDO variant, represented by green hollow square symbols, setting UBSS = 0.5, for all the datasets used in this work.



**Fig. 3.** Evolved solutions that were exclusively found by either SPEA2, represented by black hollow circle symbols, and its SDO variant, represented by green hollow square symbols, setting UBSS = 0.5, for all the datasets used in this work.

part of Tables 7 and 8) consistently outperforms the baselines, while this is not clearly the case for SSC and SCD (middle and bottom of these tables, respectively).

The following statistical analysis was performed. For each baseline method, NSGA-II and SPEA2, we perform three blocks of comparisons, one for each semantic variant, namely SDO, SSC and SCD (six blocks in total). In each block we perform a multi-group test with  $N$  configurations, where  $N = 16$  represents each of the configurations tested based on different UBSS and LBSS bound values which are compared with one of the baseline methods (NSGA or SPEA2). We use the Friedman test in our analysis, comparing based on all 6 problems and considering 50 replicates (runs) for each problem. The null hypothesis is that the median performance of all groups in the same block is the same, and we reject the null hypothesis at the  $\alpha = 0.01$  significance level. For both SSC and SCD, the null hypothesis of the multi-group tests was not rejected considering each of the baselines, with  $p$ -values above the significance level of the test. However,

the null hypothesis was rejected in the multi-group tests for the SDO configurations relative to both baselines. In these cases, a post-hoc test was conducted, once again using the Friedman test to perform  $N$  pairwise tests between each SDO configuration and the baseline method (NSGA-II and SPEA2). The Bonferroni–Dunn correction of the  $p$ -value was performed to account for the family-wise error of performing multiple comparisons. Once again, in all pairwise comparisons the null hypothesis was rejected at the  $\alpha = 0.01$  significance level, comparing each SDO configuration and the respective baseline. Based on these results, we found that the SDO method outperforms each of the baseline methods, while SSC and SCD did not. Moreover, we can see that SDO performance is robust to how it is parameterised, relative to the LBSS and UBSS values.

From this statistical analysis, it is evident that SDO behaves, in average, better compared to any of the approaches used in this work (two canonical EMO approaches and the two other semantic-based methods). To illustrate the performance of the



Table 7 (continued).

LBSS		Hypervolume							
		Average				PO front			
		UBSS				UBSS			
		0.25	0.5	0.75	1.0	0.25	0.5	0.75	1.0
Yeast <sub>2</sub>	—	0.948 ± 0.013	0.950 ± 0.009	0.947 ± 0.011	0.946 ± 0.011	<b>0.978</b>	<b>0.977</b>	0.976	0.976
	0.001	0.947 ± 0.016	0.946 ± 0.016	0.946 ± 0.016	0.950 ± 0.012	0.976	0.976	0.976	<b>0.979</b>
	0.01	0.950 ± 0.010	0.949 ± 0.009	0.949 ± 0.009	0.947 ± 0.011	<b>0.979</b>	<b>0.979</b>	<b>0.979</b>	<b>0.977</b>
	0.1	0.948 ± 0.010	0.947 ± 0.009	0.947 ± 0.009	<b>0.952 ± 0.008</b>	<b>0.977</b>	<b>0.978</b>	<b>0.978</b>	<b>0.978</b>
Abal <sub>1</sub>	—	0.841 ± 0.089	0.838 ± 0.073	0.838 ± 0.080	0.847 ± 0.071	<b>0.963</b>	0.960	<b>0.964</b>	0.961
	0.001	0.828 ± 0.074	0.830 ± 0.103	0.830 ± 0.103	0.799 ± 0.114	0.958	<b>0.963</b>	<b>0.963</b>	0.961
	0.01	0.842 ± 0.082	0.839 ± 0.083	0.839 ± 0.083	0.825 ± 0.070	0.964	<b>0.962</b>	<b>0.962</b>	<b>0.964</b>
	0.1	<b>0.864 ± 0.062</b>	0.833 ± 0.076	0.833 ± 0.076	0.837 ± 0.071	<b>0.962</b>	0.958	0.958	<b>0.964</b>
Abal <sub>2</sub>	—	0.561 ± 0.120	0.542 ± 0.110	0.534 ± 0.150	0.570 ± 0.104	0.831	0.824	<b>0.862</b>	0.830
	0.001	0.533 ± 0.131	<b>0.577 ± 0.074</b>	<b>0.577 ± 0.074</b>	<b>0.580 ± 0.092</b>	0.831	0.815	0.815	0.823
	0.01	0.572 ± 0.121	<b>0.577 ± 0.097</b>	0.577 ± 0.097	0.565 ± 0.138	0.842	0.830	0.830	<b>0.845</b>
	0.1	0.529 ± 0.127	0.574 ± 0.085	0.574 ± 0.085	0.566 ± 0.097	0.811	0.831	0.831	0.817

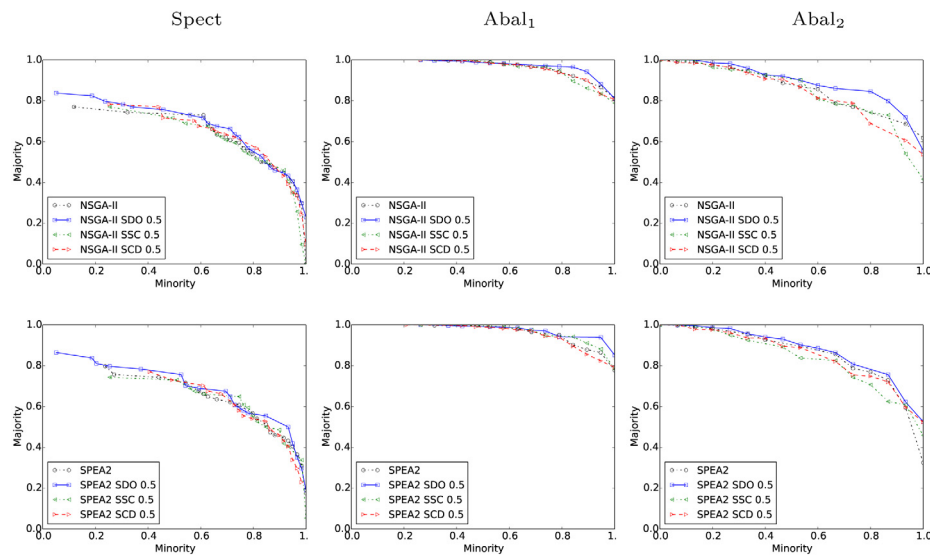


Fig. 4. Pareto fronts for Spect, Abal<sub>1</sub>, Abal<sub>2</sub> datasets using NSGA-II and its semantic-based variants (top) and using SPEA2 and its variants (bottom). Semantic-based variants set UBSS = 0.5.

proposed semantic methods, Fig. 4 shows the Pareto fronts obtained by each approach, setting UBSS at 0.5, as well as the Pareto fronts obtained by NSGA-II and SPEA2. The figure shows the fronts for the Spect, Abal<sub>1</sub> and Abal<sub>2</sub> datasets, as representative examples. From this figure, it is easy to see how SDO, represented by blue hollow squares connected by a solid line, achieves better coverage of the objective space compared to the other approaches. This is particularly clear for the Abal datasets, using either NSGA-II (top) or SPEA2 (bottom) as well as for the Spect dataset when using SPEA2 and less clear when using NSGA-II.

### 6.3. Discussion on results and why SDO works

Tables 9 and 10 show the average number of distinct solutions, which are defined as solutions that exclude duplicates, across all 50 independent runs, in the first approximated Pareto front for Generations 1, 10, 20, 30, 40 and 50 for the canonical EMO methods (top of Tables 9 and 10) and the and SDO methods (bottom part of these tables). For simplicity in our analysis, we use a single upper bound of 0.5 for the SDO method. By selecting the pivot as an individual from the sparsest region of the first front, we hope to attract new individuals to the surrounding region of sparsity, the concept of which has been discussed in greater detail in Section 4.3 and depicted in Fig. 1. As a consequence of attracting new individuals to these sparse regions, it is expected that the

number of solutions should increase in the first approximated Pareto front for subsequent generations. This is precisely what it is observed in Tables 9 and 10, although it is fair to say that the increase in the numbers of individuals in the best Pareto front is minimal going, for instance, from 8.28 distinct solutions in the last generation when using the Ion dataset and NSGA-II to 10.16 distinct solutions when using NSGA-II SDO (top-right column of Table 9). Even when this increase is small, the trend is consistent for all the datasets used in this work, regardless of using NSGA-II SDO or SPEA2 SDO. This small increase in the number of distinct solutions partially explain why our proposed semantic-based method yields better results compared with their canonical EMO algorithms. As we have articulated in Section 2, multiple studies have reported the benefits of promoting semantics in evolutionary search leading to have diversity. If the same is true for our proposed semantic-based approach, dubbed “Semantic-based Distance as an additional criteriOn” (SDO for short), then we believe that a reduction in the number of duplicated solutions should be observed. To verify and help us further understand why our method yields better results, we focus on this next.

Tables 11 and 12 show the number of duplicated solutions, averaged over 50 independent runs, in multiple generations (1, 10, 20, 30, 40 and 50). We can see that the number of duplicates substantially decreases for SDO in each generation for multiple datasets. For example, a fourth of the number of duplicated



Table 8 (continued).

LBSS		Hypervolume							
		Average				PO front			
		UBSS				UBSS			
		0.25	0.5	0.75	1.0	0.25	0.5	0.75	1.0
Yeast <sub>2</sub>	—	<b>0.949 ± 0.008</b>	0.946 ± 0.015	<b>0.948 ± 0.009</b>	<b>0.948 ± 0.009</b>	0.977	0.978	0.976	0.978
	0.001	<b>0.948 ± 0.012</b>	<b>0.948 ± 0.009</b>	<b>0.948 ± 0.009</b>	<b>0.947 ± 0.011</b>	0.978	0.977	0.977	0.977
	0.01	<b>0.948 ± 0.010</b>	<b>0.948 ± 0.011</b>	<b>0.948 ± 0.011</b>	<b>0.948 ± 0.015</b>	0.975	0.977	0.977	0.977
	0.1	<b>0.949 ± 0.010</b>	<b>0.950 ± 0.012</b>	<b>0.950 ± 0.012</b>	0.945 ± 0.036	0.977	0.978	0.978	0.977
Abal <sub>1</sub>	—	0.829 ± 0.086	<b>0.834 ± 0.072</b>	<b>0.840 ± 0.083</b>	<b>0.835 ± 0.074</b>	<b>0.961</b>	0.955	0.960	<b>0.968</b>
	0.001	0.829 ± 0.072	<b>0.834 ± 0.069</b>	<b>0.834 ± 0.069</b>	0.819 ± 0.088	0.959	0.957	0.957	0.960
	0.01	<b>0.834 ± 0.084</b>	0.800 ± 0.078	0.800 ± 0.078	0.816 ± 0.097	0.962	0.960	0.960	0.952
	0.1	<b>0.841 ± 0.085</b>	0.827 ± 0.071	0.827 ± 0.071	<b>0.842 ± 0.067</b>	<b>0.965</b>	0.955	0.955	<b>0.964</b>
Abal <sub>2</sub>	—	<b>0.572 ± 0.111</b>	<b>0.577 ± 0.103</b>	<b>0.562 ± 0.127</b>	<b>0.560 ± 0.094</b>	0.828	<b>0.837</b>	<b>0.856</b>	<b>0.839</b>
	0.001	<b>0.557 ± 0.094</b>	<b>0.578 ± 0.103</b>	<b>0.578 ± 0.093</b>	<b>0.554 ± 0.116</b>	<b>0.838</b>	<b>0.852</b>	<b>0.852</b>	<b>0.853</b>
	0.01	<b>0.569 ± 0.117</b>	<b>0.576 ± 0.105</b>	<b>0.576 ± 0.105</b>	<b>0.580 ± 0.070</b>	<b>0.838</b>	<b>0.837</b>	<b>0.837</b>	0.834
	0.1	0.535 ± 0.128	<b>0.567 ± 0.118</b>	<b>0.567 ± 0.118</b>	<b>0.586 ± 0.077</b>	0.816	0.823	0.823	<b>0.842</b>

Table 9

Average number of distinct solutions (± std. deviation) occurring at the first front of generations 1, 10, 20, 20, 40 and 50 for canonical NSGA-II and NSGA-II SDO methods.

Data set	Generation					
	1	10	20	30	40	50
	mean ± std	mean ± std	mean ± std	mean ± std	mean ± std	mean ± std
NSGA-II						
Ion	11.32 ± 2.3	9.98 ± 3.57	8.42 ± 3.44	8.38 ± 2.91	8.68 ± 2.95	8.28 ± 3.14
Spect	13.78 ± 2.14	21.36 ± 2.23	21.22 ± 2.09	19.84 ± 2.19	19.12 ± 2.4	17.34 ± 2.67
Yeast <sub>1</sub>	15.84 ± 2.21	40.66 ± 11.6	47.44 ± 13.19	51.72 ± 9.94	50.28 ± 8.85	46.5 ± 10.63
Yeast <sub>2</sub>	8.56 ± 1.7	15.18 ± 6.71	20.24 ± 6.87	24.34 ± 4.76	22.46 ± 4.25	20.84 ± 4.63
Abal <sub>1</sub>	8.7 ± 1.58	10.98 ± 1.8	10.56 ± 2.09	10.02 ± 1.87	9.58 ± 1.47	9.12 ± 1.77
Abal <sub>2</sub>	8.76 ± 1.51	6.68 ± 1.77	6.18 ± 1.44	5.7 ± 1.83	5.98 ± 1.73	7.08 ± 2.59
NSGA-II SDO 0.5						
Ion	11.32 ± 2.3	12.86 ± 2.01	12.2 ± 2.02	11.3 ± 1.58	11.34 ± 1.93	10.16 ± 1.5
Spect	13.78 ± 2.14	21.96 ± 2.13	21.42 ± 2	20.84 ± 2.28	19.78 ± 2.12	19.2 ± 1.86
Yeast <sub>1</sub>	15.84 ± 2.21	46.24 ± 5.82	54.98 ± 4.65	54.9 ± 5.34	54.56 ± 7.1	52.3 ± 6.99
Yeast <sub>2</sub>	8.56 ± 1.7	18.48 ± 3.51	23.82 ± 2.96	25.18 ± 2.57	25.22 ± 2.94	24.4 ± 3.2
Abal <sub>1</sub>	8.7 ± 1.58	11.46 ± 1.5	11.2 ± 1.7	11.04 ± 1.28	10.42 ± 1.47	10.14 ± 1.57
Abal <sub>2</sub>	8.76 ± 1.51	13.86 ± 1.85	13.86 ± 2.89	10.88 ± 4.02	9.54 ± 4.3	9.54 ± 4.79

Table 10

Average number of distinct solutions (± std. deviation) occurring at the first front of generations 1, 10, 20, 20, 40 and 50 for canonical SPEA2 and SPEA2 SDO methods.

Data set	Generation					
	1	10	20	30	40	50
	mean ± std	mean ± std	mean ± std	mean ± std	mean ± std	mean ± std
SPEA2						
Ion	11.32 ± 2.3	10.6 ± 3.45	9.94 ± 3.32	9.42 ± 2.89	8.78 ± 2.94	8.46 ± 2.72
Spect	13.78 ± 2.14	21.6 ± 2.93	20.88 ± 2.31	19.56 ± 2.05	18.76 ± 2.21	17.92 ± 2.35
Yeast <sub>1</sub>	15.84 ± 2.21	41.82 ± 12.01	48.42 ± 11.65	46.18 ± 14.32	46.14 ± 15	47.12 ± 11.44
Yeast <sub>2</sub>	8.56 ± 1.7	15.58 ± 5.96	21.7 ± 5.73	23.06 ± 4.65	22.32 ± 3.85	20.7 ± 5.32
Abal <sub>1</sub>	8.7 ± 1.58	11 ± 1.88	10.86 ± 1.96	10.14 ± 1.73	9.38 ± 1.87	9.08 ± 1.59
Abal <sub>2</sub>	8.76 ± 1.51	6.8 ± 2.02	6.76 ± 2.14	6.98 ± 3.08	6.9 ± 2.73	6.3 ± 2.32
SPEA2 SDO 0.5						
Ion	11.32 ± 2.3	13.24 ± 2.58	12.3 ± 2.22	11.2 ± 2.17	11.02 ± 1.95	10.18 ± 1.78
Spect	13.78 ± 2.14	22.1 ± 2.31	21.44 ± 2.52	21.34 ± 2.06	20.46 ± 1.89	20.12 ± 1.85
Yeast <sub>1</sub>	15.84 ± 2.21	46.52 ± 4.83	55.52 ± 4.48	56.96 ± 5.11	55.78 ± 5.1	54.48 ± 5.64
Yeast <sub>2</sub>	8.56 ± 1.7	18.14 ± 3.34	23.94 ± 3.11	25.66 ± 2.38	25.6 ± 2.47	25.36 ± 2.88
Abal <sub>1</sub>	8.7 ± 1.58	11.28 ± 1.41	11.44 ± 1.28	10.98 ± 1.29	10.16 ± 1.61	9.92 ± 1.54
Abal <sub>2</sub>	8.76 ± 1.51	14.22 ± 1.87	13.28 ± 3.15	11.08 ± 4.52	10.5 ± 4.52	9.62 ± 4.53

solutions are reported when using the Ion dataset and NSGA-II SDO compared to NSGA-II (right column in Table 11). The same is observed in other datasets such as the Spect and Yeast<sub>2</sub>, where the number of duplicates for the semantic-based method is half compared to the NSGA-II.

When we turn our attention to Table 12, reporting the duplicates when using the SPEA2 algorithm and our proposed SDO, we can observe the same trend as before: the number of duplicates is drastically reduced in the semantic-based method compared

to the canonical SPEA2 in all the datasets used in this study, except in the Yeast<sub>1</sub> dataset, where a marginally reduction is observed for the semantic-based method going from 8.51 (canonical SPEA2) to 6.52. It is important to note that our proposed SDO reduces the number of duplicates thanks to its mechanism to promote diversity through the use of another objective to be optimised, rather than having an explicit mechanism, for example, to penalise individuals already present in the population with the hope to eliminate this undesired effect during evolution.

**Table 11**Average number of duplicates ( $\pm$  std. deviation) occurring at the first Pareto front of generations 1, 10, 20, 20, 40 and 50 for canonical NSGA-II and NSGA-II SDO methods.

Data set	Generation					
	1	10	20	30	40	50
	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std
NSGA-II						
Ion	1.73 $\pm$ 2.75	29.49 $\pm$ 104.89	37.08 $\pm$ 117.41	36.18 $\pm$ 113.71	35.36 $\pm$ 106.18	43.58 $\pm$ 114.89
Spect	1.10 $\pm$ 0.55	2.51 $\pm$ 3.53	4.31 $\pm$ 10.14	7.09 $\pm$ 20.36	11.30 $\pm$ 34.19	15.11 $\pm$ 44.10
Yeast <sub>1</sub>	2.00 $\pm$ 3.72	5.09 $\pm$ 32.11	6.07 $\pm$ 31.44	6.73 $\pm$ 23.29	8.47 $\pm$ 26.10	9.77 $\pm$ 31.91
Yeast <sub>2</sub>	3.03 $\pm$ 5.67	12.49 $\pm$ 68.64	6.88 $\pm$ 37.38	8.62 $\pm$ 31.64	14.65 $\pm$ 48.69	19.27 $\pm$ 61.88
Abal <sub>1</sub>	1.15 $\pm$ 0.46	12.23 $\pm$ 56.19	25.61 $\pm$ 78.06	33.68 $\pm$ 88.88	43.86 $\pm$ 106.38	46.55 $\pm$ 111.15
Abal <sub>2</sub>	5.39 $\pm$ 12.21	74.85 $\pm$ 174.41	79.33 $\pm$ 180.00	87.72 $\pm$ 187.83	80.33 $\pm$ 181.22	64.54 $\pm$ 164.32
NSGA-II SDO 0.5						
Ion	1.73 $\pm$ 2.75	5.70 $\pm$ 25.70	6.53 $\pm$ 33.30	9.13 $\pm$ 36.99	10.40 $\pm$ 37.92	11.49 $\pm$ 40.81
Spect	1.10 $\pm$ 0.55	2.33 $\pm$ 2.64	3.47 $\pm$ 5.48	5.24 $\pm$ 11.63	6.59 $\pm$ 12.26	7.53 $\pm$ 14.85
Yeast <sub>1</sub>	2.00 $\pm$ 3.72	2.29 $\pm$ 3.56	3.30 $\pm$ 5.37	4.70 $\pm$ 12.37	5.86 $\pm$ 16.10	6.89 $\pm$ 19.75
Yeast <sub>2</sub>	3.03 $\pm$ 5.67	2.07 $\pm$ 3.32	3.35 $\pm$ 9.78	5.45 $\pm$ 17.62	8.12 $\pm$ 23.76	10.98 $\pm$ 27.64
Abal <sub>1</sub>	1.15 $\pm$ 0.46	4.46 $\pm$ 14.61	8.73 $\pm$ 36.10	13.59 $\pm$ 39.66	23.00 $\pm$ 59.09	29.88 $\pm$ 74.32
Abal <sub>2</sub>	5.39 $\pm$ 12.21	9.87 $\pm$ 39.35	19.20 $\pm$ 78.21	34.11 $\pm$ 114.74	39.46 $\pm$ 123.19	39.61 $\pm$ 120.37

**Table 12**Average number of duplicates ( $\pm$  std. deviation) occurring at the first Pareto front of generations 1, 10, 20, 20, 40 and 50 for canonical SPEA2 and SPEA2 SDO methods.

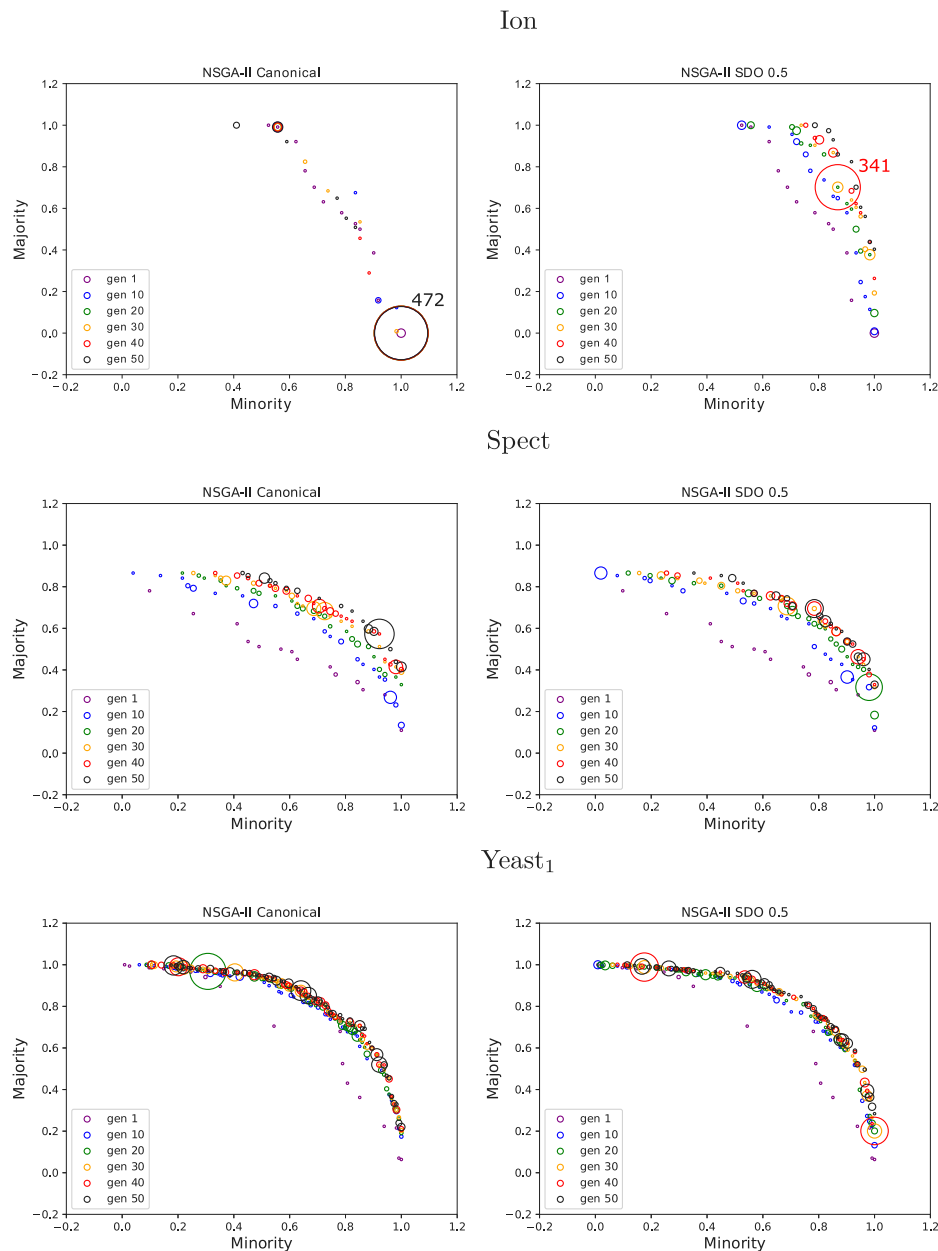
Data set	Generation					
	1	10	20	30	40	50
	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std	mean $\pm$ std
SPEA2						
Ion	1.73 $\pm$ 2.75	24.26 $\pm$ 90.64	28.76 $\pm$ 92.82	33.69 $\pm$ 101.81	38.58 $\pm$ 109.38	42.00 $\pm$ 109.99
Spect	1.10 $\pm$ 0.55	2.23 $\pm$ 2.84	4.14 $\pm$ 8.53	7.13 $\pm$ 25.50	10.93 $\pm$ 35.49	17.01 $\pm$ 46.47
Yeast <sub>1</sub>	2.00 $\pm$ 3.72	5.13 $\pm$ 32.33	6.31 $\pm$ 30.91	8.29 $\pm$ 37.45	8.69 $\pm$ 32.28	8.51 $\pm$ 27.74
Yeast <sub>2</sub>	3.03 $\pm$ 5.67	8.24 $\pm$ 54.33	6.07 $\pm$ 29.75	11.18 $\pm$ 41.72	16.48 $\pm$ 54.99	19.68 $\pm$ 64.16
Abal <sub>1</sub>	1.15 $\pm$ 0.46	9.39 $\pm$ 45.68	16.60 $\pm$ 67.21	35.25 $\pm$ 93.86	46.58 $\pm$ 110.93	48.44 $\pm$ 116.32
Abal <sub>2</sub>	5.39 $\pm$ 12.21	72.51 $\pm$ 172.07	68.04 $\pm$ 167.64	68.05 $\pm$ 167.39	70.25 $\pm$ 169.38	76.40 $\pm$ 176.26
SPEA2 SDO 0.5						
Ion	1.73 $\pm$ 2.75	4.60 $\pm$ 20.22	8.21 $\pm$ 29.47	10.38 $\pm$ 38.35	13.95 $\pm$ 48.19	15.54 $\pm$ 53.57
Spect	1.10 $\pm$ 0.55	2.10 $\pm$ 2.26	3.33 $\pm$ 4.11	4.64 $\pm$ 6.97	5.85 $\pm$ 8.66	7.25 $\pm$ 10.74
Yeast <sub>1</sub>	2.00 $\pm$ 3.72	2.17 $\pm$ 3.16	3.32 $\pm$ 5.09	4.79 $\pm$ 11.62	5.21 $\pm$ 10.52	6.52 $\pm$ 16.93
Yeast <sub>2</sub>	3.03 $\pm$ 5.67	2.34 $\pm$ 6.28	2.91 $\pm$ 4.72	4.32 $\pm$ 6.68	6.68 $\pm$ 15.05	8.01 $\pm$ 17.53
Abal <sub>1</sub>	1.15 $\pm$ 0.46	4.43 $\pm$ 15.34	8.32 $\pm$ 28.41	14.34 $\pm$ 42.79	20.03 $\pm$ 52.82	27.40 $\pm$ 71.82
Abal <sub>2</sub>	5.39 $\pm$ 12.21	9.24 $\pm$ 33.99	24.37 $\pm$ 93.08	33.17 $\pm$ 111.46	37.14 $\pm$ 118.02	41.74 $\pm$ 125.53

There is one more element worth noting from [Tables 11](#) and [12](#), which is the standard deviation of duplicated solutions. The standard deviation is notably large when compared to its respective mean, however, again these standard deviations drop significantly for the semantic method compared to the canonical method. These comparatively large standard deviations suggest that while the number of distinct individuals that experience large levels of duplication in the population is somewhat low, there may be a small subset of individuals which experience a level of duplication on an order of magnitude greater than the rest of the population. There are further ramifications of having such large standard deviations in duplication size, which will be discussed in greater detail when further analysing the plots depicted in [Figs. 5, 6, 7](#) and [8](#).

To explain how the semantic distance-based method is promoting diversity we will briefly return to some of the core functionality of the EMO framework. NSGA-II and SPEA2 are partial ordering methods, that is that when the new population is being formulated, the new population is filled with individuals from the entire approximated fronts first (that is the full Pareto front of a given dominance rank) and subsequently the remainder of the population is filled based on either the crowding distance operation, in the case of canonical methods, or by the semantic crowding-distance for the semantic distance-based methods. An important aspect of this mechanism is that the population size is fixed, therefore removing duplication from the lower dominance rank Pareto fronts allows more individuals from the higher

ranked Pareto fronts to be retained in the population at each generation. This in turn leads to a greater spread in the available genetic material for the algorithm to work with, thus promoting diversity. Furthermore, since the pivot is attracting more unique solutions to the first front, this in combination with the decreased duplication of solutions lead to greater overall performance in the semantic distance-based approach. [Figs. 5](#) and [6](#) show solutions from the first approximated Pareto front for NSGA-II and NSGA-II SDO, for the Ion, Spect and Yeast<sub>1</sub> datasets and for the Yeast<sub>2</sub>, Abal<sub>1</sub> and Abal<sub>2</sub> datasets, respectively, and likewise, [Figs. 7](#) and [8](#) for SPEA2 and SPEA 2 SDO, respectively, show solutions from the first approximated Pareto front for generations 1, 10, 20, 30, 40 and 50, represented with different coloured hollow circles. All plots represent results from a single seeded run, chosen at random, in order to avoid a biased analysis of the results. The left-hand column and right-hand column of [Figs. 5](#) to [8](#) show the results of the canonical EMO method and the SDO method, respectively.

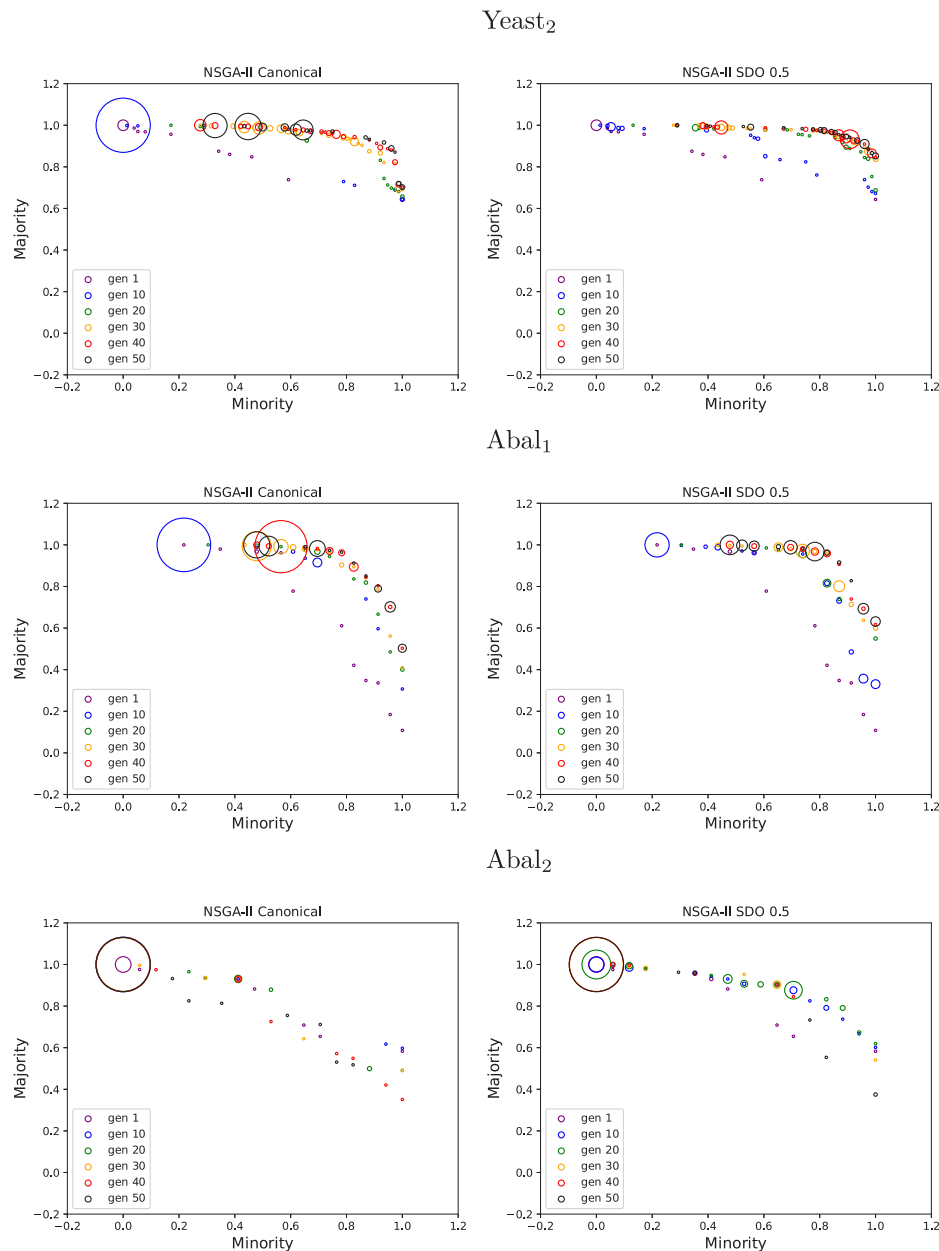
The marker size represents the number of candidate solutions found at a particular location in the objective space and thus is an indication of the frequency of duplication. Since the results are for a single run there is some variability when compared to our analysis of [Tables 9–12](#), as these results are based on the average across 50 independent runs, but in general they tend to conform. In particular Yeast<sub>2</sub> and Abal<sub>1</sub> demonstrate clearly the increase in duplication, with markers of much greater area being observed in the EMO canonical methods. The Ion and Abal<sub>2</sub>



**Fig. 5.** Duplicate frequency of individuals at first Pareto front for Ion, Spect and Yeast<sub>1</sub> dataset for generation 1, 10, 20, 30, 40 and 50 for NSGA-II and NSGA-II SDO for a single run.

datasets for NSGA-II and Abal<sub>2</sub> dataset for NSGA-II SDO exhibit rather unusual behaviours in terms of the evolution of their respective approximated Pareto fronts. It would be expected that fronts of subsequent generations ought to move towards and ideal region of maximisation, i.e. closer to point [1,1], as clearly observed with Spect data. In other words, the expected behaviour is that generation 50 (black hollow circle) is closer to [1,1] as opposed to generation 1 (purple hollow circle) being closer to [1,1], however this is not the case in with the aforementioned datasets. This can again be explained by the duplication of results. In Fig. 5 for the Ion dataset NSGA-II method (top left), the point at [1,0] has a large number of duplicates with 472 solutions at this location, as shown in the plot. However, the total population size is set at 500. Therefore as a result of the large duplication occurring at this point preferable solutions are being 'pushed' out of the solution set as the first front is exceeding the population size. This causes the approximate front to gradually recede and leads to a decrease in performance. This behaviour

can be implicitly understood from the standard deviation of the duplicates Tables 11 and 12. In particular, both Ion, Abal<sub>1</sub> and Abal<sub>2</sub> have standard deviations greater than 100 for both NSGA-II and SPEA2. Since the semantic distance-based methods reduce the overall duplication and better promote individuals to the first approximated Pareto front this issue is not as readily observed for these methods but does still occur. For instance, if we look at the semantic approach for Ion NSGA-II SDO (top-right of Fig. 5) we see that point [0.93, 0.72] grows to a total number of duplicates of 341 (as shown in the plot) at that location for generation 40 but by generation 50 all of these points have been removed as more recently created candidate solutions now dominate this point. Abal<sub>2</sub> was the only one notable dataset for the semantic method that failed to tackle this issue adequately (bottom row in Figs. 6 and 8). Even though Abal<sub>2</sub> saw a drop in the duplication averages and standard deviation from the canonical to semantic methods, the poorer performance of Abal<sub>2</sub> can be attributed to



**Fig. 6.** Duplicate frequency of individuals at first Pareto front for  $Yeast_2$ ,  $Abal_1$  and  $Abal_2$  dataset for generation 1, 10, 20, 30, 40 and 50 for NSGA-II and NSGA-II SDO for a single run.

the relatively large standard deviation in duplication average still present (Tables 11 and 12).

#### 6.4. Size of GP solutions

Bloat, increase in average tree size without a corresponding increase in performance, is a phenomenon commonly observed in GP variable length representations, such as the one used in this study. Fig. 9 shows the average number of nodes per run for each one of the 50 independent runs for each dataset and for each approach used in this work, including NSGA-II and SPEA2, shown in the left and in the right of Fig. 9, respectively, including their semantic-based variants, setting UBSS = 0.5 for each of these three variants (SDO, SSC and SCD).

We have grouped, for each of the six datasets used in this work (see Table 1), the average number of nodes evaluated by either NSGA-II/SPEA2, SDO, SSC or SCD. From Fig. 9, it is easy to observe that when we compare the semantic-based approaches (last three

box-plots, from left to right, for each of the six datasets groups) either using NSGA-II (left) or SPEA (right), SDO (second box-plot, from left to right) tends to evaluate more nodes, hence larger trees, compared to the other two semantic-based approaches (last two box-plots for each of these six groups) as well as their EMO counterparts (first box-plot for each of these groups). This is particularly visible in the  $Abal_2$  dataset.

When we compare the number of nodes evaluated by SDO vs. NSGA-II or SPEA2, left and right of Fig. 9, respectively, we can see that SDO does not evaluate many more nodes compared to the two canonical EMO approaches when using the Ion, Spect and  $Yeast_2$  datasets. From this analysis, it is interesting to note that there is a tendency to get a better Pareto front when more nodes are evaluated. For example, see in Fig. 4, how the Pareto front by the SDO approach is better compared to the other three approaches in the  $Abal_2$  dataset (right-hand side of the figure). SDO evaluates significantly more nodes compared to the other three approaches in this dataset (see right-hand side of Fig. 9).



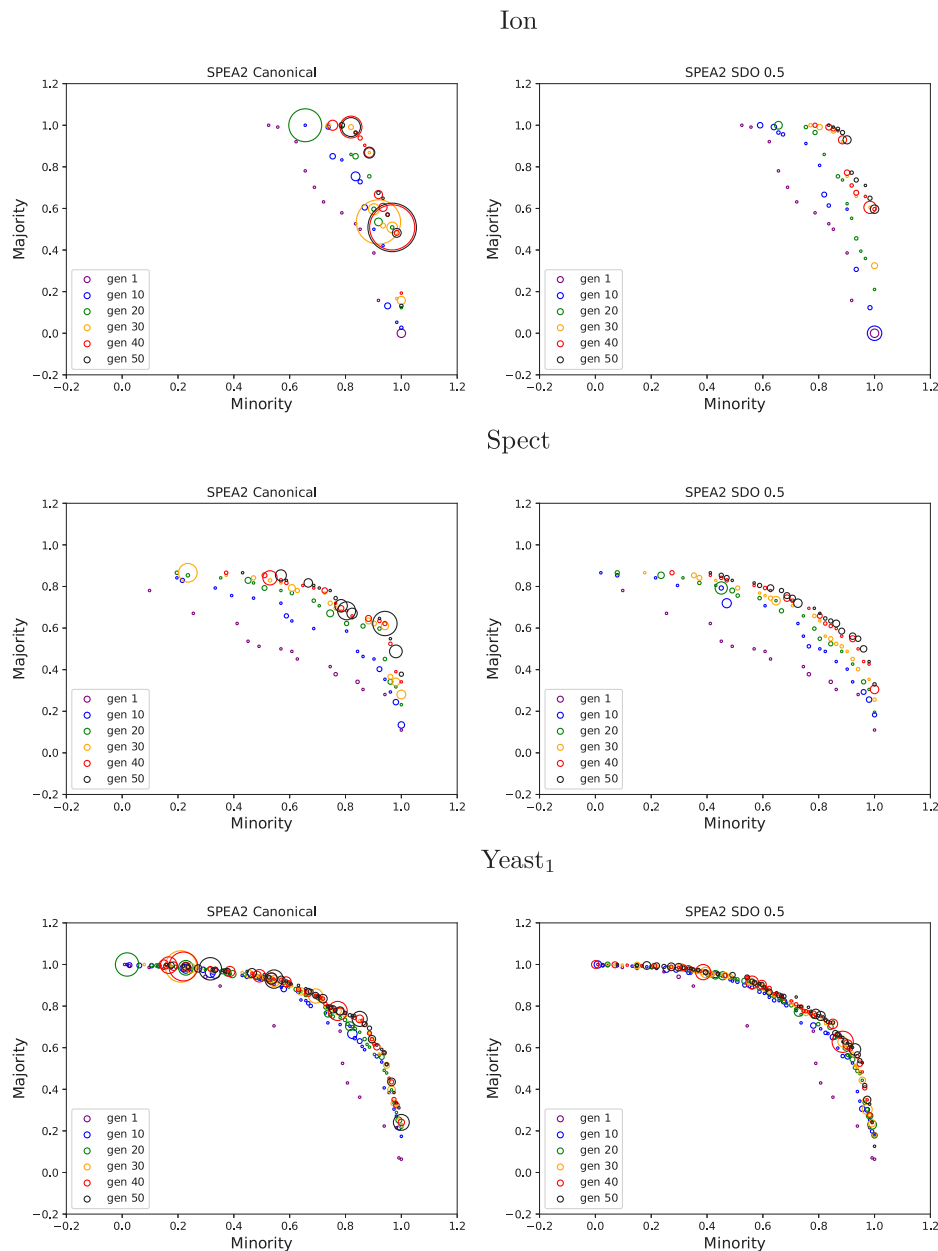


Fig. 7. Duplicate frequency of individuals at first Pareto front for Ion, Spect and Yeast<sub>1</sub> dataset for generation 1, 10, 20, 30, 40 and 50 for SPEA2 and SPEA2 SDO for a single run.

The same tendency is observed when using the Abal<sub>1</sub> dataset. SDO evaluates more nodes compared to NSGA-II (left of Fig. 9) and achieves a better Pareto front (see middle-top of Fig. 4). This analysis suggests that evaluating more nodes leads to better results in the objective space, which means this additional code growth should not be characterised as bloat.

## 7. Limitation of semantic-based distance as an additional criterion

### 7.1. Using MOEA/D

To highlight some of the limitations of the SDO approach we also look at a decomposition approach known as Multi-Objective Evolutionary Algorithm with Decomposition (MOEA/D) [9]. With MOEA/D we decompose the optimisation problem into a set of scalar optimisation problems. A scalar optimisation function  $g$ ,

Ion

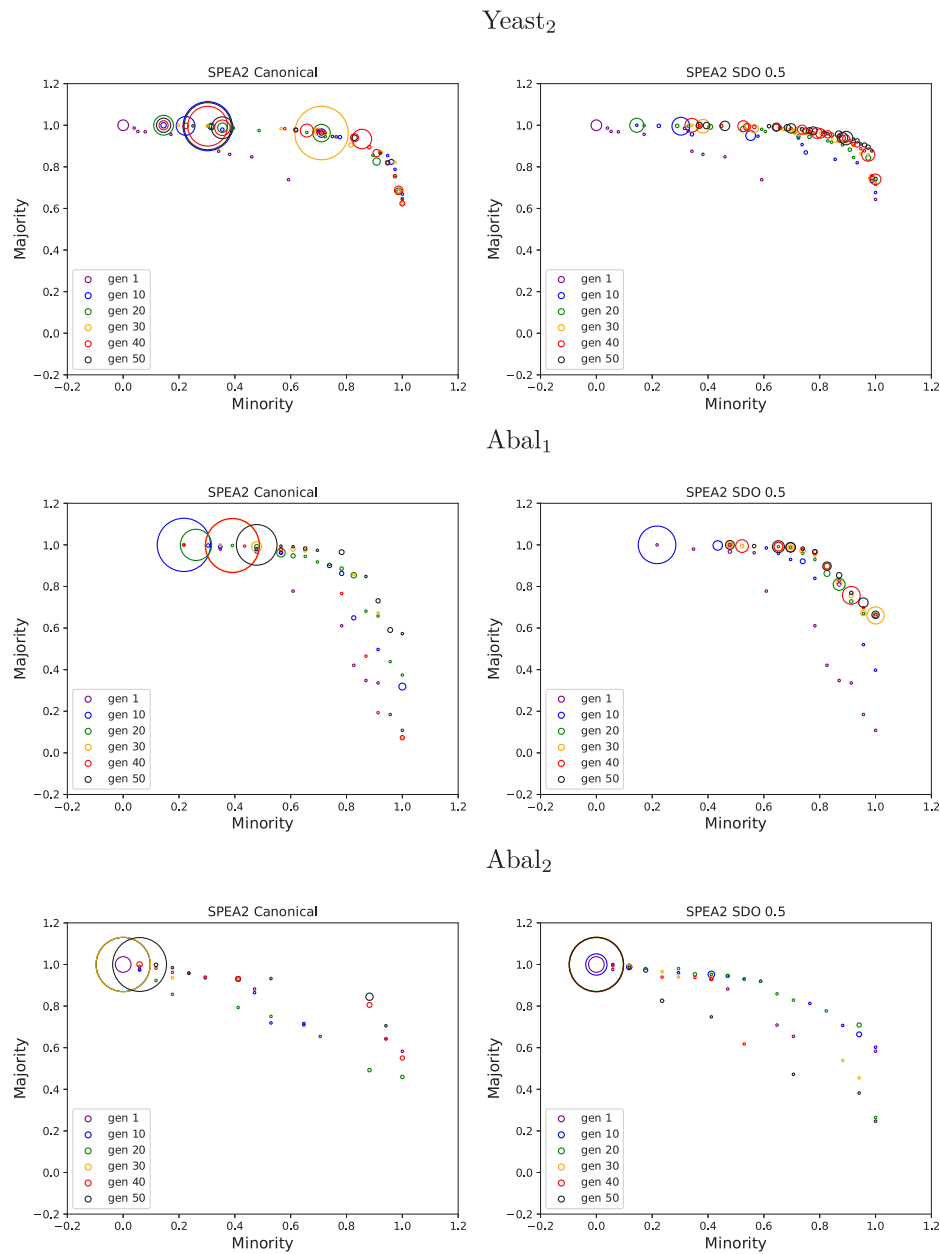
Spect

Yeast<sub>1</sub>

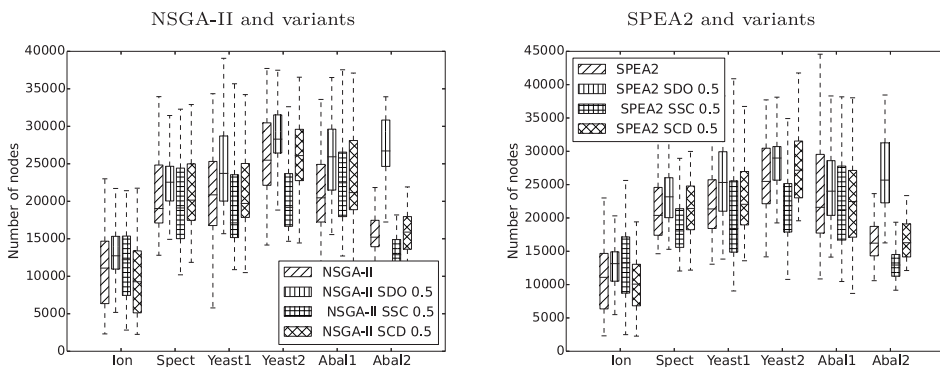
along with a uniform distribution of weight vectors  $\lambda_i$  are used to define each sub-problem. It is important to note that each sub-problem relies only on neighbouring sub-problems for evolution, that is crossover, mutation and selection only occur for individuals within a given neighbourhood. In the original MOEA/D approach, three aggregation approaches are discussed, namely the Weighted sum, Tchebycheff and Penalty-based Boundary Intersection (PBI) approach. To demonstrate that our approach is best suited to Pareto dominance-based approaches, we use one of these three MOEA/D methods. As such, the Tchebycheff approach was selected as (i) it does not require any additional parameters, unlike the PBI approach and (ii) since it performed better than the Weighted Sum approach in baseline tests.

The scalar optimisation of the Tchebycheff approach  $g^{tch}$  is given by Eq. (7)

$$\min(g^{tch}(x|\lambda)) = \max_{1 \leq j \leq m} \{\lambda_j |f_j(x) - z_j|\} \quad (7)$$



**Fig. 8.** Duplicate frequency of individuals at first Pareto front for  $Yeast_2$ ,  $Abal_1$  and  $Abal_2$  dataset for generation 1, 10, 20, 30, 40 and 50 for SPEA2 and SPEA2 SDO for a single run.



**Fig. 9.** Average size of GP individuals, computed by using the entire population per generation for each independent run, using NSGA-II (left) and SPEA2 (right) and their corresponding semantic-based variants (SDO, SSC and SCD, setting UBSS = 0.5).

**Table 13**

Average ( $\pm$  standard deviation) hypervolume of evolved Pareto-approximated fronts and PO fronts for MOEA/D over 50 independent runs.

Dataset	MOEA/D	
	Hypervolume	
	Average	PO front
Ion	0.823 $\pm$ 0.031	0.932
Spect	0.537 $\pm$ 0.024	0.652
Yeast <sub>1</sub>	0.837 $\pm$ 0.008	0.877
Yeast <sub>2</sub>	0.945 $\pm$ 0.013	0.978
Abal <sub>1</sub>	0.808 $\pm$ 0.097	0.959
Abal <sub>2</sub>	0.602 $\pm$ 0.100	0.798

where  $j = \{1, 2 \dots m\}$  are the dimensions of the objective space,  $f_j$  is the objective function and  $z_j$  is the ideal point representing the best objective function value found in the population so far for the  $j$ th objective. For a more detailed step through of the method, the original paper by Zhang et al. [9] describes the algorithm in detail.

Only one additional step is required for implementing the SDO approach in MOEA/D. The inclusion of this step can occur at any stage prior to implementing the aggregation and selection process but after mutation and crossover operations have been performed. Since, the dominance relation is not naturally used in the MOEA/D algorithm, we instead create a dummy population. That is, the child and offspring populations are joined together and the pivot is derived from this joint population as is the case of the framework used in our NSGA-II and SPEA2 approaches. Once the semantic distances have been calculated, this joint population is discarded and the MOEA/D algorithm works as expected.

## 7.2. Results for SDO in MOEA/D

As was the case in Section 6.2, the average and accumulated hypervolumes are calculated for each of the six datasets (see Table 1) and have been produced in Table 14. A fixed neighbourhood size of 20 was used for all experiments. As before the Friedman test is used to test the null hypothesis that the median performance of all groups in the same block are the same. We reject the null hypothesis at the  $\alpha = 0.01$  significance level, concluding that the median performance for all groups differs. When we perform multiple comparisons against the baseline results (using MOEA/D only), from Table 13, and each of the SDO configurations, from Table 14, we found that the baseline method outperformed the SDO method, for most datasets, except for the Abal<sub>2</sub>.

In Table 14 when looking at the average and accumulated hypervolumes, barring Abal<sub>2</sub>, we can see that none of the datasets perform better than their respective canonical results, shown in Table 13. In fact, it was found that for these datasets the canonical approach performed better. The results for Abal<sub>2</sub>, highlighted in boldface, are performing better for all UBSS and LBSS values. The seemingly conflicting results are due to the high level of variance in the canonical results, where the standard deviation is  $\pm 0.100$ . This unexplained variance is accounted for in the Friedman test.

In order to understand why MOEA/D fails to produce similar results to NSGA-II and SPEA2 when using SDO, it is important to highlight how these algorithms differ in terms of searching for solutions in objective space. Both NSGA-II and SPEA2 rely on the dominance relationship to sort preferential solutions to be retained, MOEA/D instead decomposes the multi-objective problem into sub-problems and each sub-problem is assigned a weight vector which represents a specific direction or location for that particular sub-problem to be optimised to. The weight vectors are typically initialised so that they are evenly distributed, with the ultimate aim of producing a representative spread of solutions

in objective space. Fig. 10 illustrates the distinction between decomposition (left-hand side diagram) and Pareto dominance (right-hand side diagram) based approaches. Both diagrams have three numbered regions separated by a dashed line. Region 1 represents the initially randomised solutions, Region 2 represents the solutions midway through optimisation and Region 3 represents solutions towards the end of the optimisation process when maximising a potential solution.

On the left diagram, solutions have been colour coded to highlight how they optimise towards particular localities in the objective space. For simplicity sake only 3 neighbourhoods have been represented and we will assume these three neighbourhoods are from a subset of a much larger set of neighbourhoods such that none of the individuals in each neighbourhood overlap. Each neighbourhood has been colour coded with blue, yellow and red circles. Even though the solutions are initially mixed in objective space (left diagram, Region 1), as the optimisation progresses, newly produced offspring will converge towards their respective locations in objective space based on the initial weighting (left diagram, Region 2). In the final stages, the offspring will likely converge on a single point, leading to duplication of solutions (left diagram, Region 3).

In the Pareto dominance approach (right diagram, Region 1), the selection process is determined by the dominance relation. For simplicity sake only non-dominated solutions with dominance Rank 0 have been highlighted (purple circle) along with the dominated solutions with dominance rank greater than 0 (orange circle). Under the assumption that diversity is actively being promoted, as the optimisation progresses, the solutions will become more evenly spread with the dominated solutions reducing over time and subsequently more non-dominated solutions being assigned within the population (right diagram, Regions 2 and 3).

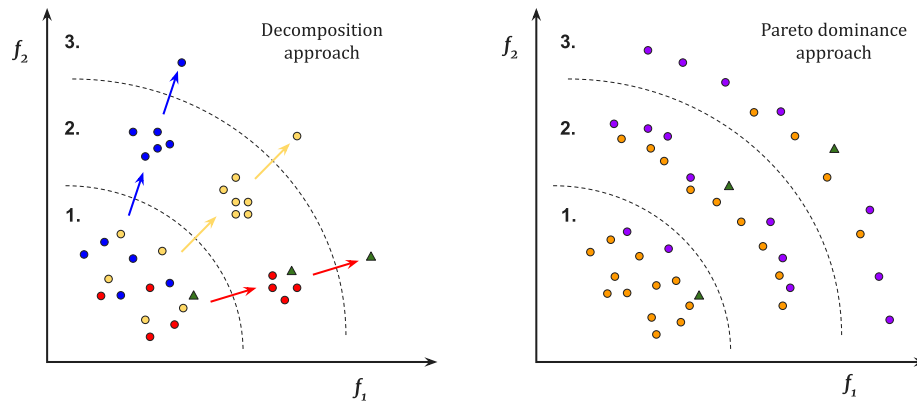
This illustrated behaviour allows for an intuitive distinction between the search mechanism in the Pareto dominance-based algorithms and the decomposition algorithm. In the Pareto dominance-based approach new solutions are free to occupy any location in the objective space as long as they satisfy the dominance relationship and crowding distance criteria, and therefore these algorithms search the objective space globally. However in the MOEA/D algorithm, search is localised, for a given sub-problem in conjunction with its weights and once the algorithm converges the diversity of solutions are constrained by the initialised weights.

Given that the pivot behaves as a point of semantic attraction, it is possible to show how this is problematic: the semantic relationship between individuals in a specific neighbourhood and the pivot may not provide beneficial updates as the selection process is determined by the aggregation function, which behaves as a localised search mechanism when the neighbourhood size is small relative to these overall population size. In other words, assuming the neighbourhood is far away from the pivot, the direction a particular sub-problem in objective space is being optimised to, will likely not correspond to the direction of attraction in semantic space and as such the aggregation function negates any benefit from drawing a semantic relationship to the pivot. To get an intuitive understanding of this, if we look at left diagram of Fig. 10, we can see that if the pivot (green triangle) is always selected in the bottom left neighbourhood (red circles), then it will not be possible to draw individuals to this neighbourhood from the other neighbourhoods (blue and yellow circles) and subsequently using the semantic relationship between the pivot and these individuals is redundant. Furthermore, as the algorithm converges, the number of duplicates in each neighbourhood increase significantly meaning the semantic distance will be the same for all individuals in a particular neighbourhood.

**Table 14**

Average ( $\pm$  standard deviation) hypervolume of evolved Pareto-approximated fronts and PO fronts for the MOEA/D semantic-based method for SDO with over 50 independent runs. Bold indicates better performance compared to the baseline MOEA/D results reported in Table 13.

	LBSS	Hypervolume							
		Average				PO front			
		UBSS							
		0.25	0.5	0.75	1.0	0.25	0.5	0.75	1.0
MOEA/D SDO									
lon	–	0.689 $\pm$ 0.039	0.702 $\pm$ 0.038	0.708 $\pm$ 0.032	0.691 $\pm$ 0.030	0.848	0.847	0.850	0.850
	0.001	0.753 $\pm$ 0.032	0.738 $\pm$ 0.038	0.727 $\pm$ 0.033	0.729 $\pm$ 0.031	0.885	0.879	0.866	0.860
	0.01	0.747 $\pm$ 0.039	0.725 $\pm$ 0.035	0.735 $\pm$ 0.036	0.727 $\pm$ 0.031	0.906	0.880	0.872	0.859
	0.1	0.749 $\pm$ 0.033	0.749 $\pm$ 0.035	0.734 $\pm$ 0.033	0.734 $\pm$ 0.032	0.885	0.892	0.865	0.869
Spect	–	0.472 $\pm$ 0.028	0.474 $\pm$ 0.027	0.474 $\pm$ 0.024	0.473 $\pm$ 0.024	0.601	0.604	0.599	0.585
	0.001	0.439 $\pm$ 0.047	0.441 $\pm$ 0.038	0.461 $\pm$ 0.040	0.458 $\pm$ 0.030	0.597	0.589	0.612	0.575
	0.01	0.456 $\pm$ 0.040	0.436 $\pm$ 0.050	0.447 $\pm$ 0.037	0.447 $\pm$ 0.035	0.588	0.582	0.611	0.609
	0.1	0.451 $\pm$ 0.051	0.444 $\pm$ 0.043	0.453 $\pm$ 0.033	0.449 $\pm$ 0.028	0.603	0.588	0.606	0.578
Yeast <sub>1</sub>	–	0.748 $\pm$ 0.036	0.743 $\pm$ 0.037	0.746 $\pm$ 0.030	0.749 $\pm$ 0.032	0.848	0.845	0.846	0.843
	0.001	0.764 $\pm$ 0.030	0.767 $\pm$ 0.035	0.776 $\pm$ 0.028	0.781 $\pm$ 0.027	0.850	0.848	0.850	0.852
	0.01	0.755 $\pm$ 0.040	0.765 $\pm$ 0.034	0.764 $\pm$ 0.028	0.768 $\pm$ 0.031	0.847	0.847	0.848	0.849
	0.1	0.758 $\pm$ 0.032	0.754 $\pm$ 0.041	0.743 $\pm$ 0.037	0.755 $\pm$ 0.037	0.846	0.846	0.840	0.844
Yeast <sub>2</sub>	–	0.833 $\pm$ 0.075	0.846 $\pm$ 0.064	0.825 $\pm$ 0.067	0.825 $\pm$ 0.060	0.968	0.959	0.958	0.956
	0.001	0.772 $\pm$ 0.092	0.800 $\pm$ 0.080	0.833 $\pm$ 0.077	0.859 $\pm$ 0.075	0.960	0.959	0.968	0.969
	0.01	0.778 $\pm$ 0.091	0.771 $\pm$ 0.073	0.794 $\pm$ 0.087	0.801 $\pm$ 0.083	0.957	0.952	0.963	0.960
	0.1	0.770 $\pm$ 0.083	0.778 $\pm$ 0.081	0.755 $\pm$ 0.086	0.766 $\pm$ 0.076	0.950	0.948	0.953	0.953
Abal <sub>1</sub>	–	0.705 $\pm$ 0.087	0.707 $\pm$ 0.099	0.700 $\pm$ 0.100	0.698 $\pm$ 0.110	0.938	0.927	0.932	0.941
	0.001	0.719 $\pm$ 0.072	0.736 $\pm$ 0.089	0.765 $\pm$ 0.068	0.794 $\pm$ 0.066	0.920	0.928	0.931	0.944
	0.01	0.678 $\pm$ 0.078	0.722 $\pm$ 0.073	0.738 $\pm$ 0.076	0.769 $\pm$ 0.068	0.882	0.908	0.899	0.936
	0.1	0.678 $\pm$ 0.086	0.672 $\pm$ 0.086	0.692 $\pm$ 0.066	0.720 $\pm$ 0.080	0.898	0.894	0.884	0.895
Abal <sub>2</sub>	–	<b>0.672 <math>\pm</math> 0.061</b>	<b>0.665 <math>\pm</math> 0.046</b>	<b>0.666 <math>\pm</math> 0.053</b>	<b>0.658 <math>\pm</math> 0.075</b>	<b>0.844</b>	<b>0.840</b>	<b>0.836</b>	<b>0.851</b>
	0.001	<b>0.666 <math>\pm</math> 0.042</b>	<b>0.660 <math>\pm</math> 0.041</b>	<b>0.676 <math>\pm</math> 0.043</b>	<b>0.676 <math>\pm</math> 0.034</b>	<b>0.851</b>	<b>0.829</b>	<b>0.852</b>	<b>0.838</b>
	0.01	<b>0.666 <math>\pm</math> 0.035</b>	<b>0.664 <math>\pm</math> 0.035</b>	<b>0.676 <math>\pm</math> 0.044</b>	<b>0.663 <math>\pm</math> 0.039</b>	<b>0.846</b>	<b>0.834</b>	<b>0.849</b>	<b>0.815</b>
	0.1	<b>0.672 <math>\pm</math> 0.046</b>	<b>0.661 <math>\pm</math> 0.036</b>	<b>0.652 <math>\pm</math> 0.040</b>	<b>0.656 <math>\pm</math> 0.046</b>	<b>0.850</b>	<b>0.831</b>	<b>0.830</b>	<b>0.828</b>



**Fig. 10.** Diagram illustrating search behaviour for decomposition and Pareto dominance based approaches when maximising a solution. Numbers 1., 2. and 3. denote typical spread of solutions in objective space at initial, intermediate and latter generations respectively. The green triangle represents a typical pivot selection.

While each individual sub-problem engages in a localised search of the objective space, when we consider all sub problems in unison MOEA/D is effectively searching globally. The real issue arises from the trade-off of exploitation versus exploration when selecting the neighbourhood size relative to the population size and specifically its incompatibility when using a pivot in this manner. Selecting a neighbourhood size of 20 was required in the baseline models to garner reasonable results and when we compare Tables 6 (results using NSGA-II and SPEA2) and Table 13 (results using MOEA/D), we can see these results are comparatively similar. If we increased the neighbourhood size so that it was substantially larger, this would likely improve exploration and make the pivot mechanism more effective, however, doing so would come at the cost of the exploitative aspects of the algorithm. It is however important to note, that using semantic distance can still be useful in decomposition approaches when we operate internally within the neighbourhood structure. In

our previous research we integrated a semantic distance based approach that semantically orders solutions prior to the selection process [41]. This method reduced duplication, as well as showing some promising results when the feasible search space were discontinuous or jagged.

### 8. Conclusions

This work proposes a new approach, named Semantic-based Distance as an additional criterion (SDO), which consists of using semantic distance values as another criterion to optimise and preferences solutions that are semantically attracted to the sparsest region of the first approximated Pareto front. We also use this distance in lieu of the crowding distance at the heart of the aforementioned EMO algorithm. Results for the new approach were tested against the canonical frameworks of NSGA-II and SPEA2 and additionally two semantic-based methods were use

as baselines, namely Semantic Similarity-based Crossover and Semantic-based Crowding Distance. It was found that SDO produced significantly better results when compared against each of these methods in terms of the hypervolume metric.

Our analysis shows that the SDO method produces more unique individuals compared to the other methods. A comparison of the first approximated Pareto fronts at specific generations showed that SDO not only attracts new individuals to the sparsest regions of the front but also reduces the amount of duplication, thus improving diversity.

### CRedit authorship contribution statement

**Edgar Galván:** Conceptualization, Supervision, Methodology, Software, Visualisation, Writing – original draft, Validation, Writing – review & editing. **Leonardo Trujillo:** Writing – review & editing. **Fergal Stapleton:** Writing – review & editing, Software, Visualisation.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6049. The opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Science Foundation Ireland. The authors wish to acknowledge the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support. We would like to thank the reviewers for the comments made on the early version of this manuscript.

### References

- [1] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [2] E. Galván, M. Schoenauer, Promoting semantic diversity in multi-objective genetic programming, in: A. Auger, T. Stützle (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*, Prague, Czech Republic, July 13–17, 2019, ACM, 2019, pp. 1021–1029, <http://dx.doi.org/10.1145/3321707.3321854>.
- [3] E. Galván-López, E. Mezura-Montes, O. Ait ElHara, M. Schoenauer, On the use of semantics in multi-objective genetic programming, in: J. Handl, et al. (Eds.), *Parallel Problem Solving from Nature – PPSN XIV: 14th International Conference*, Edinburgh, UK, September 17–21, 2016, *Proceedings*, Springer, 2016, pp. 353–363, [http://dx.doi.org/10.1007/978-3-319-45823-6\\_33](http://dx.doi.org/10.1007/978-3-319-45823-6_33).
- [4] E. Galván-López, L. Vázquez-Mendoza, L. Trujillo, Stochastic semantic-based multi-objective genetic programming optimisation for classification of imbalanced data, in: O. Pichardo-Lagunas, S. Miranda-Jiménez (Eds.), *Advances in Soft Computing*, Springer, 2016, pp. 261–272.
- [5] E. Galván, F. Stapleton, Semantic-based distance approaches in multi-objective genetic programming, in: *2020 IEEE Symposium Series on Computational Intelligence, SSCI*, 2020, pp. 149–156, <http://dx.doi.org/10.1109/SSCI47803.2020.9308386>.
- [6] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2002) 182–197.
- [7] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, *Tech. rep.*, 2001.
- [8] N.Q. Uy, N.X. Hoai, M. O'Neill, R.I. McKay, E. Galván-López, Semantically-based crossover in genetic programming: application to real-valued symbolic regression, *Genet. Program. Evol. Mach.* 12 (2) (2011) 91–119, <http://dx.doi.org/10.1007/s10710-010-9121-2>.
- [9] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731, <http://dx.doi.org/10.1109/TEVC.2007.892759>.
- [10] A. Moraglio, K. Krawiec, C.G. Johnson, Geometric semantic genetic programming, in: C.A.C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, M. Pavone (Eds.), *PPSN*, Vol. 1, in: *LNCS*, vol. 7491, Springer, 2012, pp. 21–31.
- [11] N.F. McPhee, B. Ohs, T. Hutchison, Semantic building blocks in genetic programming, in: *Proceedings of the 11th European Conference on Genetic Programming*, in: *EuroGP'08*, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 134–145, 3-540-78670-8, 978-3-540-78670-2 URL <http://dl.acm.org/citation.cfm?id=1792694.1792707>.
- [12] L. Beadle, C. Johnson, Semantically driven crossover in genetic programming, in: *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, 2008, pp. 111–116, <http://dx.doi.org/10.1109/CEC.2008.4630784>.
- [13] L. Beadle, C. Johnson, Semantically driven mutation in genetic programming, in: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, 2009, pp. 1336–1342, <http://dx.doi.org/10.1109/CEC.2009.4983099>.
- [14] Q.U. Nguyen, X.H. Nguyen, 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*, in: *EuroGP '09*, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 292–302, [http://dx.doi.org/10.1007/978-3-642-01181-8\\_25](http://dx.doi.org/10.1007/978-3-642-01181-8_25).
- [15] S. Forstenlechner, D. Fagan, M. Nicolau, M. O'Neill, Towards effective semantic operators for program synthesis in genetic programming, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, in: *GECCO '18*, ACM, New York, NY, USA, 2018, pp. 1119–1126, <http://dx.doi.org/10.1145/3205455.3205592>.
- [16] T. Dou, P. Rockett, Comparison of semantic-based local search methods for multiobjective genetic programming, *Genet. Program. Evol. Mach.* 19 (4) (2018) 535–563, <http://dx.doi.org/10.1007/s10710-018-9325-4>.
- [17] A. Moraglio, R. Poli, Topological interpretation of crossover, in: K. Deb (Ed.), *Genetic and Evolutionary Computation – GECCO 2004: Genetic and Evolutionary Computation Conference*, Seattle, WA, USA, June 26–30, 2004. *Proceedings, Part I*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 1377–1388, [http://dx.doi.org/10.1007/978-3-540-24854-5\\_131](http://dx.doi.org/10.1007/978-3-540-24854-5_131).
- [18] E. Galván-López, R. Poli, An empirical investigation of how and why neutrality affects evolutionary search, in: M. Cattolico (Ed.), *Genetic and Evolutionary Computation Conference, GECCO 2006, Proceedings*, Seattle, Washington, USA, July 8–12, 2006, ACM, 2006, pp. 1149–1156, <http://dx.doi.org/10.1145/1143997.1144180>.
- [19] E. Galván-López, R. Poli, A. Kattan, M. O'Neill, A. Brabazon, Neutrality in evolutionary algorithms... What do we know? *Evol. Syst.* 2 (3) (2011) 145–163, <http://dx.doi.org/10.1007/s12530-011-9030-5>.
- [20] R. Poli, E. Galván-López, The effects of constant and bit-wise neutrality on problem hardness, fitness distance correlation and phenotypic mutation rates, *IEEE Trans. Evol. Comput.* 16 (2) (2012) 279–300, <http://dx.doi.org/10.1109/TEVC.2011.2132726>.
- [21] R. Poli, E. Galván-López, On the effects of bit-wise neutrality on fitness distance correlation, phenotypic mutation rates and problem hardness, in: C.R. Stephens, M. Toussaint, D. Whitley, P.F. Stadler (Eds.), *Foundations of Genetic Algorithms*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 138–164.
- [22] L. Vanneschi, M. Castelli, L. Manzoni, S. Silva, A new implementation of geometric semantic GP and its application to problems in pharmacokinetics, in: K. Krawiec, A. Moraglio, T. Hu, A.Ş. Etaner-Uyar, B. Hu (Eds.), *Genetic Programming: 16th European Conference, EuroGP 2013*, Vienna, Austria, April 3–5, 2013. *Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 205–216, [http://dx.doi.org/10.1007/978-3-642-37207-0\\_18](http://dx.doi.org/10.1007/978-3-642-37207-0_18).
- [23] Q.U. Nguyen, T.A. Pham, X.H. Nguyen, J. McDermott, Subtree semantic geometric crossover for genetic programming, *Genet. Program. Evol. Mach.* 17 (1) (2016) 25–53, <http://dx.doi.org/10.1007/s10710-015-9253-5>.
- [24] C.A.C. Coello, Evolutionary multi-objective optimization: a historical view of the field, *IEEE Comput. Intell. Mag.* 1 (1) (2006) 28–36, <http://dx.doi.org/10.1109/MCI.2006.1597059>.
- [25] C.A. Coello Coello, A comprehensive survey of evolutionary-based multi-objective optimization techniques, *Knowl. Inf. Syst.* 1 (3) (1999) 269–308, <http://dx.doi.org/10.1007/BF03325101>.
- [26] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [27] S. Bleuler, M. Brack, L. Thiele, E. Zitzler, Multiobjective genetic programming: reducing bloat using SPEA2, in: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, Vol. 1, 2001, pp. 536–543, <http://dx.doi.org/10.1109/CEC.2001.934438>.
- [28] U. Bhowan, M. Johnston, M. Zhang, X. Yao, Evolving diverse ensembles using genetic programming for classification with unbalanced data, *IEEE Trans. Evol. Comput.* 17 (3) (2013) 368–386, <http://dx.doi.org/10.1109/TEVC.2012.2199119>.
- [29] H. Zhao, A multi-objective genetic programming approach to developing Pareto optimal decision trees, *Decis. Support Syst.* 43 (3) (2007) 809–826, <http://dx.doi.org/10.1016/j.dss.2006.12.011>, *Integrated Decision Support*, URL <http://www.sciencedirect.com/science/article/pii/S016792360600217X>.

- [30] L. Shao, L. Liu, X. Li, Feature learning for image classification via multiobjective genetic programming, *IEEE Trans. Neural Netw. Learn. Syst.* 25 (7) (2014) 1359–1371, <http://dx.doi.org/10.1109/TNNLS.2013.2293418>.
- [31] T.P. Pawlak, B. Wieloch, K. Krawiec, Semantic backpropagation for designing search operators in genetic programming, *IEEE Trans. Evol. Comput.* 19 (3) (2015) 326–340, <http://dx.doi.org/10.1109/TEVC.2014.2321259>.
- [32] P.T. Anh, Q.U. Nguyen, X.H. Nguyen, M. O'Neill, Examining the diversity property of semantic similarity based crossover, in: K. Krawiec, A. Moraglio, T. Hu, A.S. Eteaner-Uyar, B. Hu (Eds.), *Genetic Programming - 16th European Conference, EuroGP 2013, Vienna, Austria, April 3-5, 2013. Proceedings*, in: *Lecture Notes in Computer Science*, vol. 7831, Springer, 2013, pp. 265–276, [http://dx.doi.org/10.1007/978-3-642-37207-0\\_23](http://dx.doi.org/10.1007/978-3-642-37207-0_23).
- [33] E. Galván-López, B. Cody-Kenny, L. Trujillo, 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, 2013*, pp. 2972–2979.
- [34] K. Krawiec, T. Pawlak, Locally geometric semantic crossover: a study on the roles of semantics and homology in recombination operators, *Genet. Program. Evol. Mach.* 14 (2013) 31–63, <http://dx.doi.org/10.1007/s10710-012-9172-7>.
- [35] N.Q. Uy, E. Murphy, M. O'Neill, N.X. Hoai, Semantic-based subtree crossover applied to dynamic problems, in: *2011 Third International Conference on Knowledge and Systems Engineering, 2011*, pp. 78–84, <http://dx.doi.org/10.1109/KSE.2011.20>.
- [36] U. Bhowan, M. Zhang, M. Johnston, Multi-objective genetic programming for classification with unbalanced data, in: A. Nicholson, X. Li (Eds.), *AI 2009: Advances in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009*, pp. 370–380.
- [37] D.N. A. Asuncion, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2007, URL <http://www.ics.uci.edu/~mlern/{MLR}epository.html>.
- [38] J.R. McDonnell, R.G. Reynolds, D.B. Fogel, Genetic programming exploratory power and the discovery of functions, in: *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming, 1995*, pp. 719–736.
- [39] E.K. Burke, S. Gustafson, G. Kendall, Diversity in genetic programming: an analysis of measures and correlation with fitness, *IEEE Trans. Evol. Comput.* 8 (1) (2004) 47–62, <http://dx.doi.org/10.1109/TEVC.2003.819263>.
- [40] C.A.C. Coello, G.B. Lamont, D.A.V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.
- [41] F. Stapleton, E. Galván, Semantic neighborhood ordering in multi-objective genetic programming based on decomposition, in: *2021 IEEE Congress on Evolutionary Computation, CEC, 2021*, pp. 580–587, <http://dx.doi.org/10.1109/CEC45853.2021.9504860>.