Genetic Programming and Evolvable Machines (2022) 23:279–304
https://doi.org/10.1007/s10710-021-09426-4

GP-DMD: a genetic programming variant with dynamic management of diversity

Ricardo Nieto-Fuentes[1] · **Carlos Segura**[1]

Received: 17 March 2021 / Revised: 2 October 2021 / Accepted: 6 November 2021 /
Published online: 21 January 2022

Abstract

The proper management of diversity is essential to the success of Evolutionary Algorithms. Specifically, methods that explicitly relate the amount of diversity maintained in the population to the stopping criterion and elapsed period of execution, with the aim of attaining a gradual shift from exploration to exploitation, have been particularly successful. However, in the area of Genetic Programming, the performance of this design principle has not been studied. In this paper, a novel Genetic Programming method, Genetic Programming with Dynamic Management of Diversity (GP-DMD), is presented. GP-DMD applies this design principle through a replacement strategy that combines penalties based on distance-like functions with a multi-objective Pareto selection based on accuracy and simplicity. The proposed general method was adapted to the well-established Symbolic Regression benchmark problem using tree-based Genetic Programming. Several state-of-the-art diversity management approaches were considered for the experimental validation, and the results obtained showcase the improvements both in terms of mean square error and size. The effects of GP-DMD on the dynamics of the population are also analyzed, revealing the reasons for its superiority. As in other fields of Evolutionary Computation, this design principle contributes significantly to the area of Genetic Programming.

Keywords Genetic programming · Diversity management · Exploration · intensification · Bloat

✉ Carlos Segura
carlos.segura@cimat.mx

Ricardo Nieto-Fuentes
nifr91@gmail.com

[1] Centro de Investigación en Matemáticas, Guanajuato, Mexico

1 Introduction

Genetic Programming (GP) is a successful paradigm of Evolutionary Algorithms (EAs) [1, 2] that has been applied to a wide variety of domains [3–5]. In this paradigm, a population of computer codes or mathematical models is evolved through the iterative application of selection, genetic and replacement operators. In the broader field of Evolutionary Computation (EC), the proper balance between exploration and exploitation is considered a cornerstone for proper performance [6]. Since there is a clear relationship between this balance and the amount of diversity maintained in the population, several strategies to alter the amount of population's diversity have been devised [7]. They are usually classified in terms of the component that is altered, and they can alleviate numerous inconveniences [8], such as premature convergence and oversampling of neutral networks, among others.

In 2015 Segura et al. [9] proposed a novel design paradigm that explicitly relates the amount of diversity maintained in the population to the stopping criterion and elapsed period of execution. Subsequent studies, performed by our research group, have yielded important advances for both single-objective continuous and combinatorial optimization [10]. The main principle behind this design paradigm is that the internal operations of EAs should depend on the amount of computational resources available, and in particular, that the initial phases should focus on exploration while the final phases should be devoted to exploitation, with a gradual transition between those stages. This principle has been successfully used to design new components, such as replacement strategies [8], and crossover operators [11]. In fact, this principle was used to design the winning strategy of the extended round of Google Hash Code 2020[1], which featured more than 100,000 participants, thus indicating the potential of this principle.

In the case of GP, several authors have already noted the impact of diversity management on performance. Several ways of measuring diversity and strategies to alter the amount of diversity maintained in the population have been devised [12]. However, to our knowledge, the design principle previously described has never been applied in the area of GP. This work analyzes the hypothesis that this design principle is also useful for GP and that state-of-the-art strategies can be advanced further. Accordingly, this paper presents a new general methodology for GP that considers an explicit and dynamic management of diversity that takes into account the stopping criterion and elapsed period. The novel proposal is called Genetic Programming with Dynamic Management of Diversity and, similarly to some of the successful single-objective optimizers, it incorporates the principle discussed through a novel replacement strategy. Specifically, it uses a multi-objective selection that considers both the accuracy and simplicity. Additionally, as in the case of single-objective optimization, the diversity is managed by means of a dynamic penalization scheme that takes into account a measure of similarity between individuals.

[1] https://codingcompetitions.withgoogle.com/hashcode/.

The components of GP-DMD were specified for dealing with Symbolic Regression (SR), which is one of the most used benchmark tasks in GP. Specifically, a tree-based GP with rather standard components is used. The accuracy is considered by minimizing the *Mean Squared Error* (MSE), whereas the simplicity is addressed by minimizing the tree size. The experimental validation takes into account several distance-like functions, and includes measures of both behavioral and structural diversity. Comparisons against a large number of GP strategies, including some diversity-aware strategies, show the important benefits of our proposal. In addition to analyses related to accuracy and simplicity, the dynamics of the population in terms of several features are studied, revealing the reasons for the superior performance of GP-DMD.

The rest of this paper is organized as follows. Section 2 provides a summary of the most relevant strategies for diversity management in the field of GP, including a taxonomy for diversity measures. Our proposal, the Genetic Programming with Dynamic Management of Diversity, is described in Sect. 3. Then, the experimental validation is presented and discussed in Sect. 4. Several state-of-the-art strategies are used to illustrate the benefits of GP-DMD and, in addition to our results, some analyses related to population dynamics are provided. Finally, the main conclusion and some lines of future work are given in Sect. 5.

2 Diversity in genetic programming

Most variants of population-based optimizers lack mechanisms to systematically alter the amount of diversity maintained in the population. Since premature convergence is one of the most common drawbacks found in the design of this kind of optimizers, several diversity management techniques have been proposed in order to influence the amount of diversity maintained in the population. In most cases, they aim to increase the amount of diversity preserved in the population so the term *diversity promotion* is used [13]. In the particular case of GP, several diversity-aware proposals that are inspired by well-established diversity management strategies [14], as well as techniques that are specific to GP, have been developed [15]. Due to the particular kinds of individuals evolved in GP, measuring the diversity of a population of individuals is a complex task. This section provides a representative sample of diversity measures and strategies to alter the amount of diversity maintained in GP. Most of the techniques discussed are used to validate the proposal put forth in this paper.

2.1 Diversity measures

Several ways of calculating diversity in GP, both for analysis and design, have been devised. However, different nomenclatures have been used to refer to similar measures and probably worse, the same term has been used to refer to different measures. For instance, the term *phenotypic distance* has been used both to refer to distances based on the fitness values [16] and distances based on the mathematical

models [17]. In order to avoid misunderstandings, we present a classification of diversity measures that considers the most typical use of the terms.

Structural diversity is related to the variety of mathematical models associated with each individual. Note that in this paper, our mathematical models are trees, so the discussion is restricted to this kind of structure. Since calculating tree-based diversity is complex and expensive [1], several ways of estimating it have been devised. These approximations can be categorized into the following classes:

- In edit distances [16], a set of edit transformations and associated costs are established. Then, the edit distance is defined as the minimum-cost sequence of edit operations that transform a given tree into another one. One of the most popular is the *ed*2 distance. In this case, the two trees compared (t_1 and t_2) are first overlapped and brought to the same structure by adding null nodes. Then, the distance is defined as $ed2(t_1, t_2) = dist(p, q) + w \sum_{l=1}^{\xi} ed2(t_{1,l}, t_{2,l})$, where p and q are the roots of t_1 and t_2, ξ is the number of children of the roots, $t_{i,j}$ is the subtree number j of the root of t_i and $w \in \mathbb{R}$ is a constant. $dist(p, q)$ is usually defined as one for unequal nodes p and q and zero for equal nodes and this was the decision adopted in this paper.

  Note that the constant w sets the importance of each level. Specifically, the values $w < 1$, $w > 1$ and $w = 1$ grant more importance to the upper levels, to the lower levels or the same for all levels, respectively. The value $w = 0.5$ is broadly used in GP and is the value adopted in this paper.
- Subtree based distances take into account the set of subtrees appearing in each candidate solution to estimate their differences. The proposal by Keijzer (*kj*) [18] is probably the most popular belonging to this group. This distance is defined as follows: $kj(t_1, t_2) = |S_{t_1} \cup S_{t_2}| - |S_{t_1} \cap S_{t_2}|$, where S_{t_i} is the set of subtrees of t_i. In a more recent paper [19], a related proposal that considers both subtrees with limited sizes and their positions is devised.

Behavioral diversity is based on checking the output of the mathematical models with the training cases. There has been an increasing interest in analyzing behavioral diversity because structural diversity does not guarantee behavioral diversity [20]. These approximations can be categorized into the following classes:

- In fitness-based measures, the fitness, which summarizes in a single scalar the overall performance of the individual, is used in some way to estimate the diversity. For instance, it can be done through the variance of the fitness values appearing in the population [21].
- In semantic-based measures, the semantic ($\mathbf{s}(t)$) of a program t is defined as a vector containing, for each training case, its output or a value related to its performance. Then, this vector is used to establish the differences between individuals [22]. The *Sampling Semantic Distance* (SSD) [23]

is a popular example belonging to this category, and it is calculated as $SSD(t_1, t_2) = \frac{1}{m}(|s_1(t_1) - s_1(t_2)| + \cdots + |s_m(t_1) - s_m(t_2)|)$, where $s_i(t_j)$ is the semantic value associated to program t_j in the i-th test case and m is the total number of training cases.

Finally, note that entropy-based measurements are also quite typical, but they can be considered as special cases of the previous ones. In these kinds of schemes, the mathematical models are partitioned in some way and the spread of the population over the partition is taken into account. In order to partition the population, some of the methods previously discussed are used. For instance, in [21], edit distances to a reference tree are used to establish the partitions. Then, the entropy of the population P is calculated by taking into account the fraction of individuals belonging to each group of the partition. Thus, this is an entropy-based structural diversity. An example where the partitions are established in terms of the fitness function values, i.e., an entropy-based behavioral diversity, is presented in [24].

2.2 Diversity management strategies

Since a large amount of diversity management strategies have been devised, several taxonomies to classify them have been proposed. The taxonomy provided in [13] takes into account three independent categories: the element that is considered (lineage, genotype or phenotype), the type of selection that is modified (parent, survival or both), and the context-dependency. The taxonomy provided in [7] also takes into account the component that is modified but with a more fine-grained view. It identifies *selection-based*, *crossover/mutation-based*, *population-based* and *replacement-based* techniques, among others. Additionally, it also differentiates among maintaining, controlling and learning techniques. In the maintaining techniques, the diversity is not measured globally but a modification that aims to alter the amount of diversity maintained in the population (usually to increase it) is included. Note that schemes that measure some distances among individuals but do not use a global measure of diversity, such as the crowding schemes, belong to this group. In the controlling techniques, the diversity is measured and it is used as feedback to steer the evolution. Finally, in the learning techniques, a long-term history of the diversity is stored and used in combination with machine learning techniques to alter the optimization process. This research focuses on management strategies that can be classified as maintaining techniques and in order to further classify them, the altered component is identified.

In the *selection-based* techniques, the parent selection procedure is modified, usually with the aim of biasing the search towards uncrowded regions. Some of the most popular strategies belonging to this category are the following:

- ϵ-lexicase selection (ϵ-lex) [25] is an adaption of the well-known lexicase selection [26] to the symbolic regression case. Lexicase selects parents by iteratively considering single cases and discarding non-elite solutions, meaning that the semantic diversity is taken into account. ϵ-lex modules the pass condition with

the aim of diversifying further the selected parents for symbolic regression by considering a threshold value (ϵ) in the filter step. Several ways of setting ϵ were tested, with the Sum-MAD strategy, which is based on considering the median of absolute deviations, providing quite robust results.

- Semantic in Selection (SIS) [27] relies on performing two kinds of selections to promote quality and behavioral diversity. For each pair of parents, the first one is selected by a tournament based on the fitness value, whereas the second one is selected by sampling a pool of *ps* individuals and then selecting, among the ones with a different semantic than the first parent, the one with the best fitness value. If none of the sampled individuals generate a different semantic, the individual is selected randomly. Note that in the case of symbolic regression, a threshold value (α) is used to compare behaviors. Thus, the behavior for a test case of two candidate solutions is considered to match when their absolute difference is lower than α.
- *k*-nobelty selection (KNOB) [17] is also based on performing two kinds of selections. Specifically, selections based on lexicase (or ϵ-lexicase for continuous semantics) are used with probability $(1-k)$, whereas the remaining selections are based on novelty. Thus, the first kind of selection comprises both diversity and quality, whereas the second type is based solely on diversity. The novelty is approximated by calculating the mean Hamming distance (with the binarization scheme presented in [28]) to a sampling of the previously generated solutions. The distance is thus calculated as $hamming(t_1, t_2) = \sum_{i=1}^{m} \left((b_1(e_i(t_1)) + b_2(e_i(t_2))) \mod 2 \right)$, where $e_i(t_j)$ is the error of the tree t_j in the i-th case and the binarization $b_j(e)$ is 1 if the error e is less than or equal to the mean error of the tree t_j in every case, and 0 otherwise. Specifically, a bounded archive A that stores the last $|A|$ evaluated solutions is maintained, and *sp* individuals are sampled to estimate the novelty. Several ways of setting k were analyzed, and schemes based on a logarithmic decay excelled.

Crossover and mutation operators are quite important for the dynamics of the populations. Thus, redefining these operators by taking into account desired effects on diversity is also quite typical. *Crossover/mutation-based* strategies alter the genetic operators and/or their probabilities. One of most popular strategies that act on the variation phase is the Diverse Partner Selection with Brood Recombination (DPSBR) [29]. DPSBR alters the probabilities of crossover (p_c) and mutation (p_m) dynamically and promotes crossing semantically distant individuals by forcing certain conditions related to the expected improvement. After selecting the parents with a typical tournament based on fitness, those conditions are checked and if they are not fulfilled, alternative parents are tested. After $n/2$ attempts, where n is the population size, the crossover is abandoned and the last pair of selected parents is accepted. Since this might be indicative of a behavioral diversity that is too low, p_c is reduced and p_m is increased by $1/n$. In order to calculate the semantic distance, the same binarization scheme as in k-nobelty is used. Additionally, brood recombination is applied, meaning that for each pair of parents, multiple recombinations (*mr*) are performed.

Replacement-based schemes modify the process of selecting the survivors of the next generation. Since this phase is in charge of erasing information, it is quite a natural component to promote diversity. The following strategies are some of the most popular in this category:

- Find Only and Complete Undominated Sets (FOCUS) [30] applies a multi-objective approach, where the objective triplets consist of fitness, size and the average squared distance to other individuals. The normalized $ed1$ distance [12] is used, which is defined as $ed1(t_1, t_2) = dist(p, q) + \sum_{l=1}^{\zeta} ed1(t_{1,l}, t_{2,l})$, where p and q are the roots of t_1 and t_2, ζ is the minimum between the number of children of p and q, and $dist(p, q)$ is defined as one for unequal nodes p and q and zero for equal nodes. Note that unlike $ed2$, this distance only takes into account the overlapped nodes, so no expanded tree is created, and it is normalized by dividing by the size of the smaller tree. Distances consider the extended population made up from the union of the current population and the offspring. Then, it applies the Pareto dominance criterion to identify the non-dominated solutions, which are selected to survive.
- Age Fitness Pareto Optimization (AFPO) [31] also applies multi-objective concepts but using a tuple that consists of fitness and age. The key idea is the concept of genetic age, which is related to the number of generations that the genetic information has been in the evolutionary process. Specifically, the age of solutions of the initial population is set to 0 and at each iteration, the age of every member of the population is increased. The age of the individuals created by crossover is inherited from the oldest parent and, in addition, in each generation a new random individual with its age set to 0 is created. Note that this step is similar to the random immigrants approach [32], but, together with the novel selection process, it further favors young individuals. Finally, the non-dominated individuals survive. However, in this case, there is a target population size n meaning that if less than n non-dominated individuals exist, some dominated individuals also survive. The dominated individuals are selected through Pareto tournaments.
- Genetic Marker Density Genetic Programming (GMD-GP) [33] is also based on Pareto dominance, but in this case each individual is associated with a tuple that consists of genetic marker density and fitness. Genetic markers are the main novelty of this proposal and they are partial trees generated by traversing the generated trees from the root to a specified depth (md). Then, the partial tree generated for each individual is used to calculate its density by considering the number of times that such a tree appears in the population. The logic behind this proposal is that, for several cases, the population converges very quickly in the top levels [34]. Note that the same replacement procedure as in AFPO is applied, meaning that all nominated individuals survive and, additionally, some dominated individuals might survive to reach the target population size.

In the case of *population-based* diversity-management strategies, the main mechanism is not focused in a single evolution component, rather the typical panmictic model with a fixed size is modified, for instance, by considering a notion of

sub-population or by altering the population size dynamically. One of the most popular strategies belonging to this category in the GP area is the Age Layered Population Structure [35, 36] (ALPS) scheme. The core idea behind ALPS is to protect recently created individuals in the population from being overshadowed by fitter and more evolved individuals. Thus, only individuals with similar ages compete among them. In order to attain this aim, the population is partitioned into layers and each layer has a capacity (*lc*) and an age limit. Specifically, the age limit of layer i (age_i) is set to $agegap \times i^2$, except for the last layer, which has no age limit. Inside each layer, a generational strategy with elitism is applied, meaning that the best *el* solutions in terms of their fitness, as well as the offspring, survive. At each generation, the individuals' ages increase and they are promoted to the next layer if their age is greater than their current layer limit. Additionally, individuals in the first layer are regenerated randomly at specific generation intervals (*agegap*), meaning that short new individuals enter in the competition. Genetic operators are applied inside layers to generate offspring and the age is inherited from the oldest parent involved.

Finally, note that some hybrid diversity management strategies that combine variants of several of the schemes presented above have also been devised. For instance, GMD-GP has been combined with lexicase to simultaneously consider the behavioral and structural diversity [37].

3 Proposal

The proposal put forth in this paper was designed by considering the hypothesis that relating the diversity management to the stopping criterion and elapsed period of execution might bring additional benefits to the GP area. Specifically, this principle was used to design a novel replacement phase. This component is an adaptation of a replacement algorithm proposed by our research group, the Replacement with Multi-objective Dynamic Diversity Control (RMDDC), which is a strategy that has been applied successfully in combinatorial single-objective optimization [8]. One of the most important features of RMDDC lies in the incorporation of penalties in the replacement phase with the aim of avoiding the survival of individuals that are too similar. The definition of similarity is based on distance-like functions which are problem-dependent, whereas the notion of being too similar is dynamic. In particular, an initial threshold distance to distinguish between penalized and non-penalized individuals is established. Then, this threshold is decreased linearly during the evolution in such a way that it attains the value 0 at the end of the optimization process. Note that this means that as the evolution progresses, closer individuals are accepted with the aim of shifting gradually from exploration to exploitation. The penalization promotes the survival of less fit but diverse solutions, but at the same time, the dynamic threshold promotes the search to focus on the most promising regions during the last phases of the optimization.

In order to test the potential of adapting RMDDC to GP, and with the aim of comparing it fairly against other diversity management strategies, we implemented quite a large number of different strategies using a Simple GP (SGP) template (Algorithm 1). The template is quite general and standard. First, an initial population

Algorithm 1: Main Procedure of Simple GP

```
   Input  : n (population size), population-initialization, selection, crossover,
            mutation, replacement, stopping-criterion, evaluation
   Output : The best solution found t_best
1  P^(0) = population-initialization(n)
2  evaluation(P^(0))
3  until stopping-criterion()
4  |  P_s = selection(P^(g))
5  |  O' = crossover(P_s)
6  |  O = mutation(O')
7  |  evaluation(O)
8  |  P^(g+1) = replacement(P^(g), O)
9  |  t_best = best-individual({t_best} ∪ P^(g+1))
10 end
11 return t_best
```

($P^{(0)}$) with n individuals is created (line 1) and evaluated (line 2). Note that g denotes the number of the current generation in the pseudocode. Then, several generations are evolved until the stopping criterion is met (lines 3–10). The evolution process is as follows. First, the parents (P_s) are selected from the current population $P^{(g)}$ (line 4). Then, they are subjected to crossover (line 5) and mutation (line 6) to create the offspring (O), which is evaluated (line 7). Finally, the new population ($P^{(g+1)}$) is selected from the offspring and current population (line 8). Note also that, at each generation, the best individual (t_{best}) is updated (line 9). The best individual is returned at the end of the evolutionary process (line 11).

In this general template, the components enumerated in the pseudocode are considered as input parameters of the strategy, meaning that different GP variants and adaptations to specific applications can be developed. Additionally, when instantiating specific components, some other input parameters might be required. When integrating a specific diversity management strategy, only the components involved in the strategy are altered, whereas the remaining ones take a default behavior. In this paper, Symbolic Regression (SR) is used for comparison purposes. Thus, default components were specified for the SR task, and in particular, a tree-based GP with very standard behavior is considered. The components selected as default for SR are the following [2]:

- *Population-initialization*: The standard ramped half and half strategy is used. It combines grow and full tree generation techniques to generate n individuals.
- *Evaluation*: Each individual is evaluated with the MSE (*Mean Squared Error*) function $\frac{1}{m}\sum_i (y_i - \hat{y}_i)^2$, where m is the number of training cases, and y_i, $\hat{y}_i$ are the expected and predicted outputs for the i-th test case of the training set. Note that individuals are only evaluated if necessary, i.e., in cases where crossover or mutation alters the tree. In our experimental validation, the terms fitness and MSE are used interchangeably.

[2] The selected components are straightforward and commonly used and, although different choices are possible, the core idea is that all algorithms share the same selection in most components.

Algorithm 2: Replacement Phase of GP-DMD

Input : P (current population), O (offspring), n (number of survivors), e_e (elapsed evaluations), e_s (stopping evaluation), d_i (initial distance threshold)
Output : New Population (NP)

```
C = P ∪ O
NP = {best-individual(C)}
C = C \ NP
τ = minimum-required-distance(d_i, e_s, e_e)
while |NP| < n do
    (C_p, C_np) = classify-individuals(NP,C,τ)
    if C_np ≠ ∅ then
        nds = non-dominated-set(C_np)            # Using Accuracy and Simplicity
        t = random-sampling(nds)
    else
        t = farthest(C_p, NP)
    end
    NP = NP ∪ {t}
    C = C \ {t}
end
return NP
```

- *Selection*: n parents are selected by applying tournament selections. In each selection a pool of individuals of size `tsize` is randomly sampled from the population and the fittest individual is selected. In case of ties, they are broken at random. The selection returns an array of parents pairs P_s.
- *Crossover*: The parents pairs are subjected to the `subtree-exchange` operator with probability p_c. In this operator, two nodes, n_a and n_b, each from a parent, are selected at random. Then, the subtrees rooted at these nodes are exchanged to create the offspring. Individuals belonging to pairs that are not subjected to crossover are moved to O' without any change.
- *Mutation*: For each individual in the offspring, the `subtree-replacement` method is applied with probability p_m. In this mutation, a node n is selected randomly and the subtree rooted in n is replaced by a new tree that is generated by the `grow` method. The method returns a new offspring array O where $p_m \times |O'|$ elements are expected to be mutated.
- *Replacement*: A generational scheme with elitism is used, meaning that the next population $P^{(g+1)}$ consists of the offspring together with the best solution from the current generation population $P^{(g)}$, but only if this solution is better than any of the offspring.
- *Stopping-criterion*: The iterations are performed until a maximum number (e_s) of objective functions are evaluated.

In the novel Genetic Programming with Dynamic Management of Diversity proposal, the only component that is modified from the SGP template is the replacement phase. Algorithm 2 details the proposed replacement. The inputs are the population (P), the offspring (O), the number of desired survivors (n) and some information that is used by the penalization approach, which is the number of elapsed evaluations (e_e), the stopping criterion set as a maximum number of evaluations (e_s) and

the initial distance threshold (d_i). The aim is to select n members to form a new population (NP) for the next generation.

GP-DMD's replacement strategy initially joins the population and offspring into a set of candidates (C) (line 1). The best individual in terms of fitness is selected from the candidates to form part of the new population (line 2) and removed from the candidates (line 3). In case of ties, the smallest individual is selected and, if there is still a tie, it is broken randomly. Then, a threshold value (τ) is calculated (line 4), which is used in subsequent steps to penalize the candidates. After the previous initialization, the algorithm performs $n-1$ iterations to select the survivors from the candidates (lines 5–15) to form the new population. At each iteration, the candidates are categorized and included either in the set of penalized candidates (C_p) or the set of non-penalized candidates (C_{np}) (line 6). Specifically, any candidate whose Distance to Closest Survivor (DCS) is lower than the threshold τ is categorized as a penalized candidate; otherwise, it is categorized as non-penalized, i.e, individuals are classified depending on its distance to its closest already selected individual. If there are non-penalized candidates, a multi-objective approach based on a tuple with two objectives related to accuracy and simplicity is used for selecting a random non-dominated candidate (lines 7–9) and the penalized individuals are ignored. Otherwise, i.e., if all candidates are penalized, the farthest individual is selected (line 11). Note that this situation might indicate that the diversity is too low, so selecting the most distant individual seems promising and aligned with previous research on RMDDC. Finally, the selected candidate is included in the new population and removed from the candidates set (lines 13–14).

Note that this general replacement strategy is quite similar to the one applied in RMDDC. However, the process to select among the potential non-penalized individuals differs. In the case of RMDDC and its variants, two different strategies were analyzed. In the first variant, a multi-objective approach based on a tuple of fitness and diversity was applied and the selection process chooses survivors at random from among the non-dominated individuals. Subsequently, a second simpler strategy based only on fitness also behaved properly [38], so the step to categorize individuals into the penalized and non-penalized classes seems to be key to proper performance. In the case of GP, both the accuracy and simplicity of the models are important. Thus, in this case, the multi-objective approach is based on a tuple of objectives that considers accuracy and simplicity, respectively. The `minimum-required-distance` function is used to set the threshold that is applied to distinguish between penalized and non-penalized individuals. In this paper, this function is similar to the one applied in RMDDC, i.e., it starts from an initial distance value d_i (a parameter of GP-DMD), which is reduced linearly. Specifically, the threshold τ is set as follows: $\tau(d_i, e_e, e_s) = d_i - \frac{d_i \times e_e}{e_s}$.

In order to fully specify our proposal for a given application, some problem-dependent decisions must be made; specifically, the objectives related to accuracy and simplicity, as well as the way to calculate distances between individuals, must be defined. In order to deal with the SR task, the following decisions were made. The accuracy and simplicity are considered by minimizing the MSE and tree sizes, respectively. In the case of the distance-like function, we considered a normalized

*ed*2 distance [3], which was previously presented. In order to restrict the distance values between two given trees t_1 and t_2 to the range [0, 1], the *ed*2 distance is divided by an upper bound of the maximum attainable distance ($\max_{ed2}(t_1, t_2)$). This upper bound is calculated by considering a tree whose structure is formed by overlapping the trees of t_1 and t_2 and creating nodes in any position where at least one of the trees has a node. Then, $\max_{ed2}(t_1, t_2)$ is calculated by considering this structure and assuming that differences appear in each of the nodes. The normalized distance used is $ed2_{norm} = \frac{ed2(t_1,t_2)}{\max_{ed2}(t_1,t_2)}$

4 Experimental validation

The main focus of this paper is to show that the design principles related to the management of diversity previously discussed provide important benefits to the field of GP. In order to show the potential of GP-DMD, the experimental validation is focused in symbolic regression, a simple and well-established benchmark for GP. The goal of SR is to generate a predictive model as an analytic function, from a set of input/output pairs [39]. GP has been shown to yield impressive results with SR, and both the simplicity and accuracy are important [40]. However, simple variants of GP that evolve a single tree have not excelled, and the state-of-the-art algorithms are hybrid schemes that incorporate other kinds of procedures, such as multiple linear regressions and/or operators specifically devoted to SR but that suffer from exponential trees growth [41, 42].

The focus of this paper is not to further develop the state-of-the-art strategies for SR. Thus, these kinds of ad-hocs and hybrid methods are not considered in our experimental validation, which is focused on comparing GP-DMD against other strategies that also evolve a single tree and that provide some specific strategies to alter the degree between exploration and intensification. As it is subsequently shown, GP-DMD provides significant advances in comparison to other diversity management schemes. However, to further explore the generality of GP-DMD, additional problems should be taken into account. This is beyond the scope of this paper, so in this sense, this validation should be considered as a first proof-of-concept.

4.1 Experimental setup

In order to validate the achievements of GP-DMD, our extensive comparison considers the following proposals: ϵ-LEX [25], SIS [27], DPSBR [29], FOCUS [30], GMD-GP [15], AFPO [31], KNOBELTY [17], ALPS [35] and SGP (Algorithm 1 with default components). All these schemes were previously presented and they were implemented by changing some of the default components of Algorithm 1 to incorporate the specific changes devised in each of these strategies. Table 1 details

[3] Note that this decision is probably the most difficult one when adapting GP-DMD to other applications. In the experimental validation, some results with alternative distance-like functions are also presented.

the parameters applied in each scheme and clarifies the component that each of the methods altered with respect to SGP. These parameters are similar to those put forth in the original proposals, with some minor tweaks to adapt each proposal to the specific problems that are addressed in this paper. The common parameters used in all the algorithms are detailed in Table 2.

In order to facilitate reproducibility, free software implementations are available for public download [4], [5]. The programming languages used are Crystal (`v1.0`) for the core framework and Ruby (`v3.0`) and Python (`v3.9`) for pre and post processing scripts. Instructions and requirements for replicating the experiments presented in this paper are provided in the repository.

Experiments were performed on 25 SR problems from the benchmark proposed in [43]. Note that some preliminary executions with additional problems were performed. Since similar conclusions could be drawn and due to time restrictions, we limited the experimentation to the first 25 problems, which are among the most popular ones. The symbolic regression problems are listed in Table 3. For each problem, the training and validation sets were created as specified in their original papers, and in order to provide a fair comparison, the same data were considered for all the executions. Note that $\mathcal{E}$ indicates an equidistant point sampling, and $\mathcal{U}$ a uniform sampling. In order to facilitate future comparisons, codes to generate the training and validation sets are included in our repository.

Finally note that, since stochastic algorithms were considered in this study, each execution was repeated 30 times and comparisons were carried out by applying a set of statistical tests. Specifically, the following tests were applied, assuming a significance level of 5%. First, a Shaphiro-Wilk test was applied to check if the values of the results followed a Gaussian distribution. If so, the Levene test was used to check for the homogeneity of the variance. When similar variances were confirmed, an ANOVA test was done; otherwise, a Welch test was performed. For non-Gaussian distributions, the non-parametric Kruskal-Wallis test was applied. The statement "algorithm A is superior than algorithm B" means that the differences between them are statistically significant and that the median obtained by A is lower than the median achieved by B. Note that the same kinds of statistical tests are used to compare fitness and sizes. The following subsections include analyses related to accuracy and simplicity, as well as some studies regarding the dynamics of the population which contribute to a better understanding of the internal behavior of the set of GP strategies considered. Additionally, some variants that alter the penalization scheme of GP-DMD are tested with the aim of better understanding the implications of this particular component.

[4] https://gitlab.com/nifr91/genetic-programming
[5] http://doi.org/10.5281/zenodo.5009057

Table 1 Component modified by each method with respect to SGP and parameterization applied

Method	Altered component	Parameters
GP-DMD	Replacement	$d_i = 0.125$
ϵ-LEX	Selection	ϵ = Sum-MAD
SIS	Selection	$\alpha = 0.01$
		$ps = 3$
KNOBELTY	Selection	$k = \exp(-\lambda e_e)$
		$\lambda = 1.3 \times 10^{-4}$
		ϵ = Sum-MAD
		$\lvert A \rvert = 2000$
		$sp = 100$
DPSBR	Crossover	$mr = 5$
FOCUS	Replacement	
AFPO	Replacement	
GMD-GP	Replacement	$md = 3$
ALPS	Population	$agegap = 3$
		$lc = {}^{n}/_{10}$
		$el = 3$
		$age_i = agegap \times i^2$

Table 2 Common GP parameters used in all the algorithms tested. The nodes 'int' and 'float' are constants defined at random when created. The number of 'variables' nodes is defined by the problem, and the functions nodes are protected, e.g., when the reciprocal function ($\text{rcpl}(x) = 1/x$) is undefined, it returns 1

n	200
`Stop-criteria` (evaluations)	76,800
`tsize`	3
p_m	0.2
p_c	0.9
Terminals	Int $\in [0, 9]$, float $\in [0, 1]$, variables
Functions	Add, sub, mult, sin, cos, exp, ln, sq, rcpl

4.2 Fitness comparison

In order to validate GP-DMD in terms of accuracy, it is compared against the above mentioned methods in the 25 selected benchmark problems in terms of fitness. Table 4 summarizes the performance of the algorithms. Specifically, it shows the number of instances where the given algorithm was either the best algorithm (lowest median fitness) or it did not exhibit a statistically significant difference with the algorithm that reported the lowest median (for detailed pairwise comparisons and

Table 3 Symbolic Regression problems

Name	Objective function	# Variables	Training set	Samples	Validation set	Samples
keijzer-01	$0.3x\sin(2\pi x)$	1	$\mathcal{E}(-1,1)$	21	$\mathcal{E}(-1,1)$	2001
keijzer-06	$\sum_{i=1}^{x}\frac{1}{i}$	1	$\mathcal{E}(1,50)$	50	$\mathcal{E}(1,120)$	120
keijzer-07	$\ln x$	1	$\mathcal{E}(1,100)$	100	$\mathcal{E}(1,100)$	1000
keijzer-08	$\sqrt{x}$	1	$\mathcal{E}(0,100)$	101	$\mathcal{E}(0,100)$	1001
keijzer-09	$\ln(x+\sqrt{x^2+1})$	1	$\mathcal{E}(0,100)$	101	$\mathcal{E}(0,100)$	1001
keijzer-10	x^y	2	$\mathcal{U}(0,1)$	100	$\mathcal{E}(0,1)$	10, 001
keijzer-12	$x^4-x^3+\frac{y^2}{2}-y$	2	$\mathcal{U}(-3,3)$	20	$\mathcal{E}(-3,3)$	361, 201
korns-01	$1.57+(24.3v)$	5	$\mathcal{U}(-50,50)$	10, 000	$\mathcal{U}(-50,50)$	10, 000
korns-05	$3+2.13\ln(w)$	5	$\mathcal{U}(-50,50)$	10, 000	$\mathcal{U}(-50,50)$	10, 000
korns-08	$6.87+11\sqrt{7.23*x*c*w}$	5	$\mathcal{U}(-50,50)$	10, 000	$\mathcal{U}(-50,50)$	10, 000
koza-01	$x^4+x^3+x^2+x$	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
koza-02	x^5-2x^3+x	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
koza-03	$x^6-2x^4+x^2$	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
nguyen-01	x^3+x^2+x	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
nguyen-03	$x^5+x^4+x^3+x^2+x$	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
nguyen-04	$x^6+x^5+x^4+x^3+x^2+x$	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
nguyen-05	$\sin(x^2)\cos(x)-1$	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
nguyen-06	$\sin(x)+\sin(x+x^2)$	1	$\mathcal{U}(-1,1)$	20	$\mathcal{U}(-1,1)$	100
nguyen-07	$\ln(x+1)+\ln(x^2+1)$	1	$\mathcal{U}(0,2)$	20	$\mathcal{U}(0,2)$	100
nguyen-08	$\sqrt{x}$	1	$\mathcal{U}(0,4)$	20	$\mathcal{U}(0,4)$	100
nguyen-09	$\sin(x)+\sin(y^2)$	2	$\mathcal{U}(0,1)$	20	$\mathcal{U}(0,1)$	100
nguyen-10	$2\sin(x)\cos(y)$	2	$\mathcal{U}(0,1)$	20	$\mathcal{U}(0,1)$	100
pagie-01	$\frac{1}{1+x^{-4}}+\frac{1}{1+y^{-4}}$	2	$\mathcal{E}(-5,5)$	26	$\mathcal{E}(-5,5)$	10, 201
vladislav-leva-04	$\frac{10}{5+(x-3)^2+(y-3)^2+(z-3)^2+(v-3)^2+(w-3)^2}$	5	$\mathcal{U}(0.05,6.06)$	1, 024	$\mathcal{U}(-0.25,6.35)$	5000
vladislav-leva-06	$6\sin(x)\cos(y)$	2	$\mathcal{U}(0.1,5.9)$	30	$\mathcal{E}(-0.05,6.05)$	93, 636

the specific fitness values obtained, see the supplementary material). This is reported both for the training and validation sets. It should be noted that GP-DMD was in the group of best-performing algorithms in more instances than any other approach, and probably more importantly, when considering the validation set, it remained in this group in 23 of the 25 instances. Thus, not only is the accuracy of GP-DMD competitive, but it is attained in quite a robust way.

Figure 1 shows boxplots of the fitness in the validation set for 30 independent executions in three selected instances [6]. Note that GP-DMD achieved a very low

[6] These instances belong to different benchmark sets, so they are quite different, and their small sample sizes allow fast runs, meaning that these plots can be easily used in the future for comparison purposes.

Table 4 Number of instances for each method where there was no statistical difference with respect to the best (lowest) median for fitness in training and validation sets and model size

	GP-DMD	KNOBELTY	GMD-GP	ALPS	ϵ-LEX	FOCUS	SGP	DPSBR	SIS	AFPO
Training	21	15	14	11	11	5	4	3	3	2
Validation	23	11	10	20	8	8	9	8	8	8
Size	17	2	3	14	1	10	2	7	2	1

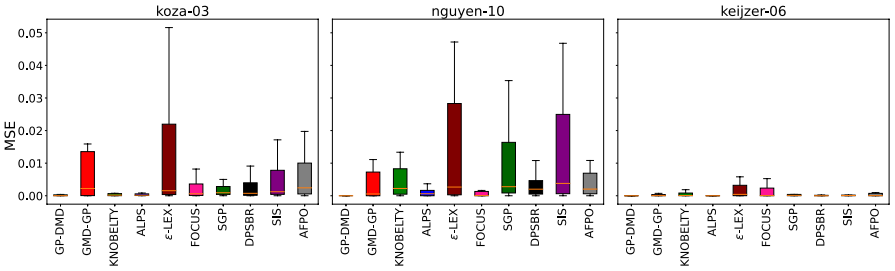

Fig. 1 Boxplots of the fitness attained in 30 independent executions for the validation set in three instances

median and the lowest variability. While other methods eventually yield good solutions, GP-DMD consistently generates good models, so the way to explore the search space provided by GP-DMD, with a more explicit management of the diversity, results in a much more robust behavior.

4.3 Solution size

Generating easily interpretable models is important in machine learning tasks [44, 45]. One key aspect related to interpretability is the size of the mathematical models (trees in this case). However, a typical drawback in the design of GP algorithms, is the phenomenon known as bloat [46], which is the uncontrolled growth of code without a significant performance improvement [30]. Avoiding this uncontrolled growth is important for several reasons. First, concise final trees provide advantages in terms of reduced computational complexity, increased generalization and easier examination of the structure. Second, in the presence of bloat, the structural diversity measures become an unreal measure of the difference between individuals, so the diversity management schemes might not behave as expected [17]. Thus, diversity management strategies, and especially those based on structural diversity, should take size into account. For the above reasons, this section is devoted to analyzing the implications of the different tested algorithms on the sizes of the trees.

Table 4 presents a summary of the results of the statistical tests when applied to compare tree sizes (number of nodes). It shows the number of instances where the given algorithm was either the best algorithm (lowest median size) or it did not exhibit a statistically significant difference with the algorithm that reported the

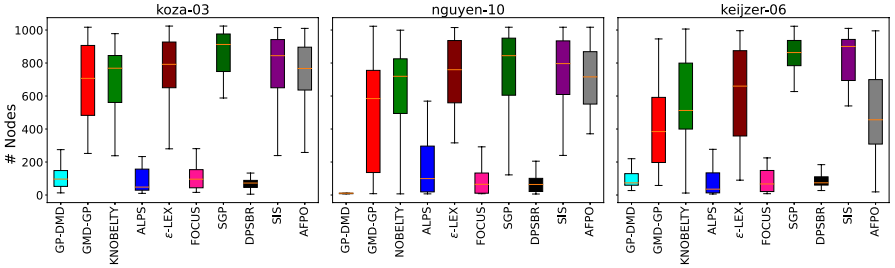

Fig. 2 Boxplots of the sizes of the models generated in the 30 independent executions

lowest median size. The clear winners in terms of size are ALPS and GP-DMD, with a clear advantage in favor of GP-DMD. In fact, only KNOBELTY, ALPS and FOCUS were able to attain lower trees (statistically significant) than GP-DMD in a few number instances. Figure 2 shows boxplots of the solution sizes for 30 independent executions in three selected instances. It shows a clear separation among ALPS, GP-DMD, DPSBR, and FOCUS and the remaining methods.

These results show that the proposed method has a positive and significant impact on the size of the solutions, which is usually considered beneficial for avoiding overfitting and improving generalization. Note that this benefit is confirmed with the performance analyses in the validation set.

4.4 Population dynamics

In order to shed some light on the reasons behind the good performance of our proposed method, a discussion of some of the properties related to the dynamics of the population is presented. To this end, above is a set of graphs with the trend, through the optimization process, of several measurements for three test cases. The behavior observed in the remaining 22 cases is quite similar.

Figure 3 shows the median among executions of the fitness for the validation set. Note that GP-DMD yields improvements in a continuous and gradual manner, which indicates that the algorithm does not suffer from premature convergence. This

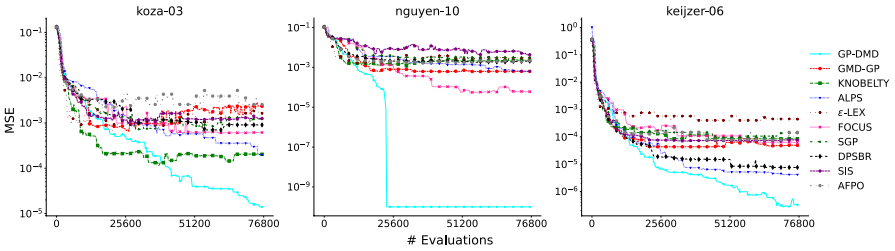

Fig. 3 Trend of the median for 30 independent executions of the fitness in the validation set (logarithmic scale)

contrasts with the remaining methods, where the improvements are not very significant for a large period of the execution.

Figure 4 depicts the median among executions of the mean size (number of nodes) of the solutions in the population. It is clear that ALPS, GP-DMD, DPSBR, and FOCUS exhibit a slow but steady increase in the sizes, whereas for the remaining methods, there is initially a fast increase in the population tree size that never shrinks, meaning they are searching much more complex models. Note also that those schemes that maintain lower sizes are faster, meaning larger populations or executions could be used to potentially improve the quality of solutions further.

Figure 5 shows the median among executions of the mean $ed2_{norm}$ distance to the closest individual in the population (DC). As expected, in the case of GP-DMD there is quite a linear decrease in this measurement. It is also important to point out that the diversity resulting from the population initialization method is not very high, and GP-DMD is capable of promoting the diversity to the desired value despite the initial diversity value. This behavior is quite different to the ones exhibited in the remaining methods, where in most cases, a relatively constant amount of diversity is maintained for a large period of the execution. This means that GP-DMD is in fact moving from exploration to exploitation, which is the main aim behind the design principle studied in this paper.

As previously stated, the behavioral diversity is also important. In order to analyze the trend in the behavioral diversity, Fig. 6 shows the median among executions of the median pair-wise fitness differences appearing in the population. A large value indicates that solutions with different kinds of behaviors are present in the population. Differently, a low value showcases a convergence in this property. Note

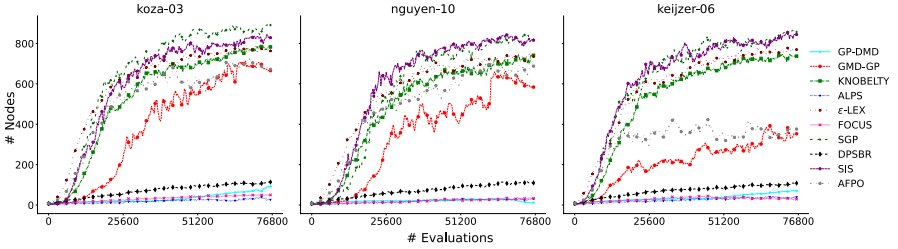

Fig. 4 Trend of the median for 30 independent executions of the mean population's tree sizes

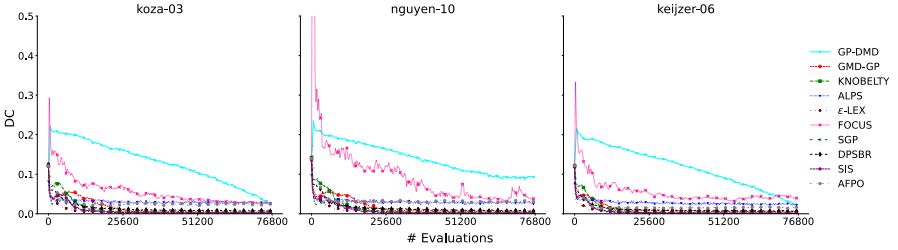

Fig. 5 Trend of the median for 30 independent executions of the DC in the population

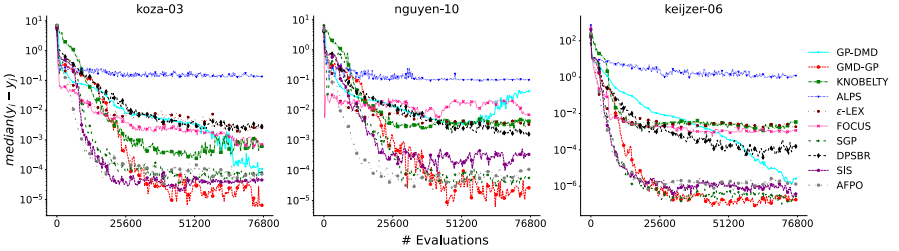

Fig. 6 Trend of the median for 30 independent executions of the median pair-wise fitness distances in the validation set (logarithmic scale)

that this does not necessarily imply a convergence in terms of the tree structures; however, presenting a too homogeneous behavior might be an indication that several of the trees are in a neutral network, which in the case of GP might be due to the appearance of introns. While GP-DMD does not explicitly control the behavioral diversity, it attains a progressive decrease in this metric. This is particularly clear in the keijzer-06 and koza-03 cases. In contrast, the other approaches tend to maintain a relatively fixed amount of this kind of diversity after a relatively small period of execution. The gradual reduction presented by GP-DMD indicates that the explicit management of structural diversity indirectly manages the behavioral diversity in a proper way in the case of GP-DMD. The gradual decrease of the median pair-wise fitness difference exhibited by GP-DMD in combination with the controlled diversity in DC, seems to indicate that the proposed algorithm is successful in promoting and managing both behavioral and structural diversities.

These trends show that the search features are quite different in the proposal put forth in this paper than in other methods. These trends are quite similar to the ones observed in the field of combinatorial optimization, meaning that these interesting features could be translated to the realm of GP and that in this case, it also provides significant benefits to the results. Specifically, there are important benefits in terms of robustness, meaning that the standard deviation of the MSE and tree size are quite low, and there were no problems with a significant degradation in performance, which is also one of the advantages found in complex combinatorial optimization problems.

4.5 Analysis of execution time

The previous experiments involved 7500 independent executions that were performed in the High Performance Computing (HPC) cluster "*Laboratorio de Supercómputo del Bajío*". Servers with two *Intel Xeon E52620 v2* processors with 6 cores at 2.10 GHz and 32 GB of DDR3 RAM were used. The stopping criterion was set by evaluations, as previously described. However, there were important variations in terms of execution times.

The analyses of the runs show that in some cases, most of the time is invested in evaluating the solutions and the time associated to the diversity management strategy is not meaningful. However, in other cases, the overhead associated to the diversity management strategy is more significant. Note that, at each generation

the evaluation stage takes $O(vmn)$, where v is the mean size of the solutions, m is the number of test cases and n is the population size. From the tested algorithms, GP-DMD presents the highest worst-case overhead complexity due to the diversity management strategy. In particular, the current implementation of the replacement operator takes $O(vn^2 + n^3)$. This is a relatively high complexity and it might be an issue for very large populations. However, for typical population sizes, such as the value 200 used in this paper, this overhead is not too significant.

The properties of the instances used in this paper are very diverse. In particular, there are instances with very short m values, such as keijzer-06, and instances with very large m, such as vladislavleva-04. Figure 7 shows the median of the time of the 30 independent executions for each algorithm. Particularly, the relation between time and number of evaluations is shown. In the vladislavleva-04, GP-DMD is the fastest algorithm. Since in this case m is much larger than n, the most computationally expensive part is the evaluation of the solutions, with complexity $O(vmn)$. Thus, in this case, the cost is proportional to the size of the trees and in fact, the relation between the time invested in the executions and the sizes analyzed in Table 4 is clear. In the case of the instances with lower m, such as keijzer-06, the overhead associated to the diversity management strategy is more important and as a result, GP-DMD is slower than some other methods, such as ALPS and FOCUS. The reason is that, while the tree maintained by GP-DMD are slightly smaller, the total cost is dominated by the $O(n^3)$ part associated to the diversity management strategy. However, in comparison to the rest of the methods, which maintain much larger trees, GP-DMD is faster. Finally, it is also important to mention that in both instances, GP-DMD is slower than several other methods in the initial phases. In these initial exploration stages, the trees in all the algorithms are small, so the time associated to the diversity management strategy dominates. However, as the evolution progresses and GP-DMD is able to maintain short trees, the advantages described above appear. In summary, it is important to mention that significant drawbacks in terms of performance might appear when considering very large populations and, less significantly, with very low training samples. Thus, for such cases, some modifications of GP-DMD might be required.

4.6 Analysis of GP-DMD configurations

This last experiment is devoted to analyze some variations of GP-DMD with the aim of better identifying the reasons behind its significant advantages. Since this research is focused on exploring the advantages of the dynamic diversity management, several dynamic and static schemes are tested. Specifically, 11 variants of GP-DMD with different features, including the baseline proposal used in previous experiments that considers a linear decrement with $d_i = 1/8$, are analyzed. Three variants test different initial d_i. Particularly, the values $\frac{1}{8}$, $\frac{1}{4}$ and $\frac{1}{2}$ are considered. Four variants use a static threshold value (τ) in the penalization approach. Thus, the `minimum-required-distance` just returns a fixed value. The tested τ values were 0, $\frac{1}{8}$, $\frac{1}{4}$ and $\frac{1}{2}$. Note that using $\tau = 0$ means that only the clones are penalized. Additionally, two schemes that consider a dynamic threshold schedule different to

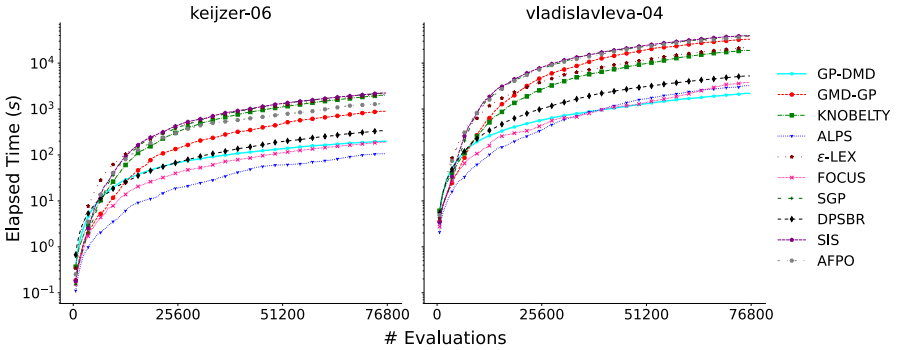

Fig. 7 Trend of the median for 30 independent executions of the elapsed time with respect to the number of evaluations (logarithmic scale)

the linear one are included. Particularly, the schedules follow a bezier curve [47] with control points (0, 0) (bezier-b) and $(e_s, 0.5)$ (bezier-t), respectively. Note that all the previous proposals consider the normalized *ed*2 distance. Finally, two additional distances are used. They are the normalized *ed*1 and the normalized *hamming* distances; the *ed*1 is a structural distance, while the *hamming* distance represents a drastic modification because it is a semantic distance. Note that in these two last cases, the linear dynamic schedule for the penalization approach used in our previous experiments is maintained.

Figure 8 shows, for all these variants, the trend of the median of several features through the execution considering 30 independent runs in the koza-03 instance. The behavior with other instances is quite similar. The subplot (A) refers to the fitness of the best solution, considering the validation set. The subplot (B) shows the mean DC distance of the population. Finally, the mean tree size is shown in subplot (C). Probably, the most obvious insight is that when using very low threshold values in the penalization, bloat appears. This is particularly clear when using $\tau = 0$, but it is also obvious in the schedule beizer-b, which maintains a low threshold for a long period. Our hypothesis is that this increase is due to the appearance of introns but more analysis is required to fully understand the reasons. However, it is clear that applying not too low thresholds is good for maintaining small trees. It is also remarkable that the DC distances follow quite precisely the schedule applied. The only exception appears when applying the Hamming distance, meaning that applying penalties in terms of semantic distances do not allow to maintain appropriate structural diversity, resulting in not so proper results.

In order to summarize the results attained with all the instances, Table 5 presents the results of the statistical tests for the size and fitness in the validation and training sets. As in previous experiments, for each variant it shows the number of instances where the corresponding model attained the lowest median among all the variants or did not exhibit a statistically significant difference with the algorithm that reported the lowest median. In terms of fitness, it is clear that using a linear dynamic threshold is superior than using a static value. In fact, for all the tested values ($\frac{1}{8}$, $\frac{1}{4}$ and $\frac{1}{2}$) the schemes considering a linear decrease attained a superior performance

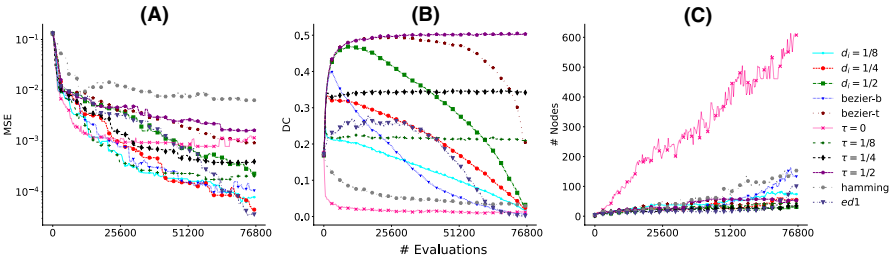

Fig. 8 Trend of the median for 30 independent executions of different GP-DMD configurations in the instance koza-03 for, (**A**) best solution fitness in validation set (logarithmic scale), (**B**) population diversity and (**C**) population tree size

than the ones using a fixed value. These results confirm that, as in combinatorial optimization, promoting a gradual shift from exploration to intensification provides important benefits. Regarding the proposals with a linear decrease, it is also clear that the initial threshold value significantly impacts its effectiveness. Using lower values provokes an increase on the number of instances with adequate fitness, but this is at the cost of increasing the size of the resulting trees. Thus, different parameterization might be used depending on the desired performance and interpretability. It is also worth noting that the comparisons among the results of these three d_i values with linear decrement, and the remaining diversity management strategies analyzed in previous experiments, reveal the superiority of the proposals with linear decrement. Thus, while the value $\frac{1}{8}$ was used to report the experimental results in the previous sections, any of the tested values allow reaching similar conclusions (for a summary of the statistical tests considering all the algorithms, see the supplementary material). Finally, using the $ed1$ distance did not result in significant changes in the results, so the proposal is robust against minor modifications in the distance-like functions. However, the Hamming distance resulted in poor results. As previously discussed, the application of this distance did not ensure the maintenance of a proper structural diversity, so in order to properly apply behavioral diversity distance-like functions in our proposal, additional research is required.

Table 5 Number of instances (for each variant) where there was no statistical difference with respect to the best (lowest) median for fitness in training and validation sets and model size

	$d_i = 1/8$	$d_i = 1/4$	$d_i = 1/2$	$\tau = 0$	$\tau = 1/8$	$\tau = 1/4$	$\tau = 1/2$	Hamming	$ed1$	bezier-b	bezier-t
Validation	23	20	11	9	18	10	6	6	15	22	6
Training	19	12	9	7	9	6	2	1	12	25	3
Size	8	15	19	1	10	19	17	3	11	4	19

5 Conclusions and future work

The proper balance between exploration and exploitation is one of the keys to designing effective Evolutionary Algorithms. A design principle proposed and studied by our research group, which is based on relating the amount of diversity maintained in the population to the elapsed period of execution and stopping criterion, has yielded significant benefits in the field of combinatorial optimization. This work studies the application of this design principle in the area of GP; specifically, this paper presents a novel GP variant, called GP-DMD, which manages diversity explicitly through a novel replacement strategy that incorporates the previously mentioned design principle and at the same time favors fit and simple solutions by applying multi-objective concepts.

The novel proposal was compared to a diverse set of well-established algorithms, including several diversity-aware techniques. Specifically, the validation is carried out using the symbolic regression task. The experimental validation shows that the approach presented allows for the generation of high-quality solutions with impressively small sizes, significantly improving the algorithm's robustness. The dynamics of the population in terms of size, fitness and diversity shows the remarkable differences in GP-DMD when compared to other related algorithms. Additionally, extensive experiments show that the method is quite robust, in the sense that small variants of the proposal also behave properly. Particularly, proposals that follow different kinds of schedules to penalize individuals and distance-like functions behave similarly. Moreover, the findings are somewhat similar to those achieved in the field of combinatorial optimization, meaning that this design principle proposed in that area could be successfully transferred to GP.

One of the weaknesses of our proposal is that it requires setting a stopping criterion based on time or evaluations in order to promote the gradual shift from exploration to exploitation. In practical terms, using a stopping criterion related to quality is beneficial in some cases, so in future work, we plan to analyze the application of the ideas explored in this paper but in a way that is not incompatible with setting a stopping criterion based on quality. Also note that, while GP-DMD is quite robust for different parameterizations, using more advanced parameter control techniques to adapt some of its internal components might bring additional benefits. Finally, we would also like to apply GP-DMD to other kinds of applications and perform some extensions to consider additional interpretability metrics that might be used in combination with the minimization of the sizes of the trees.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10710-021-09426-4.

Acknowledgements Authors acknowledge the financial support from CONACyT through the "Ciencia Básica" project no. 285599 and the support from "Laboratorio de Supercómputo del Bajio" through the project 300832 from CONACyT.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. J.R. Koza, J.R. Koza, Genetic programming: on the programming of computers by means of natural selection, volume 1. MIT press, (1992)
2. R. Poli, J. Koza, Genetic programming, in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. ed. by E.K. Burke, G. Kendall (Springer, Boston, 2014), pp. 143–185
3. K. Nag, N.R. Pal, Genetic programming for classification and feature selection. in *Evolutionary and swarm intelligence algorithms*, Springer International Publishing, Cham, (2019), pp. 119–141
4. K. Krawiec, *Behavioral Program Synthesis with Genetic Programming*, vol. 618 (Springer, 2016)
5. I. Azaria, A. Elyasaf, M. Sipper, Evolving artificial general intelligence for video game controllers, in *Genetic Programming Theory and Practice XIV*. ed. by R. Riolo, B. Worzel, B. Goldman, B. Tozier (Springer International Publishing, Cham, 2018), pp. 53–63
6. C. Segura, A.H. Aguirre, S.I.V. Pena, S.B. Rionda, The importance of proper diversity management in evolutionary algorithms for combinatorial optimization, in *NEO 2015: Results of the Numerical and Evolutionary Optimization Workshop NEO 2015 held at September 23–25 2015 in Tijuana Mexico*. ed. by O. Schutze, L. Trujillo, P. Legrand, Y. Maldonado (Springer International Publishing, Cham, 2017), pp. 121–148
7. M. Crepinsek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput. Surv. **45**(3), 1–33 (2013)
8. C. Segura, A. Hernández-Aguirre, F. Luna, E. Alba, Improving diversity in evolutionary algorithms: new best solutions for frequency assignment. IEEE Trans. Evol. Comput. **21**(4), 539–553 (2017)
9. C. Segura, S.B. Rionda, A.H. Aguirre, S.I.V. Pena, A novel diversity-based evolutionary algorithm for the traveling salesman problem. in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15*, Association for Computing Machinery, New York, NY, USA, (2015). pp. 489–496
10. J.C. Castillo, C. Segura, Differential evolution with enhanced diversity maintenance. Opt. Lett. **14**(6), 1471–1490 (2020)
11. J. Chacon, C. Segura, Analysis and enhancement of simulated binary crossover. in *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, (2018)
12. E. Burke, S. Gustafson, G. Kendall, N. Krasnogor, Advanced population diversity measures in genetic programming, in *Parallel Problem Solving from Nature: PPSN VII*. ed. by J.J.M. Guervos, P. Adamidis, H.-G. Beyer, H.-P. Schwefel, J.-L. Fernandez-Villacan (Springer, Berlin, Heidelberg, 2002), pp. 341–350
13. G. Squillero, A. Tonda, Divergence of character and premature convergence: a survey of methodologies for promoting diversity in evolutionary optimization. Inf. Sci. **329**, 782–799 (2016). (**Special issue on Discovery ScienceSpecial issue on Discovery Science**)
14. N.T. Hien, N.X. Hoai, A brief overview of population diversity measures in genetic programming. in *Proc. 3rd Asian-Pacific Workshop on Genetic Programming*, Hanoi, Vietnam, pp. 128–139. Citeseer, (2006)
15. A.R. Burks, W.F. Punch, An efficient structural diversity technique for genetic programming. in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15*, New York, NY, USA, (2015). Association for Computing Machinery, page 991–998
16. 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), 47–62 (2004)
17. U.-M. OReilly, J. Kelly, E. Hemberg, Improving genetic programming with novel exploration: exploitation control. Eur. Conf. Genet. Progr. **11451**, 64–80 (2019)
18. M. Keijzer, *Efficiently Representing Populations in Genetic Programming* (MIT Press, Cambridge, 1996), pp. 259–278

19. M. Gaudesi, G. Squillero, A. Tonda, Universal information distance for genetic programming. in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO Comp '14*, New York, NY, USA, 2014. Association for Computing Machinery, pp. 137–138
20. L. Vanneschi, M. Castelli, S. Silva, A survey of semantic methods in genetic programming. Genet. Progr. Evol. Mach. **15**(2), 195–214 (2014)
21. G. Folino, C. Pizzuti, G. Spezzano, L. Vanneschi, M. Tomassin, Diversity analysis in cellular and multipopulation genetic programming. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, (2003) vol. 1, pp. 305–311
22. N.Q. Uy, N.X. Hoai, M. ONeill, R.I. McKay, E. Galvan-Lopez, Semantically-based crossover in genetic programming: application to real-valued symbolic regression. Genet. Progr. Evol. Mach. **12**(2), 91–119 (2011)
23. Q.U. Nguyen, X.H. Nguyen, M. Oneill, A. Agapitos, An investigation of fitness sharing with semantic and syntactic distance metrics. in *European Conference on Genetic Programming*, Springer, (2012), Vol. 7244, pp. 109–120
24. J.P. Rosca, Entropy-driven adaptive representation. in *Proceedings of the workshop on genetic programming: From theory to real-world applications*, vol. 9, Tahoe City, California, USA, (1995). Citeseer
25. W. La Cava, L. Spector, and K. Danai, Epsilon-lexicase selection for regression. in *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO '16*, New York, NY, USA, (2016). Association for Computing Machinery, pp. 741–748
26. B. Metevier, A.K. Saini, L. Spector, *Lexicase Selection Beyond Genetic Programming* (Springer International Publishing, Cham, 2019), pp. 123–136
27. E. Galvan-Lopez, 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
28. P. Day, A.K. Nandi, Binary string fitness characterization and comparative partner selection in genetic programming. IEEE Trans. Evol. Comput. **12**(6), 724–735 (2008)
29. A.K. Nandi, M.W. Aslam, Z. Zhu, Diverse partner selection with brood recombination in genetic programming. Appl. Soft Comput. **67**, 558–566 (2018)
30. E.D. de Jong, R.A. Watson, J.B. Pollack, Reducing bloat and promoting diversity using multi-objective methods. in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, GECCO'01*, San Francisco, CA, USA, (2001). Morgan Kaufmann Publishers Inc, pp. 11–18
31. M. Schmidt, H. Lipson, *Age-Fitness Pareto Optimization* (Springer, New York, 2011), pp. 129–146
32. J. Grefenstette, Genetic algorithms for changing environments. in *Parallel Problem Solving from Nature 2*, Elsevier, (1992), pp. 137–144
33. A.R. Burks, W.F. Punch, An analysis of the genetic marker diversity algorithm for genetic programming. Genet. Progr. Evol. Mach. **18**(2), 213–245 (2017)
34. N.F. McPhee, N.J. Hopper, Analysis of genetic diversity through population history. in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 2, GECCO'99*, San Francisco, CA, USA, (1999). Morgan Kaufmann Publishers Inc, pp. 1112–1120
35. B. Cao, Z. Jiang, Increasing diversity and controlling bloat in linear genetic programming. in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, (2016), pp. 414–419
36. G.S. Hornby, *A Steady-State Version of the Age-Layered Population Structure EA* (Springer, US, Boston, MA, 2010), pp. 87–102
37. A.R. Burks, W.F. Punch, An investigation of hybrid structural and behavioral diversity methods in genetic programming, in *Genetic Programming Theory and Practice XIV*. ed. by R. Riolo, B. Worzel, B. Goldman, B. Tozier (Springer International Publishing, Cham, 2018), pp. 19–34
38. E.R. Ruiz, C. Segura, Memetic algorithm with Hungarian matching based crossover and diversity preservation. Comput. Sist. **22**(2), 07 (2018)
39. D.R. White, J. Mdermott, M. Castelli, L. Manzoni, B.W. Goldman, G. Kronberger, W. Jaskowski, U.-M. OReilly, S. Luke, Better GP benchmarks: community survey results and proposals. Genet. Progr. Evol. Mach. **14**(1), 3–29 (2013)
40. P. Orzechowski, W La Cava, J.H. Moore, Where are we now? a large benchmark study of recent symbolic regression methods. In *Proceedings of the Genetic and Evolutionary Computation*

Conference, GECCO '18, New York, NY, USA, (2018). Association for Computing Machinery, pp. 1183–1190
41. J. Zegklitz and P. Posik, Benchmarking state-of-the-art symbolic regression alrogirhms. Genet. Progr. Evol. Mach., pp. 1–29, (2020)
42. I. Arnaldo, K. Krawiec, U-M OReilly, Multiple regression genetic programming. in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO '14*, New York, NY, USA, (2014). Association for Computing Machinery, pp. 879–886
43. J. McDermott, D.R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaskowski, K. Krawiec, R. Harper, K. De Jong, and U-M OReilly, Genetic programming needs better benchmarks. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, GECCO '12*, New York, NY, USA, (2012). ACM, pp. 791–798
44. M. Virgolin, A. De Lorenzo, E. Medvet, F. Randone, Learning a formula of interpretability to learn interpretable formulas, in *Parallel Problem Solving from Nature: PPSN XVI*. ed. by T. Back, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, H. Trautmann (Springer International Publishing, Cham, 2020), pp. 79–93
45. M. Virgolin, A. De Lorenzo, F. Randone, E. Medvet, M. Wahde, Model learning with personalized interpretability estimation (ml-pie). In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21*, New York, NY, USA, (2021). Association for Computing Machinery, pp. 1355–1364
46. E.D. De Jong, J.B. Pollack, Multi-objective methods for tree size control. Genet. Progr. Evol. Mach. **4**(3), 211–233 (2003)
47. E.P. Bezier, S. Sioussiou, Semi-automatic system for defining free-form curves and surfaces. Comput.-Aid. Des. **15**(2), 65–72 (1983)