Multi-modal multi-objective model-based genetic programming to find multiple diverse high-quality models

E.M.C. Sijben
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands
evi.sijben@cwi.nl

T. Alderliesten
Leiden University Medical Center
Leiden, the Netherlands
t.alderliesten@lumc.nl

P.A.N. Bosman
Centrum Wiskunde & Informatica
Amsterdam, the Netherlands
peter.bosman@cwi.nl

ABSTRACT

Explainable artificial intelligence (XAI) is an important and rapidly expanding research topic. The goal of XAI is to gain trust in a machine learning (ML) model through clear insights into how the model arrives at its predictions. Genetic programming (GP) is often cited as being uniquely well-suited to contribute to XAI because of its capacity to learn (small) symbolic models that have the potential to be interpreted. Nevertheless, like many ML algorithms, GP typically results in a single best model. However, in practice, the best model in terms of training error may well not be the most suitable one as judged by a domain expert for various reasons, including overfitting, multiple different models existing that have similar accuracy, and unwanted errors on particular data points due to typical accuracy measures like mean squared error. Hence, to increase chances that domain experts deem a resulting model plausible, it becomes important to be able to explicitly search for multiple, diverse, high-quality models that trade-off different meanings of accuracy. In this paper, we achieve exactly this with a novel multi-modal multi-tree multi-objective GP approach that extends a modern model-based GP algorithm known as GP-GOMEA that is already effective at searching for small expressions.

CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**.

KEYWORDS

Genetic programming, multi-modal, multi-objective, multi-tree

ACM Reference Format:
E.M.C. Sijben, T. Alderliesten, and P.A.N. Bosman . 2022. Multi-modal multi-objective model-based genetic programming to find multiple diverse high-quality models. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528850

1 INTRODUCTION

State-of-the-art machine learning models are often difficult to interpret. This can be caused by models having many coefficients, many variables, and/or an intricate structure. The popular field of eXplainable Artificial Intelligence (XAI) aims to either develop methods that enable humans to interpret the complex machine learning model and its predictions, or to make inherently interpretable models. Models are unlikely to be inherently interpretable if they have many coefficients and/or variables. Therefore, it makes sense to restrict the size of the model. However, doing so may negatively affect model performance. Ideally, one would obtain both high performance and interpretability.

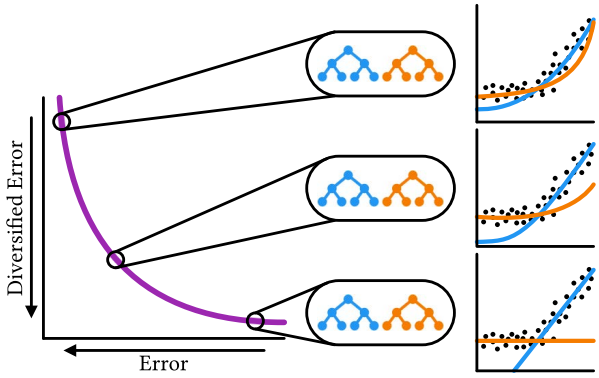


Figure 1: Visualization of our approach.

Genetic Programming (GP) is a learning algorithm that can generate expressions that are flexible in their structure, and in the operators or functions used in these expressions. This enables GP to capture non-linear relations in a compact expression, offering a useful trade-off between performance and size compared to other methods [16]. This quality is why GP has been suggested to be useful for XAI [9, 10, 23].

XAI approaches aim to give users insight into a model. This allows users to make more informed use of the model in the real world [4, 19]. However, if a model has (serious) shortcomings or other problems, although now the user may be informed about these potential problems, insight alone does not necessarily provide a solution, nor a direct way to change the model. Of course, users can always choose not to use the model. Ideally, however, users would be able to choose a model from a list of models that exhibit different qualities, to meet their specific preferences and prevent model rejection. Unfortunately, this is typically not possible because most machine learning algorithms generate only one model.

In practice, due to the amount of data available (both in terms of records and variables) and/or the complexity of the prediction task, combined with a finite capacity of learnable models, it is unlikely that the optimal model (in terms of minimum training or validation error) is unique. Even if it is, it might not be the best model according to the user. The user may have certain model requirements, or there could be conflicts between the model and expert knowledge about the domain. Additionally, it might not be

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07.
https://doi.org/10.1145/3512290.3528850

the best model simply because the data is limited and models with slightly worse performance in error on the data at hand are better if we had infinite data to train on.

Therefore, it would be very useful in practice, with the vision of a domain-expert end-user selecting the preferred model for the task at hand, to be able to search for multiple models that are all *well-performing* but are also *diverse* and potentially even describe different parts of the data better. Allowing the user to inspect such a set of different models may provide unique novel insights into the data and the process underlying the data, and support choosing a good model with additional expert knowledge in a powerful and sensible way. Thereby user agency and control are increased.

In this paper, we propose exactly this: searching, in a novel way, for *sets of multiple models* instead of a single model. Rather than using (adaptive) niching or fitness sharing, we use a multi-tree GP model. This enables us to explicitly define diversity between models and perform multi-modal search in a potentially highly multi-modal search space by finding a fixed number of modes/niches. Moreover, by defining the search in a Multi-Objective (MO) way, we can optimize for both diversity between models and model performance. Finally, by using particular notions of diversity, we can not only find models that are different, but even focus on different parts of the data in different ways, giving additional meaning and potentially practically useful dimensions to the models that will be presented to the user. We visualize our approach in Figure 1. To the best of our knowledge, this is the first paper to propose searching for diverse multi-tree models with an MO approach.

Besides searching for a set of models, the ultimate goal is to create interpretable models. While we do not directly focus on interpretability here, we do consider that smaller models are likely more interpretable than larger models. Under this assumption, it becomes interesting to look at GP approaches that are particularly well-suited to evolving small solutions. Results show that the GP variant of the Gene-pool Optimal Mixing Evolutionary Algorithm (GP-GOMEA) gives better results when searching for solutions of limited tree size than classic GP [23]. An MO version of GP-GOMEA does not yet exist, however. Therefore, we implement an MO variant of GP-GOMEA by leveraging the best practices previously employed in the design of MO-GOMEA [17].

The contribution of this paper is threefold. We 1) develop a novel approach for searching for sets of high-quality and diverse multi-tree GP models, 2) implement a version of GP-GOMEA with multi-trees and MO optimization, and 3) show the benefits of our approach by applying it to real-world data sets.

2 RELATED WORK

Diversity maintenance is a well-studied subject in GP that aims to improve diversity in the whole population to prevent premature convergence. In [11] semantic diversity is promoted by using the semantic crowding distance. The semantic crowding distance adopts the same principles as the crowding distance except it does not look at the distance in objective space but in semantic space. Not only is the regular crowding distance replaced by the semantic one, the semantic crowding distance is also added as an objective.

In [2], an extension of the age-fitness Pareto optimization [20], an MO method that optimizes age and fitness, is proposed. The idea behind this is to avoid that younger individuals, that have had less time to become fit, compete with older individuals, that have had more time to become fit. Thereby, the competition between individuals with a similar age is stimulated. A metric for tree structure similarity (genetic marker density) is used instead of age to prevent converging to a specific structure.

The FOCUS method [5] performs an MO search with three objectives: fitness, size, and average squared overlapping tree distance to other members in the population. Individuals are stimulated to move away from the central peak in the fitness landscape if the central peak becomes too crowded.

In [18] fitness sharing for GP is introduced. The general idea is to reward solutions that are different, i.e., the reward for each prediction is divided by the number of individuals in the population that give the same prediction. A distance function that reflects the structural dissimilarity of trees to extend the applicability of fitness sharing for tree-based methods is introduced in [8].

Our work distinguishes itself from the above-mentioned works because we do not primarily aim to avoid premature convergence, but we aim to present the user with a diverse set of potentially interesting models. Furthermore, the approach we present in this paper to realize this goal, is novel, for two reasons: 1) we maintain diversity within individuals rather than diversity across the population by using an MO search with multi-tree individuals, and 2) we incorporate a novel diversity objective in this approach that has particular relevance to machine learning.

3 MULTI-MODAL MULTI-OBJECTIVE MODEL-BASED GP

In this section, we present our approach to search for diverse high-quality models. It has four key components: GP-GOMEA, multi-tree individuals, MO optimization, and a particular diversity objective.

We use GP-GOMEA [23] because it is known to offer a good trade-off between performance and model size [16]. We implement multi-tree individuals, which allows us to express diversity within individuals. We choose to maintain diversity within individuals rather than maintaining diversity across the population because this allows us to more easily stimulate diversity and performance at the same time. To optimize for both diversity and performance we implement an MO variant of GP-GOMEA, where we leverage best practices of MO-GOMEA [17], and introduce a diversity objective function, which we optimize together with an error objective.

3.1 Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA)

GOMEA is a model-based Evolutionary Algorithm (EA) that has been shown to be effective in many domains such as discrete optimization [17, 21], real-valued optimization [1], and, most relevant here, GP [23]. GOMEA differs from classic EAs in that it uses a linkage model that is meant to capture the interdependencies within the genotype. This information is then used during variation to prevent building blocks (or partial solutions) from being disrupted and to effectively mix these blocks to create better solutions.

In GOMEA, a fixed-length string is used as the genotype so that genes at a certain location in the string always represent the same variables in the problem. Linkage information is represented as a

Family Of Subsets (FOS). The FOS contains subsets of genes (string indices) that are assumed to be linked.

If no linkage information is known a priori, it can be learned from the population during search. To this end, Mutual Information (MI) is often used to measure linkage among gene pairs and a so-called Linkage Tree (LT) is built to represent variable dependence relations in a hierarchical fashion. Computing joint MI for more than two genes is costly and requires large population sizes to be accurate. Therefore, the UPGMA algorithm [13] is used to approximate linkage for larger groups of genes. UPGMA is a hierarchical clustering algorithm that merges subsets, starting from singleton sets, which in GOMEA contain the individual genes. For the similarity measure in UPGMA, the pairwise average MI is used. At each merge step, the newly constructed cluster is added to the FOS, ultimately resulting in the LT. Merging is usually stopped in GOMEA when only two clusters are left. The subset containing all genes is typically not added to the FOS because using this subset would result in entire solutions being cloned during variation.

In GOMEA, every generation, each individual in the population undergoes variation through Gene-pool Optimal Mixing (GOM). GOM uses the information in the FOS to replace linked genes at the same time. Suppose individual $\mathcal{P}_i$ undergoes GOM. First, $\mathcal{P}_i$ is cloned into offspring O_i. Then, each of the subsets in the FOS is considered in a random order. For each FOS subset anew, a donor is randomly selected from the population. The values of the genes in O_i are replaced with those of the donor, but only at the positions specified by the FOS subset. If a replacement did not result in a worse fitness, the change is kept.

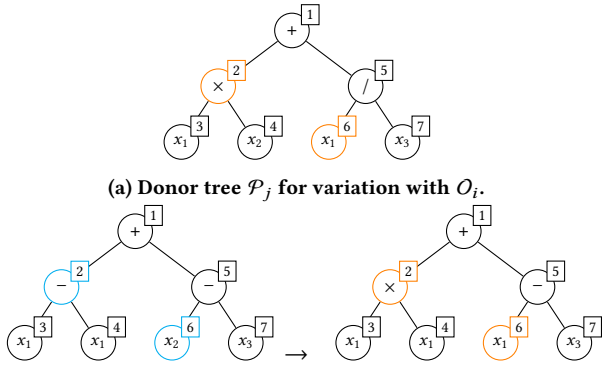
After processing all FOS elements, O_i is added to the offspring set and the next population member is considered. After processing the entire population, the population is replaced by the offspring.

3.2 GP-GOMEA & Multi-trees

GP-GOMEA is a GP variant of GOMEA [23]. In GP-GOMEA, individuals are trees that adhere to a template with fixed node positions. This enables mapping trees to strings in a fixed manner, i.e., nodes at the same position in different trees always map to the same string index. Consequently, learning an FOS and GOM variation can be straightforwardly used. However, a particular form of normalization is used on the estimated MI values to account for suprious linkage in the initial population that may occur in the case of GP trees because not every node in the GP tree is always used, nor can every node represent the same thing (leaves cannot represent functions). For more details, see [23].

Figure 2 shows an example of a variation step in GP-GOMEA, where we variate O_i using FOS subset member $\{2, 6\}$. Elements at indices 2 and 6 in selected individual O_i are replaced with those of donor tree $\mathcal{P}_j$, where j is randomly selected.

In our approach, we use a multi-tree representation. A multi-tree T consist of multiple trees t_i such that $T = (t_1, \ldots, t_n)$. A multi-tree implementation of GP-GOMEA did, however, not yet exist. To support multi-tree in GP-GOMEA, we concatenate the string representations of the n trees. Therefore, the indices count onward from one tree to another tree. For a multi-tree with 2 trees and a height of 3, this means that the first tree has a root node with index 1 and contains indices 1 through 7 as in Figure 2. The second tree has a root node with index 8 and contains indices 8 through 14.



(a) Donor tree $\mathcal{P}_j$ for variation with O_i.

(b) Recipient tree O_i before variation with $\mathcal{P}_j$.

(c) Recipient tree O_i after variation with $\mathcal{P}_j$.

Figure 2: GOM step for FOS element $\{2, 6\}$ applied to offspring O_i using donor $\mathcal{P}_j$.

Note that GOM always replaces a gene at index i with a gene in a donor solution at index i. GOM will thus not replace nodes from tree t_i of a multi-tree with nodes from t_j from a donor multi-tree where $i \neq j$. We do, however, want to be able to learn interdependencies *across* trees in a multi-tree. This will, for example, enable to learn that index 2 in the first tree is linked to index 11 in the second tree, which allows GOM to replace these nodes simultaneously. This is straightforwardly achieved because we concatenate the string representations of the n trees to achieve the string representation in GOMEA. Automatically, therefore, an FOS is learned that can represent interdependencies across trees.

However, we know, by design, that we are modelling multiple trees and that these trees separately will make a substantial contribution to the objective functions. For this reason, it is also of value to learn an FOS (an LT, to be precise) for each GP tree independently, which we propose to do. Moreover, because these GP trees are individual components of the larger multi-tree, we want to be able to exchange *entire trees* as well. For this, when we learn an LT for a single GP tree in the multi-tree representation, we do not discard the FOS element with all variable indices. The final FOS that is used for GOM, is the union of the FOS learned for the entire multi-tree and all of the FOSs learned for each GP tree separately.

3.3 MO-GP-GOMEA

Our goal is to search for trees that are both of high-quality and diverse. Using a multi-tree representation, we can explicitly express these notions over the multi-tree and optimize for them.

By properly governing that these expressions stay different throughout the search, we could design a multi-modal approach around this representation that searches for n local optima. However, as stated in the introduction, the expression with the smallest training error is not necessarily the preferred one, due to noise in the data, limited data, missing features, expert knowledge, and/or user needs. Hence, in a user-centered XAI setting, returning only the expression with the lowest training error, or even multiple expressions with the same optimal training error, is likely not satisfactory. It would be prudent to accept that our data is not perfect and explicitly also search for other properties of our expressions that represent potentially other things a domain expert could be interested in.

Therefore, we present an objective that stimulates searching for such potentially interesting properties in subsection 3.4.

Firstly, however, it is important to realize that finding multiple sets of expressions with different, conflicting properties, requires MO search. However, only a Single-Objective (SO) version of GP-GOMEA currently exists [23]. We therefore create, for the first time, an MO version of GP-GOMEA, which we base on the MO version of GOMEA originally introduced for binary variables [17].

MO-GP-GOMEA(population size N, clusters k)

1: **for** $i \in \{0, 1, \ldots, N-1\}$ **do**
2: $\mathcal{P}_i \leftarrow$ CreateRandomSolution()
3: EvaluateFitness($\mathcal{P}_i$)
4: $\mathcal{A} \leftarrow$ UpdateElitistArchive($\mathcal{P}_i$)
5: **while** ¬TerminationCriteriaSatisfied **do**
6: $\{C_0, C_1, \ldots, C_{k-1}\} \leftarrow$ ClusterPopulation($\mathcal{P}$)
7: **for** $j \in \{0, 1, \ldots, k-1\}$ **do**
8: $\mathcal{F}_j \leftarrow$ LearnLinkageModel(C_j)
9: **for** $i \in \{0, 1, \ldots, N-1\}$ **do**
10: $O_i \leftarrow$ Clone($\mathcal{P}_i$)
11: $C_j \leftarrow$ DetermineCluster(O_i, $\{C_0, C_1, \ldots, C_{k-1}\}$)
12: **if** IsExtremeCluster(C_j) **then**
13: $O_i \leftarrow$ SingleObjective-GOM(O_i, C_j,$\mathcal{F}_j$,$\mathcal{A}$)
14: **else**
15: $O_i \leftarrow$ MultiObjective-GOM(O_i, C_j,$\mathcal{F}_j$,$\mathcal{A}$)
16: $\mathcal{A} \leftarrow$ UpdateElitistArchive(O_i)
17: $\mathcal{P} \leftarrow O = \{O_0, O_1, \ldots, O_{N-1}\}$

Pseudo-code for MO GP-GOMEA is shown above, which at the top level, is essentially analogous to MO-GOMEA [17]. Every generation, the population is divided into k clusters (line 6). The clustering approach works on the basis of distance between solutions in normalized objective space, and is described in further detail in the supplementary material. For every cluster, a separate FOS is learned (line 8). Restricted mating is applied, meaning that donors for an individual must come from the same cluster as the individual itself. Then, 'extreme' clusters are identified (line 12): these are the clusters that, on average, perform best on a single objective. In extreme clusters, a single objective is optimized, that is, in GOM only one objective is considered when testing for improvements (line 13). A change is accepted if the offspring does not have worse fitness than the parent. In all other clusters, multi-objective optimization is performed, that is, in GOM both objectives are considered when testing for improvements (line 15). A change is accepted if the offspring either: 1) dominates the parent, 2) has equal objective values as the parent, 3) is not dominated by any member of the elitist archive and has different objective values than any member in the elitist archive, or 4) has the same objective values as a member in the elitist archive but is different in semantic space. When variation does not change the solution for one generation or does not improve the solution for $1 + \log_{10}(\mathrm{N})$ generations, we do an additional round of GOM called Forced Improvements (FI). This means that another round of variation is done with the donor being a random member from the elitist archive, or in the case of extreme clusters, the solution from the archive that has the best fitness for the corresponding single objective. The acceptance is the same as with regular variation, except, when the offspring has the same objective value(s) as the parent, the change is rejected. In addition, we stop iterating over the FOS once a change is accepted. If it has not changed during FI, we replace the offspring with a random member from the elitist archive, unless it originated from an extreme cluster, in which case we replace it with the solution from the archive that has the best fitness in terms of the corresponding single objective. After variation, the elitist archive is updated with the offspring (line 16).

3.4 Objectives

We optimize for two objectives, error and diversified error. For error, we use the commonplace Mean Squared Error (MSE). For diversified error, we specify an objective that fits well with the concept of GP-based XAI in practice as outlined in the introduction.

3.4.1 Error objective. The MSE is arguably the most commonly adopted objective in machine learning, especially for (symbolic) regression. In case of our multi-tree representation, to calculate the error, for each tree t_i of the multi-tree T, the MSE is calculated between the targets y associated with data points X_j and the predictions of the tree $t_i(X_j)$. We calculate the final error E of the multi-tree by summing the MSE of the individual trees, i.e.:

$$E = \sum_{i=1}^{n} MSE(t_i, X, y),$$

where

$$MSE(t_i, X, y) = \frac{1}{|X|} \sum_{j=1}^{|X|} (t_i(X_j) - y_j)^2,$$

n is the number of trees in a multi-tree, and $|X|$ is the number of records in the data set.

3.4.2 Diversified error objective. As all our experiments will be done with two trees, for now, we introduce the definition for the diversified error objective for a multi-tree with two trees. We come back to this in Section 7. We define the diversified error D as the mean of the minimum squared error of the individual trees, i.e.,:

$$D = \frac{1}{|X|} \sum_{j=1}^{|X|} \min((t_1(X_j) - y_j)^2, (t_2(X_j) - y_j)^2).$$

We choose this objective function for three reasons. Firstly, it enables finding expressions that *together* describe a data set well, i.e., expressions that describe different parts of the data set. Note that because we also optimize for error objective E, we will find expressions that not only predict the data set well together but are also optimized for their individual error. In addition, describing different parts of a data set with different models can even improve interpretability, because it resonates well with how humans intuitively understand things. Humans tend to categorize things [15]. In this way, they can dissect a problem and try to understand parts of it separately. Secondly, in combination with the error objective, we can find sets of expressions that have a similar error but have a different error distribution over the data points. Thirdly, an expert may well prefer a model that is very good in some cases, but obviously wrong in others, over a model that is moderately right and wrong in all cases. In practice, this may for instance mean fewer adjustments or further investigations: the cases in which the model

is very good do not need to be adjusted. Also, such an objective essentially finds an ensemble of complementary models that would have superior performance upon its use, when it is clear in practice for a particular case which model to follow. In summary, with this objective, we can find expressions that describe the data better as a set than a single expression would be able to, but at the same time, we also find sets containing expressions with a similar individual error that are different from each other.

Alternatively, one could define a diversity objective that maximizes the distance between the predictions of the trees within a multi-tree. However, this also stimulates expressions that do not perform well and are just very different. By contrast, using D, we do not stimulate diversity merely for the sake of diversity.

Finally, while D is proposed with the goal of offering interesting alternative models, D is a non-smooth non-convex function, which would be unsuitable for most typical machine learning approaches. It is a key strength of GP, being an evolutionary machine learning approach, that it can handle such functions.

4 EXAMPLES ON SYNTHETIC DATA

In this section, we present two examples of the use of our approach on synthetic data. We generate two simple regression data sets, with Gaussian Noise (GN) added. We perform a single run with our approach, using multi-tree individuals with $n = 2$. We multi-objectively optimize the error E and the diversified error D. The allowed symbols are the functions $+, -, \times, \div_p$, Ephemeral Random Constants (ERCs), and the input variables that appear in the synthetic data set. We use a maximum tree height of 3 (7 nodes), and perform 30 generations with a population size of 1000. We choose this population size because, on the one hand, these are quite simple problems, but on the other hand, we need to search in three 'directions'; the two extremes and the trade-off between those two.

4.1 Example 1: multi-modal data

We generate a multi-modal data set with one input variable X, and target variable Y using the following functions:

$$\begin{aligned} f(X) &= X^2 + GN(0, 10) \\ g(X) &= 2 \cdot X + GN(0, 10) \end{aligned}$$

We generate 100 data points with $Y = f(X)$ where X is drawn randomly from the interval $X[0, 10]$, and generate 40 data points with $Y = g(X)$ where X is drawn from the same interval.

We show the approximation front of the error and diversified error found by our approach on this data set in Figure 3. We also show the predicted Y of the two expressions of three multi-tree models along the front, against X, as well as the expressions themselves. As can be observed, our approach can effectively model multi-modal data sets. In particular, we see that the expressions of multi-tree C closely resemble $f(X)$ and $g(X)$, despite the uneven number of data points per function. The expressions of multi-tree A fall in between $f(X)$ and $g(X)$, because this minimizes the MSE. Multi-tree B is somewhere in between A and C.

4.2 Example 2: hidden variable

We generate a data set with hidden variable H, two input variables X_1 and X_2, and target variable Y, i.e., H itself is not in the data set.

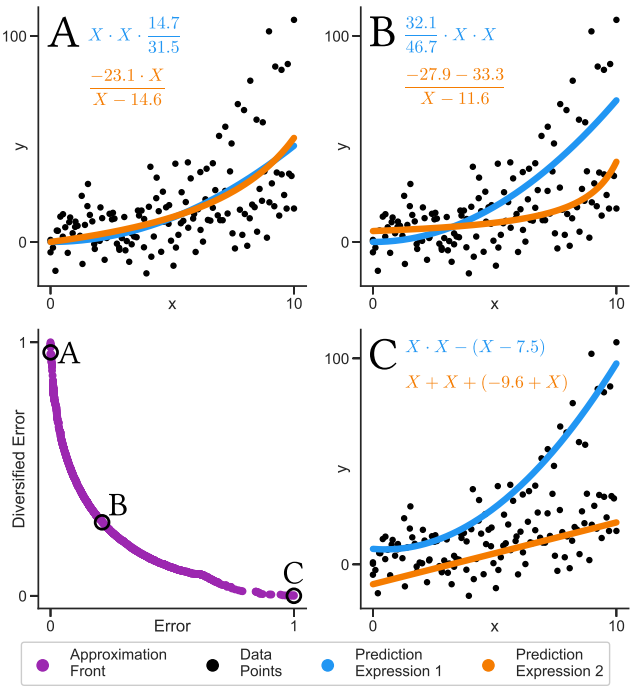


Figure 3: The approximation front of our approach on the synthetic multi-modal data set. The predictions of the expressions of multi-trees A, B and C are visualized.

Target variable Y is equal to hidden variable H. Input variables X_1 and X_2 are flawed 'estimations' of H. More specifically, the data set is generated using the following functions:

$$\begin{aligned} X_1 &= H + GN(0, 0.5) \\ X_2 &= H + GN(0, 0.5) \\ Y &= H \end{aligned}$$

We generate 100 data points by drawing H randomly from the interval $[0, 10]$. We show the approximation front of the error and diversified error found by our approach on this data set in Figure 4. Furthermore, we show the two expressions, and their individual MSE, of two multi-tree models along the front. The parameters are as described above, except here we do not use ERCs. By chance, due to the randomness of the Gaussian noise, X_1 has a slightly smaller error for predicting Y. Therefore, expressions that simply predict X_1 have a lower MSE than expressions that predict X_2. However, X_1 and X_2 are almost equally good. Our approach is able to retrieve multi-tree model B that represents this, with one expression using X_1 and the other expression using X_2.

The same principle applies to more elaborate data sets as well. Data sets may have multiple input variables or combinations of input variables that are good predictors of the target variable. Expressions using different combinations of the variables and functions might have a similar error, but if one of the functions is only slightly better, or if one of the solutions is easier to find when optimizing for MSE only, the focus will be on this solution. Our approach explicitly stimulates to explore different possibilities and is more likely to find multiple of these solutions.

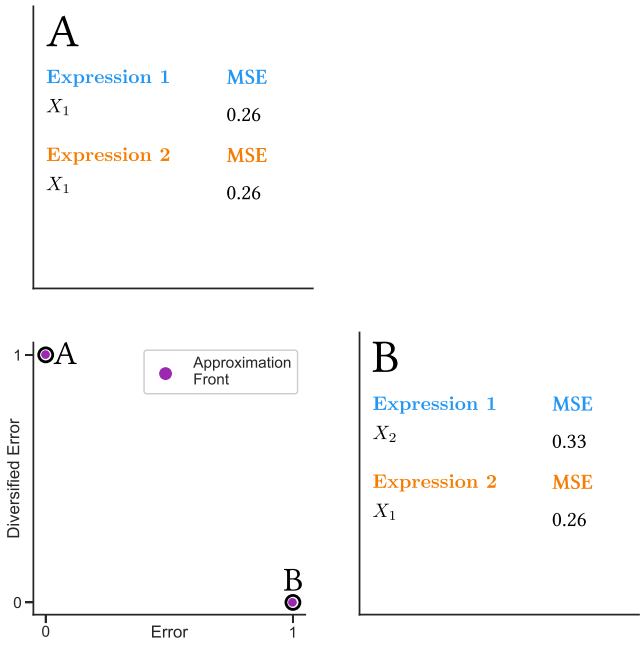


Figure 4: The approximation front of our approach on the synthetic hidden variable data set. The two expressions, and the individual MSE of multi-trees A and B are shown.

5 EVALUATION ON BENCHMARK DATA

In this section, we evaluate our approach on real-world data sets. We first describe our experimental setup before reporting our results.

5.1 Experimental setup

We evaluate three approaches: our multi-modal multi-tree MO-GP-GOMEA approach, multi-tree SO optimization with GP-GOMEA, and multi-tree NSGA-II [6]. We compare with SO optimization to evaluate whether our approach can find solutions at the extremes of the approximation front that are as good as when optimizing for each objective individually.

NSGA-II is a state-of-the-art MO EA, and perhaps the most popular MO EA in literature. Note that the GP version of NSGA-II does not use tree templates but has a variable tree length (with a maximum total size) like in traditional GP. We compare with NSGA-II to evaluate whether our approach can find approximation fronts as good as, or better than, a state-of-the-art MO GP algorithm.

We evaluate these approaches on three well-known data sets with a regression task, of which we show the properties in Table 1. The Boston housing data set contains information concerning housing in the area of Boston Mass [14], where the goal is to predict the median house value in an area. The concrete compressive strength data set contains information concerning the components and age of concrete [24], where the goal is to predict the strength of concrete. The yacht hydrodynamics data set contains information on the characteristics of yachts [12], where the goal is to predict the resistance of sailing yachts.

We initialize the population with multi-tree individuals with $n = 2$. We multi-objectively optimize the two objective functions described in subsection 3.4: E and D. Algorithm parameter and experiment settings are given in Table 2. Each run is repeated 30 times with a different random seed. We report the median over these runs and the statistical significance of the Wilcoxon signed-rank test (with $\alpha = 0.05$ and Bonferroni correction). For NSGA-II we use a crossover proportion of 0.5, a mutation proportion of 0.5, a tournament size of 4, and ramped half-and-half initialization.

Table 1: Properties of benchmark data sets.

Data set (abbreviation)	Features	Samples
Boston housing (b)	13	506
Concrete compressive strength (c)	9	1030
Yacht hydrodynamics (y)	6	308

Table 2: Algorithm parameter and experiment settings.

Setting	I	II	III
Functions	$+, -, \times, \div_p$		
Terminal	variables, ERC		
Train-test split	75%-25%		
For SO and Ours: tree height	3	4	5
For NSGA-II: maximum tree size	7	15	31
Time (s)	1200	7200	10800
CPU	AMD EPYC 7282		

Furthermore, we use a population size of 5000 for SO optimization, and a population size of 15000 in MO optimization. For our approach and NSGA-II, we use a population size that is 3 times larger than for SO, because MO needs to optimize three 'directions', namely the two extremes and the trade-off between those two. For our approach, we use 7 clusters in MO-GP-GOMEA.

For both objectives, we report the objective values of the extreme solutions found by the three approaches. We also report the Hyper-Volume (HV) of our approach and that of NSGA-II. HV is a measure that indicates the volume covered by the approximation front with respect to a reference point. We compute the HV by first combining all approximation fronts found by the different approaches over all runs, and then extracting the non-dominated solutions, i.e., we take the front of fronts. Then, we take the minimum and maximum values in each objective from this front, and we use them to normalize all approximation fronts. Finally, we get the HV by computing the surface area covered by the front with respect to reference point $[1.1, 1.1]$. Note that this can mean that HV values of 0 can be reported, indicating that the median run did not find any multi-tree solution near the best found solutions.

5.2 Results

We report the results of the experiment setting II in Tables 3 and 4. Table 3 shows that generally our approach has a significantly bigger HV than NSGA-II. Table 4 shows that the best performing solution of our approach generally has no significant difference in diversified error D compared with SO, and is significantly better than NSGA-II. Similar findings can be observed in Table 4 regarding error E. The supplementary material includes the results of experiment I and III. These results generally show the same pattern as in experiment II. For the error objective, however, our approach is significantly better than SO in experiment I, whereas SO is significantly better in many

Table 3: Median HV results for experiment setting II. A triangle symbol next to the reported median value indicates significant superiority (better (=bigger) HV) to the approach with the name in the same color as the triangle.

Data set	Split	Ours		NSGA-II
b	Train	0.61	▲	0.33
b	Test	0.00		0.00
c	Train	0.57	▲	0.00
c	Test	0.32	▲	0.00
y	Train	0.11	▲	0.00
y	Test	0.00	▲	0.00

Table 4: Median best diversified error D and median best error E for experiment setting II. A down-pointing triangle next to the reported median value indicates significant superiority (better (=smaller) objective value) to the approach with the name in the same color as the down-pointing triangle.

Data set	Split	Ours		NSGA-II	SO	
			D			
b	Train	6.97	▼	9.58	6.77	▼
b	Test	13.29		13.86	13.24	
c	Train	31.98	▼ ▼	46.67	34.30	▼
c	Test	32.71	▼	46.48	35.79	▼
y	Train	0.96	▼	3.36	1.10	▼
y	Test	1.44	▼	2.94	1.66	▼
			E			
b	Train	39.17	▼	48.96	39.33	▼
b	Test	51.13	▼	61.78	51.38	▼
c	Train	191.79	▼	279.83	194.37	▼
c	Test	211.41	▼	274.80	195.65	▼ ▼
y	Train	7.17	▼	24.39	7.23	▼
y	Test	8.87	▼	19.74	8.23	▼

cases in experiment III. The median values however, are similar. From the above, we conclude that we can effectively find a high-quality approximation front that includes the extreme solutions.

6 EXAMPLES ON REAL-WORLD DATA

In this section, we take a closer look at some results of our approach, using the yacht and the Boston housing data sets as examples. These examples illustrate how our approach could be useful to users. For both examples, we perform a single run and describe the expressions found by our approach. We use the settings as described in our experimental setup in the previous section, with tree height 3.

6.1 Example 1: Yacht data set

In Figure 5, we show the approximation front of the error and diversified error found by our approach on the yacht data set. We also show the predicted resistance $\hat{R}$ of the two expressions of three multi-tree models along the front, against the Froude number F, as well as the expressions themselves. The Froude number is a ratio between the speed of the yacht and the gravity, which is known to influence the (wave) resistance of a yacht. P is the prismatic coefficient, which is a measure of how quick the breadth of the yacht changes or in other words a measure for the fullness of the ends of the yacht.

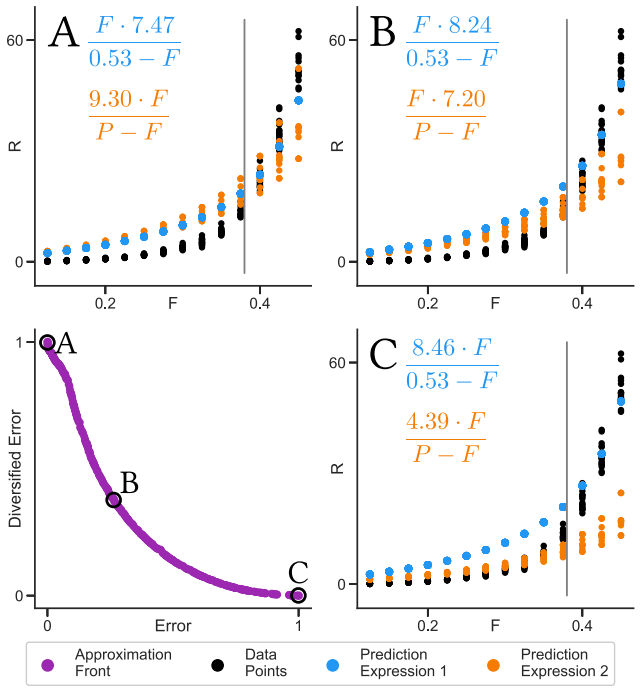


Figure 5: The approximation front of our approach on the yacht data set. The predictions $\hat{R}$ of the expressions of multi-trees A, B, and C are visualized against F.

We see that the expressions of multi-tree C, which is an extreme solution, having the best diversified error D, describe the data well together: expression 1 describes the data well for $F > 0.38$, while expression 2 describes the data well for $F < 0.38$. This corresponds well to findings in literature: according to the theory of Chapman, as well as experimental results, the resistance increases drastically when the Froude number exceeds approximately 0.38 [3, 22].

The expressions of multi-tree A, on the other hand, which is an extreme solution, having the best error E, both try to model the whole data set, which corresponds to having the lowest MSE. However, neither accurately describes the data for either $F < 0.38$ or $F > 0.38$. The predictions of the expressions of multi-tree B are visually somewhere in between the predictions of A and C.

6.2 Example 2: Boston housing data set

In Figure 6, we show the approximation front of the error and diversified error found by our approach on the Boston housing data set. Furthermore, we show the two expressions, and their individual MSE, of three multi-tree models along the front.

Expression 2 of multi-tree A has the lowest error. However, depending on the task, this expression might not be the expression that best accommodates the needs of the user. Assume the user is a real estate investor that is interested in the factors that influence the house price. Now suppose that the user knows that a new highway will be built in the near future. The user wants to take this into account when buying new houses and wants to predict how the highway might affect the value of houses in that area, to predict how much profit can be made. Therefore, the user employs expression 1 of multi-tree B, which uses x_8, the index of the accessibility

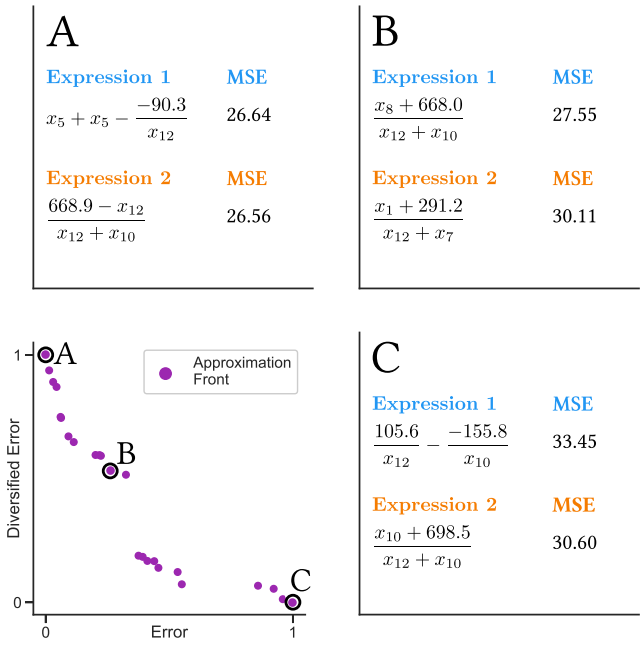


Figure 6: The approximation front of our approach on the Boston housing data set. The two expressions, and the individual MSE of multi-trees A, B, and C are shown. x_1 is the proportion of land with big lots, x_5 is the average number of rooms per residence, x_7 is the weighted distance to five employment centers, x_8 is the accessibility to radial highways, x_{10} is the ratio of pupils to teachers of a town, and x_{12} is the percentage of adults without high school education.

to radial highways, assuming the new highway only effects x_8 and not the other variables, instead of expression 2 of multi-tree A that does not use x_8 and is only slightly different in MSE. This shows how it can be useful to search for multiple expressions that are different but have a similar MSE. If the user had used SO optimization to generate expressions, the user would not have had this choice, because only one model would have been presented.

7 DISCUSSION

We evaluated our approach with multi-trees where $n = 2$. To use our approach with multi-trees where $n > 2$, we need to generalize the diversified error objective function D. In principle, this can be done by simply taking the minimum error over all trees. For the extreme solution that optimizes this generalization, individual trees will still represent different parts of the data more closely. Toward the other extreme, the sum of MSE values, however, this generalization does not necessarily have the same effect as for $n = 2$. To realize this also for $n > 2$, we specify *two* diversified error functions, D_1 and D_2. D_1 is the aforementioned generalization, and D_2 is the average *pairwise* mean of the minimum squared error of the trees:

$$D_1 = \frac{1}{|X|}\sum_{j=1}^{|X|} \min((t_1(X_j) - y_j)^2, \ldots, (t_n(X_j) - y_j)^2)$$

$$D_2 = \frac{2}{n \cdot (n-1)} \sum_{i=1}^{n-1} \sum_{l=i+1}^{n} D_p(t_i, t_l),$$

where

$$D_p(t_i, t_l) = \frac{1}{|X|}\sum_{j=1}^{|X|} \min((t_i(X_j) - y_j)^2, (t_l(X_j) - y_j)^2).$$

Note that for $n = 2$ the objective functions are equal, i.e., $D = D_1 = D_2$. D_1 stimulates sets of expressions that describe a data set well together. D_2 stimulates sets of expressions that have similar error but have a different error distribution over the data points. We explain why D_2 stimulates a different kind of diversity with an example in Figure 7. In this example, either t_2 or t_3 is closer to every target value $y_1, \ldots, y_5$ than t_1. Therefore, t_1 does not decrease D_1, even though it adds diversity with respect to t_2 and t_3. D_2, in contrast, takes into account the diversity that t_1 adds.

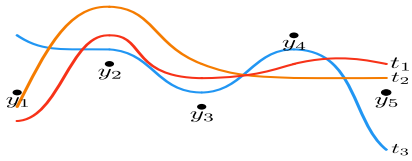


Figure 7: A multi-model with $n = 3$ and target values $y_1, \ldots, y_5$.

When $n > 2$, one could 1) use either D_1 or D_2, depending on their needs, 2) perform multiple runs, some of which optimize D_1, and some of which optimize D_2, or 3) perform a run which optimizes both D_1 and D_2 along E with MO optimization. This may however make final model selection more complicated.

Additionally, our approach increases computational requirements because we increase the number of fitness evaluations. We compute the fitness for multiple trees, multiple objective functions, and iterate over more FOS elements.

Finally, an important next step is to evaluate our approach in a real-world setting where users are provided with multiple expressions to choose from. For example, it could be researched how people interact with the solutions that our approach finds, and how to present multiple solutions to a user. Another aspect to study is how to aggregate and/or combine the results of our approach with cross-validation, given that you would get multiple fronts of models. Finally, the interleaved multi-start scheme [7] could be implemented such that users do not have to choose a population size when using our approach.

8 CONCLUSION

In this work, we have presented a novel multi-modal multi-tree MO GP approach that extends a modern model-based GP algorithm known as GP-GOMEA. We presented experimental evidence on synthetic and real-world data that showed that our approach can generate multiple diverse high-quality expressions that include expressions that excel in different notions of quality. Our approach could be a promising approach to allow users to inspect different models, which could lead to novel insights into the data and the process underlying the data. Providing the user with options in this manner, possibly in combination with additional expert knowledge, could help support them in choosing a good model for the task at hand in a powerful and sensible way.

ACKNOWLEDGMENTS

This research was funded by the European Commission within the HORIZON Programme (TRUST AI Project, Contract No.: 952060).

REFERENCES

[1] Anton Bouter, Tanja Alderliesten, Cees Witteveen, and Peter A. N. Bosman. 2017. Exploiting Linkage Information in Real-Valued Optimization with the Real-Valued Gene-Pool Optimal Mixing Evolutionary Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Berlin, Germany) *(GECCO '17)*. Association for Computing Machinery, New York, NY, USA, 705–712. https://doi.org/10.1145/3071178.3071272

[2] Armand R. Burks and William F. Punch. 2015. An Efficient Structural Diversity Technique for Genetic Programming. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (Madrid, Spain) *(GECCO '15)*. Association for Computing Machinery, New York, NY, USA, 991–998. https://doi.org/10.1145/2739480.2754649

[3] R.B. Chapman. 1972. Hydrodynamic drag of semisubmerged ships. *Journal of Basic Engineering* 94 (1972), 879–884. Issue 4. https://doi.org/10.1115/1.3425581

[4] European Commission, Content Directorate-General for Communications Networks, and Technology. 2019. *Ethics guidelines for trustworthy AI*. Publications Office. https://doi.org/10.2759/177365

[5] Edwin D. de Jong, Richard A. Watson, and Jordan B. Pollack. 2001. Reducing Bloat and Promoting Diversity Using Multi-Objective Methods. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (San Francisco, California) *(GECCO'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 11–18.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Transactions on Evolutionary Computation* 6, 2 (April 2002), 182–197. https://doi.org/10.1109/4235.996017

[7] Arkadiy Dushatskiy, Marco Virgolin, Anton Bouter, Dirk Thierens, and Peter AN Bosman. 2021. Parameterless Gene-pool Optimal Mixing Evolutionary Algorithms. *arXiv preprint arXiv:2109.05259* (2021).

[8] Anikó Ekárt and Sandor Z. Németh. 2000. A Metric for Genetic Programs and Fitness Sharing. In *Proceedings of the European Conference on Genetic Programming*. Springer-Verlag, Berlin, Heidelberg, 259–270.

[9] Benjamin P. Evans, Bing Xue, and Mengjie Zhang. 2019. What's inside the Black-Box? A Genetic Programming Method for Interpreting Complex Machine Learning Models. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 1012–1020. https://doi.org/10.1145/3321707.3321726

[10] Leonardo Augusto Ferreira, Frederico Gadelha Guimarães, and Rodrigo Silva. 2020. Applying genetic programming to improve interpretability in machine learning models. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[11] Edgar Galván and Marc Schoenauer. 2019. Promoting Semantic Diversity in Multi-Objective Genetic Programming. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 1021–1029. https://doi.org/10.1145/3321707.3321854

[12] J Gerritsma, R Onnink, and A Versluis. 1981. Geometry, resistance and stability of the Delft systematic yacht hull series. *International shipbuilding progress* 28, 328 (1981), 276–297.

[13] Ilan Gronau and Shlomo Moran. 2007. Optimal implementations of UPGMA and other common clustering algorithms. *Inform. Process. Lett.* 104, 6 (2007), 205–210.

[14] David Harrison and Daniel L Rubinfeld. 1978. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management* 5, 1 (1978), 81–102. https://doi.org/10.1016/0095-0696(78)90006-2

[15] Herbert J. Klausmeier. 1992. Concept Learning and Concept Teaching. *Educational Psychologist* 27, 3 (1992), 267–286. https://doi.org/10.1207/s15326985ep2703$_$1 arXiv:https://doi.org/10.1207/s15326985ep2703_1

[16] William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H Moore. 2021. Contemporary symbolic regression methods and their relative performance. *arXiv preprint arXiv:2107.14351* (2021).

[17] Ngoc Hoang Luong, Han La Poutré, and Peter A.N. Bosman. 2014. Multi-Objective Gene-Pool Optimal Mixing Evolutionary Algorithms. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation* (Vancouver, BC, Canada) *(GECCO '14)*. Association for Computing Machinery, New York, NY, USA, 357–364. https://doi.org/10.1145/2576768.2598261

[18] R I (Bob) McKay. 2000. Fitness Sharing in Genetic Programming. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation* (Las Vegas, Nevada) *(GECCO'00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 435–442.

[19] Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/

[20] Michael D. Schmidt and Hod Lipson. 2010. Age-Fitness Pareto Optimization. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (Portland, Oregon, USA) *(GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 543–544. https://doi.org/10.1145/1830483.1830584

[21] Dirk Thierens and Peter A.N. Bosman. 2011. Optimal Mixing Evolutionary Algorithms. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (Dublin, Ireland) *(GECCO '11)*. Association for Computing Machinery, New York, NY, USA, 617–624. https://doi.org/10.1145/2001576.2001661

[22] Ernest O Tuck. 1987. Wave resistance of thin ships and catamarans. Report T8701. *Applied Mathematics Department, The University of Adelaide* (1987), 15.

[23] Marco Virgolin, Tanja Alderliesten, Cees Witteveen, and Peter AN Bosman. 2021. Improving model-based genetic programming for symbolic regression of small expressions. *Evolutionary Computation* 29, 2 (2021), 211–237.

[24] I.-C. Yeh. 1998. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research* 28, 12 (1998), 1797–1808. https://doi.org/10.1016/S0008-8846(98)00165-3