Improving Generalization of Evolutionary Feature Construction with Minimal Complexity Knee Points in Regression

Hengzhe Zhang[1(✉)], Qi Chen[1], Bing Xue[1], Wolfgang Banzhaf[2], and Mengjie Zhang[1]

[1] Centre for Data Science and Artificial Intelligence and School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
hengzhe.zhang@ecs.vuw.ac.nz

[2] Department of Computer Science and Engineering in the College of Engineering and BEACON Center, Michigan State University, East Lansing, MI 48824, USA

Abstract. Genetic programming-based evolutionary feature construction is a widely used technique for automatically enhancing the performance of a regression algorithm. While it has achieved great success, a challenging problem in feature construction is the issue of overfitting, which has led to the development of many multi-objective methods to control overfitting. However, for multi-objective methods, a key issue is how to select the final model from the front with different trade-offs. To address this challenge, in this paper, we propose a novel minimal complexity knee point selection strategy in evolutionary multi-objective feature construction for regression to select the final model for making predictions. Experimental results on 58 datasets demonstrate the effectiveness and competitiveness of this strategy when compared to eight existing methods. Furthermore, an ensemble of the proposed strategy and existing model selection strategies achieves the best performance and outperforms four popular machine learning algorithms.

Keywords: Knee Point · Multi-criteria Decision-Making · Genetic Programming · Evolutionary Feature Construction · Symbolic Regression

1 Introduction

Evolutionary feature construction is an emerging topic that has achieved significant success in enhancing machine learning pipelines [32]. Formally, evolutionary feature construction methods aim to create a set of features $\Phi(X)$ to improve the learning performance of a machine learning algorithm $\mathcal{A}$ on a dataset (X, Y). Among all evolutionary feature construction methods, genetic programming (GP)-based feature construction is one of the most popular choices because its variable-length, flexible representation is a natural approach for feature construction. However, despite its considerable success, a significant challenge in this area is the problem of overfitting [2].

M. Giacobini et al. (Eds.): EuroGP 2024, LNCS 14631, pp. 142–158, 2024.
https://doi.org/10.1007/978-3-031-56957-9_9

Since evolutionary algorithms are gradient-free optimization techniques, many works are able to optimize non-differentiable complexity measures to strike a balance between training accuracy and model complexity. Many works use a multi-objective optimization framework to balance the trade-off between learning performance and model size, VC-dimension [10], input-output distance correlation [33], or Rademacher complexity [8]. This paper focuses on multi-objective feature construction using the size of GP trees as the complexity measure because of its simplicity. In simple terms, multi-objective feature construction considers both training accuracy/cross-validation score and tree size in the evaluation and selection process. Finally, a Pareto front with different levels of trade-off is obtained for users to select the appropriate model.

However, when confronted with a front consisting of solutions with varying trade-offs, one key issue is how to select the final model from the set of non-dominated solutions, a problem known as multi-objective decision-making. Most existing approaches use the model with the highest training accuracy [8], but this model may still exhibit significant complexity as it is an extreme point in the front.

In the domain of multi-objective decision-making [7], when no explicit preference is given, a common strategy is to identify a knee point [44]. The knee point is a point where a marginal improvement in one objective results in a substantial degradation in other objectives. Based on this definition, it is evident that multiple knee points may exist within the front.

Existing work in multi-objective GP often selects a single knee point based on the most significant trade-off among solutions in the front [38], which is intuitive when domain knowledge is lacking. However, for evolutionary feature construction, we hypothesize that among all knee points, the one with minimal complexity may provide better generalization performance, aligning with the philosophy of Occam's razor [36]. This hypothesis is based on the idea that a substantial increase in complexity required to improve training accuracy may be indicative of overfitting. Therefore, choosing the knee point with minimal complexity is a sensible option to avoid overfitting.

To clarify this, we present a real-world example of a front in Fig. 1 based on the dataset "OpenML_228", which is a case of severe overfitting when applying some non-dominated solutions to the test set. Here, several knee points exist on the Pareto front. However, the model at the knee point with the largest bend angle, denoted as 'B,' performs poorly, with a test relative squared error (RSE) exceeding 0.48. In contrast, the knee point with minimal complexity, denoted as 'D', demonstrates reasonable performance with a test RSE of approximately 0.16. Moreover, this example shows that for knee points on the left side of knee point 'D', there is a gradual increase in test RSE, as indicated by the color of those points, suggesting that overfitting may occur before reaching the complexity level of the traditional knee point. Therefore, selecting the knee point with the largest bend angle, as in the traditional approach, may not be ideal.

1.1 Goals

In this paper, we propose a minimal complexity knee point selection (MCKP) strategy for selecting the final model in multi-objective GP-based feature con-

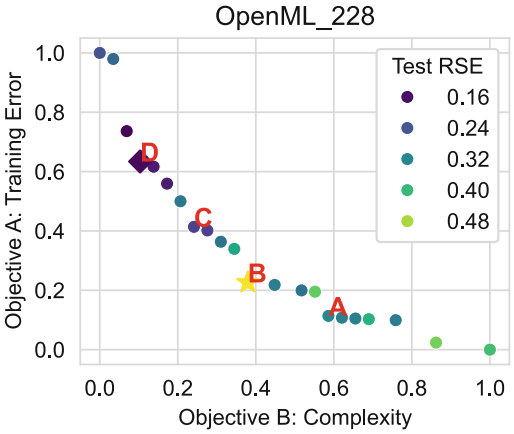


Fig. 1. Visualization of knee points on the front. The numbers in the legend represent relative squared error (RSE) on the test set, with darker colors indicating better performance, and yellow points represent extremely worse RSE. Both objectives are normalized by extreme objective values in the front. Knee points are annotated by red letters. The figure is for post-hoc analysis only and cannot be used for model selection. (Color figure online)

struction for regression[1]. Firstly, our approach uses a clustering algorithm to automatically determine the angle threshold, thereby identifying a set of angle-based knee points. Subsequently, among all knee points, we select the one with minimal complexity as the final model. The main objectives are summarized as follows:

- To favor models with potentially strong generalization performance, we propose a minimal complexity selection strategy to select the final model from the Pareto front.
- To determine a set of candidate knee points from the front, we propose a clustering-based method to automatically determine the angle threshold for knee points.
- To validate the effectiveness of the proposed strategy, we compare the MCKP strategy with seven commonly used model selection strategies in the multi-objective framework on 58 datasets.

1.2 Organization

The remainder of this paper is structured as follows: Sect. 2 reviews related work on knee point selection and overfitting control. Section 3 introduces the details of the proposed algorithm. Section 4 provides the experimental settings, and Sect. 5 shows the experimental results. Section 6 includes further analysis of the proposed strategy. Finally, we conclude the paper and outline future directions in Sect. 7.

[1] Source code: https://anonymous.4open.science/r/Knee-GP/.

2 Related Work

2.1 Knee Point Selection

In real-world applications, users often need to select a single solution from the Pareto front as the final solution, which is known as multi-objective decision-making [6]. When no specific preference exists, a common approach is to select the knee point, which is a point where improving one objective significantly decreases another. However, there is no formal, clear definition of a knee point since it depends on the specific context. In machine learning, it could mean the decrease in complexity if the training RSE score improves by 0.1 or 0.01. Due to this ambiguity, several knee point selection strategies exist in the field of multi-objective optimization [30], broadly classified as trade-off information-based and geometry property-based methods [19].

For trade-off information-based knee point selection strategies, a representative example is the utility function. Assuming there are M normalized minimization objectives $f_1, \ldots, f_M$, the trade-off between two points x_i and x_j on the Pareto front can be computed as follows [29]:

$$T\left(x_i, x_j\right) = \frac{\sum_{m=1}^{M} \max\left[0, f_m\left(x_j\right) - f_m\left(x_i\right)\right]}{\sum_{m=1}^{M} \max\left[0, f_m\left(x_i\right) - f_m\left(x_j\right)\right]}$$

This equation calculates the ratio of gain and loss when changing objective values. Then, the utility value of point x_i is defined as the minimum trade-off value $T\left(x_i, x_j\right)$ among all possible x_j on the Pareto front:

$$\mu\left(x_i, S\right) = \min_{x_j \in S} T\left(x_i, x_j\right)$$

Thus, an individual with a high utility value for any changes is considered a knee point [29].

Regarding geometry property-based knee point selection strategies, two representative examples are as follows:

- Angle-based Method [11]: For a bi-objective optimization task, the angle method calculates the angle between the line formed by the current point x and the left point x^L and the line formed by the current point x and the right point x^R. For simplicity, we can first calculate two angles $\theta^{\mathrm{L}} = \arctan \frac{f_2\left(\mathbf{x}^{\mathrm{L}}\right) - f_2(\mathbf{x})}{f_1(\mathbf{x}) - f_1\left(\mathbf{x}^{\mathrm{L}}\right)}$ and $\theta^{\mathrm{R}} = \arctan \frac{f_2(\mathbf{x}) - f_2\left(\mathbf{x}^{\mathrm{R}}\right)}{f_1\left(\mathbf{x}^{\mathrm{R}}\right) - f_1(\mathbf{x})}$, as shown in Fig. 2. The bend angle is then defined as the difference between θ^{L} and θ^{R}, i.e., $\theta\left(\mathbf{x}, \mathbf{x}^L, \mathbf{x}^R\right) = \theta^L - \theta^R$. The point with the largest bend angle $\theta\left(\mathbf{x}, \mathbf{x}^L, \mathbf{x}^R\right)$ is chosen as the knee point.
- Distance To Extreme Line [31]: The distance to extreme line method identifies the knee point on the Pareto front by finding the point with the maximum distance from a line $\mathcal{L}\left(p_1^*, p_2^*\right)$, where $\mathcal{L}\left(p_1^*, p_2^*\right)$ represents the line connecting two extreme points p_1^* and p_2^* on the Pareto front.

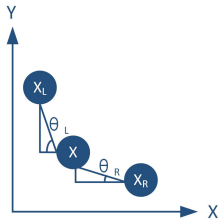


Fig. 2. Angle-based knee point calculation.

In the GP domain, knee point-based selection methods have been used for determining important features [37] and important individuals for knowledge transfer [38]. However, their application in selecting the final model based on the trade-off between training accuracy and model complexity remains limited. This paper explores this aspect.

2.2 Evolutionary Feature Construction

Evolutionary feature construction has been widely used to enhance learning performance and can be categorized into three categories: wrapper-based, filter-based, and embedded methods.

- Wrapper-based methods evaluate features based on a specific learning algorithm, such as KNN [28] and decision trees [43]. These methods can achieve good performance with that specific learning algorithm. However, the wrapper-based method can sometimes lead to overfitting because it directly optimizes accuracy or cross-validation scores on the training data.
- Filter-based methods use general metrics that are independent of any learning algorithm to evaluate features, such as purity [22], which is inexpensive and can generalize to different kinds of algorithms. However, these features may not have optimal performance on a specific learning algorithm.
- Embedded methods construct features during the learning process, with symbolic regression [9] being a typical example.

This paper focuses on wrapper-based methods due to their effectiveness. The problem of overfitting in wrapper-based methods is the issue we aim to address in this paper.

2.3 Overfitting Control for Genetic Programming

GP-based symbolic regression and feature construction methods have achieved great success in recent years. However, a significant challenge in applying evolutionary feature construction in real-world scenarios is its susceptibility to overfitting on limited or noisy training data [1,34]. To address this challenge, various approaches have been explored. Some incorporate metrics from statistical machine learning theory, such as Tikhonov regularization [24], VC dimen-

sion [10], or Rademacher complexity [8], as additional optimization objectives. Others adopt overfitting control techniques from other machine learning domains, including auxiliary fitness functions [4], modular architecture [39], semantic hoist mutation [40], multi-task learning [5], feature selection [9], ensemble learning [41], and random sampling [16].

Among these overfitting control techniques, multi-objective GP is widely used in state-of-the-art symbolic regression and evolutionary feature construction algorithms [8,17]. Typically, one objective is set as the training accuracy, and the other objective is the model size. However, a challenge arises in that a front of models with different levels of training performance and complexity is available at the end of evolution. When domain experts are available, they can inspect these models and select the best one. However, in many cases where domain experts are not available, many existing algorithms simply choose the model with the best training accuracy [8,17], which is evidently suboptimal and worth further investigation.

3 The Proposed Method

3.1 Algorithm Framework

Overall, this paper focuses on evolutionary feature construction based on multi-tree GP with a linear regression model. The optimization objectives are leave-one-out cross-validation loss and tree sizes. The evolutionary process follows a common framework of evolutionary feature construction, which includes the following stages:

- Population Initialization: Initially, a set of individuals is randomly initialized using the ramped half-and-half method [3]. Each multi-tree GP individual starts with a single randomly initialized GP tree, representing a constructed feature. Although only one GP tree is initialized in each individual, additional GP trees can be added using genetic operators during offspring generation [20].
- Individual Evaluation: For each individual, the evaluation process first transforms the training data using the features constructed by all trees within a GP individual. Then, a linear regression model is trained on the constructed features to calculate training errors using a leave-one-out cross-validation scheme on the training data [42]. Along with the training error, we also compute the tree size, which is the sum of the sizes of all GP trees within an individual.
- Parent Selection: After obtaining objective values, parents are selected using the domination-based binary tournament selection operator in NSGA-II [12]. The general idea is that, for a pair of randomly selected individuals, the non-dominated solution is given the first priority, and then the individual with the better crowding distance is considered if the two individuals are non-dominated with respect to each other.
- Offspring Generation: Offspring are generated by using random subtree crossover and random subtree mutation on GP trees. Moreover, the random

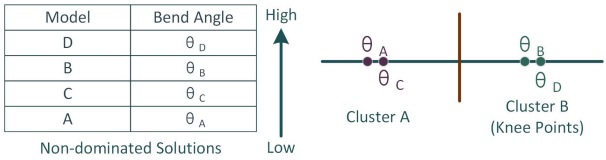


Fig. 3. Automatic determination of the knee point threshold through clustering.

tree addition and random tree deletion operators [20] are used to enable the construction of more than one feature. The crossover operator, mutation operator, and addition/deletion operator are applied sequentially with their respective probabilities.

– Environmental Selection: In this stage, non-dominated sorting with crowding distance [12] is used to select surviving individuals from a combination of parent and offspring individuals.

The evaluation, parent selection, offspring generation, and environmental selection are performed iteratively until the termination criterion is met, resulting in a front of solutions with various trade-offs between model complexity and training accuracy. Subsequently, we can use the knee point selection strategy to identify the final model from this front. The predictions on unseen data are bounded within the range of the training data to avoid overly large extrapolations because we have seen that bounding the predictions, i.e., using decision trees [35] and k-nearest neighbor [18] as the base learner, can provide good generalization performance.

3.2 Minimal-Complexity Knee Selection

In this paper, we first calculate the bend angle θ_Δ for each point x using its adjacent points x_{i-1} and x_{i+1} in the objective space of the front. Then, we can identify knee points based on a threshold of bend angles. However, using a static threshold is challenging, as finding a fixed value suitable for all datasets is difficult. Thus, as illustrated in Fig. 3, K-Means is applied to automatically determine the threshold by clustering non-dominated solutions into k groups based on angles, where k is a hyperparameter. The cluster corresponding to the largest angle is chosen and denoted by $\mathcal{C}_{max}$, representing the cluster of knee points. Within that cluster $\mathcal{C}_{max}$, the model with the minimum complexity is selected as the final model. If the number of points is less than the required number of clusters, then the point with the largest bend angle is selected. The pseudo-code is presented in Algorithm 1, which primarily consists of two stages:

– Angle Calculation (Lines 2–7): The angles are calculated based on the bend angle calculation method introduced in Sect. 2.1.
– Minimal-Complexity Knee Selection (Lines 9–15): After using clustering techniques to determine the threshold for knee points, the knee point with minimal complexity is chosen as the final model.

Algorithm 1. Minimal Complexity Knee Selection

Input: $\mathbf{x}$: Points from Pareto front, k: Number of clusters
Output: Φ_{min}: Selected model with minimum complexity

$\boldsymbol{\Theta} \leftarrow \{\}$ ▷ Initialize angle set
for $i = 1$ to $len(\mathbf{x}) - 1$ **do** ▷ Iterate through points
  $\theta_L \leftarrow \arctan\left(\frac{\mathbf{x}[i-1][1]-\mathbf{x}[i][1]}{\mathbf{x}[i][0]-\mathbf{x}[i-1][0]}\right)$ ▷ Calculate left angle
  $\theta_R \leftarrow \arctan\left(\frac{\mathbf{x}[i][1]-\mathbf{x}[i+1][1]}{\mathbf{x}[i+1][0]-\mathbf{x}[i][0]}\right)$ ▷ Calculate right angle
  $\theta_\Delta \leftarrow \theta_L - \theta_R$ ▷ Compute the bend angle
  $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} \cup \{\theta_\Delta\}$
end for
$\mathcal{C} \leftarrow \text{KMeans}(\boldsymbol{\Theta}, k)$ ▷ Cluster the angles
$\theta_{max} \leftarrow -\infty$
for $\mathcal{C}_j$ in $\mathcal{C}$ **do** ▷ Find cluster with max angle
  if $\frac{1}{|\mathcal{C}_j|}\sum_{\theta \in \mathcal{C}_j} \theta > \theta_{max}$ **then**
    $\theta_{max} \leftarrow \frac{1}{|\mathcal{C}_j|}\sum_{\theta \in \mathcal{C}_j} \theta$
    $\mathcal{C}_{max} \leftarrow \mathcal{C}_j$
  end if
end for
$\Phi_{min} \leftarrow \text{argmin}_{\Phi \in \mathcal{C}_{max}} \text{Complexity}(\Phi)$ ▷ Select features with minimum complexity
return Φ_{min}

4 Experimental Settings

4.1 Datasets

The datasets consist of real-world datasets from the Penn Machine Learning Benchmark (PMLB) [26], which is a curated list of datasets from OpenML. Synthetic datasets are excluded because they are less prone to overfitting. After excluding the synthetic datasets, 58 datasets remains.

4.2 Evaluation Protocol

To obtain reliable results, we conduct 30 independent runs with different random seeds. In each run, to simulate situations where training samples are scarce, only 100 training instances are used as the training data for feature construction [25], and the remaining data are used for testing. To further increase the difficulty, 20 random variables generated from $\mathcal{N}(0, 1)$ are appended to the datasets. To eliminate magnitude differences between different datasets, RSE is used to evaluate the performance of a model on test data. After conducting 30 independent runs, the signed rank test at a significance level of 0.05 was used to examine statistical differences among algorithms.

Table 1. Parameter settings for MCKP-GP.

Parameter	Value
Maximal Population Size	100
Number of Generations	50
Crossover and Mutation Rates	0.9 and 0.1
Tree Addition Rate	0.5
Tree Deletion Rate	0.5
Initial Tree Depth	0–3
Maximum Tree Depth	10
Initial Number of Trees	1
Maximum Number of Trees	20
Elitism (Number of Individuals)	1
Functions	+, −, *, AQ, Sqrt, Max, Min, Negative, Abs, ReLU, Gaussian

4.3 Parameter Settings

The parameter settings are shown in Table 1, which are common settings for GP. For instance, the crossover rate is set significantly higher than the mutation rate to facilitate the exchange of building blocks. To prevent zero-division errors, we employ the analytical quotient (AQ) [23], defined as $AQ = \frac{a}{\sqrt{1+(b^2)}}$ for given inputs a and b. We use *ReLU* and *Gaussian* because they have shown good performance in neuroevolution [15]. The range for ephemeral random constants is set to $[-5\tau, 5\tau]$, where τ represents the maximum absolute value of input variables [35].

4.4 Baseline Algorithms

The baseline algorithms include five popular knee point selection strategies:

- Angle Knee Selection (AKS) [6]: AKS identifies the final model by selecting the point with the maximum angle formed by it, its left neighbor, and its right neighbor. The angle calculation method is introduced in Sect. 2.
- Four Angle Knee Selection (FAKS) [6]: FAKS is similar to AKS, but considers the maximum angle formed by four adjacent points instead of two adjacent points to determine the knee.
- Bended Angle Knee Selection (BAKS) [11]: BAKS is similar to AKS, but uses the two extreme points as reference points for angle calculation, rather than using two adjacent points.
- Utility Function Knee Selection (UFKS) [29]: UFKS selects the individual with the highest utility value as the final model. The utility function is introduced in Sect. 2.
- Distance To Extreme Line Knee Selection (DELKS) [44]: In this method, the model with the maximum Euclidean distance from the extreme line is chosen as the final model.

In addition to knee point selection methods, we also compare:

- Best Training Accuracy (BTA) [8,17]: BTA selects the model with the best training accuracy/lowest training error from the front as the final model.
- Best Harmonic Mean Rank (HMR) [14]: HMR ranks models based on the harmonic mean of accuracy rank (r_a) and model size rank (r_m), using the formula $\frac{1}{r_a^{-1}+r_m^{-1}}$. The model with the best harmonic mean rank is chosen as the final model. This method is used for ranking models discovered by different algorithms in the GECCO 2022 symbolic regression competition [14], but it is also applicable for ranking models within a front.
- Standard GP (STD-GP): STD-GP is a standard GP algorithm that does not consider model size as an additional objective.

Except for STD-GP, all model selection methods follow the same multi-objective evolutionary process, differing only in how they select the final model from the Pareto front.

5 Experimental Results

In this section, we validate the effectiveness of the proposed minimal complexity knee point selection strategy by comparing it with other model selection strategies. Additionally, we inspect the Pareto front and conduct parameter sensitivity analysis to further demonstrate the effectiveness of the proposed method.

5.1 Comparison of Model Selection Strategies

In this section, we present experimental results of test RSE when employing different model selection strategies, as detailed in Table 2. There are two points to highlight from the results.

First, the proposed MCKP strategy significantly improves the generalization performance of standard GP on 32 datasets and degrades it on 11 datasets, indicating that MCKP effectively enhances generalization performance. Traditional knee point selection strategies also outperform standard GP to varying degrees. In comparison, BTA improves performance on only one dataset while worsening it on four datasets. Thus, existing methods for selecting the best training performance are not effective for controlling overfitting, and knee point selection strategies are better options.

Second, the experimental results show that using the knee point with minimal complexity outperforms the AKS strategy on 20 datasets and underperforms on 7 datasets. This is an interesting finding because except for MCKP, other knee point selection strategies show similar behaviors to each other, as most of them exhibit similar performance on more than 50 out of 58 datasets. Ideally, it would be great to know which strategy performs well on which dataset, as instance space analysis techniques show [21]. However, it is a very difficult task because overfitting is not only related to the number of instances but also the noise in data, which is an unknown property. Thus, an alternative way is to combine

Table 2. Statistical comparison of test RSE across various model selection strategies. ("+", "∼", and "−" indicate that using the method in a row performs better than, similar to, or worse than using the method in a column.)

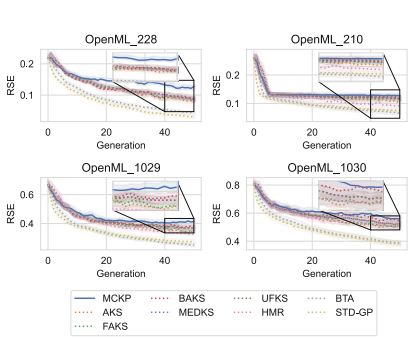
	AKS	FAKS	BAKS	MEDKS
MCKP	20(+)/31(∼)/7(−)	15(+)/36(∼)/7(−)	10(+)/37(∼)/11(−)	10(+)/40(∼)/8(−)
AKS	—	0(+)/58(∼)/0(−)	0(+)/56(∼)/2(−)	0(+)/54(∼)/4(−)
FAKS	—	—	0(+)/55(∼)/3(−)	0(+)/56(∼)/2(−)
BAKS	—	—	—	0(+)/58(∼)/0(−)
MEDKS	—	—	—	—
UFKS	—	—	—	—
HMR	—	—	—	—
BTA	—	—	—	—
	UFKS	HMR	BTA	STD-GP
MCKP	11(+)/37(∼)/10(−)	16(+)/30(∼)/12(−)	31(+)/15(∼)/12(−)	32(+)/15(∼)/11(−)
AKS	0(+)/55(∼)/3(−)	2(+)/51(∼)/5(−)	23(+)/28(∼)/7(−)	21(+)/29(∼)/8(−)
FAKS	0(+)/56(∼)/2(−)	2(+)/52(∼)/4(−)	22(+)/28(∼)/8(−)	24(+)/26(∼)/8(−)
BAKS	0(+)/58(∼)/0(−)	3(+)/52(∼)/3(−)	24(+)/29(∼)/5(−)	24(+)/27(∼)/7(−)
MEDKS	0(+)/58(∼)/0(−)	3(+)/52(∼)/3(−)	24(+)/28(∼)/6(−)	26(+)/26(∼)/6(−)
UFKS	—	3(+)/52(∼)/3(−)	23(+)/29(∼)/6(−)	25(+)/27(∼)/6(−)
HMR	—	—	24(+)/31(∼)/3(−)	27(+)/26(∼)/5(−)
BTA	—	—	—	1(+)/53(∼)/4(−)

models selected by two strategies and make an ensemble prediction. By doing so, we hope the model can benefit from two models, which will be shown in Sect. 6.

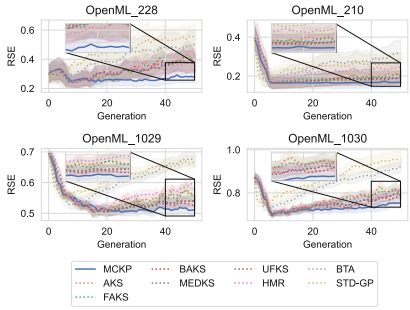
To further analyze the behavior of different selection strategies, we plot both the evolutionary training curve and the corresponding test curve of these selection methods on four representative datasets. The training curves are shown in Fig. 4a, and they reveal that MCKP has a significantly lower training curve compared to other methods. This aligns with our assumption because MCKP favors the simplest knee point, which has higher training error than traditional knee points. However, as shown in Fig. 4b, other strategies may overfit on datasets like "OpenML_228", whereas MCKP handles overfitting well on these datasets. Thus, in practical scenarios where domain knowledge suggests potentially severe overfitting, considering MCKP for model selection can mitigate the risk of overfitting.

5.2 Visualization of Pareto Fronts

In Sect. 1, we introduced the minimal knee point selection method using an example of the final front. Here, we provide more results on various datasets in Fig. 5. The training error and complexity in these figures are normalized according to the best and worst objective values achieved by non-dominated individuals. These results highlight that knee points with the largest bend angles are not good in many cases. For example, on the "OpenML_210" dataset, the traditional knee point, labeled as point "B", has a relatively high RSE of around



(a) Training RSE of the models selected by different model selection methods.



(b) The corresponding test RSE of the selected models using different model selection methods.

Fig. 4. Evolutionary plots of training RSE and test RSE for the selected models.

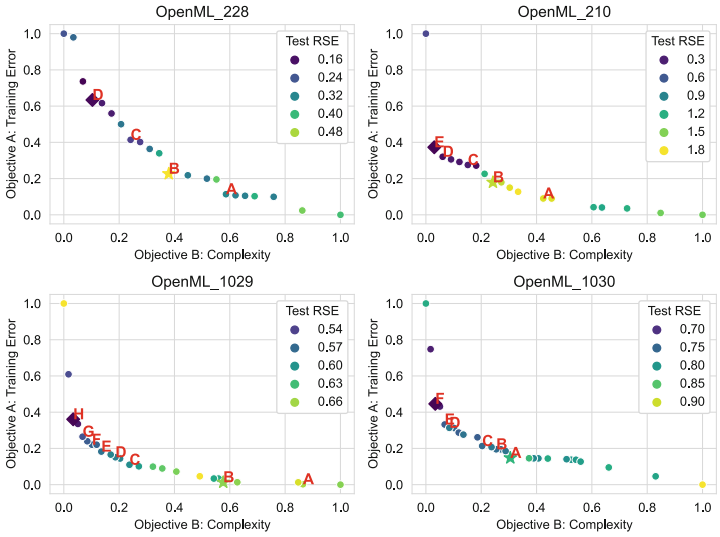


Fig. 5. Visualization of knee points on Pareto fronts. The numbers in the legend represent normalized test MSE, where lower values are better. Knee points are annotated by red letters. The "star" point denotes the traditional knee point, and the "diamond" point represents the minimal complexity knee point. Yellow points represent models with extremely high test errors. (Color figure online)

1.5, whereas the knee point with minimal complexity can achieve a lower RSE of approximately 0.3. Similar trends can also be observed in other figures, suggesting that selecting the knee point with minimal complexity is a better option in many cases.

Table 3. Statistical comparison of test RSE for different numbers of clusters.

	3	5
2	9(+)/42(∼)/7(−)	15(+)/34(∼)/9(−)
3	—	2(+)/55(∼)/1(−)

Table 4. Statistical comparison of test RSE for different model selection strategies

	AKS+HMR	MCKP	AKS	HMR
MCKP+HMR	9(+)/46(∼)/3(−)	12(+)/44(∼)/2(−)	22(+)/35(∼)/1(−)	20(+)/34(∼)/4(−)
AKS+HMR	—	14(+)/38(∼)/6(−)	10(+)/47(∼)/1(−)	4(+)/53(∼)/1(−)
MCKP	—	—	20(+)/31(∼)/7(−)	16(+)/30(∼)/12(−)
AKS	—	—	—	2(+)/51(∼)/5(−)

5.3 Impact of the Number of Clusters

The number of clusters is a hyperparameter that needs to be set when determining the threshold of knee points. Table 3 presents the impact of different numbers of clusters on final performance. As shown in the results, compared to using a cluster number of 3, using a cluster number of 2 can improve performance on nine datasets but can also degrade it on seven datasets. Using a cluster number of 5 improves performance compared to using a cluster number of 3 on one dataset but worsens it on two datasets. In summary, using the default parameter of 3 is a reasonable choice, although using a cluster number of 2 is good as well.

6 Further Analysis

In this section, considering the conclusions from the previous section, we first conduct experiments to ensemble different types of knee points to achieve better performance. Following that, we compare GP with the proposed knee point selection strategy to several popular interpretable machine learning algorithms.

6.1 Post-Hoc Analysis of Ensemble Learning

Even though the MCKP strategy is better than other methods, like HMR, on 16 datasets, it is worth noting that MCKP is still worse than HMR on around 12 datasets. Given that MCKP and other model selection techniques have varying advantages on different datasets, we propose using ensemble learning to enhance performance. In this section, we focus on combining MCKP and HMR because AKS, FAKS, BAKS, MEDKS, and UFKS have similar test RSE to HMR on most datasets, indicating that they select similar models from the front. The experimental results of RSE, presented in Table 4, demonstrate that combining MCKP and HMR through ensemble learning performs better than using MCKP alone on 12 datasets and is worse on only 2 datasets. This indicates that ensemble learning can combine the advantages of two different model selection strategies and

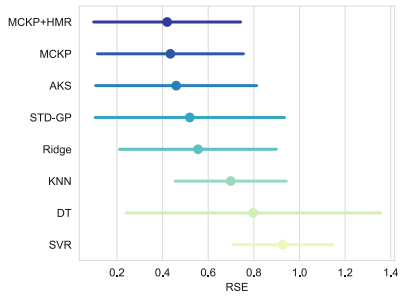


Fig. 6. Median RSE of different learning methods.

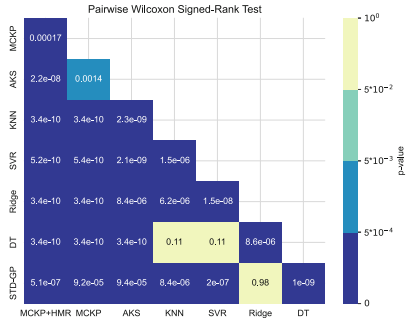


Fig. 7. Pairwise statistical comparison of different learning methods.

achieve better performance. Moreover, we also tried to combine AKS and HMR, but its performance is significantly worse than combining MCKP and HMR on 9 datasets, and only better on 3 datasets. These results demonstrate that it is important to incorporate selection strategies with different behaviors to achieve good performance. AKS and HMR have similar test RSE on 51 datasets, and their similar behavior in model selection results in fewer improvements compared to combining MCKP and HMR.

6.2 Comparisons with Other Machine Learning Algorithms

To further validate the effectiveness of the proposed method, we compare it with popular machine learning algorithms, including support vector regression (SVR), k-nearest neighbor (KNN), Ridge, and decision tree (DT) [27]. The experimental results of RSE are presented in Fig. 6, and the pairwise signed rank test with Benjamini-Hochberg correction is shown in Fig. 7. These results indicate that GP outperforms popular machine learning algorithms when dealing with sample-limited and noisy datasets. Furthermore, the proposed knee point selection strategy further enhances the advantages of GP, especially the ensemble knee point selection strategy.

7 Conclusions

In this paper, we propose a minimal complexity strategy, MCKP, to select the final model from knee points on the front of training accuracy and model size in order to improve the generalization performance of GP-based evolutionary feature construction algorithms. Experimental results on 58 datasets show that MCKP outperforms existing knee point-based model selection strategies and the strategy that selects the model with the best training accuracy/lowest training error in controlling severe overfitting. Given that we usually do not know which dataset is prone to overfitting, we also propose an ensemble strategy, combining

MCKP with traditional knee point-based model selection strategies, which yields the best performance.

In this work, we simply ensemble different models selected by different strategies to combine the advantages of these various models. Future work could delve deeper into determining the most suitable scenario for using each selection strategy. The analysis could follow a similar approach to the one employed in analyzing symbolic regression benchmarks [13].

References

1. Agapitos, A., Loughran, R., Nicolau, M., Lucas, S., O'Neill, M., Brabazon, A.: A survey of statistical machine learning elements in genetic programming. IEEE Trans. Evol. Comput. **23**(6), 1029–1048 (2019)
2. Al-Sahaf, H., et al.: A survey on evolutionary machine learning. J. R. Soc. N. Z. **49**(2), 205–228 (2019)
3. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications. Morgan Kaufmann Publishers Inc. (1998)
4. Bi, Y., Xue, B., Zhang, M.: Dual-tree genetic programming for few-shot image classification. IEEE Trans. Evol. Comput. **26**(3), 555–569 (2021)
5. Bi, Y., Xue, B., Zhang, M.: Learning and sharing: a multitask genetic programming approach to image feature learning. IEEE Trans. Evol. Comput. **26**(2), 218–232 (2021)
6. Branke, J., Deb, K., Dierolf, H., Osswald, M.: Finding knees in multi-objective optimization. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 722–731. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30217-9_73
7. Chaudhuri, S., Deb, K.: An interactive evolutionary multi-objective optimization and decision making procedure. Appl. Soft Comput. **10**(2), 496–511 (2010)
8. Chen, Q., Xue, B., Zhang, M.: Rademacher complexity for enhancing the generalization of genetic programming for symbolic regression. IEEE Trans. Cybern. **52**(4), 2382–2395 (2022)
9. Chen, Q., Zhang, M., Xue, B.: Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. IEEE Trans. Evol. Comput. **21**(5), 792–806 (2017)
10. Chen, Q., Zhang, M., Xue, B.: Structural risk minimization-driven genetic programming for enhancing generalization in symbolic regression. IEEE Trans. Evol. Comput. **23**(4), 703–717 (2018)
11. Deb, K., Gupta, S.: Understanding knee points in bicriteria problems and their implications as preferred solution principles. Eng. Optim. **43**(11), 1175–1204 (2011)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
13. de França, F.O.: Transformation-interaction-rational representation for symbolic regression: a detailed analysis of SRBench results. ACM Trans. Evol. Learn. (2023)
14. de Franca, F., et al.: Interpretable symbolic regression for data science: analysis of the 2022 competition. arXiv preprint arXiv:2304.01117 (2023)
15. Gaier, A., Ha, D.: Weight agnostic neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)

16. Gonçalves, I., Silva, S.: Balancing learning and overfitting in genetic programming with interleaved sampling of training data. In: Krawiec, K., Moraglio, A., Hu, T., Etaner-Uyar, A.Ş, Hu, B. (eds.) EuroGP 2013. LNCS, vol. 7831, pp. 73–84. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37207-0_7
17. La Cava, W., Moore, J.H.: Learning feature spaces for regression with genetic programming. Genet. Program Evolvable Mach. **21**, 433–467 (2020)
18. La Cava, W., Silva, S., Danai, K., Spector, L., Vanneschi, L., Moore, J.H.: Multidimensional genetic programming for multiclass classification. Swarm Evol. Comput. **44**, 260–272 (2019)
19. Li, K., Nie, H., Gao, H., Yao, X.: Posterior decision making based on decomposition-driven knee point identification. IEEE Trans. Evol. Comput. **26**(6), 1409–1423 (2021)
20. Muñoz, L., Trujillo, L., Silva, S., Castelli, M., Vanneschi, L.: Evolving multidimensional transformations for symbolic regression with M3GP. Memetic Comput. **11**, 111–126 (2019)
21. Muñoz, M.A., et al.: An instance space analysis of regression problems. ACM Trans. Knowl. Discov. Data (TKDD) **15**(2), 1–25 (2021)
22. Neshatian, K., Zhang, M., Andreae, P.: A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. IEEE Trans. Evol. Comput. **16**(5), 645–661 (2012)
23. Ni, J., Drieberg, R.H., Rockett, P.I.: The use of an analytic quotient operator in genetic programming. IEEE Trans. Evol. Comput. **17**(1), 146–152 (2012)
24. Ni, J., Rockett, P.: Tikhonov regularization as a complexity measure in multiobjective genetic programming. IEEE Trans. Evol. Comput. **19**(2), 157–166 (2014)
25. Nicolau, M., Agapitos, A.: Choosing function sets with better generalisation performance for symbolic regression models. Genet. Program Evolvable Mach. **22**(1), 73–100 (2021)
26. Olson, R.S., La Cava, W., Orzechowski, P., Urbanowicz, R.J., Moore, J.H.: PMLB: a large benchmark suite for machine learning evaluation and comparison. BioData Min. **10**, 1–13 (2017)
27. Orzechowski, P., La Cava, W., Moore, J.H.: Where are we now? A large benchmark study of recent symbolic regression methods. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1183–1190 (2018)
28. Peng, B., Wan, S., Bi, Y., Xue, B., Zhang, M.: Automatic feature extraction and construction using genetic programming for rotating machinery fault diagnosis. IEEE Trans. Cybern. **51**(10), 4909–4923 (2020)
29. Rachmawati, L., Srinivasan, D.: Multiobjective evolutionary algorithm with controllable focus on the knees of the pareto front. IEEE Trans. Evol. Comput. **13**(4), 810–824 (2009)
30. Ramirez-Atencia, C., Mostaghim, S., Camacho, D.: A knee point based evolutionary multi-objective optimization for mission planning problems. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1216–1223 (2017)
31. Schütze, O., Laumanns, M., Coello, C.A.C.: Approximating the knee of an MOP with stochastic search algorithms. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 795–804. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87700-4_79
32. Telikani, A., Tahmassebi, A., Banzhaf, W., Gandomi, A.H.: Evolutionary machine learning: a survey. ACM Comput. Surv. (CSUR) **54**(8), 1–35 (2021)
33. Vanneschi, L., Castelli, M.: Soft target and functional complexity reduction: a hybrid regularization method for genetic programming. Expert Syst. Appl. **177**, 114929 (2021)

34. Vanneschi, L., Castelli, M., Silva, S.: Measuring bloat, overfitting and functional complexity in genetic programming. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 877–884 (2010)
35. Virgolin, M., Alderliesten, T., Bosman, P.A.: On explaining machine learning models by evolving crucial and compact features. Swarm Evol. Comput. **53**, 100640 (2020)
36. Zhang, B.T., Muhlenbein, H., et al.: Evolving optimal neural networks using genetic algorithms with occam's razor. Complex Syst. **7**(3), 199–220 (1993)
37. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling. IEEE Trans. Cybern. **51**(4), 1797–1811 (2020)
38. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Collaborative multifidelity-based surrogate models for genetic programming in dynamic flexible job shop scheduling. IEEE Trans. Cybern. **52**(8), 8142–8156 (2021)
39. Zhang, H., Chen, Q., Xue, B., Banzhaf, W., Zhang, M.: Modular multi-tree genetic programming for evolutionary feature construction for regression. IEEE Trans. Evol. Comput. (2023)
40. Zhang, H., Chen, Q., Xue, B., Banzhaf, W., Zhang, M.: A semantic-based hoist mutation operator for evolutionary feature construction in regression. IEEE Trans. Evol. Comput. (2023). https://doi.org/10.1109/TEVC.2023.3331234
41. Zhang, H., Zhou, A., Chen, Q., Xue, B., Zhang, M.: SR-Forest: a genetic programming based heterogeneous ensemble learning method. IEEE Trans. Evol. Comput. (2023)
42. Zhang, H., Zhou, A., Qian, H., Zhang, H.: PS-tree: a piecewise symbolic regression tree. Swarm Evol. Comput. **71**, 101061 (2022)
43. Zhang, H., Zhou, A., Zhang, H.: An evolutionary forest for regression. IEEE Trans. Evol. Comput. **26**(4), 735–749 (2021)
44. Zhang, X., Tian, Y., Jin, Y.: A knee point-driven evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **19**(6), 761–776 (2014)