Automatically Choosing Selection Operator Based on Semantic Information in Evolutionary Feature Construction

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,qi.chen,bing.xue,mengjie.zhang}@ecs.vuw.ac.nz

[2] Department of Computer Science and Engineering, Michigan State University, East Lansing 48824, USA
banzhafw@msu.edu

Abstract. In recent years, genetic programming-based evolutionary feature construction has shown great potential in various applications. However, a critical challenge in applying this technique is the need to select an appropriate selection operator with great care. To tackle this issue, this paper introduces a novel approach that leverages the Thompson sampling technique to automatically choose the optimal selection operator based on semantic information of genetic programming models gathered during the evolutionary process. The experimental results on a standard symbolic regression benchmark containing 37 datasets show that the proposed adaptive operator selection algorithm outperforms expert-designed operators, demonstrating the effectiveness of the adaptive operator selection algorithm.

Keywords: Genetic Programming · Evolutionary Feature Construction · Adaptive Operator Selection

1 Introduction

Automated feature construction is an important technique in the machine learning domain and has achieved significant success in various applications [1,2]. Formally, given a dataset $\{X, Y\}$, the objective of automated feature construction is to develop a set of features $\Phi_1(X), \ldots, \Phi_m(X)$ that enhance the performance of a learning algorithm on the given dataset. The effectiveness of automated feature construction techniques has been well-demonstrated by deep learning [2]

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-981-99-7022-3_36.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 385–397, 2024.
https://doi.org/10.1007/978-981-99-7022-3_36

and kernel methods [3]. However, the interpretability of the constructed features remains a notable point of criticism within the field, demanding further investigation and deliberation [4].

In recent years, interpretable automated feature construction techniques, particularly those based on genetic programming (GP), have demonstrated impressive performance for enhancing ensemble learning algorithms [5,6], compared to learning with original features. The variable-length representation and gradient-free search mechanism make GP suitable for exploring flexible high-order features, like $(x_1 + x_2) * x_3$, on non-differentiable machine learning algorithms. Based on the evaluation methods, evolutionary feature construction methods can be categorized into filter-based [7], wrapper-based [1,8], and embedded methods [9,10]. Filter-based methods do not rely on any specific machine learning algorithm, making them efficient and generalize well to different learning algorithms [7]. On the other hand, wrapper-based methods evaluate the constructed features on a specific learning algorithm, which may lead to better features at the cost of higher computation time [8]. Finally, embedded methods integrate the feature construction into model learning, with GP-based symbolic regression being a representative example [9].

To improve the search effectiveness, numerous selection operators have developed for GP, which are used in GP to select promising individuals for crossover and mutation to generate new solutions, playing a crucial role in driving the evolutionary progress. Representative examples include standard tournament selection [11], clustering tournament selection [12], lexicase selection [13], and multi-dimensional archive of phenotypic elites (MAP-Elites) [14]. Each of these operators demonstrates unique advantages in different scenarios, such as dynamic selection pressure adjustment [12], specialist preservation [15], and diversity enhancement for ensemble learning [14]. However, selecting the most appropriate selection operators in real-world tasks is challenging because suitable operators vary with different optimization landscapes or phases, often unknown in advance. Recent work has shown that GP performs well using tournament selection for the first 10% of generations, then lexicase selection for the rest [16]. Therefore, an adaptive operator selection (AOS) algorithm for the selection operator is needed.

There are two potential approaches for automatic operator selection. First, operators can be selected based on historical knowledge [17], also known as algorithm recommendation. However, obtaining historical knowledge requires running numerous experiments in advance. Furthermore, the best operator may change during the evolutionary process. Therefore, adaptive operator selection may be a better choice [18]. Given the success of AOS techniques in selecting genetic operators for continuous numerical optimization problems [19–21], particularly AOS based on the multi-armed bandit and dynamic Thompson sampling [21], this paper explores the feasibility of automatically selecting the optimal selection operators during evolution. However, in GP, relying only on the improvement of fitness values may not provide sufficient rewards to selection operators. Thus, this paper explores the use of GP semantics to design an effec-

tive AOS method, where the semantics of each GP program refers to the output values of each GP individual [22].

Goals: The main goal of this paper is to develop an AOS method for determining selection operators in GP-based feature construction. The specific objectives of this work are as follows:

1. Developing a portfolio of selection operators for AOS in evolutionary feature construction.
2. Proposing a semantic-based AOS method using dynamic Thompson sampling to adaptively determine an appropriate selection operator during the evolutionary process.
3. Evaluating the effectiveness of different selection operators and credit assignment strategies on 37 datasets.

2 Related Work

2.1 Multi-armed Bandit

The multi-armed bandit is a reinforcement learning technique that aims to balance the exploration and exploitation of different options, also known as "arms", based on past rewards [21,23,24]. In the context of GP, selection operators can be considered arms. In each generation, an operator with the highest estimated rewards is chosen, and applying this operator to select two parents is a trial. The goal is to find the optimal selection operator for GP through trials. Numerous multi-armed bandit algorithms have been developed for various scenarios, and two key techniques are particularly useful for GP.

- Dynamic Multi-armed Bandit [23,24]: In GP, the optimal selection operator may change during the evolution process. Therefore, the multi-armed bandit algorithm should have the ability to forget long-term history and focusing on recent knowledge in order to adapt to these changes, which is known as the dynamic multi-armed bandit. This is achieved via a forgetting mechanism through explicit drift detection algorithm [24] or simple decay over time [21].
- Thompson Sampling [21]: GP is a population-based optimization algorithm, and it requires to have a sampling algorithm that can generate multiple trials of selection operators simultaneously. Thus, it is desirable to have an explicit reward distribution for each selection operator, and each trial can sample a value from each distribution to determine which selection operator to choose. This process is known as Thompson sampling. Compared to using the upper confidence bound and expected improvement, Thompson sampling allows for trying different selection operators in each round, which is more naturally suited for GP.

2.2 Automatic Operator Selection

In the evolutionary computation domain, numerous genetic operators have been developed, and studies have shown that combining the advantages of different

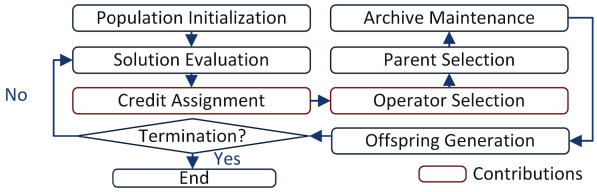


Fig. 1. Workflow of the proposed algorithm.

operators is beneficial for addressing optimization problems [25]. Instead of simple hybridization, automatic operator selection has become a hot topic in the evolutionary computation (EC) domain, aiming to dynamically choose the optimal operator at each stage [20]. One pioneering approach is probability matching, which adjusts the probability of selecting each individual based on reward distribution [19], where the reward is typically defined as the improvement in fitness values, either in a real-valued form [18] or a boolean-valued form [21]. However, probability matching does not consider accumulative reward distribution. To address this limitation, adaptive purist was proposed to accumulate reward during the evolutionary process, leading to improved operator selection performance [19]. Building upon this idea, a dynamic multi-armed bandit algorithm with the Page-Hinkley test was proposed to further enhance operator selection effectiveness [23,24]. Under the framework of fitness-rate-rank-based multi-armed bandit (FRRMAB) [20], dynamic Thompson Sampling [21], deep reinforcement learning [18,26], and other methods have been developed. While numerous approaches have been proposed for automatic operator selection, most of them primarily focus on solving numerical optimization problems and emphasize the selection of genetic operators. For GP-based feature construction algorithms, the effectiveness of automatic operator selection methods for selection operators still requires further investigation.

3 The New Algorithm

3.1 Model Representation

In this paper, we focus on evolutionary feature construction for a linear regression model due to its simplicity and effectiveness. Specifically, each GP individual consists of m GP trees, representing m constructed features $\phi_1, \ldots, \phi_m$. Based on these constructed features, a linear model is trained to make predictions for the given data. To ensure accurate and robust predictions, the final predictions are made by an ensemble model that incorporates the top-$|A|$ individuals obtained during the evolutionary process, where $|A|$ is an algorithm parameter referring to the ensemble size.

3.2 Algorithm Framework

The algorithm follows a general framework of GP, as illustrated in Fig. 1, where credit assignment and operator selection are the key components for the new AOS method to determine the best selection operator. The main components of the proposed algorithm are described as follows:

- Population Initialization: During the initialization stage, GP trees are initialized using the ramped half-and-half method [11]. Specifically, each GP individual with m trees is randomly generated, with each tree representing a constructed feature.
- Solution Evaluation: In the evaluation stage, all GP individuals are evaluated using ridge regression. Specifically, m trees construct m features, and these constructed features are then fed into a linear model to make predictions for the given data. Fitness is determined by the R^2 score on training data, with leave-one-out cross-validation to avoid overfitting by selecting a regularization coefficient from $\{0.1, 1, 10\}$.
- Credit Assignment: This phase updates the reward distribution of operators based on evaluation results. The details of the credit assignment are presented in Sect. 3.4.
- Operator Selection: Selection operators are sampled to select pairs of individuals for crossover and mutation. For a population of n individuals, it needs to sample $\frac{n}{2}$ operators. In this paper, lexicase selection and tournament selection are defined as candidate operators since they are commonly used in GP.
- Parent Selection: At this stage, GP individuals are selected using the $\frac{n}{2}$ sampled selection operators to select promising individuals.
- Archive Maintenance: In addition to selecting offspring, the top individuals in the population and the archive A are compared, and the top $|A|$ individuals are stored in the archive to form an ensemble model.
- Offspring Generation: Offspring generation is a stage where new GP individuals are generated using random subtree crossover and guided subtree mutation operator [6]. In this paper, each individual has m GP trees, and thus genetic operators are invoked m times for each individual to ensure sufficient variations.

3.3 Selection Operators

This paper considers two widely used selection operators:

1. Tournament Selection: The tournament selection operator randomly samples t individuals from the population, where t is the tournament size, and selects the best as the parent. Here, $t = 7$ is used according to common settings in GP literature.
2. Lexicase Selection [13]: The lexicase selection operator iteratively constructs filters to progressively narrow down the selection pool until one individual remains. In each round, the filter is constructed as $min_{\Phi \in P}\mathcal{L}_k(\Phi) + \epsilon_k$, with ϵ_k as the median absolute deviation of the loss on the k-th instance among all individuals $\Phi \in P$.

Intuitively, tournament selection tends to converge by favoring individuals with higher overall fitness values. In contrast, lexicase selection emphasizes improving fitness on different instances, thus promoting diversity. Thus, using an AOS method to choose between these can simultaneously improve the overall accuracy and the accuracy on tough instances, leading to a superior ensemble model. This idea is inspired by AdaBoost, where some learners have good overall accuracy while others focus on hard instances. The ensemble model can then achieve good accuracy on all instances.

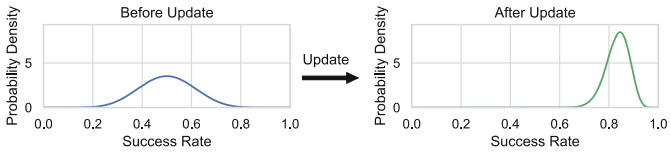


Fig. 2. Credit assignment update in multi-armed bandit using beta distribution.

3.4 Dynamic Multi-armed Bandit

To apply the dynamic multi-armed bandit in GP, two components, credit assignment and operator selection, must be carefully designed: credit assignment allocates rewards to each operator, and operator selection samples operators based on estimated rewards. This section delves into these components.

Credit Assignment: Credit assignment is a stage where rewards are assigned to each selection operator, involving two main questions:

- **How to define a successful trial?** In this paper, a successful trial for a selection operator is defined as having any improvement in one dimension of the semantics compared to the best semantics among all parents. Semantics $(\Phi(X_1), \ldots, \Phi(X_N))$ refer to the output values of each GP individual Φ, which can determine a loss vector $(\mathcal{L}_{\Phi,1}, \ldots, \mathcal{L}_{\Phi,N})$. At each generation, all individuals in the population $\Phi \in P$ can collectively form the best loss vector, where each element corresponds to the minimum loss value that individuals in P achieve on each training instance. This vector is denoted as $\{\min_{\Phi \in P} \mathcal{L}_{\Phi,i} | i \in [1, N]\}$. For a new individual Φ^+, if $\exists_{i \in [1,N]} \mathcal{L}_{\Phi^+,i} < \min_{\Phi \in P} \mathcal{L}_{\Phi,i}$, it is considered a successful trial and is rewarded with one point. This reward strategy is based on the principle that if a new individual outperforms all existing individuals on a data sample, it indicates that useful knowledge has been discovered, allowing the new individual to achieve the best performance on that sample, even if average fitness does not increase.
- **How to update estimated reward distribution?** As shown in Fig. 2, in order to use Thompson sampling, $k = 2$ beta distributions $\theta = (\theta_1, \ldots, \theta_k)$ are defined for k selection operators with two sets of parameters $\alpha = (\alpha_1, \ldots, \alpha_k)$ and $\beta = (\beta_1, \ldots, \beta_k)$. All these parameters are initialized to one. After obtaining a successful trial for each operator, the α parameters are updated, i.e.,

$\alpha_i = \alpha_i + 1$ and $\beta_i = \beta_i$. Otherwise, if the trial is unsuccessful, the β parameters are updated, i.e., $\alpha_i = \alpha_i$ and $\beta_i = \beta_i + 1$. Due to the changing dynamics of the evolutionary process, the reward distribution for k operators may change. Therefore, weight decay is applied to all distributions. After each round of updating, the distribution parameters α, β are decayed by a decay factor γ, which is set to 0.9 in this paper. In order to prevent the probability from diminishing to an extremely low value, which could lead to an operator never being chosen in the future, the decayed value is restricted to a minimum of 1.

Operator Selection: Once the parameters of all selection operators have been updated, the selection operators are sampled based on the probabilities associated with each selection operator in the operator selection stage. Specifically, the probability of choosing selection operator i is defined in Eq. (1) [21], where $\Gamma(x)$ is the gamma function, that is, $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$.

$$\mathcal{P}^{\text{Beta}}(\theta_i) = \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \theta_i^{\alpha_i - 1} (1 - \theta_i)^{\beta_i - 1}, \tag{1}$$

After applying the selected operator to select an individual, the selected operator is marked associated with the selected individual to be able to make credit assignments.

Table 1. Parameter settings for GP.

Parameter	Value
Maximal Population Size	30D (500)
Number of Generations	200
Ensemble Size	30
Crossover and Mutation Rates	0.9 and 0.1
Maximum Tree Depth	10
Initial Tree Depth	0–2
Number of Trees in An Individual	10
Elitism (Number of Individuals)	1
Functions	Add, Sub, Mul, AQ, Sin, Cos, Abs, Max, Min, Negative

4 Experimental Settings

4.1 Experimental Dataset

The experimental datasets are obtained from the Penn Machine Learning Benchmark (PMLB) [27] [1]. Due to the constraints of computational resources, we

[1] Details of Datasets: https://epistasislab.github.io/pmlb/

selected datasets with fewer than 5000 instances. Additionally, we only evaluate performance on real-world datasets. Based on these criteria, 37 datasets are finally selected. Specifically, the number of instances in these datasets ranges from 47 to 3848, and the number of dimensions falls between 2 and 124.

4.2 Parameter Settings

The parameters follow the conventions established in the GP literature, as outlined in Table 1. The population size is set to 30 times the number of original features, with a maximum limit of 500. To address the issue of zero-division errors, we replace the division operator with the analytical quotient operator [28]. The analytical quotient operator is defined as $AQ\,(a, b) = \frac{a}{\sqrt{1+b^2}}$, where a and b are two parameters.

4.3 Evaluation Protocol

The experiments are conducted on the New Zealand e-science infrastructure (NeSI), which consists of a cluster of AMD EPYC 7713 CPUs. For the evaluation protocol, each algorithm is independently tested on each dataset for 30 runs. The comparisons between algorithms are performed using the Wilcoxon signed-rank test. For each run, the datasets are split into training and test sets in an 80:20 ratio. The performance of an algorithm is evaluated using the R^2 score as the performance metric based on the test set.

4.4 Baseline Algorithms

This work considers three baseline selection operators within GP-based feature construction algorithms:

- Lexicase [13]: Only the automatic epsilon lexicase selection operator is used in GP.
- Tournament: Only the tournament selection operator is used in GP.
- TR/LS [16]: TR/LS is a heuristic operator selection strategy designed by GP experts. Tournament selection is used in the first q generations to avoid hyper-selection, and lexicase selection is used in the remaining generations. In the original paper of TS/LS [16], q is set to 10% of the total generations. Therefore, q is set to 10 in this paper.

Moreover, two different credit assignment strategies are studied to determine the best one for GP:

- Semantics: Any improvement achieved by the selection operator over a value in the vector of squared errors of parents is considered a successful improvement. This is the credit assignment strategy used in this paper.
- Fitness: Any improvement achieved by the selection operator over the best R^2 score of parents is considered a successful improvement.

5 Experimental Results

5.1 Comparison Between Selection Operators

Test Score: The experimental results using different selection operators are presented in Table 2 [2]. The results demonstrate that AOS significantly outperforms tournament selection and lexicase selection operators on 16 and 9 datasets, respectively, while not performing worse on any dataset. These results highlight the advantages of combining different selection operators in the evolutionary process. Interestingly, AOS also outperforms TL/LS, a hybrid operator designed by GP experts. As shown in Table 2, AOS performs better than the TR/LS operator on 6 datasets, similar on 30 datasets, and worse on only one dataset. These results suggest that AOS can provide an advantage over the heuristic operator selection strategy designed by GP experts.

Table 2. Comparison of R^2 scores for different selection operators.

	TR/LS	Tournament	Lexicase
AOS	6(+)/30(∼)/1(-)	16(+)/21(∼)/0(-)	9(+)/28(∼)/0(-)
TR/LS	—	11(+)/25(∼)/1(-)	5(+)/30(∼)/2(-)
Tournament	—	—	0(+)/26(∼)/11(-)

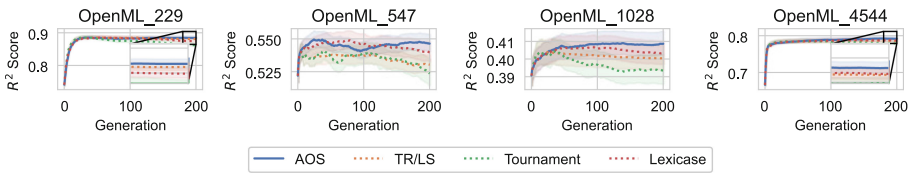


Fig. 3. Evolutionary plots of test R^2 scores using four different selection operators.

To gain further insights into the advantage of using AOS over deterministic operators, we plot the curve of test R^2 scores for representative datasets in Fig. 3. The results demonstrate that AOS can improves test R^2 scores in later generations, whereas tournament selection operators suffer from severe overfitting, leading to degraded test R^2 scores in later generations. Therefore, in the following sections, we focus on analyzing the reasons behind the improved generalization ability of the ensemble models made by AOS.

[2] Detailed Results: https://tinyurl.com/AOS-GP-Supplementary-Material

Operator Selection Patterns: To understand why AOS outperforms TS/LS, lexicase selection and tournament selection, we analyze the selection ratios of different operators during the evolutionary process on four datasets, as shown in Fig. 4. The results indicate that the lexicase selection operator performs well and is selected more frequently than the tournament selection operator. For instance, on the "OpenML_547" dataset, the lexicase selection operator has selected an average of 181 times at the last generation, while the tournament selection operator is selected only 29 times on average. Although the proportion of tournament selection operators is relatively small compared to the lexicase selection operator, this small proportion is not negligible considering the significant improvements achieved with using AOS compared with using lexicase selection alone, as presented in Fig. 3.

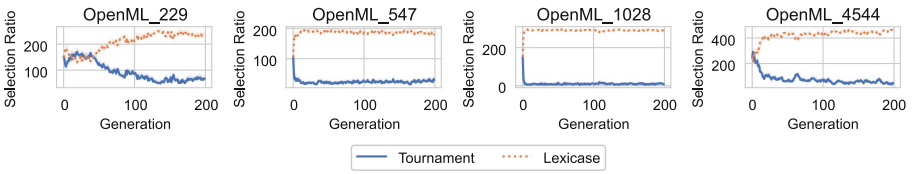


Fig. 4. Selection ratios of different operators during the evolutionary process.

Cosine Distance: To further demonstrate the reasons behind the superior performance of AOS, we introduce the average cosine distance. The average cosine distance is used as a metric to measure the complementarity of different models in the archive, which is crucial for ensemble learning [14]. A larger cosine distance indicates greater complementarity. The results in Fig. 5 demonstrate that the adaptive selection operator achieves the greatest cosine distance by adaptively balancing lexicase and tournament selections. Although lexicase selection fosters a high level of diversity, incorporating a small proportion of tournament selection appears to enhance it further. This may be because lexicase selection can suffer from hyper-selection [29], where a superior individual dominating the other individuals can be chosen up to 90% of the time [29]. In such cases, introducing a moderate proportion of tournament selection may improve archive diversity. In other cases, the high usage of lexicase selection ensures a high level of population diversity for discovering well-performing models on different training instances, thereby forming a strong ensemble learning model.

5.2 Comparison of Credit Assignment Strategies

To demonstrate the superiority of the proposed credit assignment strategy, this section compares the effectiveness of two different credit assignment strategies for GP. The comparison results between semantics-based credit assignment and fitness-based credit assignment are presented in Fig. 6a. The results indicate that utilizing semantic information for credit assignment leads to significantly better

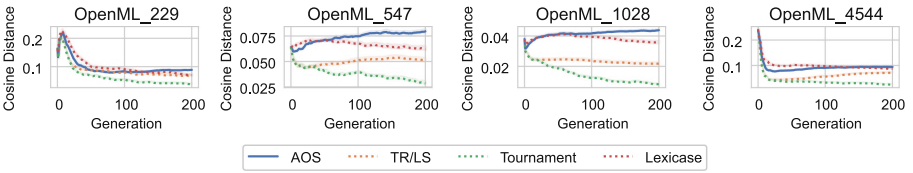
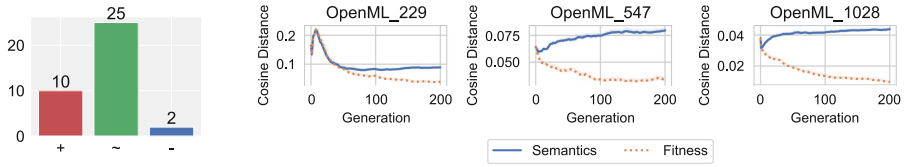


Fig. 5. Cosine distance of archived individuals.

results on 10 datasets while performing worse on 2 datasets. This performance can be attributed to the fact that credit assignment based on semantics encourages the discovery of solutions with diverse semantics, thereby facilitating the formation of a high-quality ensemble model. The evolutionary plots of the cosine distance of archived individuals are shown in Fig. 6b, clearly demonstrating the advantage of the semantic-based credit assignment strategy in terms of diversity maintenance.



(a) Test R^2 scores. (b) Cosine distance.

Fig. 6. (a). Statistical comparison of test R^2 scores. ("+"/red bar indicates that for a dataset, the semantic-based credit assignment strategy outperforms the fitness-based credit assignment strategy.) (b). Evolutionary plots of cosine distances. (Color figure online)

6 Conclusions

This work aims to automate the determination of the optimal selection operator during the process of evolutionary feature construction. To achieve this, we use the Thompson sampling technique to sample selection operators based on their estimated rewards, where the reward is defined as an improvement in semantics. The experimental results on 37 datasets demonstrate that the proposed method outperforms using sole lexicase selection, sole tournament selection and a manually designed hybrid selection operator, highlighting the advantages of employing AOS. However, it should be noted that this paper is limited to the use of AOS for determining selection operators. In future research, it would be valuable to extend this framework to the selection of genetic operators and environmental selection operators in order to further enhance the performance.

References

1. La Cava, W., Singh, T.R., Taggart, J., Suri, S., Moore, J.H.: Learning concise representations for regression by evolving networks of trees. In: ICLR (2018)
2. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
3. Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. Neural Netw. **12**(2), 181–201 (2001)
4. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. **1**(5), 206–215 (2019)
5. Zhang, H., Zhou, A., Zhang, H.: An evolutionary forest for regression. IEEE Trans. Evol. Comput. **26**(4), 735–749 (2022)
6. 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)
7. 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)
8. Virgolin, M., Alderliesten, T., Bosman, P.A.: On explaining machine learning models by evolving crucial and compact features. Swarm Evol. Comput. **53**, 100640 (2020)
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. Zhang, H., Zhou, A., Qian, H., Zhang, H.: PS-Tree: a piecewise symbolic regression tree. Swarm Evol. Comput. **71**, 101061 (2022)
11. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Stat. Comput. **4**(2), 87–112 (1994)
12. Xie, H., Zhang, M.: Parent selection pressure auto-tuning for tournament selection in genetic programming. IEEE Trans. Evol. Comput. **17**(1), 1–19 (2012)
13. La Cava, W., Helmuth, T., Spector, L., Moore, J.H.: A probabilistic and multi-objective analysis of lexicase selection and ε-lexicase selection. Evol. Comput. **27**(3), 377–402 (2019)
14. Zhang, H., Chen, Q., Tonda, A., Xue, B., Banzhaf, W., Zhang, M.: Map-elites with cosine-similarity for evolutionary ensemble learning. In: EuroGP. pp. 84–100. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-29573-7_6
15. Helmuth, T., Pantridge, E., Spector, L.: Lexicase selection of specialists. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1030–1038 (2019)
16. Xu, M., Mei, Y., Zhang, F., Zhang, M.: Genetic programming with lexicase selection for large-scale dynamic flexible job shop scheduling. IEEE Trans. Evol., Comput. (2023)
17. Tian, Y., Peng, S., Zhang, X., Rodemann, T., Tan, K.C., Jin, Y.: A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks. IEEE Trans. Artif. Intell. **1**(1), 5–18 (2020)
18. Tian, Y., Li, X., Ma, H., Zhang, X., Tan, K.C., Jin, Y.: Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. IEEE Trans. Emerg. Top. Comput, Intell (2022)

19. Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: GECCO, pp. 1539–1546 (2005)
20. Li, K., Fialho, A., Kwong, S., Zhang, Q.: Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **18**(1), 114–130 (2013)
21. Sun, L., Li, K.: Adaptive operator selection based on dynamic Thompson sampling for MOEA/D. In: Bäck, T., et al. (eds.) PPSN 2020. LNCS, vol. 12270, pp. 271–284. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58115-2_19
22. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric semantic genetic programming. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012. LNCS, vol. 7491, pp. 21–31. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32937-1_3
23. DaCosta, L., Fialho, A., Schoenauer, M., Sebag, M.: Adaptive operator selection with dynamic multi-armed bandits. In: GECCO, pp. 913–920 (2008)
24. Belluz, J., Gaudesi, M., Squillero, G., Tonda, A.: Operator selection using improved dynamic multi-armed bandit. In: GECCO, pp. 1311–1317 (2015)
25. Wang, C., Deng, Y., Li, X., Xin, Y., Gao, C.: A label-based nature heuristic algorithm for dynamic community detection. In: PRICAI, pp. 621–632. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29911-8_48
26. Zhen, H., Gong, W., Wang, L.: Evolutionary sampling agent for expensive problems. IEEE Trans. Evol., Comput. (2022)
27. 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), 1–13 (2017)
28. 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)
29. Helmuth, T., McPhee, N.F., Spector, L.: The impact of hyperselection on lexicase selection. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, pp. 717–724 (2016)