Investigating Combined Algorithm Selection and Hyperparameter Optimization for Fairness

Zhiang Chen
University College Dublin
Dublin, Ireland
zhiang.chen@ucdconnect.ie

Mark Connor
University College Dublin
Dublin, Ireland
mark.connor@ucd.ie

Sudarshan Pant
University College Dublin
Dublin, Ireland
sudarshan.pant@ucd.ie

Michael O'Neill
University College Dublin
Dublin, Ireland
m.oneill@ucd.ie

Abstract

As machine learning algorithms are increasingly employed for critical decision-making, ensuring algorithmic fairness has become imperative. Fairness-aware Automated Machine Learning has emerged as a flexible and effective method for enhancing both model accuracy and fairness. However, most existing studies focus on hyperparameter optimization for a single model. In this study, we frame the problem as a multi-objective Combined Algorithm Selection and Hyperparameter Optimization (CASH) problem, aiming to jointly optimize both accuracy and fairness across a diverse set of machine learning algorithms and their corresponding hyperparameters. To address this challenge, we apply Multi-Objective Grammatical Evolution (MOGE). The results demonstrate that MOGE not only effectively identifies models that achieve higher fairness and accuracy, while also exploring the trade-offs between accuracy and fairness efficiently.

CCS Concepts

• **Applied computing** → **Multi-criterion optimization and decision-making**; • **Computing methodologies** → **Supervised learning by classification**.

Keywords

AutoML, Algorithmic Fairness, Multi-objective Optimization

ACM Reference Format:
Zhiang Chen, Mark Connor, Sudarshan Pant, and Michael O'Neill. 2025. Investigating Combined Algorithm Selection and Hyperparameter Optimization for Fairness. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3712255.3726608

1 Introduction

The widespread adoption of machine learning has raised concerns about algorithmic fairness, as models often replicate or amplify biases, leading to systematical disadvantage for certain demographic groups[7]. While fairness-aware AutoML has emerged as an effective approach due to its efficiency and flexibility[15], previous research has primarily focused on how a single model's hyperparameter optimization (HPO) affects its fairness, neglecting the potential benefits of broader algorithm selection. Different machine learning models have unique characteristics, and searching for a wider range of models has the potential to achieve higher performance and fairness outcomes.

This leads to the central research question of this paper: *How does combined algorithm selection and hyperparameter Optimization (CASH) affect model accuracy and fairness?* We address this by formulating the problem as a multi-objective CASH task that simultaneously optimizes accuracy and fairness. We apply multi-objective grammatical evolution (MOGE) to search the model configuration while generating a set of Pareto optimal solutions that balance both objectives. The main contributions of this paper include: (1) To the best of our knowledge, this paper is the first to integrate algorithmic fairness into the CASH problem. (2) We develop a novel, human-understandable grammar to represent a feasible search space, ranging from high-level algorithm selection down to low-level hyperparameter optimization. (3) We apply MOGE to explore the Pareto frontier of model configurations, thereby enabling more effective fairness-accuracy trade-offs.

2 Background

Current fairness-aware AutoML research primarily focuses on HPO for a single model. These methods incorporate fairness in three ways: (1) Treating it as a constraint [10, 12]; (2) Including it as a weighted penalty in the objective function [4, 10, 11]; or (3) Considering it as an additional objective in multi-objective optimization [3, 13, 14]. While constraint and weighted methods require difficult condition setting and only produce single solutions [15], multi-objective methods are more flexible and generate Pareto frontiers that better explore accuracy-fairness trade-offs. Our interest in applying MOGE is motivated by three reasons: (1) Evolutionary algorithms excel at searching complex model configuration spaces

GECCO '25 Companion, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3726608

[8]; (2) They can freely optimize diverse hierarchical model configurations [6]; (3) using grammar allows us to define the search space for model configurations more flexibly. Meanwhile, existing work has largely overlooked the potential impact of algorithm selection. Therefore, we frame the problem as a multi-objective CASH problem and implement MOGE to create a Pareto frontier of diverse model configurations which seek to enhance accuracy and fairness.

3 Methodology

3.1 Combined Algorithm Selection and Hyper-Parameter Optimization for Accuracy and Fairness

First, we define the task of selecting model configurations to enhance accuracy and fairness as a multi-objective CASH problem:

$$\underset{A\in\mathcal{A},\lambda\in\Lambda(A)}{\arg\max}\quad F(A,\lambda) = \underset{A\in\mathcal{A},\lambda\in\Lambda(A)}{\arg\max}\quad (f_1(A,\lambda), f_2(A,\lambda)) \tag{1}$$

Here, $\mathcal{A} = \{A(1), \ldots, A(n)\}$ denotes a set of candidate machine learning models. For each model $A \in \mathcal{A}$, λ is a specific hyperparameter configuration configuration specific to the model A in the set of hyperparameter vectors $\Lambda(A)$. $F(A,\lambda)$ is a set of objective functions, where $f_1(A,\lambda)$ denotes the accuracy, and $f_2(A,\lambda)$ denotes the specified fairness metric.

We use the differences between two commonly used fairness metrics for different groups to measure the degree of unfairness, specifically denoted by **Average Odds Difference (AOD)** and **Statistical Parity Difference (SPD)**:

$$\begin{aligned}\text{AOD} = \frac{1}{2}\Big(&\left|P(\hat{Y}=1 \mid S=a, Y=1) - P(\hat{Y}=1 \mid S=b, Y=1)\right| \\ &+ \left|P(\hat{Y}=1 \mid S=a, Y=0) - P(\hat{Y}=1 \mid S=b, Y=0)\right|\Big)\end{aligned} \tag{2}$$

$$\text{SPD} = \left|P(\hat{Y}=1 \mid S=a) - P(\hat{Y}=1 \mid S=b)\right| \tag{3}$$

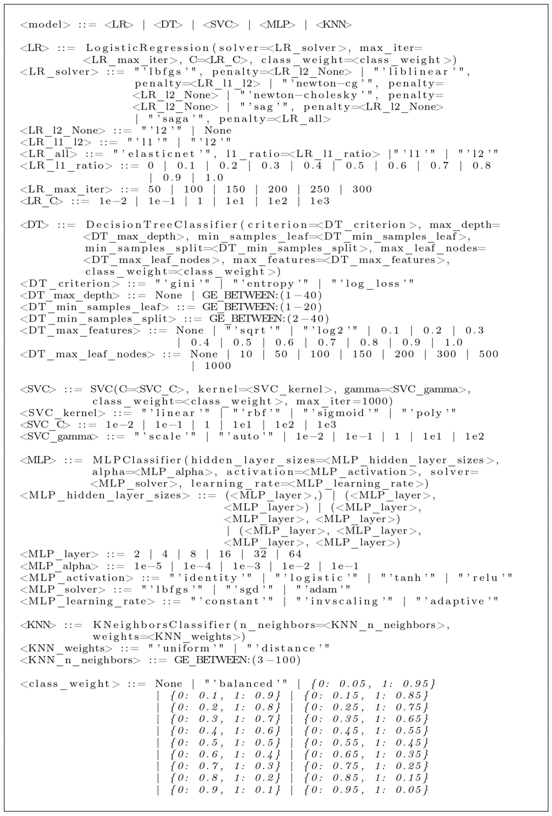
We define Y as the ground truth outcome for binary classification tasks, $\hat{Y}$ as the classifier's predicted outcome, S as the sensitive attribute, and use a and b to represent two different groups. In our experiment, accuracy is the objective we aim to maximize, while the fairness metric (AOD or SPD) is the objective we seek to minimize.

3.2 Multi-Objective Grammatical Evolution

In this paper, we employ grammatical evolution (GE) [9], a genetic programming algorithm, which uses context-free grammar to describe the search space of model configurations. We use MOGE with NSGA-II search operator to solve multi-objective CASH problem for accuracy and fairness. We selected five widely used machine learning models: Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), and Support Vector Classifier (SVC). The grammar for model configurations search space is shown in Fig. 1.

4 Experiment

We conducted experiments on three public real-world binary classification datasets from three different domains: (1)**Adult - Census Income**; (2) **COMPAS - Correctional Offender Management Profiling for Alternative Sanctions**; and (3) **Heart Disease Indicators**. All datasets were preprocessed by normalizing numerical features and encoding categorical variables, then split into a training set (60%), a validation set (20%), and a testing set (20%) using stratified sampling to maintain class and sensitive attribute ('Sex') distributions.



```
<model> ::= <LR> | <DT> | <SVC> | <MLP> | <KNN>

<LR> ::= LogisticRegression(solver=<LR_solver>, max_iter=
         <LR_max_iter>, C=<LR_C>, class_weight=<class_weight>)
<LR_solver> ::= "'lbfgs'", penalty=<LR_l2_None> | "'liblinear'",
                penalty=<LR_l1_l2> | "'newton-cg'", penalty=
                <LR_l2_None> | "'newton-cholesky'", penalty=
                <LR_l2_None> | "'sag'", penalty=<LR_l2_None>
                | "'saga'", penalty=<LR_all>
<LR_l2_None> ::= "'l2'" | None
<LR_l1_l2> ::= "'l1'" | "'l2'"
<LR_all> ::= "'elasticnet'", l1_ratio=<LR_l1_ratio> |"'l1'" | "'l2'"
<LR_l1_ratio> ::= 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8
                  | 0.9 | 1.0
<LR_max_iter> ::= 50 | 100 | 150 | 200 | 250 | 300
<LR_C> ::= 1e-2 | 1e-1 | 1 | 1e1 | 1e2 | 1e3

<DT> ::= DecisionTreeClassifier(criterion=<DT_criterion>, max_depth=
         <DT_max_depth>, min_samples_leaf=<DT_min_samples_leaf>,
         min_samples_split=<DT_min_samples_split>, max_leaf_nodes=
         <DT_max_leaf_nodes>, max_features=<DT_max_features>,
         class_weight=<class_weight>)
<DT_criterion> ::= "'gini'" | "'entropy'" | "'log_loss'"
<DT_max_depth> ::= None | GE_BETWEEN:(1-40)
<DT_min_samples_leaf> ::= GE_BETWEEN:(1-20)
<DT_min_samples_split> ::= GE_BETWEEN:(2-40)
<DT_max_features> ::= None | "'sqrt'" | "'log2'" | 0.1 | 0.2 | 0.3
                      | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0
<DT_max_leaf_nodes> ::= None | 10 | 50 | 100 | 150 | 200 | 300 | 500
                        | 1000

<SVC> ::= SVC(C=<SVC_C>, kernel=<SVC_kernel>, gamma=<SVC_gamma>,
          class_weight=<class_weight>, max_iter=1000)
<SVC_kernel> ::= "'linear'" | "'rbf'" | "'sigmoid'" | "'poly'"
<SVC_C> ::= 1e-2 | 1e-1 | 1 | 1e1 | 1e2 | 1e3
<SVC_gamma> ::= "'scale'" | "'auto'" | 1e-2 | 1e-1 | 1 | 1e1 | 1e2

<MLP> ::= MLPClassifier(hidden_layer_sizes=<MLP_hidden_layer_sizes>,
          alpha=<MLP_alpha>, activation=<MLP_activation>, solver=
          <MLP_solver>, learning_rate=<MLP_learning_rate>)
<MLP_hidden_layer_sizes> ::= (<MLP_layer>,) | (<MLP_layer>,
                             <MLP_layer>) | (<MLP_layer>,
                             <MLP_layer>, <MLP_layer>)
                             | (<MLP_layer>, <MLP_layer>,
                             <MLP_layer>, <MLP_layer>)
<MLP_layer> ::= 2 | 4 | 8 | 16 | 32 | 64
<MLP_alpha> ::= 1e-5 | 1e-4 | 1e-3 | 1e-2 | 1e-1
<MLP_activation> ::= "'identity'" | "'logistic'" | "'tanh'" | "'relu'"
<MLP_solver> ::= "'lbfgs'" | "'sgd'" | "'adam'"
<MLP_learning_rate> ::= "'constant'" | "'invscaling'" | "'adaptive'"

<KNN> ::= KNeighborsClassifier(n_neighbors=<KNN_n_neighbors>,
          weights=<KNN_weights>)
<KNN_weights> ::= "'uniform'" | "'distance'"
<KNN_n_neighbors> ::= GE_BETWEEN:(3-100)

<class_weight> ::= None | "'balanced'" | {0: 0.05, 1: 0.95}
                   | {0: 0.1, 1: 0.9} | {0: 0.15, 1: 0.85}
                   | {0: 0.2, 1: 0.8} | {0: 0.25, 1: 0.75}
                   | {0: 0.3, 1: 0.7} | {0: 0.35, 1: 0.65}
                   | {0: 0.4, 1: 0.6} | {0: 0.45, 1: 0.55}
                   | {0: 0.5, 1: 0.5} | {0: 0.55, 1: 0.45}
                   | {0: 0.6, 1: 0.4} | {0: 0.65, 1: 0.35}
                   | {0: 0.7, 1: 0.3} | {0: 0.75, 1: 0.25}
                   | {0: 0.8, 1: 0.2} | {0: 0.85, 1: 0.15}
                   | {0: 0.9, 1: 0.1} | {0: 0.95, 1: 0.05}
```

Figure 1: The BNF grammar adopted in the study. showing the high-level architecture that transitions from algorithm selection to hyperparameter optimization across multiple layers of the grammar.

To explore the effectiveness of MOGE in balancing model accuracy and fairness, we compare MOGE with five other baselines: (1). Model performance of candidate models (LR, DT, KNN, MLP, SVC) with default hyperparameters (Default); (2). Model performance when optimizing accuracy alone using a single-objective grammatical evolution (SOGE) (Acc Only); (3). Model performance when optimizing fairness alone using a single-objective grammatical evolution (SOGE) (SPD Only or AOD Only); (4). Model performance when using random search; (5)Exponentiated Gradient Reduction [1], a classical in-processing fairness method which returns the best classifier under certain fairness constraints implemented via the AI Fairness 360 toolkit [2]. Each setup was repeated 30 times using 30 different seeds for each dataset. We conducted experiments using the PonyGE2 library (version 0.2.0) [5]. The hyperparameters we have modified for PonyGE2 are presented in Table 1, while all other hyperparameters remain at their default settings.

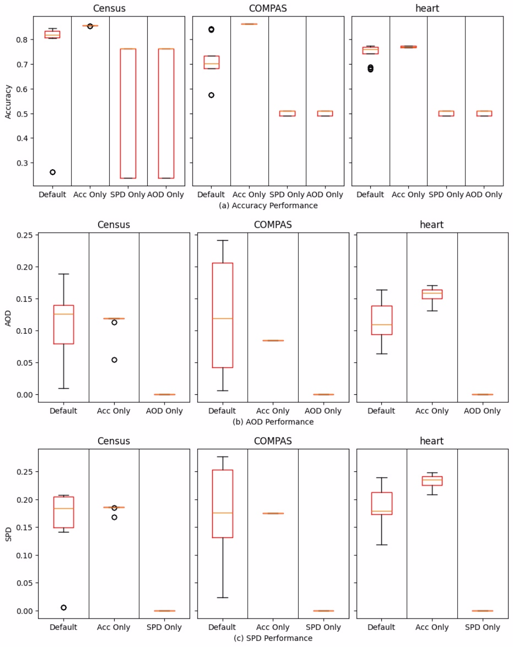
Table 1: Hyperparameter for Grammatical Evolution

Hyperparameter	Value
Generations	250
Population	250
Crossover probability	0.75
Mutation probability	0.1

5 Results

5.1 Limitations of Default Models and SOGE

Figure 2 shows the results of default models and SOGE. We found that the default models perform very differently, and their performance also varies across different datasets. Meanwhile, SOGE outperforms default models in achieving its target (whether accuracy or fairness). However, Accuracy-focused SOGE often overlooks fairness and sometimes even results in a significant reduction compared default models. Notably, we observed that when optimizing for fairness, regardless of which fairness metric is chosen, SOGE tends to produce an oversimplified model that classifies all instances as positive or negative (an all-1 or all-0 classifier) to achieve absolute fairness (i.e., AOD or SPD equal to zero). This finding underscores the importance of jointly considering both accuracy and fairness during the optimization process. Interestingly, we observed that the models that achieved optimal fairness or accuracy varied across the three datasets. This interplay highlights the limitations of single-model HPO and underscores the necessity of addressing the CASH problem.



(a) Accuracy Performance

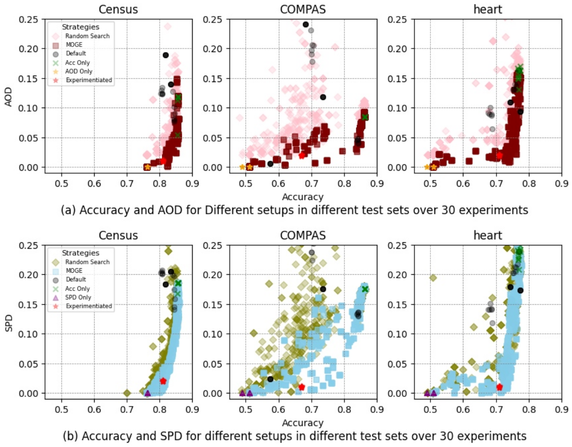
(b) AOD Performance

(c) SPD Performance

Figure 2: Accuracy, AOD, and SPD for different test datasets over 30 experiments for default models and SOGE setups. Note that lower fairness metric values are more optimal, while higher values of accuracy are more desirable.

5.2 Comparison of MOGE with Other Baselines

Compared to single-objective setups, MOGE can effectively consider and balance both objectives simultaneously. Table 2, 3, and Fig. 3 show the results of all setups. We find that the solution with the highest accuracy in MOGE is similar to the results obtained from SOGE focused on accuracy. When comparing MOGE to SOGE focused on fairness, we observe that MOGE tends to select oversimplified classifier with higher accuracy preferentially. However, MOGE generates a spectrum of solutions spanning the Pareto frontier between single-objective extremes, offering users a repository of trade-offs to balance fairness and accuracy for their specific context. Interestingly, we observed that all Pareto fronts included at least three different algorithm types. This mix highlights why multi-objective CASH matters: sticking to one model type risks missing better performance, while letting models "compete" across types naturally uncovers richer trade-offs.



(a) Accuracy and AOD for Different setups in different test sets over 30 experiments

(b) Accuracy and SPD for different setups in different test sets over 30 experiments

Figure 3: Accuracy, SPD, and AOD for different setups in different test datasets over 30 experiments. Note that lower fairness metric values are more optimal, while higher values of accuracy are more desirable.

While random search can also produce a set of solutions, its results tend to be scattered throughout the search space and often lie further from the ideal point (i.e., accuracy of 1 and fairness of 0). Consequently, random search achieves lower average performance than MOGE, highlighting MOGE's superior effectiveness. Compared to Exponentiated Gradient Reduction, although it offers a substantial fairness improvement over the default model, we were pleasantly surprised to find that MOGE still identified multiple solutions that are both more accurate and fairer. This outcome highlight the competitive potential of the MOGE. We can select the appropriate models from a wide range of options that balance accuracy and fairness based on specific circumstances.

6 Conclusion and Future Work

In this paper, we implemented MOGE to investigate how the multi-objective CASH affects model accuracy and fairness. By comparing the results of SOGE and default models, we found that while

Table 2: Mean and standard deviation of accuracy and AOD for different setups across different test datasets over 30 experiments. Note that lower fairness metric values are more optimal, while higher values of accuracy are more desirable.

Dataset	**Setups**	**Test Accuracy**	**Test AOD**
Census	Default	0.71 ± 0.23	0.11 ±0.06
	Acc Only	0.85 ± 0.01	0.12 ± 0.02
	AOD Only	0.61 ± 0.25	0.00 ± 0.00
	MOGE	0.79 ± 0.04	0.02 ± 0.04
	Random Search	0.75 ± 0.15	0.13 ± 0.10
	Exponentiated	0.81 ± 0.01	0.01 ± 0.01
COMPAS	Default	0.71 ± 0.09	0.12 ± 0.09
	Acc Only	0.86 ± 0.00	0.08 ± 0.00
	AOD Only	0.51 ± 0.04	0.00 ± 0.00
	MOGE	0.77 ± 0.14	0.06 ± 0.03
	Random Search	0.64 ± 0.11	0.12 ± 0.09
	Exponentiated	0.67 ± 0.01	0.02 ± 0.01
Heart	Default	0.75 ± 0.03	0.12 ± 0.03
	Acc Only	0.77 ± 0.01	0.16 ± 0.01
	AOD Only	0.50 ± 0.01	0.00 ± 0.00
	MOGE	0.71 ± 0.09	0.06 ± 0.05
	Random Search	0.67 ± 0.11	0.09 ± 0.06
	Exponentiated	0.71 ± 0.01	0.02 ± 0.01

Table 3: Mean and standard deviation of accuracy and SPD for different setups across different test datasets over 30 experiments. Note that lower fairness metric values are more optimal, while higher values of accuracy are more desirable.

Dataset	**Setups**	**Test Accuracy**	**Test SPD**
Census	Default	0.71 ± 0.23	0.15 ± 0.08
	Acc Only	0.85 ± 0.01	0.18 ± 0.01
	AOD Only	0.60 ± 0.26	0.03 ± 0.08
	MOGE	0.83 ± 0.03	0.08 ± 0.05
	Random Search	0.75 ± 0.15	0.15 ± 0.11
	Exponentiated	0.82 ± 0.01	0.02 ± 0.01
COMPAS	Default	0.71 ± 0.09	0.17 ± 0.09
	Acc Only	0.86 ± 0.00	0.18 ± 0.00
	AOD Only	0.50 ± 0.01	0.00 ± 0.00
	MOGE	0.75 ± 0.14	0.11 ± 0.07
	Random Search	0.64 ± 0.11	0.14 ± 0.10
	Exponentiated	0.66 ± 0.01	0.01 ± 0.01
Heart	Default	0.75 ± 0.03	0.19 ± 0.04
	Acc Only	0.77 ± 0.01	0.23 ± 0.01
	AOD Only	0.50 ± 0.01	0.00 ± 0.00
	MOGE	0.71 ± 0.08	0.10 ± 0.08
	Random Search	0.67 ± 0.11	0.13 ± 0.08
	Exponentiated	0.72 ± 0.01	0.01 ± 0.01

SOGE outperforms default models in achieving its specific objectives (accuracy or fairness), it neglects the other. In contrast, MOGE generates a set of Pareto-optimal solutions that balance both accuracy and fairness, incorporating at least three different types of models within the Pareto front. Furthermore, MOGE outperforms random search and Exponentiated Gradient Reduction by producing solutions with higher accuracy and fairness, thereby proving its superior effectiveness. Future work will expand to more datasets and tasks, explore richer model configurations, and investigate alternative search operators beyond NSGA-II to further enhance fairness-accuracy trade-offs.

Acknowledgments

This publication has emanated from research conducted with the financial support of Taighde Éireann - Research Ireland under Grant number 21/FFP-A/9255.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International conference on machine learning*. PMLR, 60–69.

[2] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. 2019. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development* 63, 4/5 (2019), 4–1.

[3] Antonio Candelieri, Andrea Ponti, and Francesco Archetti. 2024. Fair and green hyperparameter optimization via multi-objective and multiple information source Bayesian optimization. *Machine Learning* 113, 5 (2024), 2701–2731.

[4] André F Cruz, Pedro Saleiro, Catarina Belém, Carlos Soares, and Pedro Bizarro. 2021. Promoting fairness through hyperparameter optimization. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1036–1041.

[5] Michael Fenton, James McDermott, David Fagan, Stefan Forstenlechner, Erik Hemberg, and Michael O'Neill. 2017. Ponyge2: Grammatical evolution in python. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1194–1201.

[6] Florian Karl, Tobias Pielok, Julia Moosbauer, Florian Pfisterer, Stefan Coors, Martin Binder, Lennart Schneider, Janek Thomas, Jakob Richter, Michel Lang, et al. 2023. Multi-objective hyperparameter optimization in machine learning—An overview. *ACM Transactions on Evolutionary Learning and Optimization* 3, 4 (2023), 1–50.

[7] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)* 54, 6 (2021), 1–35.

[8] Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. 2016. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016* (Denver, Colorado, USA) *(GECCO '16)*. ACM, New York, NY, USA, 485–492. doi:10.1145/2908812.2908918

[9] Michael O'Neill and Conor Ryan. 2001. Grammatical evolution. *IEEE Transactions on Evolutionary Computation* 5, 4 (2001), 349–358.

[10] Valerio Perrone, Michele Donini, Muhammad Bilal Zafar, Robin Schmucker, Krishnaram Kenthapadi, and Cédric Archambeau. 2021. Fair bayesian optimization. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 854–863.

[11] Robin Schmucker, Michele Donini, Valerio Perrone, and Cédric Archambeau. 2020. Multi-objective multi-fidelity hyperparameter optimization with application to fairness. (2020).

[12] Saeid Tizpaz-Niari, Ashish Kumar, Gang Tan, and Ashutosh Trivedi. 2022. Fairness-aware configuration of machine learning libraries. In *Proceedings of the 44th International Conference on Software Engineering*. 909–920.

[13] Ana Valdivia, Javier Sánchez-Monedero, and Jorge Casillas. 2021. How fair can we go in machine learning? Assessing the boundaries of accuracy and fairness. *International Journal of Intelligent Systems* 36, 4 (2021), 1619–1643.

[14] David Villar and Jorge Casillas. 2021. Facing many objectives for fairness in machine learning. In *International Conference on the Quality of Information and Communications Technology*. Springer, 373–386.

[15] Hilde Weerts, Florian Pfisterer, Matthias Feurer, Katharina Eggensperger, Edward Bergman, Noor Awad, Joaquin Vanschoren, Mykola Pechenizkiy, Bernd Bischl, and Frank Hutter. 2024. Can fairness be automated? Guidelines and opportunities for fairness-aware AutoML. *Journal of Artificial Intelligence Research* 79 (2024), 639–677.