# Leveraging Integer Linear Programming to Learn Optimal Fair Rule Lists

Ulrich Aïvodji[1], Julien Ferry[2*], Sébastien Gambs[3], Marie-José Huguet[2], and Mohamed Siala[2]

[1] École de Technologie Supérieure, Montréal, Canada
[2] LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
jferry@laas.fr
[3] Université du Québec à Montréal, Montréal, Canada

**Abstract.** Fairness and interpretability are fundamental requirements for the development of responsible machine learning. However, learning optimal interpretable models under fairness constraints has been identified as a major challenge. In this paper, we investigate and improve on a state-of-the-art exact learning algorithm, called `CORELS`, which learns rule lists that are certifiably optimal in terms of accuracy and sparsity. Statistical fairness metrics have been integrated incrementally into `CORELS` in the literature. This paper demonstrates the limitations of such an approach for exploring the search space efficiently before proposing an Integer Linear Programming method, leveraging accuracy, sparsity and fairness jointly for better pruning. Our thorough experiments show clear benefits of our approach regarding the exploration of the search space.

**Keywords:** Fairness · Interpretability · Rule Lists · Machine Learning

## 1 Introduction

The combination of the availability of large datasets as well as algorithmic and computational progress has led to a significant increase in the performance of machine learning models. Despite their usefulness for numerous applications, the use of such models also raises several issues when their outcome impacts individuals' lives (*e.g.*, credit scoring or scholarships granting). Fairness and interpretability are key properties for the development of trustworthy machine learning and have become legal requirements defined in legislative texts [14].

The interpretability of a machine learning model is defined in [10] as "the ability to explain or to present in understandable terms to a human". This definition is quite general, and its precise instantiation depends on the task at hand, the context considered and the target of the explanation. Several methods have been proposed to explain machine learning models' predictions, which can be categorized into two main families. On one side, *black-box explanations* [15] can be useful in non-sensitive contexts to provide *a posteriori* explanations of

[*] First author.

a black-box, but can be manipulated [25]. On the other side, *transparent-box design* [22] aims at building inherently interpretable models (*e.g.*, rule-based or tree-based models of reasonable size) [13, 22].

Fairness is a central requirement for high-stake decision systems. Indeed, learning algorithms try to extract useful correlations from the training data but real-world datasets may include negative biases that should not be captured (*e.g.*, historical discrimination). Several fairness notions have been proposed to address this issue [6, 7, 27, 20]. Among them, statistical fairness metrics ensure that a given statistical measure has similar values between groups as determined by the value of a sensitive feature. They are widely used as they can implement legal requirements and are easily quantifiable. Several approaches to fair learning have emerged in the literature, categorized into three main families. *Preprocessing* techniques [21] directly modify the training data to remove undesirable correlations so that any classifier trained on this data does not learn such correlations. *Postprocessing* approaches [16] modify the outputs of a previously trained classifier to meet some fairness criteria. Finally, *algorithmic modification* techniques [28] directly incorporate the fairness requirements into the learning algorithm and output a model satisfying a given fairness definition. In this paper, we focus on statistical fairness metrics using algorithmic modification approaches, which usually offer the best trade-offs between accuracy and fairness [6].

While many heuristic approaches for learning have been proposed, exact approaches offer a considerable advantage as a lack of optimality can have societal implications [4]. For instance, `CORELS` [3, 4] produces rule lists that are certifiably optimal in terms of accuracy and sparsity. It relies on a branch-and-bound algorithm leveraging several dedicated bounds to prune the search space efficiently. `FairCORELS` [1, 2] is a bi-objective extension of `CORELS` handling both statistical fairness and accuracy. `FairCORELS` consists in an $\epsilon$-constraint method that leverages `CORELS`' original search tree and bounds for the accuracy objective and considers the fairness objective as a constraint. However, handling such constraints modifies the set of acceptable solutions, which makes the exploration considerably harder. Indeed, learning optimal interpretable machine learning models under constraints (*e.g.,* fairness constraints) has been identified as one of the main technical challenges towards interpretable machine learning [26].

In this paper, we address this issue and propose a method that harnesses the fairness constraints to efficiently prune the search space and optionally guide exploration. More precisely, we argue that `CORELS`' original bounds are not sufficient to efficiently explore the search space in this bi-objective setup. To address this, we design Integer Linear Programming (ILP) models combining both accuracy and fairness requirements for well-known statistical fairness metrics. These models are incorporated into `FairCORELS` through effective pruning mechanisms and can also be used to guide the exploration towards fair and accurate rule lists. Our large experimental study using three datasets with various fairness measures and requirements demonstrates clear benefits of the proposed approaches in terms of search exploration, memory consumption and learning quality.

The outline of the paper is as follows. First, we provide the relevant background and notations in Section 2. Then in Section 3, after describing the fair learning algorithm used, we discuss the theoretical claims motivating the necessity of efficient pruning. Afterwards, in Section 4, we propose pruning approaches based on ILP models, before evaluating empirically their efficiency and quality in Section 5 through a large experimental study. Finally, we conclude in Section 6.

## 2 Technical Background & Notations

In this section, we introduce the necessary background as well as the different notations used throughout the paper.

### 2.1 Rule Lists & Associated Notations

In supervised machine learning, the purpose of a classification problem is to learn a classifier function that maps as accurately as possible an input space to an output space. We use $\mathcal{F} = \{f_1, \dots, f_G\}$ to denote a set of $G$ binary features, all of them take their value in $\{0, 1\}$. The training data, denoted by $\mathcal{E} = \{e_1, \dots, e_M\}$, is a set of $M$ examples. The examples in $\mathcal{E}$ are partitioned into $\mathcal{E}^+$ and $\mathcal{E}^-$, which correspond respectively to positive examples and negative ones. Precisely, an example $e_j \in \mathcal{E}$ is represented as a 2-tuple $(x_j, y_j)$, in which $x_j \in \{0, 1\}^G$ denotes the value vector for all binary features associated with the example and $y_j \in \{0, 1\}$ is the label indicating its class. We have $e_j \in \mathcal{E}^+$ if $y_j = 1$ and $e_j \in \mathcal{E}^-$ if $y_j = 0$.

We consider classifiers that are expressed as *rule lists* [24], which are formed by an ordered list of *if-then* rules, followed by a default prediction. More precisely, a *rule list* is a tuple $d = (\delta_d, q_0)$ in which $\delta_d = (r_1, r_2, \dots, r_k)$ is $d$'s *prefix*, and $q_0 \in \{0, 1\}$ is a *default prediction*. A prefix is an ordered list of $k$ distinct association rules $r_i = a_i \to q_i$. Each rule $r_i$ is composed of an *antecedent* $a_i$ and a *consequent* $q_i \in \{0, 1\}$. Each antecedent $a_i$ is a Boolean assertion over $\mathcal{F}$ evaluating either to true or false for each possible input $x \in \{0, 1\}^G$. If $a_i$ evaluates to true for example $e_j$, we say that rule $r_i$ *captures* $e_j$. Similarly, if at least one of the rules in $\delta_d$ captures $e_j$, we say that prefix $\delta_d$ captures example $e_j$. Rule list 1.1 predicts whether a given individual has a [low] or [high] salary. Its prefix is composed of five rules, and its default decision is [low].

**Rule list 1.1.** Example rule list found by `FairCORELS` on the Adult Income dataset.

```
if [occupation:Blue-Collar] then [low]
else if [occupation:Service] then [low]
else if [capital gain: > 0] then [high]
else if [not(workclass:Government)] then [low]
else if [education:Masters/Doctorate] then [high]
else [low]
```

Using a rule list $d = (\delta_d, q_0)$ to classify an example $e$ is straightforward as rules in $\delta_d$ are applied sequentially. If $e$ is not captured by prefix $\delta_d$, then the default prediction $q_0$ is returned. Finally, remark that rule list $((), q_0)$ is well defined, and simply consists of a default prediction (hence representing a constant classifier).

**Table 1.** Summary of four statistical fairness metrics widely used in the literature.

| Metric | Statistical Measure | Mathematical Formulation |
|---|---|---|
| Statistical Parity (SP) | Probability of Positive Prediction | $\left\lvert \frac{TP^c_{\mathcal{E},p} + FP^c_{\mathcal{E},p}}{\lvert\mathcal{E}^p\rvert} - \frac{TP^c_{\mathcal{E},u} + FP^c_{\mathcal{E},u}}{\lvert\mathcal{E}^u\rvert} \right\rvert \leq \epsilon$ |
| Predictive Equality (PE) | False Positive Rate | $\left\lvert \frac{FP^c_{\mathcal{E},p}}{\lvert\mathcal{E}^p \cap \mathcal{E}^-\rvert} - \frac{FP^c_{\mathcal{E},u}}{\lvert\mathcal{E}^u \cap \mathcal{E}^-\rvert} \right\rvert \leq \epsilon$ |
| Equal Opportunity (EOpp) | False Negative Rate | $\left\lvert \frac{FN^c_{\mathcal{E},p}}{\lvert\mathcal{E}^p \cap \mathcal{E}^+\rvert} - \frac{FN^c_{\mathcal{E},u}}{\lvert\mathcal{E}^u \cap \mathcal{E}^+\rvert} \right\rvert \leq \epsilon$ |
| Equalized Odds (EO) | PE and EOpp | Conjunction of PE and EOpp |

### 2.2 Statistical Fairness

The rationale of statistical fairness notions is to ensure that a given statistical measure has similar values between several protected groups, defined by the value(s) of some sensitive feature(s) of $\mathcal{F}$. The underlying principle is that such sensitive features (*e.g.*, race, gender, ...) should not influence predictions. While the exact formulation of such metrics would enforce equality for the given measure over the protected groups, a common relaxation consists of bounding the difference. Depending on the particular value being equalized across groups, several metrics have been proposed in the literature. In this paper, we consider the four most commonly used metrics: Statistical Parity [12] (SP), Predictive Equality [9] (PE), Equal Opportunity [16] (EOpp) and Equalized Odds [16] (EO).

Let $\mathcal{E}$ denote a training set and $c$ a classifier. Throughout the paper, we assume that $\mathcal{E}$ is partitioned into two groups: a protected group $\mathcal{E}^p$ and an unprotected group $\mathcal{E}^u$ (this partition depends on the value of the sensitive feature(s)). Let also $\epsilon \in [0,1]$ denote the unfairness tolerance (*i.e.*, the maximum acceptable value for the unfairness measure). Thus, the fairness requirement gets harder as $\epsilon$ gets smaller. For a classifier $c$, among a group $\mathcal{E}^h$, with $h \in \{p,u\}$, we denote by $TP^c_{\mathcal{E},h}$ the number of true positives, $TN^c_{\mathcal{E},h}$ the number of true negatives, $FP^c_{\mathcal{E},h}$ the number of false positives and $FN^c_{\mathcal{E},h}$ the number of false negatives. Table 1 gives the definition of the four metrics considered.

## 3 CORELS & FairCORELS

`CORELS` [4] is a state-of-the-art supervised learning algorithm that outputs a certifiably optimal rule list minimizing the following objective function on a given training dataset $\mathcal{E}$:

$$\mathsf{obj}(d,\mathcal{E}) = \mathsf{misc}(d,\mathcal{E}) + \lambda \cdot K_d, \tag{1}$$

in which $\mathsf{misc}(d,\mathcal{E}) \in [0,1]$ denotes the training classification error of the rule list $d$, $K_d$ is the length of $d$ (*i.e.*, number of association rules in $d$) and $\lambda$ is a regularization hyper-parameter for sparsity. `CORELS` is a branch-and-bound algorithm, representing the search space of rule lists $\mathcal{R}$ as a prefix tree. Each node is a prefix in this tree, and each child node is an extension of its parent, obtained by adding exactly one rule at the end of the parent's prefix. Finally, the root node corresponds to the empty prefix. Each node is a possible solution (*i.e.*, rule list), obtained by adding a default decision (based on majority prediction) to the prefix associated with this node. While this search space corresponds to an exhaustive enumeration of the candidate solutions, CORELS leverages several bounds to prune it efficiently. Thanks to these bounds, along with several smart data structures, CORELS is able to find optimal solutions with a reasonable amount of time and memory. The set of antecedents $A$ is pre-mined and given as input to the algorithm. While CORELS is agnostic to the rule mining procedure used as preprocessing, an overview of existing techniques can be found in [8].

FairCORELS [1, 2] is a bi-objective extension of CORELS jointly addressing accuracy and statistical fairness, integrating several metrics from the literature. Formally, given a statistical fairness notion, whose violation by a rule list $d$ on dataset $\mathcal{E}$ is quantified by an unfairness function $\mathsf{unf}(d, \mathcal{E})$ and a maximum acceptable violation $\epsilon$, FairCORELS solves the following optimization problem:

$$\begin{aligned} \underset{d \in \mathcal{R}}{\arg\min} \quad & \mathsf{obj}(d, \mathcal{E}) \\ \text{such that} \quad & \mathsf{unf}(d, \mathcal{E}) \leq \epsilon \end{aligned} \tag{2}$$

FairCORELS is presented in Algorithm 1. In this algorithm, $d^c$ denotes the current best solution and $z^c$ is its objective value. Moreover, a priority queue $Q$ of prefixes is used to store its exploration frontier. The priority queue ordering defines the exploration heuristic. The function $\mathsf{b}(\delta, \mathcal{E})$ (coming from the CORELS algorithm) gives an objective lower bound for any rule list built upon prefix $\delta$ on the dataset $\mathcal{E}$. At each iteration of the main loop, a prefix $\delta$ is removed from the priority queue (Line 4). When the lower bound of $\delta$ is less than the current best objective value (Line 5), two operations are considered. First, the rule list $d$ formed by prefix $\delta$ along with a default prediction is accepted as a new best solution if it improves the current best objective value while respecting the unfairness tolerance (Line 9). Second, extensions of $\delta$ using the antecedents not involved in $\delta$'s rules are added to the queue (Line 12).

The constrained optimization formulation of the fair learning problem used in FairCORELS allows for the construction of different trade-offs between accuracy and fairness using a simple $\epsilon$-constraint method [23]. However, the fairness constraints modify the set of acceptable solutions and the resulting search space is considerably harder to work with. Indeed, CORELS' original bounds are less efficient as the fairness constraint gets stronger. In addition, some data structures used by CORELS to speed up the exploration are no longer usable. For instance, a prefix permutation map that reduces considerably the running time and the memory consumption [3, 4] does not apply anymore. This symmetry-aware map ensures that only the best permutation of each set of rules containing the same antecedents is kept. However, it cannot be used within FairCORELS without sacrificing optimality. Indeed, a given permutation may allow for better objective function values than others but may not lead to solutions meeting the fairness

**Algorithm 1** `FairCORELS`

**Input**: Training data $\mathcal{E}$ with set of pre-mined antecedents $A$; unfairness tolerance $\epsilon$; initial best known rule list $d^0$ such that $\mathsf{unf}(d^0, \mathcal{E}) \leq \epsilon$

**Output**: $(d^*, z^*)$ in which $d^*$ is a rule list with the minimum objective function value $z^*$ such that $\mathsf{unf}(d^*, \mathcal{E}) \leq \epsilon$

```
$(d^c, z^c) \leftarrow (d^0, \mathsf{obj}(d^0, \mathcal{E}))$
$Q \leftarrow queue(())$    ▷ Initially the queue contains the empty prefix ()
while $Q$ not empty do    ▷ Stop when the queue is empty
    $\delta \leftarrow Q.pop()$
    if $\mathsf{b}(\delta, \mathcal{E}) < z^c$ then
        $d \leftarrow (\delta, q_0)$    ▷ Set default prediction $q_0$ to minimize training error
        $z \leftarrow \mathsf{obj}(d, \mathcal{E})$
        if $z < z^c$ and $\mathsf{unf}(d, \mathcal{E}) \leq \epsilon$ then
            $(d^c, z^c) \leftarrow (d, z)$    ▷ Update best rule list and objective
        for $a$ in $A \setminus \{a_i \mid \exists r_i \in \delta, r_i = a_i \rightarrow q_i\}$ do ▷ Antecedent $a$ not involved in $\delta$
            $r \leftarrow (a \rightarrow q)$    ▷ Set $a$'s consequent $q$ to minimize training error
            $Q.push(\delta \cup r)$    ▷ Enqueue extension of $\delta$ with $r$
$(d^*, z^*) \leftarrow (d^c, z^c)$
```

requirement. In this situation, one could miss solutions that exhibit lower objective function values and meet the fairness requirement. Since we are interested in preserving the guarantee of optimality, we cannot use such a data structure. However, we note that a weaker permutation map can be designed and used without losing the guarantee of optimality (we precisely do that later in Section 5.3). Overall, both observations motivate the need for a new pruning approach, leveraging both the objective function value and the fairness constraint to efficiently explore `FairCORELS`' search space.

## 4 The Proposed Pruning Approach

This section presents our proposition to prune the search space by reasoning about the number of well-classified examples and fairness. The main idea is to discard prefixes that cannot improve the current objective while satisfying the fairness requirement before being treated. To realize this, one has to guarantee that for any prefix discarded, none of its extensions can satisfy both requirements, which is the purpose of Section 4.1. Afterwards, Section 4.2 exploits this property in the presentation of our proposition.

### 4.1 A Sufficient Condition to Reject Prefixes

Let $\mathcal{E}$ be a training set and $d$ be a rule list. We use $W^d_{\mathcal{E}}$ to denote the number of examples of dataset $\mathcal{E}$ well classified by $d$:

$$\begin{aligned} W^d_{\mathcal{E}} &= TP^d_{\mathcal{E},p} + TP^d_{\mathcal{E},u} + TN^d_{\mathcal{E},p} + TN^d_{\mathcal{E},u} && (3)\\ &= TP^d_{\mathcal{E},p} + TP^d_{\mathcal{E},u} + |\mathcal{E}^p \cap \mathcal{E}^-| - FP^d_{\mathcal{E},p} + |\mathcal{E}^u \cap \mathcal{E}^-| - FP^d_{\mathcal{E},u} && (4)\end{aligned}$$

We slightly extend the notation introduced in Section 2. For a prefix $\delta$, among a group $\mathcal{E}^h$ with $h \in \{p, u\}$, we denote by $TP^{\delta}_{\mathcal{E},h}$ (respectively $TN^{\delta}_{\mathcal{E},h}$, $FP^{\delta}_{\mathcal{E},h}$ and $FN^{\delta}_{\mathcal{E},h}$) the number of true positives (respectively true negatives, false positives and false negatives) among the examples of $\mathcal{E}$ captured by $\delta$. Similarly, we define $W^{\delta}_{\mathcal{E}}$ as the number of examples well classified by $\delta$, among the examples of $\mathcal{E}$ that $\delta$ captures. Clearly, $W^{\delta}_{\mathcal{E}} = TP^{\delta}_{\mathcal{E},p} + TP^{\delta}_{\mathcal{E},u} + TN^{\delta}_{\mathcal{E},p} + TN^{\delta}_{\mathcal{E},u}$.

We define $\sigma(\delta)$ to be the set of all rule lists whose prefixes start with $\delta$: $\sigma(\delta) = \{(\delta_d, q_0) \mid \delta_d \text{ starts with } \delta\}$. Formally, we say that $\delta_d$ starts with $\delta$ (a prefix of length $K$) if and only if the $K$ first rules of $\delta_d$ are precisely those of $\delta$, appearing in the same order.

Consider $d = (\delta_d, q_0)$ such that $d \in \sigma(\delta)$. On the one hand, some examples of $\mathcal{E}$ cannot be captured by $\delta$. On the other hand, all examples of $\mathcal{E}$ captured by $\delta$ are captured by $\delta_d$ and have the same prediction as with $\delta$.

**Proposition 1.** *Given a prefix $\delta$, a rule list $d \in \sigma(\delta)$ and $h \in \{p, u\}$, we have:*

$$TP^{\delta}_{\mathcal{E},h} \leq TP^{d}_{\mathcal{E},h} \leq |\mathcal{E}^h \cap \mathcal{E}^+| - FN^{\delta}_{\mathcal{E},h}$$
$$FP^{\delta}_{\mathcal{E},h} \leq FP^{d}_{\mathcal{E},h} \leq |\mathcal{E}^h \cap \mathcal{E}^-| - TN^{\delta}_{\mathcal{E},h}$$

*Proof. The lower bounds are an immediate consequence of the fact that all examples captured by $\delta$ are captured by $d$'s prefix and have the same predictions that in $\delta$. Concerning the upper bounds, we show the proof for the first inequality as the second can be proven using a similar argument. Define $T$ as the set of examples in $\mathcal{E}^h \cap \mathcal{E}^+$ that are not determined by $\delta$. When constructing $d$ from $\delta$, the maximum possible augmentation of true positives within protected group $h$ is to predict all the examples correctly in $T$. The size of the set containing true positives of $\delta$ and $T$ is equal to $|\mathcal{E}^h \cap \mathcal{E}^+| - FN^{\delta}_{\mathcal{E},h}$. Hence the upper bound.* □

As a consequence of Proposition 1, $W^{d}_{\mathcal{E}} \geq W^{\delta}_{\mathcal{E}}$. We now define four integer decision variables that are used in our Integer Linear Programming (ILP) models. These variables are used to model the confusion matrix of any rule list whose prefix starts with $\delta$ as well as to define constraints modelling accuracy and fairness requirements over such matrix.

$$x^{TP_{\mathcal{E},p}} \in [TP^{\delta}_{\mathcal{E},p}, |\mathcal{E}^p \cap \mathcal{E}^+| - FN^{\delta}_{\mathcal{E},p}],\ x^{TP_{\mathcal{E},u}} \in [TP^{\delta}_{\mathcal{E},u}, |\mathcal{E}^u \cap \mathcal{E}^+| - FN^{\delta}_{\mathcal{E},u}],$$
$$x^{FP_{\mathcal{E},p}} \in [FP^{\delta}_{\mathcal{E},p}, |\mathcal{E}^p \cap \mathcal{E}^-| - TN^{\delta}_{\mathcal{E},p}],\ x^{FP_{\mathcal{E},u}} \in [FP^{\delta}_{\mathcal{E},u}, |\mathcal{E}^u \cap \mathcal{E}^-| - TN^{\delta}_{\mathcal{E},u}].$$

Consider the following constraint in which $L$ and $U$ are two integers such that $0 \leq L \leq U \leq |\mathcal{E}|$:

$$L \leq x^{TP_{\mathcal{E},p}} + x^{TP_{\mathcal{E},u}} + |\mathcal{E}^p \cap \mathcal{E}^-| - x^{FP_{\mathcal{E},p}} + |\mathcal{E}^u \cap \mathcal{E}^-| - x^{FP_{\mathcal{E},u}} \leq U. \quad (5)$$

We define $ILP(\delta, \mathcal{E}, L, U)$ to be the ILP model defined by the four variables $x^{TP_{\mathcal{E},p}}, x^{FP_{\mathcal{E},p}}, x^{TP_{\mathcal{E},u}}, x^{FP_{\mathcal{E},u}}$ and Constraint (5).

**Proposition 2.** *Given a prefix $\delta$ and $0 \leq L \leq U \leq |\mathcal{E}|$, if $ILP(\delta, \mathcal{E}, L, U)$ is unsatisfiable then we have:*

$$\nexists d \in \sigma(\delta) \mid L \leq W^{d}_{\mathcal{E}} \leq U$$

*Proof. Assume that there exists some $d \in \sigma(\delta)$ such that $L \leq W_{\mathcal{E}}^{d} \leq U$. Then, $x^{TP_{\mathcal{E},p}} = TP_{\mathcal{E},p}^{d}$, $x^{TP_{\mathcal{E},u}} = TP_{\mathcal{E},u}^{d}$, $x^{FP_{\mathcal{E},p}} = FP_{\mathcal{E},p}^{d}$ and $x^{FP_{\mathcal{E},u}} = FP_{\mathcal{E},u}^{d}$ is a solution to $ILP(\delta, \mathcal{E}, L, U)$. Indeed, Constraint (5) is satisfied by hypothesis, and the bounds of the four variables are respected due to Proposition 1 and the fact that $d$ is an extension of $\delta$. Finally, if $\exists d \in \sigma(\delta) \mid L \leq W_{\mathcal{E}}^{d} \leq U$, then $ILP(\delta, \mathcal{E}, L, U)$ is satisfiable, which completes the proof by contrapositive.* □

In the following paragraph, we show how the $ILP(\delta, \mathcal{E}, L, U)$ model can be extended to include the different considered statistical fairness metrics (defined in Table 1). For the sake of conciseness, we detail the procedure for the Statistical Parity metric and provide the key elements for the three other metrics. Note that propositions similar to Proposition 3 can be adapted and proved for the three other metrics, following the same reasoning.

**Integrating Statistical Parity.** We introduce a constant $C_1 = \epsilon \times |\mathcal{E}^p| \times |\mathcal{E}^u|$ and the following constraint:

$$-C_1 \leq |\mathcal{E}^u| \times (x^{TP_{\mathcal{E},p}} + x^{FP_{\mathcal{E},p}}) - |\mathcal{E}^p| \times (x^{TP_{\mathcal{E},u}} + x^{FP_{\mathcal{E},u}}) \leq C_1. \tag{6}$$

Let $ILP_{SP}(\delta, \mathcal{E}, L, U, \epsilon)$ be the Integer Linear Programming model defined by the four variables $x^{TP_{\mathcal{E},p}}, x^{FP_{\mathcal{E},p}}, x^{TP_{\mathcal{E},u}}, x^{FP_{\mathcal{E},u}}$ and Constraints (5) and (6).

**Proposition 3.** *Given a prefix $\delta$, an unfairness tolerance $\epsilon \in [0,1]$, and $0 \leq L \leq U \leq |\mathcal{E}|$, if $ILP_{SP}(\delta, \mathcal{E}, L, U, \epsilon)$ is unsatisfiable then we have:*

$$\nexists d \in \sigma(\delta) \mid L \leq W_{\mathcal{E}}^{d} \leq U \text{ and } \mathsf{unf}_{SP}(d, \mathcal{E}) \leq \epsilon$$

*Proof. Assume that there exists some $d \in \sigma(\delta)$ such that $L \leq W_{\mathcal{E}}^{d} \leq U$ and $\mathsf{unf}_{SP}(d, \mathcal{E}) \leq \epsilon$. First, observe that Constraint (6) is equivalent to the mathematical formulation of the Statistical Parity condition defined in Table 1. Indeed, $\mathsf{unf}_{SP}(d, \mathcal{E}) \leq \epsilon$ if and only if $-C_1 \leq |\mathcal{E}^u| \times (TP_{\mathcal{E},p}^{d} + FP_{\mathcal{E},p}^{d}) - |\mathcal{E}^p| \times (TP_{\mathcal{E},u}^{d} + FP_{\mathcal{E},u}^{d}) \leq C_1$. Then, $x^{TP_{\mathcal{E},p}} = TP_{\mathcal{E},p}^{d}$, $x^{TP_{\mathcal{E},u}} = TP_{\mathcal{E},u}^{d}$, $x^{FP_{\mathcal{E},p}} = FP_{\mathcal{E},p}^{d}$ and $x^{FP_{\mathcal{E},u}} = FP_{\mathcal{E},u}^{d}$ is a solution to $ILP_{SP}(\delta, \mathcal{E}, L, U, \epsilon)$. Finally, if $\exists d \in \sigma(\delta) \mid L \leq W_{\mathcal{E}}^{d} \leq U$ and $\mathsf{unf}_{SP}(d, \mathcal{E}) \leq \epsilon$, then $ILP_{SP}(\delta, \mathcal{E}, L, U, \epsilon)$ is satisfiable, which completes the proof by contrapositive.* □

**Integrating Other Statistical Fairness Metrics.** Consider a prefix $\delta$, an unfairness tolerance $\epsilon \in [0,1]$ and $0 \leq L \leq U \leq |\mathcal{E}|$. We define the following useful constants $C_2 = \epsilon \times |\mathcal{E}^u \cap \mathcal{E}^-| \times |\mathcal{E}^p \cap \mathcal{E}^-|$, and $C_3 = \epsilon \times |\mathcal{E}^p \cap \mathcal{E}^+| \times |\mathcal{E}^u \cap \mathcal{E}^+|$.

*Predictive Equality.* Consider the following constraint:

$$-C_2 \leq |\mathcal{E}^u \cap \mathcal{E}^-| \times x^{FP_{\mathcal{E},p}} - |\mathcal{E}^p \cap \mathcal{E}^-| \times x^{FP_{\mathcal{E},u}} \leq C_2. \tag{7}$$

Let $ILP_{PE}(\delta, \mathcal{E}, L, U, \epsilon)$ be the ILP model defined by the four variables $x^{TP_{\mathcal{E},p}}, x^{FP_{\mathcal{E},p}}, x^{TP_{\mathcal{E},u}}, x^{FP_{\mathcal{E},u}}$ and Constraints (5) and (7). If $ILP_{PE}(\delta, \mathcal{E}, L, U, \epsilon)$ is unsatisfiable, then: $\nexists d \in \sigma(\delta) \mid L \leq W_{\mathcal{E}}^{d} \leq U$ and $\mathsf{unf}_{PE}(d, \mathcal{E}) \leq \epsilon$.

*Equal Opportunity.* Consider the following constraint:

$$-C_3 \leq |\mathcal{E}^p \cap \mathcal{E}^+| \times x^{TP_{\mathcal{E},u}} - |\mathcal{E}^u \cap \mathcal{E}^+| \times x^{TP_{\mathcal{E},p}} \leq C_3. \tag{8}$$

Let $ILP_{EOpp}(\delta, \mathcal{E}, L, U, \epsilon)$ be the ILP model defined by the four variables $x^{TP_{\mathcal{E},p}}, x^{FP_{\mathcal{E},p}}, x^{TP_{\mathcal{E},u}}, x^{FP_{\mathcal{E},u}}$ and Constraints (5) and (8). If $ILP_{EOpp}(\delta, \mathcal{E}, L, U, \epsilon)$ is unsatisfiable, then: $\nexists d \in \sigma(\delta) \mid L \leq W^d_{\mathcal{E}} \leq U$ and $\mathsf{unf}_{EOpp}(d, \mathcal{E}) \leq \epsilon$.

*Equalized Odds.* Since the Equalized Odds metric is the conjunction of Equal Opportunity and Predictive Equality, we simply use the conjunction of Constraints (7) and (8) to integrate it.

Let $ILP_{EO}(\delta, \mathcal{E}, L, U, \epsilon)$ be the ILP model defined by the four variables $x^{TP_{\mathcal{E},p}}, x^{FP_{\mathcal{E},p}}, x^{TP_{\mathcal{E},u}}, x^{FP_{\mathcal{E},u}}$ and Constraints (5), (7) and (8). If $ILP_{EO}(\delta, \mathcal{E}, L, U, \epsilon)$ is unsatisfiable then: $\nexists d \in \sigma(\delta) \mid L \leq W^d_{\mathcal{E}} \leq U$ and $\mathsf{unf}_{EO}(d, \mathcal{E}) \leq \epsilon$.

### 4.2 Integration Within `FairCORELS`

We have proposed a sufficient condition to reject prefixes that do not respect a given fairness metric within a requirement of well-classified examples. One can use this property to reject prefixes before being they are treated in the main loop of `FairCORELS`. This pruning idea can be integrated using two approaches.

The first one called the *eager* approach, checks the sufficient condition before adding an extension of a prefix to the priority queue (before Line 12 with $\delta \cup r$ being the prefix given in the ILP). The second approach called the *lazy* approach, checks the sufficient condition when a prefix is removed from the priority queue and passed the branch and bound lower bound test at Line 5 with $\delta$ being the prefix tested. If the corresponding ILP (called with valid bounds) is unsatisfiable, then the prefix $\delta$ being tested can safely be discarded since no rule list whose prefix starts with $\delta$ can satisfy the conjunction of fairness and well-classified examples requirements. The difference between the two approaches can be seen as the trade-off between memory consumption and computational time. Indeed, given the same inputs and exploration strategies, the *eager* approach consumes less memory than the *lazy* approach as it prunes prefixes before adding them to the queue. However, it requires more calls to the ILP solver.

Finally, we also consider using the ILP models to guide exploration. To realize this, we add an objective to the previously defined ILP, maximizing $x^{TP_{\mathcal{E},p}} - x^{FP_{\mathcal{E},p}} + x^{TP_{\mathcal{E},u}} - x^{FP_{\mathcal{E},u}}$. The ILP is then called as in the *eager* approach, just before adding an extension of a prefix to the priority queue (before Line 12). Whenever it is unsatisfiable, the corresponding prefix is pruned. However, when it is satisfiable, we additionally get the best accuracy reachable (*e.g.*, a lower bound on the objective function value) while also meeting the fairness constraint and improving the objective function. We use this value to order the priority queue $Q$ and define the *ILP-Guided* search heuristic. Intuitively, it guides the exploration towards the prefixes whose fairness may conflict least with accuracy (those with highest ILP objective function).

When building the ILP models, we use tight lower and upper bounds on the number of well-classified examples, whose computations are detailed hereafter.

*Lower Bound Computation.* Let $L(k, d, \mathcal{E}) = |\mathcal{E}| \cdot (1 - (\mathsf{misc}(d, \mathcal{E}) + \lambda \cdot (K_d - k)))$.

**Proposition 4.** *Consider a rule list $d_2$. A rule list $d_1 = (\delta_{d_1}, q_0)$ has better objective value on $\mathcal{E}$ than $d_2$ if and only if $W_{\mathcal{E}}^{d_1} > L(|\delta_{d_1}|, d_2, \mathcal{E})$, in which $|\delta_{d_1}|$ is the length of $d_1$'s prefix.*

*Proof.* $\mathsf{obj}(d_1, \mathcal{E}) < \mathsf{obj}(d_2, \mathcal{E}) \iff \mathsf{misc}(d_1, \mathcal{E}) + \lambda \cdot K_{d_1} < \mathsf{misc}(d_2, \mathcal{E}) + \lambda \cdot K_{d_2}$
$\iff |\mathcal{E}| \cdot (1 - \mathsf{misc}(d_1, \mathcal{E})) > |\mathcal{E}| \cdot (1 - (\mathsf{misc}(d_2, \mathcal{E}) + \lambda \cdot (K_{d_2} - |\delta_{d_1}|)))$
$\iff W_{\mathcal{E}}^{d_1} > L(|\delta_{d_1}|, d_2, \mathcal{E})$ □

Consider the prefix $\delta$ and the current best solution $d^c$ of the main loop. Let $d = (\delta_d, q_0) \in \sigma(\delta)$. Using Proposition 4, we have $d$ has a better objective value than $d^c$ if and only if $W_{\mathcal{E}}^{d} > L(|\delta_d|, d^c, \mathcal{E}) \geq L(|\delta|, d^c, \mathcal{E})$ because $|\delta_d| \geq |\delta|$. Therefore $L(|\delta|, d^c, \mathcal{E})$ is a valid lower bound for the ILP, ensuring that rule list $d$ improves over the current best objective value.

*Upper Bound Computation.* We leverage two observations to compute a tight value $U(\delta, \mathcal{E})$ such that $\forall d \in \sigma(\delta), W_{\mathcal{E}}^{d} \leq U(\delta, \mathcal{E})$. First, the examples captured and misclassified by $\delta$ will always be misclassified for any $d \in \sigma(\delta)$. Second, among the examples not captured by $\delta$, some may conflict (*i.e.*, have the same features vector associated with different labels) and can never be simultaneously predicted correctly. This computation corresponds to the Equivalent Points Bound of CORELS (described in details in Section 3.14 of [4]).

## 5 Experimental Study

The purpose of this section is two-fold. First, after describing our experimental setup, we show the efficiency of the proposed pruning approaches using two biased datasets and the four considered fairness metrics of Table 1. Afterwards, we demonstrate the scalability of our method as well as its complementarity with a new prefix permutation map, using a larger real-world dataset.

### 5.1 Experimental Protocol

We implement and solve the ILP models in C++ using the ILOG CPLEX 20.10 solver[4], with an efficient memoisation mechanism. Sensitive features are used for measuring and mitigating unfairness but are not used in the model's construction in order to prevent disparate treatment [28]. For each dataset, we generate 100 different training sets by randomly selecting 90% of the dataset's instances, with reported values being averaged over the 100 instances. Test values are measured on the remaining 10% instances for each random split. All experiments are run

[4] Source code of this enhanced version of the FairCORELS Python package is available on https://github.com/ferryjul/fairCORELSV2. The use of the CPLEX solver is possible but not mandatory, as our released code also embeds an open-source solver (whose configuration has been tuned to handle our pruning problem efficiently). This solver is Mistral-2.0 [17, 19], in its version used for the Minizinc Challenge 2020.

on a computing grid over a set of homogeneous nodes using Intel Xeon E5-2683 v4 Broadwell @ 2.1GHz CPU.

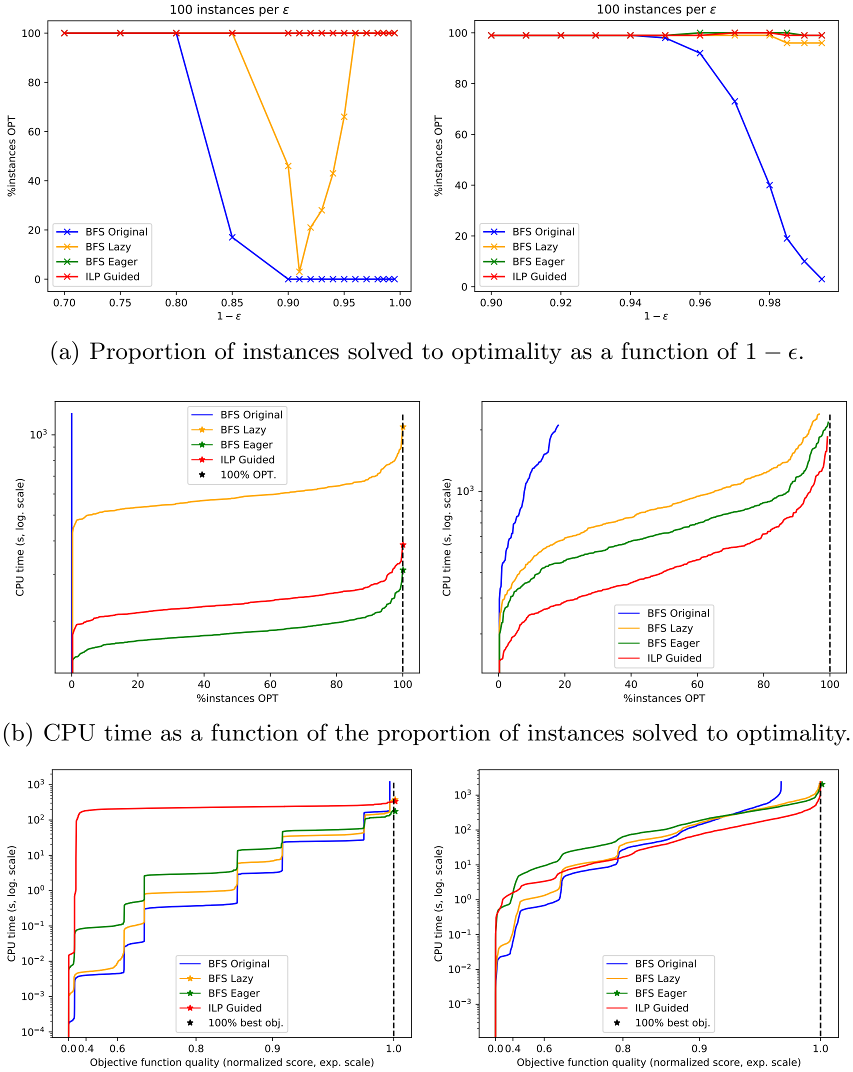
We use three exploration heuristics: a best-first search *ILP-Guided*, a best-first search guided by CORELS's objective and a Breadth-First-Search (BFS). The former inherently comes with an *eager* pruning. For the latter two, we compare the *original* FairCORELS (no ILP pruning), as well as *lazy* and *eager* integrations of our pruning approach. Then, we evaluate the seven exploration settings. However, results for the three best-first searches guided by CORELS's objective are omitted because they consistently provided worst performances (considering all evaluated criteria) than the BFS with equivalent pruning integration. This can be explained by the fact that this approach guides exploration towards accurate solutions first, which conflicts with fairness in practice.

### 5.2 Evaluation of the Proposed ILP-based Pruning Approaches

To empirically assess the effectiveness of our proposed pruning on FairCORELS, we perform experiments for the four metrics of Table 1 using two well-known classification tasks of the literature with several fairness requirements. The first task consists in predicting which individuals from the COMPAS dataset [5] will re-offend within two years. We consider race (African-American/Caucasian) as the sensitive feature. Features are binarized using one-hot encoding for categorical ones and quantiles (with 5 bins) for numerical ones. Rules are generated as single features without minimum support. The resulting preprocessed dataset contains 18 rules and 6150 examples.

The second task consists in predicting whether individuals from the German Credit dataset [11] have a good or bad credit score. We consider age (low/high) as the sensitive feature, with both groups separated by the median value. Features are binarized using one-hot encoding for categorical ones and quantiles (2 bins) for numerical ones. Rules are generated as single features with minimum support of 0.25 or conjunctions of two features with minimum support of 0.5. Gender-related features were excluded. The resulting preprocessed dataset contains 49 rules and 1000 examples. For experiments on the COMPAS (respectively German Credit) dataset, the maximum running time is set to 20 minutes (respectively 40 minutes). For each experiment, the maximum memory use is fixed to 4 Gb. Due to the limited space available, we detail our evaluation for the Statistical Parity metric. Results for all other metrics show similar trends.

Figure 1(a) displays the proportion of instances solved to optimality as a function of the fairness requirement (which gets harder as $1-\epsilon$ increases) to illustrate the joint action of CORELS' bounds and the proposed ILP-based pruning. For low fairness requirements, all evaluated methods reach optimality, thanks to the action of CORELS' bounds. However, these bounds are less effective for strong fairness requirements, and without the ILP pruning, optimality can hardly be reached. Conversely, the higher the value of $1-\epsilon$, the larger the pruning of the search space. Hence, optimality is reached most of the time when performing an eager pruning (*eager* BFS or *ILP-Guided*). This joint effect is particularly visible with the *lazy* BFS approach on the COMPAS dataset.



(a) Proportion of instances solved to optimality as a function of $1 - \epsilon$.

(b) CPU time as a function of the proportion of instances solved to optimality.

(c) Solving time as a function of the objective function quality normalized score.

**Fig. 1.** Experimental results (left: COMPAS, right: German Credit).

Figures 1(b) and 1(c) are generated using high fairness requirements (unfairness tolerances ranging between 0.005 and 0.02). Figure 1(b) presents the solving time as a function of the proportion of instances solved to optimality (lower is better). It shows a clear dominance of the proposed pruning approaches. For COMPAS, the *original* FairCORELS does not prove optimality to any of the instances, whereas all pruning methodologies prove optimality to all instances. For German Credit, similar trends are observed. Overall, the *eager* approach ap-

**Table 2.** Learning quality evaluation ($\epsilon \in [0.005, 0.05]$): Proportion of instances for which each method led to the best train (resp. test) accuracy, and average violation of the fairness constraint at test time.

| Dataset | UNF | BFS Original | | | BFS Lazy | | | BFS Eager | | | ILP Guided | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Train Acc | Test Acc | Unf viol. | Train Acc | Test Acc | Unf viol. | Train Acc | Test Acc | Unf viol. | Train Acc | Test Acc | Unf viol. |
| COMPAS dataset | SP | .951 | .971 | **.009** | **1** | .98 | **.009** | **1** | **.981** | **.009** | **1** | .98 | **.009** |
| | PE | .927 | .956 | **.033** | **1** | **.977** | .034 | **1** | **.977** | .034 | **1** | **.977** | .034 |
| | EOpp | .941 | .961 | **.03** | **1** | .98 | .031 | **1** | **.983** | .031 | **1** | **.983** | .031 |
| | EO | .897 | .934 | **.035** | .997 | .974 | .036 | **1** | **.976** | .036 | **1** | .974 | .036 |
| German Credit dataset | SP | .567 | **.799** | **.045** | .994 | .77 | **.045** | **.999** | .783 | **.045** | .996 | .779 | **.045** |
| | PE | .967 | .914 | .138 | **1** | .914 | **.137** | **1** | .914 | .138 | .997 | **.927** | .138 |
| | EOpp | .683 | .816 | .056 | .99 | .799 | .055 | **1** | .806 | .055 | .991 | **.829** | **.054** |
| | EO | .52 | .759 | **.158** | .979 | .751 | .161 | .997 | .741 | .16 | **1** | **.771** | .159 |

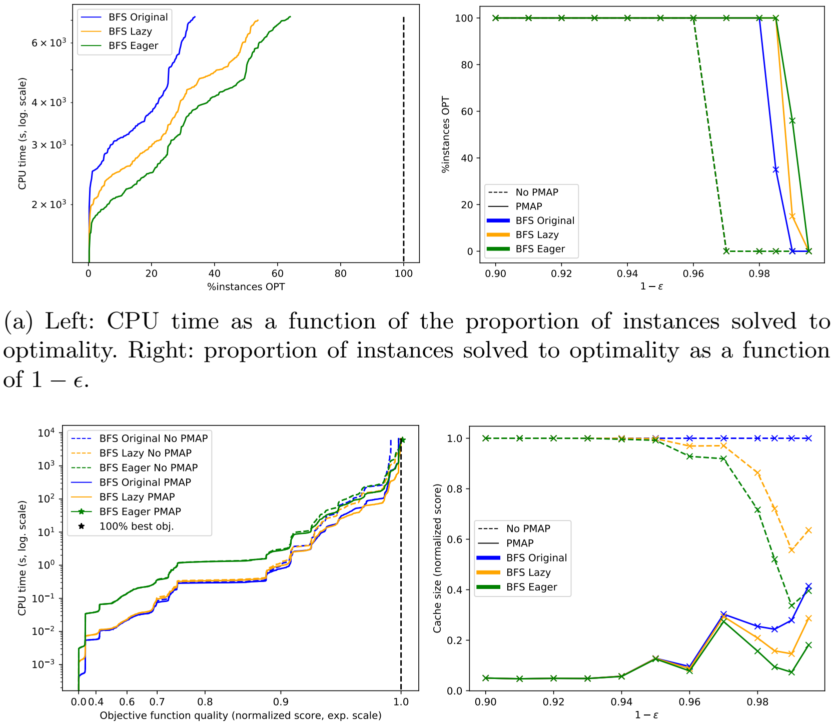
pears more suitable to prove optimality, as it keeps the size of the queue as small as possible. For experiments with German Credit, the *ILP-Guided* approach effectively speeds up convergence and proof of optimality by guiding exploration towards fair and accurate solutions. This is not the case when using COMPAS, but the approach is still able to reach the best solutions, thanks to the performed pruning. Figure 1(c) shows the learning time as a function of the objective function quality (normalized objective score proposed in [18]). The proposed pruning allows finding better solutions within the time and memory limits after a slow start. Indeed, the pruning slows the beginning of the exploration, but pays off, given enough time, by effectively limiting the growth of the priority queue. The *lazy* approach is faster than the *eager* one at the beginning of the exploration. However, this trend is inverted given sufficient time. Again, the *ILP-Guided* approach speeds up convergence on German Credit, but worsens it on COMPAS.

Finally, the reported results illustrate the efficiency of the proposed pruning approaches to speed up the exploration of the prefix tree. The *lazy* approach less slows exploration at the beginning, but the *eager* approach gives better results given sufficient time. The *ILP-Guided* strategy showed an ability to speed up convergence, but its performances depend on the problem at hand.

Test results are reported in Table 2, and suggest that building optimal models does not result in worsening accuracy nor fairness generalization.

### 5.3 Scalability and Complementarity with the Permutation Map

As discussed in Section 3, a prefix permutation map speeds up the `CORELS` algorithm by leveraging symmetries but cannot be used within `FairCORELS` without compromising optimality. We modify it to enforce a weaker symmetry-breaking mechanism while maintaining the guarantee of optimality. More precisely, the proposed new prefix permutation map (PMAP) considers that two prefixes of equal length are equivalent if and only if they have exactly the same confusion matrix and their rules imply the same antecedents. It pushes a new prefix to the priority queue $Q$ (Line 12) only if $Q$ contains no equivalent prefix.



(a) Left: CPU time as a function of the proportion of instances solved to optimality. Right: proportion of instances solved to optimality as a function of $1 - \epsilon$.

(b) Left: CPU time as a function of the objective function score. Right: relative cache size as a function of $1 - \epsilon$.

**Fig. 2.** Results of our experiments on the Adult Income dataset.

To evaluate the scalability of our pruning approaches, we consider Adult Income [11], a larger dataset that gathers records of individuals from the 1994 U.S. census. We consider the task of predicting whether an individual earns more than $50,000$$ per year, with gender (male/female) being the sensitive attribute. Categorical attributes are one-hot encoded and numerical ones are discretized using quantiles (3 bins). The resulting dataset contains $48,842$ examples and 47 rules (attributes or their negation), with a minimum support of 0.05. We consider only the Statistical Parity metric, as the three others do not conflict strongly with accuracy in this setting as observed in Figure 1(a) of [1]. Experiments are performed with and without the new PMAP. The maximum running time is set to two hours, with a maximum memory use of 8 Gb. Results for the *ILP-Guided* approach are excluded as they show no clear improvement over the *eager* pruning, suggesting that the guidance was not beneficial overall.

Results are summarized in Figure 2. The left plot of Figure 2(a) shows the proportion of instances solved to optimality, for $\epsilon \in [0.005, 0.02]$. For these strong fairness requirements, the approaches not using the new PMAP were never able to prove optimality (as can be seen in the right plot) and are not represented.

**Table 3.** Learning quality evaluation (Adult Income dataset, $\epsilon \in [0.005, 0.1]$)

| | | BFS Original | | | BFS Lazy | | | BFS Eager | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | Map Type | Train Acc | Test Acc | Test Unf viol. | Train Acc | Test Acc | Test Unf viol. | Train Acc | Test Acc | Test Unf viol. |
| All | No PMAP | .938 | .942 | **-.004** | .963 | .966 | **-.004** | .964 | .967 | **-.004** |
| | PMAP | .966 | .97 | **-.004** | .998 | .987 | **-.004** | **1** | **.989** | **-.004** |
| $< 0.02$ | No PMAP | .815 | .835 | **.0** | .89 | .907 | .001 | .892 | .91 | .001 |
| | PMAP | .897 | .91 | .001 | .993 | .96 | .001 | **1** | **.968** | .001 |

The complementarity with our pruning approach is particularly visible, with the methods using both the PMAP and the ILP pruning having the best performances, both in terms of objective function quality (Figure 2(b), left plot) and proof of optimality. This is also observed in terms of memory use in Figure 2(b) (right plot). Indeed, the PMAP considerably reduces the size of the queue, leveraging the prefix tree symmetries. However, its effect is weakened for strong fairness constraints. The use of the ILP pruning mitigates this trend and for very strong fairness requirements, the *eager* pruning alone proposes lower memory consumption than the PMAP alone, to reach the same solutions. Finally, learning quality results are provided in Table 3 and confirm these observations. More precisely, they consistently show that the approaches improving train accuracy also improve test accuracy, without impacting fairness violation.

## 6 Conclusion

We propose effective ILP models leveraging accuracy and fairness jointly to prune the search space of `FairCORELS`. Our large experimental study shows clear benefits of our approach to speed-up the learning algorithm on well-known datasets from the literature. This gain is illustrated on three dimensions: achieving better training objective function values (without loss of the learning quality), using less memory footprint (*i.e.*, reduced cache size) and certifying optimality in limited amounts of time and memory. Combined with a proposed simple data structure, the ILP pruning approaches allow the learning of optimal rule lists under fairness constraints for datasets of realistic size.

Thanks to the declarative nature of our pruning approach, our framework is flexible and can simultaneously handle multiple fairness criteria for any number of sensitive groups. Indeed, each group's confusion matrix is modelled using two variables in our ILP. Considering more than two groups would require declaring additional variables, along with desired constraints using these variables.

Overall, our work illustrates the fact that statistical fairness and accuracy, when considered jointly, can be leveraged to reduce the scope of acceptable solutions efficiently. In the future, it would be interesting to pursue this line of work by considering other learning algorithms and machine learning requirements.

Guiding the exploration by leveraging on the ILP models (as attempted with the *ILP-Guided* approach) also seems to be a promising direction.

## References

1. Aïvodji, U., Ferry, J., Gambs, S., Huguet, M.J., Siala, M.: Learning fair rule lists. arXiv preprint arXiv:1909.03977 (2019)
2. Aïvodji, U., Ferry, J., Gambs, S., Huguet, M.J., Siala, M.: Faircorels, an open-source library for learning fair rule lists. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. p. 4665–4669. CIKM '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3459637.3481965
3. Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., Rudin, C.: Learning certifiably optimal rule lists. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 35–44. KDD '17, Association for Computing Machinery (2017). https://doi.org/10.1145/3097983.3098047
4. Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., Rudin, C.: Learning certifiably optimal rule lists for categorical data. Journal of Machine Learning Research **18**(234), 1–78 (2018), http://jmlr.org/papers/v18/17-716.html
5. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. propublica (2016). ProPublica, May **23** (2016)
6. Barocas, S., Hardt, M., Narayanan, A.: Fairness and Machine Learning (2019), http://www.fairmlbook.org
7. Caton, S., Haas, C.: Fairness in machine learning: A survey. arXiv preprint arXiv:2010.04053 (2020)
8. Chikalov, I., Lozin, V., Lozina, I., Moshkov, M., Nguyen, H.S., Skowron, A., Zielosko, B.: Logical Analysis of Data: Theory, Methodology and Applications, pp. 147–192. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-28667-4_3
9. Chouldechova, A.: Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. Big data **5**(2), 153–163 (2017). https://doi.org/10.1089/big.2016.0047
10. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)
11. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
12. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. p. 214–226. ITCS '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2090236.2090255
13. Freitas, A.A.: Comprehensible classification models: A position paper. SIGKDD Explor. Newsl. **15**(1), 1–10 (mar 2014). https://doi.org/10.1145/2594473.2594475
14. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI Magazine **38**(3), 50–57 (2017)
15. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5) (aug 2018). https://doi.org/10.1145/3236009
16. Hardt, M., Price, E., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: Advances in Neural Information Processing Systems. vol. 29. Curran Associates, Inc. (2016), https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf

17. Hebrard, E.: Mistral, a constraint satisfaction library. Proceedings of the Third International CSP Solver Competition **3**(3), 31–39 (2008)
18. Hebrard, E., Siala, M.: Explanation-based weighted degree. In: Salvagnin, D., Lombardi, M. (eds.) Integration of AI and OR Techniques in Constraint Programming - 14th International Conference, CPAIOR 2017, Padua, Italy, June 5-8, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10335, pp. 167–175. Springer (2017). https://doi.org/10.1007/978-3-319-59776-8_13
19. Hebrard, E., Siala, M.: Solver engine (2017), https://www.cril.univ-artois.fr/CompetitionXCSP17/files/Mistral.pdf
20. Ignatiev, A., Cooper, M.C., Siala, M., Hebrard, E., Marques-Silva, J.: Towards formal fairness in machine learning. In: Simonis, H. (ed.) Principles and Practice of Constraint Programming - 26th International Conference, CP 2020, Louvain-la-Neuve, Belgium, September 7-11, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12333, pp. 846–867. Springer (2020). https://doi.org/10.1007/978-3-030-58475-7_49, https://doi.org/10.1007/978-3-030-58475-7_49
21. Kamiran, F., Calders, T.: Data preprocessing techniques for classification without discrimination. Knowledge and Information Systems **33**(1), 1–33 (2012). https://doi.org/10.1007/s10115-011-0463-8
22. Lipton, Z.C.: The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue **16**(3), 31–57 (jun 2018). https://doi.org/10.1145/3236386.3241340
23. Miettinen, K.: Nonlinear multiobjective optimization, International Series in Operations Research & Management Science, vol. 12. Springer, Boston, MA (2012). https://doi.org/10.1007/978-1-4615-5563-6
24. Rivest, R.L.: Learning decision lists. Machine learning **2**(3), 229–246 (1987). https://doi.org/10.1007/BF00058680
25. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019). https://doi.org/10.1038/s42256-019-0048-x
26. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges. arXiv preprint arXiv:2103.11251 (2021)
27. Verma, S., Rubin, J.: Fairness definitions explained. In: Proceedings of the International Workshop on Software Fairness. p. 1–7. FairWare '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3194770.3194776
28. Zafar, M.B., Valera, I., Gomez Rodriguez, M., Gummadi, K.P.: Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In: Proceedings of the 26th International Conference on World Wide Web. p. 1171–1180. WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2017). https://doi.org/10.1145/3038912.3052660