Extending Genetic Process Discovery to Reveal Unfairness in Processes

Muskan Muskan, Felix Mannhardt[0000-0003-1733-777X], and Boudewijn van Dongen[0000-0002-3978-6464]

Eindhoven University of Technology, Eindhoven, the Netherlands

Abstract. Fairness is an essential consideration for most processes in an organization since an equitable treatment of people involved in a process is often mandated by the rules or regulations. It is also desired from a social sustainability perspective. Many processes have a social impact on the actors performing the process activities and on the subjects affected by the process. We focus on the latter case in which a group of process subjects, such as citizens or patients, experiences unfair bias or discrimination during the execution of the process. Obvious instances of such discrimination in processes are negative decisions, but any change in process behavior for a certain group may be a symptom of unfairness. Process mining has been proposed as a method to analyze such unfairness. However, when considering the classical process discovery of a single overall process model, such hidden biases may get disregarded since they are relatively rare occurrences. To address unfairness in processes through process mining, we first need to reveal it in the process model. Towards this goal, we contribute a fairness-aware process discovery approach that extends a genetic algorithm with new quality measures for group fairness. We tested the approach on a set of synthetic but realistic benchmark datasets containing controlled cases of unfairness. The results indicate that in several cases our approach succeeds in revealing hidden biases against certain groups, which would remain hidden in state-of-the-art process discovery. We consider this as an initial step towards a comprehensive analysis of unfairness in processes.

Keywords: Evolutionary Tree Miner · Fairness · Genetic Algorithm.

1 Introduction

Most business processes have a real-life impact on people. Consider a loan application process at a bank. The subject of each process case, the applicant, is an individual, and the decisions taken on the loan as well as the decision on which activities are performed bear direct consequences for them. Fairness [3] in such decision-making, both automated and non-automated, is an essential consideration for organizations from the viewpoint of the *social sustainability* of processes as well as often mandated by regulations.

Discrimination against process subjects is not only a matter in banking but is also present in other domains in which individuals are the main subject of a process, e.g., administration (citizens), healthcare (patients), or human resources (employees). Taking one of the group fairness notions, the decision to reject a loan based on the group membership of the applicant, e.g., their nationality, may be clearly unfair. However, in the enactment of processes, there are further decisions that possibly constitute unfair discrimination [11]. The repetition of certain activities, e.g., multiple background checks of the applicant, or the execution of certain activities, e.g., extensive documentation with wet-signed paperwork, may require the investment of additional resources (time and money) from the individual compared to other process cases. Conversely, skipping certain activities in a process case may also be considered unfair, e.g., skipping certain interviews that are part of a hiring process that leads to a rejection without due process.

Process mining can be used to scrutinize decision-making in the enactment of business processes and provide insight into the *social sustainability* of processes. An event log of the processes' control flow together with detailed information on process subjects may be sufficient information to reveal unfair discrimination. Indeed, process mining has been proposed to detect unfairness in processes [12]. Pohl et al. [12] show that it is possible to leverage traditional techniques and fairness definitions by building tabular representations of decisions or situations from event logs. However, such an approach is not positioned in the standard setting of process discovery in which the process is analyzed on the basis of a discovered process model. For process discovery to be useful in revealing unfairness it needs to clearly represent the discriminating behavior in the process model.

In this paper, we show that directly using process discovery for this purpose can be a challenge. Often, process discovery algorithms disregard the infrequently occurring discriminating process behavior as noise or fail to distinguish it from actual noise [8]. As a single process model is often the departure point of an analysis, this shortcoming of the process discovery may inhibit the detection and analysis of potential unfairness in processes. Therefore, we address the problem of revealing unfair discrimination experienced by subjects of a process through process discovery of a process model from an event log. We take on the setting of group fairness [3,12] in which the event log can be partitioned along the groups into sublogs. Then, the goal is to obtain a single process model that can be used to analyse unfair discrimination in process behavior towards one of the groups, which represents likely a minority of the cases in the event log.

Towards addressing this group fairness issue, we propose two novel quality measures that indicate how well the representation of process behavior in a process model is balanced among subsets of cases recorded in an event log. We integrate our proposed measures into the Evolutionary Tree Miner (ETM) [4], a genetic process discovery algorithm, and test if we can steer the process discovery to yield models that better represent the, possibly unfair, process behavior of the minority group. An experiment with a motivational example as well as four synthetic event logs recently proposed for fairness analysis [11] shows that process discovery algorithms, indeed, hide some kinds of unfair behavior. Our proposed measures can steer the ETM to discover a process model with a more balanced log fitness towards the groups and provides, in several cases, more useful models for analysing unfairness in processes.

The remainder of this paper is structured as follows. Section 2 further motivates our work with an example. Section 3 introduces two quality measures for fairness and Section 4 presents the results of the ETM.

2 Motivation

As indicated, we assume to be given a partition of an event log L into two sublogs L_a and L_b such that $L = L_a \cup L_b$. For our work, we take the standard definition of an event log as a finite multiset over finite sequences of activities. Given a set A of activities, $L \in \mathcal{B}(A^*)$ is an event log. Here, $\mathcal{B}(X)$ is the set of all multisets over set X. To motivate our approach, consider a process with activities $A = \{A, B, C, D, E, Y, N\}$ and a partition of event log L^1 into:

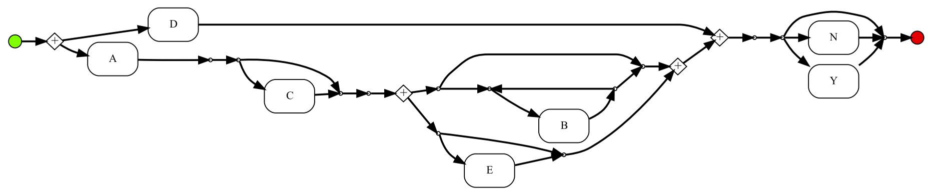
$$
\begin{aligned}
L_a^1 = [&\langle A, B, D, E, N\rangle^{60}, & L_b^1 = [&\langle A, B, D, E, B, N\rangle^{5}, && (1)\\
&\langle A, B, D, E, Y\rangle^{40}, & &\langle A, B, D, E, B, Y\rangle^{1}, && (2)\\
&\langle A, C, D, E, N\rangle^{60}, & &\langle A, C, D, E, B, N\rangle^{5}, && (3)\\
&\langle A, C, D, E, Y\rangle^{40}, & &\langle A, C, D, E, B, Y\rangle^{1}, && (4)\\
& & &\langle A, D, E, N\rangle^{5}, && (5)\\
& & &\langle A, D, E, Y\rangle^{1}, && (6)\\
&\langle D, A, B\rangle^{10}] & &\langle D, A, B\rangle^{1}] && (7)
\end{aligned}
$$

Note that the traces in L_b^1 are less frequent in the overall log $L^1 = L_a^1 \cup L_b^1$ and also show distinct process behavior that is not visible in L_a^1. In a real-world process, this log may be recorded by a hiring or loan application process and the partition may be the result of splitting the log into two groups based on *gender*, *race*, or *nationality* of the applicant. To illustrate it further, the applicant may not get invited for an interview, and/or a background check might be done twice for them. In our log, this is shown by the skipping of activities B and C, and another occurrence of B later in the process. Note that we also consider that some other, unrelated, infrequent behavior, which could be considered noise, namely $< D, A, B >$ may be contained as is often the case in real-world scenarios.

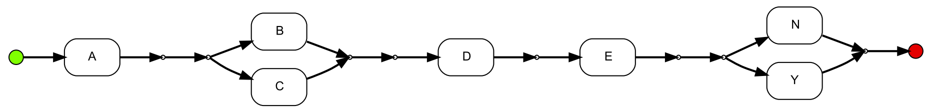
Applying process discovery to the overall event log L_1, for instance the Inductive Miner [9] and the ETM [4], returns the process models shown in Figure 1. Both Inductive Miner and ETM do not yield a desired model for such event logs since the occurrence of the discriminatory behavior is usually a minority, which may be disregarded as noise by the algorithm. Moreover, when changing the noise filtering parameters to include all behavior other infrequent behavior that could be considered actual noise may obstruct the discovered model.

3 Equalised Log Fitness using Genetic Process Discovery

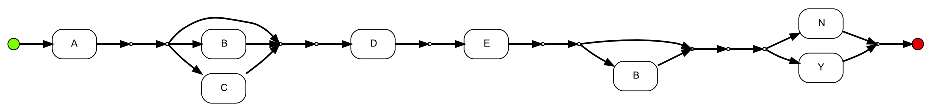
We describe the proposed quality measures and how they can be used with the ETM approach to discover process models revealing unfairness. First, we briefly recall the ETM, then, we define the two measures.



(a) Inductive Miner without filtering ($n = 0$).



(b) Evolutionary Tree Miner and Inductive Miner with filtering ($n = 0.2$).



(c) Model discovered by our approach showing both kinds of unfair discrimination.

Fig. 1: Three process models discovered on L_1 by process discovery algorithms.

3.1 Evolutionary Tree Miner

The use of genetic or evolutionary algorithms for process discovery has been considered very early with the Genetic Miner [2,10] providing the first approach to discover Petri nets based on a genetic algorithm. The ETM [4] improved on this by providing a comprehensive framework for process discovery considering many different quality dimensions [5] such as *log fitness*, *precision*, *simplicity*, and *generalization*. The performance on these measures determines the survival of the candidates onto the next generation, hence, we refer to the combination of all dimensions as *survival fitness*. By using *process trees* as a block-structured representation, the search space of the ETM consists of only sound process models. This facilitates a more efficient exploration of the search space by guiding the *mutation* operations typical of a genetic algorithm [6]. Additionally, it allows more efficient computation of the proposed quality measures, for instance, computing the *log fitness* of a model via an *alignment* [1] between the event log and the model is faster for process trees and has seen recent improvements [13]. A recent development in genetic process discovery is X-Processes [7]. While it utilises a different survival fitness function and model representation, our quality measures can still applied to X-Processes. We leave a comparison of different genetic algorithms, including fairness considerations, for future work. As the primary goal of this paper is to demonstrate the feasibility of our approach and the relevance of our problem statement.

We briefly recall the approach followed by ETM as summarized by Figure 2 taken from [4]. The ETM framework starts with *creating* an initial population of process tree models. Such population is typically random but could also be

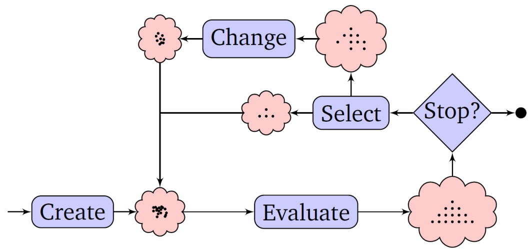


Fig. 2: The genetic algorithm in the ETM framework as illustrated in [4].

guided by the frequency of traces [6]. The main part of the generation loop is to evaluate the quality of all the models in the population according to a survival fitness function. Please note that here, *survival fitness* does not refer only to *log fitness* mentioned earlier but is determined as the weighted average of all selected quality measures, which should be normalized. Based on the evaluation of the population ETM checks a *stopping* criteria, e.g., a preset limit on the number of iterations of the loop also denoted as generations. Now, the main genetic part of the algorithm is concerned with selecting which candidate models to keep from the population and which of them to change through different operations. We are not further discussing the details of the change operations, which are explained in detail in [4], since they are not relevant to the presentation of our novel quality measure.

3.2 Proposed measures for equalising log fitness between groups

The quality of a process model is multi-faceted as shown in [4,5] which means that all of the quality measures are important and should be considered for process discovery. However, as a starting point for our research we selected *log fitness* as the most relevant measure to consider for uncovering unfair behavior. In fact, *log fitness* measures how much of the event log behavior is actually modelled and without unfair behavior being modelled it cannot be revealed. Of course, other quality dimensions are still relevant, e.g., *precision* is necessary such that we can better indicate in which part of the process unfair behavior was observed. Our quality measure can be used together with the regular *precision* measure and the investigation of additional measures for fairness is left for future work.

To ensure that the likely, infrequent unfair behavior is considered during the genetic search, we designed quality measures that aim to equalise the log fitness measured for both of the sublogs. We tested two formulas based on different considerations. The first formula for equalised log fitness is based on a normalised absolute difference EF_d and the second is based on a geometric argumentation EF_g. To define the measures, we briefly introduce *alignment costs* from which log fitness is typically determined and which we leverage directly. Given a process model M[1] and an event log L, an alignment [1] determines a model trace with the minimum number of mismatches to a log trace $\sigma \in L$. Each mismatch is typically assigned a cost, e.g., a unit cost of one, and the overall cost of aligning an individual trace to the model can be obtained. For our measure, we only require such cost function $cost(\sigma, M) \in \mathbb{Q}$ to be provided. Since the sublogs may be of different sizes, we are interested in the average alignment cost of a trace in a (sub) log.

Definition 1 (Average alignment cost). *Let L be an event log and let M be a process model. Let $cost(\sigma, M)$ be a function that returns the alignment cost, i.e., the cost of mismatches over some cost function, between any trace $\sigma \in L$ and the process model M. We denote with $avgCost(L, M) \in \mathbb{Q}$ the arithmetic mean of the cost of all traces in L:*

$$avgCost(L, M) = \frac{1}{|L|}\left(\sum_{\sigma \in L} cost(\sigma, M)\right)$$

We can now compute the average costs over all individual traces for each sublog and present our two quality measures in Definitions 2 and 3. Note that the two measures behave differently to small variations in average costs between the two group, as explained in more detail later.

Definition 2 (Equalised log fitness based on difference). *Let $L = L_a \cup L_b$ be an event log with a given partition into L_a and L_b. Let M be a process model. Let $\epsilon \in \mathbb{Q}_+$ be a small positive, non-zero constant. We define the equalised log fitness measure based on absolute difference $EF_d \in \mathbb{Q}$ in cost as:*

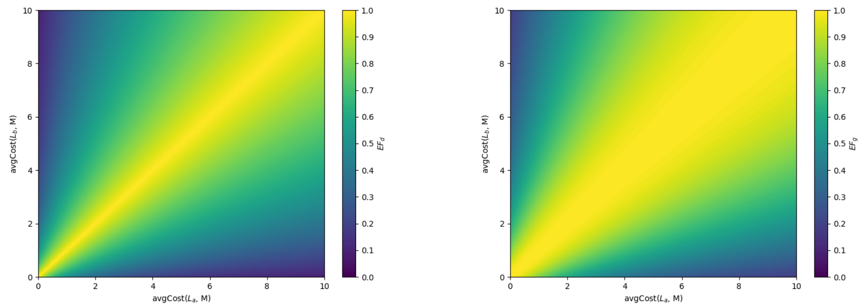
$$EF_d(L, M) = 1 - \left|\frac{avgCost(L_a, M) - avgCost(L_b, M)}{avgCost(L_a, M) + avgCost(L_b, M) + \epsilon}\right|$$

Definition 3 (Equalised log fitness based on geometry). *Let $L = L_a \cup L_b$ be an event log with a given partition into L_a and L_b. Let M be a process model. Let $\epsilon \in \mathbb{Q}_+$ be a small positive, non-zero constant. We define the equalised log fitness measure $EF_g \in \mathbb{Q}$ to be twice the angle in a right triangle spanned by both costs:*

$$EF_g(L, M) = \frac{2(avgCost(L_a, M) + \epsilon)(avgCost(L_b, M) + \epsilon)}{(avgCost(L_a, M) + \epsilon)^2 + (avgCost(L_b, M) + \epsilon)^2}$$

The addition of ϵ in Definitions 2 and 3, prevents undefined behaviour when both average costs are zero. In such cases, the equalised log fitness will be a perfect score of 1. The formula for EF_d may be more suitable than a normalized absolute difference if we want to penalize the differences while also considering

[1] In the ETM process trees are used as formalism but alignments make very few assumptions on the model only requiring every model trace to eventually properly finish. We do not further introduce process trees as our measure is independent of the modelling formalism used.



(a) Absolute difference EF_d (b) Geometric argument EF_g

Fig. 3: Values obtained for both EF_d and EF_g for cost varying from 0 to 10.

the overall magnitude of the two values, giving a more nuanced similarity metric than just the absolute difference alone. On the other hand, the formula for EF_g was obtained by following a geometric argumentation, i.e., it corresponds to $\sin(2\theta)$ where angle θ is in a right triangle spanned by both the average costs. This can be rewritten to the form in Definition 3 based on the identity $\sin(2\theta) = 2\sin\theta\cos\theta$.

Figures 3a and 3b visualise the proposed measures with $\epsilon = 1$ for the ranges of an average cost between 0 and 10 for each of the groups to facilitate their interpretation and show their unique characteristics. The graph for EF_d in Figure 3a has a narrower yellow region, corresponding to an almost perfect score, whereas the graph for EF_g in Figure 3b shows a broader area with a near-perfect score. EF_d hence punishes small differences more severely, which might be desirable in some scenarios. In contrast, EF_g has a relatively slow decrease of our measure for slightly different cost values. Consequently, we get a lower threshold for equalised log fitness measure, which might be desirable in certain real life applications where the notion of fairness is not so strict.

4 Evaluation

We implemented both proposed measures and added them to the ETM framework in ProM[2]. Then, we evaluated whether they, indeed, can help to reveal unfairness in processes based on event logs. The complete results are published as supplementary material[3].

[2] An updated ETM is available: https://github.com/promworkbench/EvolutionaryTreeMiner/tree/etm-fairness and is added to ProM 6.14.

[3] Results are available at: https://dx.doi.org/10.5281/zenodo.13364712

4.1 Setup

Our evaluation builds on a recently published dataset [11] in which four exemplary processes, hiring, hospital, lending, and renting, are modeled, and unfair discrimination is injected in a controller manner. For each of the event logs, we have a ground truth attribute that indicates whether discrimination was applied. In this experiment, we leverage this attribute to create a partition of the event log into L_a, the majority cases with no protected attributes, and L_b, the minority cases with unfairness problems based on the protected attributes. In all cases, L_b represents approximately 30% of the data. In a real-life scenario, the partitioning could be done automatically using one or more suitable case attributes, e.g., gender or age.

The process discovery experiments were performed on the four event logs obtained from the benchmark dataset. Two process discovery algorithms were used: the ETM and the Inductive Miner infrequent (IMf). The regular ETM and the IMf were tested against the ETM having our equalised log fitness measures EF_d and EF_g. For evaluation, fitness and precision measures were computed on the overall logs and the two sublogs L_a and L_b.

The ETM results were computed over 3 repetitions with each 1000 generations for the benchmark log to mitigate the effect of randomness in the genetic algorithm. For the motivational example, we only used 300 generations once as it was sufficient to obtain stable results over this relatively small event log. The models for IMf were computed once for both of the event logs since it is deterministic. We use the default weight settings for the ETM, i.e., weight 10.0 for log fitness, weight 5.0 for precision and weight 1.0 for both generalisation and simplicity. The weight for the equalised log fitness measures was empirically chosen as 3.0 based only on the motivational example without any further optimisations.

4.2 Results

The performance of the discovered process models is evaluated by comparing the log fitness and precision values on our benchmark dataset. An overview of the results is presented in Table 1. Our primary goal is to minimize the difference in the log fitness between the two sublogs: L_a (the majority) and L_b (the minority or protected group), while still considering regular quality criteria such as log fitness and precision in the genetic search.

The hiring and lending logs show better performance with both our measures, especially with EF_d, as the fitness between the majority and minority sublogs is more equalised, with only a difference of 0.04 and 0.08. Surprisingly, for the hiring event log, our measures gives an overall better fitting model on both L_a and L_b sublogs at the expense of precision. For the lending event log, our approach gives similar high fitness on L_b and slightly worse fitness on L_a. However, our measure, EF_g, gives slightly better precision. In case of the hospital log, we discovered that the simple process model is already perfectly fitting both sublogs. Only for the renting log, our approach resulted in a decrease in the log fitness with an

Table 1: Result of four process discovery approaches: ETM with our proposed measures (EF_d, EF_g), without our measures (ETM) in the default configuration, and Inductive Miner infrequent (IMf) as a reference for a non-genetic approach. Average log fitness (fit_M) of the discovered models (M) is reported for both majority (L_a) and minority (L_b) sublogs, along with their difference ($\Delta_{\mathit{fit}_M}(L_a, L_b) = \mathit{fit}_M(L_a) - \mathit{fit}_M(L_b)$), and the precision ($prc$) is computed on the entire log.

Log	Metric	Model (M)			
		ETM	EF_d	EF_g	IMf
Hiring	$\mathit{fit}_M(L_a)$	0.89	**0.92**	0.91	0.93
	$\mathit{fit}_M(L_b)$	0.74	**0.88**	0.83	0.85
	$\Delta_{\mathit{fit}_M}(L_a, L_b)$	0.15	**0.04**	0.08	0.08
	$prc_M(L)$	**0.93**	0.55	0.71	0.71
Lending	$\mathit{fit}_M(L_a)$	**0.97**	0.91	0.91	1.00
	$\mathit{fit}_M(L_b)$	0.90	0.90	0.87	0.97
	$\Delta_{\mathit{fit}_M}(L_a, L_b)$	0.07	**0.01**	0.04	0.03
	$prc_M(L)$	0.82	0.76	**0.84**	0.90
Hospital	$\mathit{fit}_M(L_a)$	1.00	0.99	1.00	1.00
	$\mathit{fit}_M(L_b)$	1.00	0.99	1.00	0.97
	$\Delta_{\mathit{fit}_M}(L_a, L_b)$	0.00	0.00	0.00	0.03
	$prc_M(L)$	**1.00**	0.95	0.94	1.00
Renting	$\mathit{fit}_M(L_a)$	**0.95**	0.87	0.90	0.97
	$\mathit{fit}_M(L_b)$	**0.89**	0.77	0.83	0.86
	$\Delta_{\mathit{fit}_M}(L_a, L_b)$	**0.06**	0.10	0.07	0.11
	$prc_M(L)$	0.72	0.73	**0.79**	0.86

increase in precision for both sublogs. Upon further investigated, we found out that it is likely due to difference in the way discrimination was introduced in this log, i.e., by the difference in the frequency of occurrence of some events. Furthermore, the renting log is different from the other logs, as it has two subprocesses. First, the process of getting the apartment after signing the contract. Second, the process of keeping the apartment by paying the rent on time, which is a repeating process. Additionally, the second sub-process is dependent on the how often the late payment is accepted. Nonetheless, this is a side-affect of our approach that sometimes, it gives a reduced fitness on the sublog L_a compared to the model obtained with the regular ETM. Note that, this may be possible to mitigate by decreasing the weight of our measure EF from 3.0 or increasing the weight of log fitness from 10.0. However, the main aim of our approach is to have a discovered process model that can readily and clearly show the hidden discrimination.

To evaluate, whether the discovered process models are better suited to analyse unfair discrimination, we compared the process models discovered by regular ETM and our approach using the EF_d measure. We used EF_d instead of the EF_g

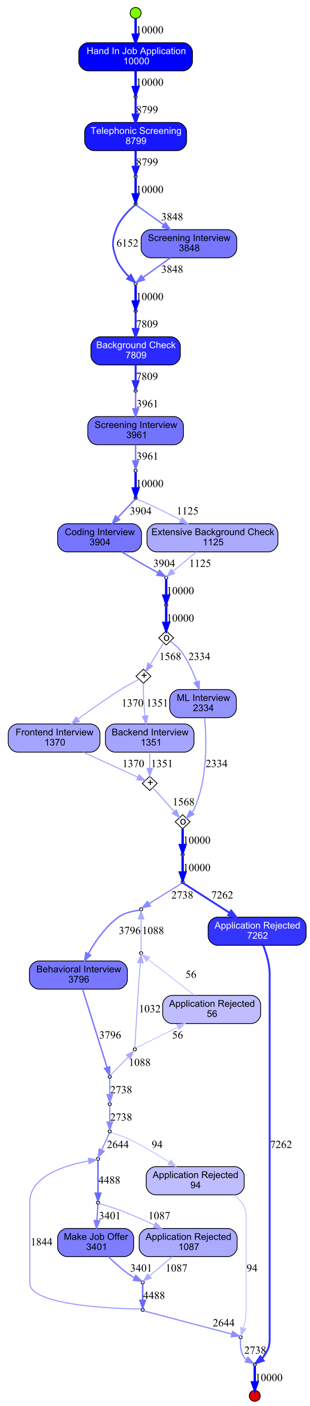
since it achieved the best balance in fitness. The frequency annotated models are shown in Figure 4. Clearly, differences in the discovered process models obtained with and without our measure are visible. Such as, in contrast to the ETM, the model returned by EF_d shows the possibility of skipping various activities such as, Telephonic interview (A, Fig. 4), Background check (B, Fig. 4), Extensive background check (C, Fig. 4), and Coding interview. Additionally, it also shows an unusual behavior, i.e., the second occurrence of Make job offer. In fact, all these indicate the discriminatory behavior that were initially introduced in this unfair dataset. Such as higher rejection rates, less interview opportunities and hence less job offers for the protected group [11].

5 Conclusion

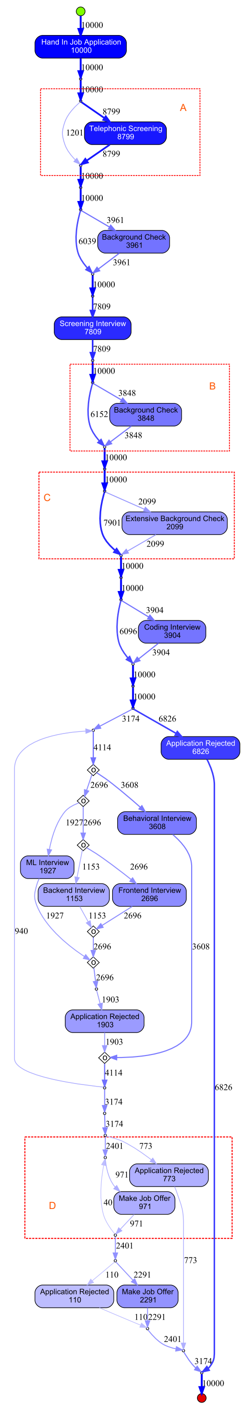
This paper introduced two novel quality measures, denoted as *equalised log fitness*, that determine the balance in log fitness towards a process model for a partition of an event log into two sublogs. Our aim with this work was to more easily reveal any unfair discrimination in the process when using a single process model discovered from the overall log. By leveraging the measures, we guided a genetic process discovery approach towards returning a process model that strikes a better balance in representing the behavior of both groups. We evaluated our approach with a motivational example and a, recently published, benchmark event log. Our approach, indeed, improved the process model for some of the benchmark event logs and can re-discover all the unfair process behavior injected into the motivational example. The resulting process models showcased the hidden unfair behavior as well as improved the overall performance of the model for these logs.

Not all discrimination from the benchmark event logs was detected since our measures focus on *log fitness* which mainly detects missing or additional events. However, discrimination can also occur in terms of changes in the frequency of occurrence of some events, which would require a stochastic perspective on fairness in processes. This stochastic perspective is a possible avenue for extending this research by including a broader understanding of fairness and would help to reveal and explain the discrimination in some of our benchmark event logs such as the renting case. The dataset could also be varied by including more than two subsets of the event log, leading to a multi-class or multi-group problem with the aim of achieving a balanced log fitness among all of them. Additionally, the weight of the equalised fitness measures can be optimised or fine-tuned better with respect to the other quality measures. Finally, a more robust fairness metric can be investigated, and compared with the existing measures used in machine learning such as demographic parity, and equality of odds. Plus, the role of precision in such a fairness measure can further be discovered.

Acknowledgments. We thank Yi-Chiau Li for several discussions on fairness in the used benchmark dataset, which have greatly influenced this work.



(a) Regular ETM



(b) ETM including EF_d

Fig. 4: Process models generated by ETM after 1000 generations with default configuration as well as including the quality measure EF_d on the hiring event log.

References

1. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. WIREs Data Mining Knowl. Discov. **2**(2), 182–192 (2012)
2. van der Aalst, W.M.P., de Medeiros, A.K.A., Weijters, A.J.M.M.: Genetic process mining. In: ICATPN. Lecture Notes in Computer Science, vol. 3536, pp. 48–69. Springer (2005)
3. Barocas, S., Hardt, M., Narayanan, A.: Fairness and machine learning: Limitations and opportunities. MIT press (2023)
4. Buijs, J.: Flexible evolutionary algorithms for mining structured process models. Ph.D. thesis, Mathematics and Computer Science (2014)
5. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. Int. J. Cooperative Inf. Syst. **23**(1) (2014)
6. van Eck, M.L., Buijs, J.C.A.M., van Dongen, B.F.: Genetic process mining: Alignment-based process model mutation. In: Business Process Management Workshops. Lecture Notes in Business Information Processing, vol. 202, pp. 291–303. Springer (2014)
7. Fantinato, M., Peres, S.M., Reijers, H.A.: X-processes: Process model discovery with the best balance among fitness, precision, simplicity, and generalization through a genetic algorithm. Inf. Syst. **119**, 102247 (2023)
8. Koschmider, A., Kaczmarek, K., Krause, M., van Zelst, S.J.: Demystifying noise and outliers in event logs: Review and future directions. In: Business Process Management Workshops. Lecture Notes in Business Information Processing, vol. 436, pp. 123–135. Springer (2021)
9. Leemans, S.J.J.: Robust Process Mining with Guarantees - Process Discovery, Conformance Checking and Enhancement, Lecture Notes in Business Information Processing, vol. 440. Springer (2022)
10. de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: an experimental evaluation. Data Min. Knowl. Discov. **14**(2), 245–304 (2007)
11. Pohl, T., Berti, A., Qafari, M.S., van der Aalst, W.M.P.: A collection of simulated event logs for fairness assessment in process mining. In: BPM (Demos / Resources Forum). CEUR Workshop Proceedings, vol. 3469, pp. 87–91. CEUR-WS.org (2023)
12. Pohl, T., Qafari, M.S., van der Aalst, W.M.P.: Discrimination-aware process mining: A discussion. In: ICPM Workshops. Lecture Notes in Business Information Processing, vol. 468, pp. 101–113. Springer (2022)
13. Rocha, E.G., van der Aalst, W.M.P.: Polynomial-time conformance checking for process trees. In: BPM. Lecture Notes in Computer Science, vol. 14159, pp. 109–125. Springer (2023)