Fair E^3: Efficient Welfare-Centric Fair Reinforcement Learning

Cyrus Cousins
Brown University
Department of Computer Science
Providence, RI
cyrus_cousins@brown.edu

Kavosh Asadi
Amazon

Michael L. Littman
Brown University
Department of Computer Science
Providence, RI

Abstract

As the negative societal consequences of machine learning systems run amok have become increasingly apparent, fair machine learning methods have seen increased attention for tasks like facial recognition, medical care and diagnosis, and employment hiring decisions. Despite this positive trend, most attention on the theory side has been focused on fair supervised and unsupervised settings, whereas second-order impact of machine learning applications, such as the runaway positive feedback loops in settings like predictive policing, are more naturally posed in the setting of *reinforcement learning*. We propose a novel, welfare-centric, fair reinforcement-learning setting, in which the agent enjoys *vector-valued* reward from a set of beneficiaries. Given a *welfare function* $\mathcal{W}(\dots)$, the task is to select a policy π that is favorable to all beneficiaries, in the sense that it optimizes the welfare of the value of the beneficiaries from state s_0, i.e., $\operatorname{argmax}_\pi \mathcal{W}\big(V_1^\pi(s_0), V_2^\pi(s_0), \dots, V_g^\pi(s_0)\big)$. We show that, in this setting, both per-beneficiary exploration and per-beneficiary policy optimization are insufficient to identify the welfare-optimal policy. Whether an individual action is a mistake depends on the context of *subsequent actions*, therefore the standard PAC-MDP framework does not readily generalize to fair reinforcement learning. Consequently, we develop a stronger learning model, wherein at each timestep an agent either takes an *exploration action* or outputs an *exploitation policy*. We require that each exploitation policy be ε-welfare optimal, and the number of exploration steps be polynomial in all relevant parameters. We reduce PAC-MDP learning to this framework, showing that our framework is sufficiently challenging so as to be interesting, and define the fair E^3 learner to operate under this model, thus demonstrating that fair reinforcement learning is tractable.

Keywords: Fairness, Welfare, Vector-Valued MDPs, Learning, Planning

1 Introduction

Fair machine learning (ML) methods have recently seen increasing attention for tasks like facial recognition [Lohr, 2018] and employment hiring decisions [Kleinberg et al., 2018]. Despite this positive trend, most attention on the theory side has been focused on fair supervised [Agarwal et al., 2018] and unsupervised [Chierichetti et al., 2018] ML, whereas second-order impact of ML models, such as the runaway feedback loops in settings like predictive policing [Ensign et al., 2018]are more naturally posed in reinforcement learning (RL) settings. We apply ideas from welfare-centric supervised learning [Cousins, 2021, 2022] to reinforcement learning (RL) settings; in particular, we assume an agent receives a vector-valued reward signal from a set of *beneficiaries*, each representing, e.g., different racial, gender, religious groups, and the task is to learn a single policy that treats beneficiaries fairly.

We argue it is not the role of the algorithm designer to dictate *what fairness means* in the sense of *how to compromise between beneficiaries*, but rather to optimize for a *given fairness notion* (ideally one agreed upon by society, government, political philosophers, and other interested parties), as encapsulated by a metric of *societal welfare*. In supervised learning, doing so is relatively straightforward, as we generally maximize the welfare of *expected per-beneficiary utility*, so in RL, we take utility to be the standard *geometrically discounted value function* w.r.t. each beneficiary's reward. In general, optimizing welfare is referred to as the *social planner's problem*, so in a sense our work addresses this problem in the context of RL.

While optimizing the welfare of beneficiary value functions is a well specified goal for a *planning algorithm*, or when studying the asymptotic behavior of a learning algorithm, when considering the process of fair learning in an *unknown MDP*, it is imperative that we also ask the right questions as to *how quickly* we can learn and if we can guarantee that our learning algorithm behaves fairly in all but a finite number of steps. To this end, we generalize the PAC-MDP framework to a novel *adversarial fair MDP learning* framework, which represents a substantially more difficult learning task. Nevertheless, we show that an algorithm inspired by the classic E^3 algorithm [Kearns and Singh, 2002], which we call fair E^3, is capable of adversarial fair MDP learning.

In fair learning settings, quantifying whether an action is fair is substantially more difficult than quantifying whether an action is optimal to a single agent, because fairness depends on the context of how the agent behaves overall (i.e., tradeoffs between beneficiaries should be balanced). Consequently, in our model, the agent must output fair policies when it is capable of doing so. When it is not, the agent can output only exploration actions, and our concept of learnability requires that the agent with high probability always outputs ε-optimal fair policies, while taking only a bounded number of exploration actions over its infinite lifetime. We allow an adversary to move the agent arbitrarily after it outputs a policy. At any step, the adversary is allowed to select a new welfare function, representing changing societal ideals of how fairness should work, and the agent is expected to output either an exploration action, or a policy optimizing said welfare function. Of course, the adversary is free to leave the agent's position and welfare function unchanged with no degradation in the upper-bounds we prove.

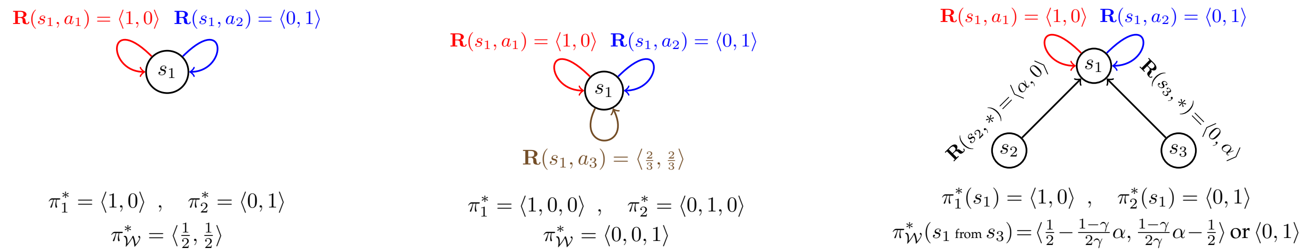
Contributions

1. We frame the traditionally egocentric challenge of reinforcement learning as a social problem, where the actions taken by an agent impact a set of *beneficiaries*.
2. Using ideas from vector-valued reinforcement learning, econometrics, and social welfare theory, we establish *welfare optimal* policies over the value functions of the set of beneficiaries. We focus on finite-state fully-observable MDPs with bounded reward and γ-geometric value-discounting, but our philosophy and methodology can be generalized.
3. We introduce a learning framework in which the agent only observes the consequences of its actions during exploration, and that an adversary can freely move the agent whenever the agent outputs an exploitation policy. During exploitation, the agent is expected to correctly return a near-welfare optimal policy from the current state with high probability, and is also expected to exploit in all but a finite number of exploratory steps. This specific decoupling of exploration and exploitation conditions is particularly conducive to studying the multi-beneficiary RL setting, in which the optimal policy may in general depend on the starting state.
4. We show that our learning framework, which we refer to as the adversarial fair MDP learning framework, is stronger (more difficult) than the well-known probably-approximately correct (PAC) MDP reinforcement learning framework.
5. In section 3, we present the *fair* E^3 algorithm, and prove that it is an adversarial fair MDP learner.

2 Illuminating Examples

We first consider a few simple examples (visualized in fig. 1) that illustrate that intuition from the standard scalar-reward RL setting can be misleading. In these examples, we consider only the relatively straightforward *egalitarian welfare* (i.e., minimum utility, $\mathcal{W}_{\mathrm{Egal}}(a, b) = \min(a, b)$) function, on stateless (single-state) MDPs with deterministic rewards. Despite this barren setting, we find that the standard RL algorithms exhibit misbehavior in learning.

Example 1 (Symmetric 2-Arm Bandit; fig. 1a)**.** *We first consider a 2-arm bandit, with reward* $\mathbf{R}(a_1) = \langle 1, 0 \rangle$*, and* $\mathbf{R}(a_2) = \langle 0, 1 \rangle$*. Here, the unique optimal Markovian policy is* $\pi^*_{\mathcal{W}} = \langle \frac{1}{2}, \frac{1}{2} \rangle$*.*


$\mathbf{R}(s_1, a_1) = \langle 1, 0 \rangle$
$\mathbf{R}(s_1, a_2) = \langle 0, 1 \rangle$
s_1
$\pi_1^* = \langle 1, 0 \rangle \;,\quad \pi_2^* = \langle 0, 1 \rangle$
$\pi_{\mathcal{W}}^* = \langle \frac{1}{2}, \frac{1}{2} \rangle$
$\mathbf{R}(s_1, a_1) = \langle 1, 0 \rangle$
$\mathbf{R}(s_1, a_2) = \langle 0, 1 \rangle$
s_1
$\mathbf{R}(s_1, a_3) = \langle \frac{2}{3}, \frac{2}{3} \rangle$
$\pi_1^* = \langle 1, 0, 0 \rangle \;,\quad \pi_2^* = \langle 0, 1, 0 \rangle$
$\pi_{\mathcal{W}}^* = \langle 0, 0, 1 \rangle$
$\mathbf{R}(s_1, a_1) = \langle 1, 0 \rangle$
$\mathbf{R}(s_1, a_2) = \langle 0, 1 \rangle$
s_1
$\mathbf{R}(s_2, *) = \langle \alpha, 0 \rangle$
$\mathbf{R}(s_3, *) = \langle 0, \alpha \rangle$
s_2
s_3
$\pi_1^*(s_1) = \langle 1, 0 \rangle \;,\quad \pi_2^*(s_1) = \langle 0, 1 \rangle$
$\pi_{\mathcal{W}}^*(s_1 \text{ from } s_3) = \langle \frac{1}{2} - \frac{1-\gamma}{2\gamma}\alpha, \frac{1-\gamma}{2\gamma}\alpha - \frac{1}{2} \rangle$ or $\langle 0, 1 \rangle$

(a) Symmetric 2-Arm Bandit (b) Compromise 3-Arm Bandit (c) Asymmetric Start Bandit MDP

Figure 1: Small MDPs that exhibit surprising behavior under multi-beneficiary objectives.

There are several surprises here:

1. The (unique) optimal policy is *stationary*, but not *deterministic*, i.e., it is *stochastic*.
2. Policy iteration iteratively selects the greedy welfare-optimal policy, i.e., selects the policy

$$\pi^{(i)} \leftarrow \operatorname*{argmax}_{\pi} \mathcal{W}\left(\mathop{\mathbb{E}}_{\pi}[\mathbf{R}_1(s_0, a) + \gamma \mathbf{V}_1^{\pi^{(i-1)}}(s_1)], \ldots, \mathop{\mathbb{E}}_{\pi}[\mathbf{R}_g(s_0, a) + \gamma \mathbf{V}_g^{\pi^{(i-1)}}(s_1)] \right) \; .$$

   This would be optimal under the (false) assumption that the *value function remains fixed*, and is *convergent* for linear (value) MDP objectives. However, policy iteration for the egalitarian welfare objective, initiated at either deterministic policy, *oscillates* between $\pi(s) = \langle 1, 0 \rangle$ and $\pi(s) = \langle 0, 1 \rangle$. which leads to repeated *overcorrections* for initial policy unfairness, hence the oscillatory behavior.

We now consider an extension, which has a third option, which is not preferable to either beneficiary, but is more effective as a compromise than any mixture of the first two options.

Example 2 (Compromise 3-Arm Bandit; fig. 1b)**.** *We next consider a 3-arm bandit, with reward* $\mathbf{R}(a_1) = \langle 1, 0 \rangle$, $\mathbf{R}(a_2) = \langle 0, 1 \rangle$, *and* $\mathbf{R}(a_3) = \langle \frac{2}{3}, \frac{2}{3} \rangle$. *The optimal policies for beneficiary 1, beneficiary 2, and the welfare objective,* $\pi_1 = \langle 1, 0, 0 \rangle$, $\pi_2 = \langle 0, 1, 0 \rangle$, *and* $\pi_{\mathcal{W}} = \langle 0, 0, 1 \rangle$, *respectively.*

This is perhaps not hugely surprising, but it is notable nonetheless, as it starkly illustrates just how different a welfare optimal policy may be from any individual beneficiary's optimal policy. In particular, despite there being unique optimal stationary policies π_1^*, π_2^*, and $\pi_{\mathcal{W}}^*$, clearly, $\pi_{\mathcal{W}}^*$ is not a linear combination of π_1^* and π_2^*. Furthermore, at every state, the probability of selecting any action never exceeds 0 in more than one of these optimal policies. This also has implications for how the MDP is explored; for instance if beneficiaries one and two are independently allowed to run a UCB-style algorithm, neither will even bother to fully explore a_3, thus even together they don't collect enough information for welfare-optimal planning. We thus conclude that not only the planning, but also the exploration aspect of RL must explicitly consider welfare objectives.

We now extend the 2-armed bandit example by adding two additional states, which represent disparate starting conditions for the two beneficiaries.

Example 3 (Symmetric 2-Arm Bandit, with Asymmetric Starting Conditions; fig. 1c)**.** *This MDP has three states, but once* s_1 *is reached, it is never left, thus becoming a 2-armed bandit.*

From s_1, neither beneficiary is privileged, but from s_2, beneficiary 1 begins by receiving α utility, and from s_3, beneficiary 2 begins by receiving α utility. To make things fair, we need to select $\pi(s_1)$ to benefit the underprivileged group, i.e., starting from s_2, have $\pi(s_1)$ choose action 1 with probability x such that $\frac{\gamma}{1-\gamma}x + \alpha = \frac{\gamma}{1-\gamma}(1 - x)$, thus $2\frac{\gamma}{1-\gamma}x = \frac{\gamma}{1-\gamma} - \alpha \implies x = \frac{1}{2} - \frac{1-\gamma}{2\gamma}\alpha$. This makes sense: larger α need to subtract a larger privilege correction term from $\frac{1}{2}$, and after a certain point, negative x (which is of course impossible) would be required to compensate for disparate starting conditions.

3 The Fair E^3 Algorithm

We now introduce the *fair* E^3 *algorithm* and bound its sample complexity, thus showing that it is a adversarial fair adversarial MDP learner. We provide pseudocode in Algorithm 1. The key to understanding the algorithm is that the state space is divided into three sets: the unknown set $\mathcal{S}_{\text{unk}}$, the outer-known set $\mathcal{S}_{\text{out}}$, and the inner-known set $\mathcal{S}_{\text{inn}}$. Initially, all states are unexplored, placed in $\mathcal{S}_{\text{unk}}$. After visiting a state and taking all actions sufficiently many times using balanced wandering [Kearns and Singh, 2002], it is placed in either $\mathcal{S}_{\text{out}}$ if with nonnegligible probability $\geq E$ it is possible to reach $\mathcal{S}_{\text{unk}}$ in $\leq T$ steps, or $\mathcal{S}_{\text{inn}}$ if it is not possible to reach the unknown set with nonnegligible probability. Note that E and T will be determined so as to ensure adversarial fair MDP learnability.

Algorithm 1 Fair E^3 and the Adversarial Fair MDP Setting

```
~ Fair $E^3$ Agent Code ~

procedure AgentInit($\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{R}, \boldsymbol{T}, \gamma \rangle, \epsilon, \delta, \mathrm{R}_{\max}$)
  $T \leftarrow \lceil \frac{1}{1-\gamma} \ln \frac{3\mathrm{R}_{\max}}{\varepsilon(1-\gamma)} \rceil$; $t \leftarrow 0$        ▷ Init. escape time; timer
  $E \leftarrow \frac{\varepsilon}{2T\mathrm{R}_{\max}}$        ▷ Compute escape probability threshold
  $\mathrm{M} \leftarrow \left\lceil \ln\left(\frac{|\mathcal{S}||\mathcal{A}|\left(2^{|\mathcal{S}|} - 2 + g\right)}{\delta}\right) \max\left(\frac{1}{2\beta^2}, \frac{\mathrm{R}_{\max}^2}{2\alpha^2}\right) \right\rceil$
  $\forall s \in \mathcal{S}, a \in \mathcal{A} : m_{s,a} \leftarrow 0$        ▷ Per-$(s,a)$ visitation counters
  $\mathcal{S}_{\text{unk}} \leftarrow \mathcal{S}$; $\mathcal{S}_{\text{out}} \leftarrow \emptyset$; $\mathcal{S}_{\text{inn}} \leftarrow \emptyset$        ▷ Init. all states to unknown
  $\hat{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \hat{\mathbf{R}}, \hat{\boldsymbol{T}}, \gamma \rangle \leftarrow \langle \mathcal{S}, \mathcal{A}, s \mapsto \mathbf{0}, s \mapsto \mathbb{1}_s, \gamma \rangle$
end procedure

procedure AgentStep($s, \mathcal{W}(\cdot)$)
  if $s' \in \mathcal{S}_{\text{unk}}$ then        ▷ Successful escape
    $t \leftarrow 0$
    return $a_{\text{xpr}} \leftarrow \operatorname{argmin}_{a \in \mathcal{A}} m_{s,a}$        ▷ Balanced walk step
  end if
  if $t > 0$ then        ▷ Ongoing attempt to escape to $\mathcal{S}_{\text{unk}}$
    $t \leftarrow t - 1$
    return $a_{\text{xpr}} \leftarrow \pi_{\text{esc}}(s, t)$        ▷ Explore $a$ from escape policy $\pi$
  end if
  if $s \in \mathcal{S}_{\text{inn}}$ then        ▷ Return exploit policy
    return $\pi_{\text{xpt}} \leftarrow \operatorname{argmax}_{\pi \in \Pi_{\mathcal{M}}} \mathcal{W}\left(\hat{V}_1^\pi(s), \hat{V}_2^\pi(s), \ldots, \hat{V}_g^\pi(s)\right)$
  else        ▷ $s \in \mathcal{S}_{\text{out}}$, begin escape attempt
    $t \leftarrow T$
    return $a_{\text{xpr}} \leftarrow \pi_{esc}(s, t)$
  end if
end procedure

procedure AgentSarsUpdate($s, a, \boldsymbol{r}, s'$)
  if $s \in \mathcal{S}_{\text{unk}}$ then
    $m_{s,a} \leftarrow m_{s,a} + 1$        ▷ Increment visitation count
    $(\boldsymbol{E}_{s,a,m_{s,a},\mathrm{S}}, \boldsymbol{E}_{s,a,m_{s,a},\mathrm{R}}) \leftarrow (s', \boldsymbol{r})$        ▷ Add to XP buffer
    if $\min_{a \in \mathcal{A}} m_{s,a} = \mathrm{M}$ then        ▷ State $s$ is learned
      $\forall a, s : \hat{\boldsymbol{T}}_{s,a,s'} \leftarrow \frac{1}{\mathrm{M}} \sum_{i=1}^{\mathrm{M}} \mathbb{1}_{s'}(\boldsymbol{E}_{s,a,i,\mathrm{S}})$
      $\forall a : \quad \hat{\boldsymbol{R}}_{s,a} \leftarrow \frac{1}{\mathrm{M}} \sum_{i=1}^{\mathrm{M}} (\boldsymbol{E}_{s,a,i,\mathrm{R}})$
      $\pi_{esc} \leftarrow \operatorname{argmax}_{\pi \in \Pi_T} \sum_{s \in \mathcal{S}} \mathbb{P}\left(\bigvee_{i=1}^T s_i \in \mathcal{S}_{\text{unk}} \middle| \pi, s_1 = s\right)$
      $\mathcal{S}_{\text{unk}} \leftarrow \mathcal{S}_{\text{unk}} \setminus \{s\}$
      $\mathcal{S}_{\text{out}} \leftarrow \left\{ s \in \mathcal{S} \setminus \mathcal{S}_{\text{unk}} \middle| \mathbb{P}\left(\bigvee_{i=1}^T s_i \in \mathcal{S}_{\text{unk}} \middle| \pi_{esc}, s_1 = s\right) \geq E \right\}$
      $\mathcal{S}_{\text{inn}} \leftarrow \mathcal{S} \setminus (\mathcal{S}_{\text{unk}} \cup \mathcal{S}_{\text{out}})$
    end if
  end if
end procedure

~ Fair-Adversarial-MDP Interaction Loop ~

procedure AgentEnvironmentInteract($\mathcal{M}, \epsilon, \delta, \mathrm{R}_{\max}$)
  AgentInit($\mathcal{M}, \epsilon, \delta, \mathrm{R}_{\max}$)
  $s \leftarrow$ Adversary        ▷ Adversarially select initial state
  while True do
    $\mathcal{W}(\cdot) \leftarrow$ Adversary        ▷ Adversarial welfare $\mathcal{W}(\cdot)$
    $z \leftarrow$ AgentStep($s, \mathcal{W}(\cdot)$)
    if $z \in \mathcal{A}$ then        ▷ Explore Action
      $s', \boldsymbol{r} \leftarrow \mathcal{M}(s, z)$
      AgentSarsUpdate($s, a, \boldsymbol{r}, s'$)
      $s \leftarrow s'$
    else if $z \in \Pi_{\mathcal{M}}$ then        ▷ Exploit Policy
      $s \leftarrow$ Adversary        ▷ Adversarial subsequent state
    end if
  end while
end procedure
```

As in the classic E^3 algorithm, within $\mathcal{S}_{\text{inn}}$, if all Chernoff bounds hold simultaneously, the value functions of the empirical MDP approximate the value functions of the true MDP, and furthermore, due to a Lipschitz assumption on welfare functions, optimizing welfare in the empirical MDP ε-optimizes welfare in the true MDP. Therefore, at each time step, if the agent is in $\mathcal{S}_{\text{inn}}$, it outputs a near optimal policy, otherwise if it is in the outer-known set, it begins an escape attempt, which either proceeds for T steps or until a state in $\mathcal{S}_{\text{unk}}$ is reached; if the agent is in $\mathcal{S}_{\text{unk}}$, it executes an action learning more about the unknown state, possibly moving it into a known set.

Definition 1. *Let* $\mathrm{TVD}(x, y)$ *denote the* total variation distance *between probability distributions* x, y. *An* $(\alpha,\ \beta)$ *uniform approximation* $\mathcal{M}' = \langle S, A, T', R', \gamma \rangle$ *of a vector-reward MDP* $\mathcal{M} = \langle S, A, T, R, \gamma \rangle$ *is an MDP that satisfies:*

1. $\forall (s, a) \quad \mathrm{TVD}\left(T'(\cdot|s, a), T(\cdot|s, a)\right) \leq \alpha$, *&*
2. $\forall (s, a) \quad |R'(s, a) - R(s, a)| \leq \beta$.

Lemma 1. *Suppose MDP* $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{R}, \boldsymbol{T}, \gamma \rangle$, *& let*

$$m \doteq \mathrm{M}(|\mathcal{S}|, |\mathcal{A}|, g, \mathrm{R}_{\max}, \alpha, \beta, \delta) \doteq \left\lceil \ln\left(\frac{|\mathcal{S}||\mathcal{A}|\left(2^{|\mathcal{S}|} - 2 + g\right)}{\delta}\right) \max\left(\frac{1}{2\beta^2}, \frac{\mathrm{R}_{\max}^2}{2\alpha^2}\right) \right\rceil \ .$$

Now if $\hat{\mathcal{M}}$ *is estimated from taking each (state, action) pair at least* m *times, then, with probability at least* $1 - \delta$, $\hat{\mathcal{M}}$ *is an* α-β *uniform approximation of* $\mathcal{M}$.

Theorem 1. *Algorithm 1* ε-δ *fair-adversarial-MDP learns* $\mathcal{M}$ *for all* $\delta \leq \frac{\epsilon}{2}$ *with sample complexity*

$$m_{\mathcal{M}}(\ldots) \leq \left\lceil \frac{|\mathcal{S}||\mathcal{A}| \cdot \mathrm{M}(|\mathcal{S}|, |\mathcal{A}|, g, \mathrm{R}_{\max}, \alpha, \frac{\beta}{\lambda}, \frac{\delta}{2}) + 3 \ln \frac{3}{\delta}}{T} \right\rceil \in \mathrm{Poly}\left(|\mathcal{S}|, |\mathcal{A}|, g, \mathrm{R}_{\max}, \lambda, \varepsilon, \frac{1}{\delta}, \frac{1}{\gamma}\right) \ .$$

Essentially, fair E^3 differs very little from E^3, as it will always seek to explore any state that is reachable with nonnegligible probability. However, it must explore each state more times than standard E^3 to account for learning *vector-valued rewards*, and the *exploitation* aspect changes greatly, as it must output *policies* that are nearly *welfare-optimal*, rather than just *actions*.

4 Discussion

We now note that in sensitive contexts, the decision to learn a policy from scratch is rather radical, and many suboptimal actions will likely be taken during the learning process. However, this isn't specific to fairness, but is rather an inherent problem in reinforcement learning in sensitive settings. In medical contexts, Thomas et al. [2019] learn starting from a *reference policy*, and seek to improve the policy while guaranteeing the learned policy is *no worse than* the reference. While this is laudable and reasonable in high-risk or sensitive settings, when *fairness between groups* is a concern, it is inherently a conservative approach (i.e., one which reinforces the status quo; in some sense comparison to reference is an *argumentum ad traditionem*), whereas starting *ex-nihilo* solely depends on the *structure of the MDP* and the *learning algorithm*, rather than existing societal bias which may be encoded in a reference policy.

Still, we note that suboptimal exploration actions taken during exploration could adversely affect one group or another in an unfair way, and in practice this is extremely important and should be monitored and controlled for. However, unlike in some bandit settings, where the cost of exploration may be unfairly borne by one group or another, in our algorithm, the escape policy and balanced walk actions are both completely independent of the reward structure in the MDP, and are thus inherently fairness agnostic. We note also that the number of suboptimal actions can be further reduced by a more careful analysis; for instance the sample complexity of learning transition matrices is much smaller when they are *sparse*, admit a *factoring*, or destination distributions are *far from uniform*, and the simple complexity of learning rewards is much smaller when the variance of rewards is also considered, or when per-beneficiary rewards are not-independent. We are hopeful hopeful that future work will lead to adversarial fair MDP learners that makes fewer suboptimal actions over the course of learning (i.e., have improved sample complexity).

In Conclusion We motivate and define a welfare-centric concept of fair reinforcement learning. Naïve approaches, like planning via policy iteration, and turn-based exploration strategies (as in multi-task learning settings) do not yield successful fair RL agents, even asymptotically. However, we show that under mild regularity conditions on the welfare function, it is possible to learn in the adversarial fair MDP framework while making polynomially many mistakes (algorithm 1, theorem 1). Our method adopts the classic E^3 algorithm, which is an appropriate fit, as its exploration strategy is actually independent of the reward function. As a result, the only change we require is attempting each action more often during exploration to account for the larger number of parameters that must be learned.

References

A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. Fair clustering through fairlets. *arXiv preprint arXiv:1802.05733*, 2018.

C. Cousins. An axiomatic theory of provably-fair welfare-centric machine learning. In *Advances in Neural Information Processing Systems*, 2021.

C. Cousins. Uncertainty and the social planner's problem: Why sample complexity matters. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian. Runaway feedback loops in predictive policing. In *Conference on Fairness, Accountability and Transparency*, pages 160–171. PMLR, 2018.

M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

J. Kleinberg, J. Ludwig, S. Mullainathan, and A. Rambachan. Algorithmic fairness. In *AEA papers and proceedings*, volume 108, pages 22–27, 2018.

S. Lohr. Facial recognition is accurate, if you're a white guy. *New York Times*, 9(8):283, 2018.

P. S. Thomas, B. Castro da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019.