On Welfare-Centric Fair Reinforcement Learning

Cyrus Cousins[*] cbcousins@umass.edu
Kavosh Asadi Amazon
Elita Lobo[*] elobo@umass.edu
Michael L. Littman[†] mlittman@cs.brown.edu

Abstract

We propose a welfare-centric fair reinforcement-learning setting, in which an agent enjoys *vector-valued* reward from a set of beneficiaries. Given a *welfare function* $\mathrm{W}(\cdot)$, the task is to select a policy $\hat{\pi}$ that approximately optimizes the welfare of their value functions from start state s_0, i.e., $\hat{\pi} \approx \operatorname{argmax}_\pi \mathrm{W}\big(\mathbf{V}_1^\pi(s_0), \mathbf{V}_2^\pi(s_0), \ldots, \mathbf{V}_g^\pi(s_0)\big)$. We find that welfare-optimal policies are stochastic and start-state dependent. Whether *individual actions* are mistakes depends on the *policy*, thus mistake bounds, regret analysis, and PAC-MDP learning do not readily generalize to our setting. We develop the *adversarial-fair KWIK* (Kwik-Af) learning model, wherein at each timestep, an agent either takes an *exploration action* or outputs an *exploitation policy*, such that the number of exploration actions is bounded and each exploitation policy is ε-welfare optimal. Finally, we reduce PAC-MDP to Kwik-Af, introduce the *Equitable Explicit Explore Exploit* (E^4) learner, and show that it Kwik-Af learns.

Keywords: Fair RL · Vector-Valued MDP · PAC-MDP · KWIK Learning

1 Introduction

As the negative societal consequences of machine learning (ML) run amok become increasingly apparent, fair ML methods have seen increased attention for tasks like facial recognition (Buolamwini and Gebru, 2018; Cook et al., 2019; Cavazos et al., 2020) and hiring (Kleinberg et al., 2018; Raghavan et al., 2020). Despite this positive trend, most attention on the theory side has been focused on fair supervised (Agarwal et al., 2018; Thomas et al., 2019; Cousins, 2021) and unsupervised (Chierichetti et al., 2017; Chhabra et al., 2021) learning, whereas the second-order societal-welfare impact of ML models, such as the runaway positive feedback loops in settings like predictive policing (Ensign et al., 2018; Alikhademi et al., 2021), are more naturally posed as reinforcement learning (RL) problems.

We apply ideas from welfare-centric supervised learning to the RL setting; in particular, we assume an agent receives a vector-valued reward signal from a set of *beneficiaries*, each representing, e.g., different racial, gender, or religious groups, and the task is to learn a single policy that treats beneficiaries fairly. We argue that it is not our role as algorithm designers to dictate *what fairness means*, or how one should *compromise among beneficiaries*, but rather we should seek to optimize for a given fairness notion (ideally one agreed upon by society, government, impacted groups, political philosophers, and other interested parties), as encapsulated by a metric of *societal welfare*. In supervised learning, this is relatively straightforward, as we generally maximize the welfare of *expected per-beneficiary value* (Cousins, 2023b), and in our setting, we take utility to be the standard *geometrically discounted reward* (value) for each beneficiary. In general, decision making to optimize 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 *planning* and *asymptotic learning*, we also ask *how quickly* we can learn to act fairly in an unknown MDP. 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 the agent's policy

[*]University of Massachusetts Amherst, College of Information and Computer Sciences
[†]Brown University, Department of Computer Science

(i.e., tradeoffs among beneficiaries should be balanced). To address this issue, we combine ideas from the PAC-MDP framework and KWIK (Know What It Knows) learning (Li et al., 2011) to create the *adversarial fair KWIK MDP* learning framework (KWIK-AF). We require a KWIK-AF agent to explicitly output at each step either a *fair policy* or an *exploration action*, and with high probability the agent must always output ε-optimal fair policies while taking only a bounded number of exploration actions over its infinite lifetime. For the sake of generality, 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. Finally, we introduce an algorithm inspired by the classic E^3 algorithm of Kearns and Singh (2002), which we call *Equitable Explicit Explore Exploit* (E^4), and show that it is a KWIK-AF learner.

We summarize our contributions below.

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*, each with their own reward function.
2. Using ideas from vector-valued RL, econometrics, and social welfare theory, we establish the goal of learning policies to optimize the *welfare* of per-beneficiary expected discounted rewards.
3. Section 3 introduces the *adversarial fair KWIK MDP* (KWIK-AF) learning framework, in which an agent learns only from *exploration actions*, and an adversary moves the agent when the agent outputs an *exploitation policy*. W.h.p., a learner must output only ε-optimal exploitation policies and take polynomially many exploration actions. We assess *policies* rather than *actions*, since welfare-optimal policies may be *stochastic* and *start-state dependent*, thus actions can not be assessed without context.
4. In section 4, we first present efficient welfare-optimal planning routines, then we discuss exploration in fair RL, define the E^4 algorithm, and show that it is a KWIK-AF learner.

1.1 Related Work

With the rapid adoption of ML algorithms, authors such as Thomas et al. (2019) note that it is imperative to ensure such algorithms are well-behaved, and do not perpetuate harmful biases. Many works study fairness in supervised and unsupervised learning with various fairness definitions. The welfare-centric approach has recently seen success as a generic solution to fair compromise among the wants and needs of various groups, but has thus far been studied primarily in supervised learning (Cousins, 2021; 2022; 2023b; Cousins et al., 2024). Defining fairness in the RL setting is particularly challenging due to the sequential nature of RL decision-makers (Thomas et al., 2019; Jabbari et al., 2017), as we must also decide how fair decisions should be distributed over time.

There is a rich body of literature on multi-objective sequential decision making, which arises naturally in bandit settings (Metevier et al., 2019; Chen et al., 2020), and more generally in planning and RL (Roijers et al., 2013). One approach to fairness is to optimize some objective subject to *fairness constraints*, usually requiring approximate parity among groups. In *contextual bandit settings*, Metevier et al. (2019) learn and plan while (probabilistically) satisfying various fairness constraints. Similarly, Wen et al. (2021) show guarantees for learning and planning in MDPs under parity constraints on per-group value functions, Satija et al. (2021) propose finding policies that improve returns while also satisfying certain group fairness constraints, and Satija et al. (2022) generalize their setting by allowing not just rewards, but also the transition function, to differ among groups.

In welfare-centric RL, the final objective is a (nonlinear) function of per-group objectives (value functions). Assuming monotonicity of welfare, the optimal policy lies on the Pareto frontier of feasible utility vectors (Van Moffaert and Nowé, 2014). Lizotte et al. (2012) show how to identify globally dominated actions in the multi-objective case under linear function approximation, and Siddique et al. (2020); Yu et al. (2023) study the problem of learning welfare-optimal for multi-objective deep RL. Cousins et al. (2022) consider welfare objectives in a similar tabular setting, and Fan et al. (2023) show how to plan for Nash social welfare, by leveraging its differentiability and linearizability. This work theoretically treats fair RL, in particular analyzing *sample complexity*, which has been a core tool for studying exploration in RL (Kearns and Singh, 2002; Kakade, 2003; Li et al., 2011).

1.2 Background

We now present relevant background material on RL, fairness, and welfare-centric ML.

Reinforcement Learning Reinforcement learning (RL) is the study of an environment and an agent that learns to maximize reward through environmental interaction. The Markov decision process (Puterman, 1994), or MDP, is the standard mathematical formalism of RL. Standard single-beneficiary MDPs are specified by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathrm{R}, \gamma \rangle$. Here the *environment* is described by the *state set* $\mathcal{S}$, *action set* $\mathcal{A}$, and *transition function* $\mathbf{P}(s,a) : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, where $\mathcal{P}(\mathcal{S})$ denotes the probability simplex over $\mathcal{S}$. The *agent's goals* or *desires* are then encoded by the *reward function* $\mathrm{R}(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$, which may be randomized, and the *discount rate* γ, which *geometrically downweights* future rewards, representing a preference for near-term rewards over delayed gratification.

The standard goal in the RL problem is to learn a policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ that can achieve high sums of future discounted rewards. An important concept in RL is the value function, defined as

$$\mathrm{V}^{\pi}(s) \doteq \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim \mathbf{P}(s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \mathrm{R}(s_t, a_t) \,\middle|\, s_0 = s \right] = \mathop{\mathbb{E}}_{\substack{a_0 \sim \pi(s_0) \\ s_1 \sim \mathbf{P}(s_0, a_0)}} \left[\mathrm{R}(s_0, \pi(s_0)) + \gamma \mathrm{V}^{\pi}(s_1) \,\middle|\, s_0 = s \right] .$$

The value function $\mathrm{V}^{\pi}(s)$ describes the *expected utility* of the following policy π at state s. RL often adopts a *egocentric view*, in which the scalar-valued reward function $\mathrm{R}(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is intrinsic to the agent, who selfishly wishes to optimize their wellbeing (as measured by the value function).

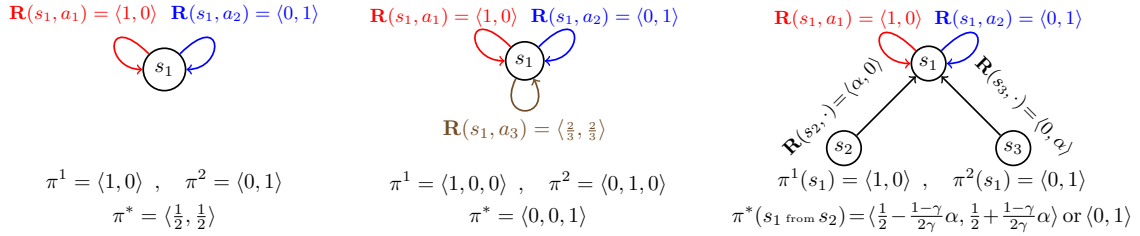
On Welfare In this paper, we are interested in *vector-valued* or *multi-beneficiary* MDPs, denoted $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{R}, \gamma \rangle$. The state set $\mathcal{S}$, action set $\mathcal{A}$, transition function $\mathbf{P}$, and discount factor γ are exactly as in the standard RL setting. We consciously use the term *beneficiary* to explicitly extricate the *passive nature* of preferences of those impacted by the system from the *active role* played by the *agent*. In this setting, there exist g beneficiaries, each with a corresponding reward function $\mathbf{R}_i$ and value function $\mathbf{V}_i^{\pi}$. In this work, we define the utility of a beneficiary to be the standard RL target of their geometrically discounted accumulated reward (value). The *scale* of reward is a crucial quantity, which we measure as $R_{\max} \doteq \max_{s \in \mathcal{S}, a \in \mathcal{A}} \|\mathbf{R}(s,a)\|_{\infty}$, thus utility is limited to the range $\left[0, \frac{R_{\max}}{1-\gamma}\right]$.

A *welfare function* $\mathrm{W}(\cdot) : \mathbb{R}_{\geq 0}^{g} \to \mathbb{R}_{\geq 0}$ summarizes the *utility* of all beneficiaries as a cardinal value, thus establishing a *preference* or *ranking* over policies, and our goal is select a policy to *maximize welfare*. For example, the *utilitarian welfare* and *egalitarian welfare* of value vector $\boldsymbol{v}$ are defined as

$$\mathrm{W}_1(\boldsymbol{v}) = \mathrm{W}_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_g) = \frac{1}{g} \sum_{i=1}^{g} \boldsymbol{v}_i \quad \text{and} \quad \mathrm{W}_{-\infty}(\boldsymbol{v}) = \mathrm{W}_{-\infty}(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_g) = \min_{i \in 1, \ldots, g} \boldsymbol{v}_i \ .$$

Utilitarian welfare draws on classical ideas of utilitarian philosophy (Bentham, 1789; Mill, 1863), wherein all members of society should be treated equally, and the goal is to maximize overall utility. On the other hand, *egalitarian welfare* draws from Rawlsian theory (Rawls, 1971; 2001), where the idea is that society should seek to uplift its most disadvantaged (or impoverished) members. Both can be interpreted through a mechanism design or game theoretic lens (Cousins, 2023a), wherein a Dæmon creates a society populated by the beneficiaries, and an Angel then banishes the Dæmon to join the society. If the Angel uniformly randomly selects who the Dæmon becomes, the Dæmon should maximize utilitarian welfare to maximize their expected utility. However, if the Angel adversarially selects the worst-off beneficiary, the Dæmon should instead maximize egalitarian welfare.

Utilitarian welfare is "fair" in the sense that it treats everyone *ostensibly equally*, however, it has no preference for *equity*, and can thus incentivize high utility for some beneficiaries at the cost of low utility for others. On the other hand, egalitarian welfare is "fair" in the sense that it allocates resources optimally to help those most in need first, however, beneficiaries that are difficult or impossible to satisfy may be catered to exclusively, at the expense of all others. Between these extremes, *prioritarian* welfare concepts (Parfit, 1997; Arneson, 2000) seek a middle ground, incentivizing equity by prioritizing the needs of disadvantaged people, but not to the extreme degree of egalitarianism. We now describe two prioritarian families, both of which contain utilitarian and egalitarian welfare as extreme cases, as well as a continuum of intermediate cases.



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

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

Definition 1.1 (Power-Mean Welfare)**.** *We define the power-mean family* $\mathrm{W}_p(\boldsymbol{v})$*, for power* $p \leq 1$*, for any utility vector* $\boldsymbol{v} \in \mathbb{R}^g_{\geq 0}$*, as*

$$\mathrm{W}_p(\boldsymbol{v}) \doteq \sqrt[p]{\frac{1}{g}\sum_{i=1}^{g} \boldsymbol{v}_i^p} \ , \quad \mathrm{W}_{-\infty}(\boldsymbol{v}) \doteq \min_{i \in 1,\ldots,g} \boldsymbol{v}_i \ , \quad \textit{or} \quad \mathrm{W}_0(\boldsymbol{v}) \doteq \sqrt[g]{\prod_{i=1}^{g} \boldsymbol{v}_i} \ .$$

Definition 1.2 (Gini Social Welfare)**.** *Given a decreasing stochastic weight vector* $\boldsymbol{w} \in \triangle^g$ *(i.e.,* $1 \geq \boldsymbol{w}_1 \geq \boldsymbol{w}_2 \geq \cdots \geq \boldsymbol{w}_g \geq 0$ *s.t.* $\|\boldsymbol{w}\|_1 = 1$*), the* Gini social welfare *of utility vector* $\boldsymbol{v} \in \mathbb{R}^g_{\geq 0}$ *is*

$$\mathrm{W}_{\boldsymbol{w}}(\boldsymbol{v}) \doteq \sum_{i=1}^{g} \boldsymbol{w}_i \boldsymbol{v}_i^{\uparrow} \ ,$$

where $\boldsymbol{v}^{\uparrow}$ *denotes the entries in* $\boldsymbol{v}$ *in ascending sorted order.*

These classes are intuitive from a prioritarian perspective, as the marginal gain of utility is larger for low-utility groups than for high-utility groups. This preference for equity is captured by the Pigou (1912)-Dalton (1920) transfer principle, which states that equitable redistribution of utility should never decrease welfare. Cardinal welfare theory provides axiomatizations for both the power-mean (Debreu, 1959; Gorman, 1968; Cousins, 2021; 2023b) and Gini (Weymark, 1981; Gajdos and Weymark, 2005) classes. For technical reasons, we assume the welfare function must be λ-$\|\cdot\|_\infty$ *Lipschitz continuous*, and *concavity* is often convenient for planning, but our methods treat any welfare function that meets these conditions. Cousins (2023b) shows that the power-mean family is Lipschitz continuous except when $p \in [0, 1)$, and the entire Gini family is 1-$\|\cdot\|_\infty$ Lipschitz continuous.

In the context of fair RL, our goal is, roughly speaking, to learn a policy to maximize the welfare of MDP $\mathcal{M}$ from start state s_0. In other words, we want to find $\hat{\pi}$ to approximate π^*, where

$$\mathrm{W}\left(\mathbf{V}_1^{\hat{\pi}}(s_0), \ldots, \mathbf{V}_g^{\hat{\pi}}(s_0)\right) \geq \operatorname*{argmax}_{\pi^* \in \Pi_{\mathcal{M}}} \mathrm{W}\left(\mathbf{V}_1^{\pi^*}(s_0), \ldots, \mathbf{V}_g^{\pi^*}(s_0)\right) - \varepsilon \ .$$

In section 3, we define how agents interact with their environments and receive feedback in this setting, and we make precise what it means to learn to plan fairly.

2 Illuminating Examples

Here we present a few simple examples (visualized in figure 1) to illustrate that intuition from the standard scalar-reward RL setting can be misleading. We consider the simple *egalitarian welfare* objective for two beneficiaries, on essentially stateless (single recurrent state) MDPs. Even in this elementary setting, we draw surprising conclusions as to the nature of welfare-optimal policies π^* (as compared to per-beneficiary optimal policies π^1 and π^2) and the behavior of RL algorithms (in both planning and exploration). Section 2.1 presents these simple MDPs, and section 2.2 then discusses the challenges of evaluating fair policies and the learners that produce them.

2.1 Simple Multi-Beneficiary MDPs

We first consider a basic 2-armed bandit, in which the beneficiaries prefer different arms. We then extend our analysis to allow for a third "compromise" arm. Finally, we also allow for additional transient states that immediately reward one of the beneficiaries to represent an "unfair start," wherein one beneficiary or the other is "privileged," and fair agents must learn to compensate.

One might expect, or at least hope, that convenient properties from standard RL would be preserved in the fair-RL setting. In particular, one might expect the following.

1. We need only consider deterministic stationary policies, i.e., we can assume that there always exists an optimal policy that is deterministic and stationary.
2. We can explore by letting individually beneficiaries take turns controlling the agent (thus mitigating potentially challenging learning problems with well-studied techniques).
3. A single policy is optimal from all starting states.

Unfortunately, *none of these properties hold* in the welfare setting. The examples of this section are presented to disabuse the reader of such notions.

Example 2.1 (Symmetric 2-Armed Bandit; Figure 1a)**.** *Suppose a 2-armed bandit with reward* $\mathbf{R}(s_1, a_1) \doteq \langle 1, 0\rangle$ *and* $\mathbf{R}(s_1, a_2) \doteq \langle 0, 1\rangle$*. The unique welfare-optimal stationary policy is* $\pi^* = \langle \frac{1}{2}, \frac{1}{2}\rangle$*.*

There are several surprises here:
1. The (unique) optimal policy is *stationary* (see lemma 3.1), but not *deterministic* (i.e., stochastic).
2. Policy iteration iteratively selects the greedy welfare-optimal policy, i.e., selects the policy

$$\pi^{(t+1)} \leftarrow \operatorname*{argmax}_{\pi \in \Pi_{\mathcal{M}}} \mathrm{W}\left(\mathop{\mathbb{E}}_{\pi, s_1}\left[\mathbf{R}_1(s_0, \pi(s_0)) + \gamma \mathbf{V}_1^{\pi^{(t)}}(s_1)\right], \ldots, \mathop{\mathbb{E}}_{\pi, s_1}\left[\mathbf{R}_g(s_0, \pi(s_0)) + \gamma \mathbf{V}_g^{\pi^{(t)}}(s_1)\right]\right) , \quad (1)$$

where $\pi^{(t)}$ is the policy selected at iterate t. This would optimize the policy in one step if updating the policy did not impact the value function, and this strategy is *convergent* for linear (value) MDP objectives. However, policy iteration for the egalitarian welfare objective, initiated at either deterministic policy, *oscillates* between $\pi(s_1) = \langle 1, 0\rangle$ and $\pi(s_1) = \langle 0, 1\rangle$ for any $\gamma \geq \frac{1}{2}$. This occurs since, assuming $\pi^{(t)}(s_1) = \langle 1, 0\rangle$, the (stale) value function is $\mathbf{V}^{\pi^{(t)}}(s_1) = \langle \frac{1}{1-\gamma}, 0\rangle$, thus taking $\pi^{(t+1)}(s_1) = \langle 0, 1\rangle$ maximizes egalitarian welfare at $\min(\frac{\gamma}{1-\gamma}, 1) = 1$ in (1). In other words, each iteration *overcorrects* for initial policy unfairness, yielding oscillatory behavior. Notably, for $\gamma < \frac{1}{2}$, the oscillation is damped and (1) actually converges to the optimal stochastic policy, but this is case-specific, and policy iteration is not in general a valid planning strategy for welfare objectives.

We now consider an extension to this MDP that includes a third arm (action), which is not preferred by either beneficiary, but is more effective as a compromise than any mixture of the first two arms.

Example 2.2 (Compromise 3-Armed Bandit; Figure 1b)**.** *Suppose reward* $\mathbf{R}(s_1, a_1) \doteq \langle 1, 0\rangle$*,* $\mathbf{R}(s_1, a_2) \doteq \langle 0, 1\rangle$*, and* $\mathbf{R}(s_1, a_3) \doteq \langle \frac{2}{3}, \frac{2}{3}\rangle$*. The unique optimal stationary per-beneficiary and welfare-optimal policies are* $\pi^1 = \langle 1, 0, 0\rangle$*,* $\pi^2 = \langle 0, 1, 0\rangle$*, and* $\pi^* = \langle 0, 0, 1\rangle$*, respectively.*

This example starkly illustrates how different the welfare-optimal policy π^* may be from the per-beneficiary optimal policies π^1 and π^2. Observe that π^* is not a linear combination of π^1 and π^2; these policies are totally disjoint, as no two optimal policies will ever prescribe the same action.

This divergence in optimal policies also has implications for how the MDP should be explored. For instance, if beneficiaries 1 and 2 are independently allowed to run a UCB-style algorithm (Auer et al., 2008), in all likelihood, neither will even bother to adequately a_3, thus even together they do not collect the appropriate information for welfare-optimal planning. We can conclude that, not only the planning, but also the exploration aspect of RL is "more than the sum of its parts," as under welfare objectives, there is an obligation to explore the MDP more thoroughly.

We now extend the 2-armed bandit example further by adding two additional states, which represent disparate starting conditions that favor one beneficiary or the other.

Example 2.3 (Symmetric 2-Armed Bandit, with Asymmetric Starting Conditions; Figure 1c)**.** *Suppose an MDP with recurrent state s_1 and transient states s_2 and s_3, thus the environment is a 2-armed bandit from s_1. Any action from s_2 yields reward α to beneficiary 1, and any action from s_3 yields reward α to beneficiary 2. Upon reaching state s_1, the MDP is identical to example 2.1.*

From s_1, neither beneficiary is privileged, and the recurrent MDP matches example 2.1, but from s_2 or s_3, some beneficiary begins with an advantage of α utility. The unique optimal stationary policy thus selects $\pi^*(s_1)$ to benefit the disadvantaged group. Starting from s_2, to achieve equity, we require

$$\tfrac{\gamma}{1-\gamma}\pi^*(s_1,a_1)+\alpha=\mathbf{V}_1^{\pi^*}(s_2)=\mathbf{V}_2^{\pi^*}(s_2)=\tfrac{\gamma}{1-\gamma}(1-\pi^*(s_1,a_1))\ ,\quad \textit{thus}\quad \pi^*(s_1)=\langle\tfrac{1}{2}-\tfrac{1-\gamma}{2\gamma}\alpha,\tfrac{1}{2}+\tfrac{1-\gamma}{2\gamma}\alpha\rangle$$

or $\pi^*(s_1)=\langle 0,1\rangle$ if equality is infeasible. By symmetry, starting at s_3 swaps these action probabilities.

2.2 On Evaluation and Optimality of Fair Learners

When we consider example 2.3, two extremely subtle points arise as to how we are to *evaluate the performance* of a learner and the actions it makes. First, for any $\alpha<\frac{\gamma}{\gamma-1}$, welfare-optimal stationary policies are stochastic at s_1, (i.e., actions a_1 and a_2 are both taken with nonzero probability). It is thus impossible to determine whether *individual actions* taken by a learner are fair *in isolation*, and a simple mistake-bound style of analysis thus seems inapplicable. This issue is not unique to fair RL, as it arises whenever *no deterministic policy is optimal*, for instance in game-theortic multi-agent RL settings (Buşoniu et al., 2010), and also with various constrained (Prashanth and Ghavamzadeh, 2016) or risk-averse (Wang and Chapman, 2022) RL objectives. In particular, evaluating a policy requires the *probability distribution over actions*, not just the individual actions taken over the course of executing the policy. An issue more specific to our fair RL setting is that the optimal policy π^* depends on the *start state*, so it is meaningless to decompose the learning process into a sequence of individual decisions at each timestep and evaluate them independently, as this erases the *context* (i.e., the start state of welfare-optimal actions) in which decisions are made.

The next section explores these issues further and derives an appropriate learning model that evaluates agents — not just on individual actions, but on their ability to output nearly welfare-optimal policies. Evaluating fair RL agents is deceptively tricky, particularly due to the contextual nature of start-state dependent policies. Due to the complexity of introducing the context of a starting state, we adopt an *adversarial setting*, in which many design decisions — in particular those regarding episodic versus continuous learning, choice of start states or distributions, and the welfare function — are made adversarially. Section 4 then shows that even in this general adversarial setting, fair learning remains possible and algorithmically practical.

3 A Model of Adversarial Fair Reinforcement Learning

In this section, we review the PAC-MDP framework, explain why a straightforward generalization to fairness-sensitive settings is troublesome, and define the KWIK-AF framework. Our guarantees are similar to the classical E^3 policy-centric guarantees of Kearns and Singh (2002), but are adapted to *adversarial* state and welfare-function selection. Both are important to the welfare-centric RL setting, as the adversary can be used to model how policies generated by the agent are actually used (and thus how they impact society), as well as shifts in human fairness concepts over time. Furthermore, we show constructively via reduction that our setting is at least as hard as the PAC-MDP framework. Before further describing the learning setting, we lead with a key lemma that allows us to restrict our attention to start-state dependent stationary stochastic MDPs.

Lemma 3.1 (Optimality of Stationary Policies: Lemma 3.1 of Siddique et al. (2020))**.** *For any start state $s_0\in\mathcal{S}$, there exists some $\mathrm{W}(\cdot)$-optimal policy*

$$\pi^*\doteq\operatorname*{argmax}_{\pi\in\Pi_{\mathcal{M}}}\mathrm{W}\left(\mathbf{V}_1^{\pi}(s_0),\ldots,\mathbf{V}_g^{\pi}(s_0)\right)$$

that is a stationary stochastic policy*, i.e., given the current state s_t, $\pi^*(s_t)$ may prescribe distributions over actions, but they may not depend on the history other than s_0 (i.e., not on $s_1,s_2,\ldots,s_{t-1}$).*

3.1 Motivation

Our framework introduces two major ideas. First, we explicitly model the explore-exploit tradeoff by requiring learners to either take *exploration actions* when uncertain about how to behave, *or* to output *exploitation policies* when they can ε-optimally plan from the current state. In particular, we require that, with high probability, the agent takes a *bounded* (usually polynomial) number of exploration actions, and every exploitation policy is *ε-welfare-optimal*. Second, many decisions in our learning model are made adversarially, and thus our model encompasses a plethora of related settings, including episodic, continual, teacher-assisted, fair, and single-beneficiary (or egocentric) RL settings. Consequently, our algorithms and analysis can be directly applied to these more specific settings. The central motivation for our policy-centric framework is that simple per-action regret or mistake bounds don't translate to the fair RL setting. This is because, as discussed in section 2.2, it is not possible to evaluate the optimality of *individual actions* of a fair learner, as they may be stochastic, and they may also depend on the *context* of the start state s_0.

Ideally, we could still ensure the agent behaved ε-optimally during learning, however, because fair policies are inherently contextual, it also does not make sense to have the learner follow its own policies *at each timestep*, as these policies may disagree, so from where would we even measure suboptimality? While resetting the start state at each step ignores historical context, indefinitely using the agent's original start state puts *too much emphasis* on the past, as from a geometric discounting perspective, we are only planning for optimal behavior in a geometric-length episode, and as time progresses, the start state should become irrelevant in any recurrent MDP. In either case, the agent behaves poorly in some sense during learning; one may consider example 2.3 starting from state s_2, where keeping the start state indefinitely favors beneficiary 2, whereas resetting it each step favors beneficiary 1 (as their initial privilege is never addressed).

There are many reasonable ways to resolve this issue, but we wish not to limit our framework by committing to one of them. For example, running the agent's policy for a geometric-length episode before returning control to the agent (to choose either an exploration action or to output another exploitation policy) would ensure that *behavior* during policy execution is fair. However, even here, reasonable design decisions abound: After a policy execution episode, should we continue from the current state, or start afresh from a new state? If we restart, should the start state be drawn i.i.d., or might its distribution change over time? Should the welfare function be fixed, or could it too change over time to reflect evolving societal values or shifting demographics? Rather than adopt some fixed control flow, we require agents to behave ε-optimally against a largely adversarial system.

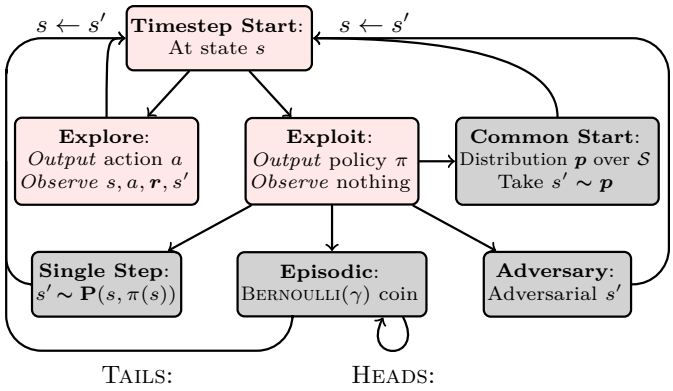
Essentially, the adversary provides modular flexibility to fairness-sensitive decisions and parameters, and robustness against a learning agent exploiting the structure of the learning procedure. This preëmpts fairness issues arising from a limited model, by requiring that the agent itself must operate under *general* (adversarial) conditions, which modelers may select to fit domain-specific conditions and ideals of fairness. Furthermore, while *exploitation policies* are guaranteed to be *ε-welfare-optimal*, how they are *actually used* is equally important to fairness. In this context, *adversarial state selection* may be interpreted as taking arbitrary real-world actions informed by the agent's policy, which should be ε-optimal, before returning control to the agent.

3.2 MDP Policy Agents and the Adversarial Fair KWIK Framework

We now define the MDP policy agent, which codifies how a learner interacts with its environment. This interface is more complicated than standard PAC-MDP learners, because it explicitly models both exploration and exploitation, but this complexity is necessary to disambiguate good from bad decisions in rich environments where individual actions do not suffice.

Definition 3.2 (MDP Policy Agent)**.** *An* MDP policy agent *interacts with an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathrm{R}, \gamma \rangle$ by starting from some start state s_0. At any timestep t, at state s_t, the agent then produces either an exploration action or an exploitation policy $z \in \mathcal{Z}$ from the space*

$$\mathcal{Z} \doteq \underbrace{\mathcal{A}}_{\textsc{Exploration Action}} \bigcup \underbrace{\Pi_{\mathcal{M}}}_{\textsc{Exploitation Policy}}.$$




```
1: procedure DeployAgent(S, A, γ, ε, δ, R_max, s)
2:   AgentInit(S, A, γ, R_max, ε, δ)
3:   loop
4:     W(·) ← Adversary                ▷ Adversarial welfare
5:     z ← AgentStep(s, W(·))
6:     if z ∈ A then                   ▷ Explore action
7:       s', r ← M(s, z)               ▷ Take action z in M
8:       AgentSarsUpdate(s, a, r, s'); s ← s'
9:     else if z ∈ Π_M then            ▷ Exploit policy
10:      case Single Step
11:        s ~ P(s, z(s))
12:      case Episode
13:        while Bernoulli(γ) do
14:          s ~ P(s, z(s))
15:      case Common Start
16:        s ~ p
17:      case Adversary
18:        s ← Adversary               ▷ Adversarial state
```

Figure 2: Illustration of MDP Policy Agent control flow. The flowchart (left) describes the control flow of the pseudocode (right). An MDP policy agent must implement the AgentInit(...), AgentStep(...), and AgentSarsUpdate(...) subroutines to interact with the environment.

If the agent outputs an exploration action a_t*, it is executed in state* s_t *of* $\mathcal{M}$ *to produce reward* $r_{t+1} \sim \mathrm{R}(s_t, a_t)$ *and subsequent state* $s_{t+1} \sim \mathbf{P}(s_t, a_t)$*, and the agent observes* $\langle s_t,\ a_t\ r_{t+1},\ s_{t+1} \rangle$.

Alternatively, if the agent outputs an exploitation policy π_t*, a new state* s_{t+1} *is produced, and the agent observes the new state, but the agent* does not *observe any reward or action. There are many reasonable models for selecting* s_{t+1}*, and we propose any of the following models:*

1. ***Single Step****: The agent's policy is run for a single step, yielding* $s_{t+1} \sim \mathbf{P}(s_t, \pi_t(s_t))$*;*
2. ***Episode:*** *Let* $k \sim$ *Geometric*$(1-\gamma)$*,* $s'_0 = s_t$*, and* $s'_{i+1} \sim \mathbf{P}(s'_i, \pi_t(s'_i))$*, then take* $s_{t+1} = s'_k$*;*
3. ***Common Start:*** *Given some* start-state distribution $\boldsymbol{p} \in \mathcal{P}(\mathcal{S})$*, we take* $s_{t+1} \sim \boldsymbol{p}$*; or*
4. ***Adversary****:* s_{t+1} *is selected adversarially.*

Figure 2 illustrates the control flow of this system. Note that definition 3.2 resembles the interface of the standard E^3 algorithm, in which an agent takes individual exploration actions until it is ready to output an ε-optimal policy from its current state, at which point it terminates. We require our agents to continue operating after producing an exploitation policy, and based on the discussion of section 3.1, we present several reasonable modes of operation following an agent producing policies, but all are encompassed by *adversarial choice* of subsequent state. To describe *successful* or *efficient* MDP-policy agents, we define the *policy-KWIK* class, which resembles the KWIK framework for supervised learning (Li et al., 2011), in the sense that the agent is issued "queries" (what to do at the current state), and the agent may either say "I don't know" to receive information (i.e., issue an exploration action to receive reward and transition samples), or answer the query (give a policy).

Definition 3.3 (Policy-KWIK Learner)**.** *An MDP policy agent is a* policy-KWIK learner *with sample complexity* $\mathrm{m}(|\mathcal{S}|, |\mathcal{A}|, \gamma, R_{\max}, \varepsilon, \delta)$ *if, for any error tolerance* $\varepsilon > 0$ *and failure probability* $\delta \in (0, 1)$*, the following pair of conditions hold with probability at least* $1 - \delta$*.*

1. ***Exploration condition:*** *The number of exploration actions is bounded, i.e.,*
$$\sum_{t=1}^{\infty} \mathbb{1}_{\mathcal{A}}(z_t) \leq \mathrm{m}\big(|\mathcal{S}|, |\mathcal{A}|, \gamma, R_{\max}, \varepsilon, \delta\big) \ .$$
2. ***Exploitation condition:*** *All exploitation policies are* ε*-optimal, i.e., for all* t*, if* $z_t \in \Pi_{\mathcal{M}}$*, then*
$$\mathrm{V}^{z_t}(s_t) \geq \sup_{\pi^* \in \Pi_{\mathcal{M}}} \mathrm{V}^{\pi^*}(s_t) - \varepsilon \ .$$

Definitions 3.2 and 3.3 explicitly delineate between exploration and exploitation. In particular, the agent output space $\mathcal{Z}$ is explicitly factored into *exploration actions*, which are used to take a single step and learn from the environment, and *exploitation policies*, through which the agent demonstrates that it knows how to act ε-optimally from the current state. This is codified in conditions 1 and 2 of

definition 3.3, as condition 1 requires that an agent may not take too many exploration actions, and condition 2 requires that each policy an agent dares to output must be ε-optimal.

Theorem 3.4 (Policy-KWIK and PAC-MDP Learners)**.** *Every policy-KWIK learner that outputs deterministic policies with polynomial sample complexity is a PAC-MDP learner, in the sense that executing an exploitation policy or exploration action at each timestep produces no more than* $\mathrm{m}(|\mathcal{S}|, |\mathcal{A}|, \gamma, R_{\max}, \varepsilon, \delta)$ *total mistakes (i.e., ε-suboptimal actions) with probability at least* $1 - \delta$*.*

Proof Sketch. Essentially, this result follows by noting that a policy-KWIK learner can be converted to a PAC-MDP learner by executing each *exploration action* a, or $\pi(s)$ for each *exploitation policy* π at state s, and in doing so, with probability at least $1 - \delta$, no exploitation action is ε-suboptimal. See appendix A for full proof of this result. □

Group-Fair Models of Reinforcement Learning Definitions 3.2 and 3.3 describe standard (scalar-valued) learning settings, so we now generalize them to definitions 3.5 and 3.6 to model efficient fair learning with welfare objectives for multiple beneficiaries.

Definition 3.5 (Adversarial Fair MDP Policy Agent)**.** *At each timestep t, at state s_t, the adversary presents a welfare function $\mathrm{W}_t(\cdot)$ from some class $\mathcal{W}$. The agent then produces either an exploration action or an exploitation policy $z \in \mathcal{Z}$ from the space*

$$\mathcal{Z} \doteq \underbrace{\mathcal{A}}_{\textsc{Exploration Action}} \bigcup \underbrace{\Pi_{\mathcal{M}}}_{\textsc{Exploitation Policy}} .$$

At this point, if the agent selected an exploration action a_t, the action is executed in state s_t of the MDP to produce reward $\boldsymbol{r}_{t+1} \sim \mathbf{R}(s_t, a_t)$ and subsequent state $s_{t+1} \sim \mathbf{P}(s_t, a_t)$, and the agent observes the tuple $\langle s_t, a_t, \boldsymbol{r}_{t+1}, s_{t+1} \rangle$. Alternatively, if the agent selected exploitation policy π_t, the adversary then selects the next state s_{t+1}, and the agent does not *observe any reward or action.*

Definition 3.6 (Kwik-Af Learner)**.** *An agent is Kwik-Af over welfare class $\mathcal{W}$ with sample complexity* $\mathrm{m}(|\mathcal{S}|, |\mathcal{A}|, \gamma, R_{\max}, g, \varepsilon, \delta)$ *if, for any error tolerance $\varepsilon > 0$ and failure probability $\delta \in (0, 1)$, the following pair of conditions hold with probability at least $1 - \delta$:*

1. ***Exploration condition:*** *The number of exploration actions is bounded, i.e.,*

$$\sum_{t=1}^{\infty} \mathbb{1}_{\mathcal{A}}(z_t) \leq \mathrm{m}\big(|\mathcal{S}|, |\mathcal{A}|, \gamma, R_{\max}, g, \varepsilon, \delta\big) .$$

2. ***Exploitation condition:*** *All exploitation policies are ε-optimal (with respect to the welfare function $\mathrm{W}_t \in \mathcal{W}$ provided by the adversary at each timestep t), i.e., for all t, if $z_t \in \Pi_{\mathcal{M}}$, then*

$$\mathrm{W}_t\big(\mathbf{V}_1^{z_t}(s_t), \mathbf{V}_2^{z_t}(s_t), \ldots, \mathbf{V}_g^{z_t}(s_t)\big) \geq \sup_{\pi^* \in \Pi_{\mathcal{M}}} \mathrm{W}_t\big(\mathbf{V}_1^{\pi^*}(s_t), \mathbf{V}_2^{\pi^*}(s_t), \ldots, \mathbf{V}_g^{\pi^*}(s_t)\big) - \varepsilon .$$

In other words, the key differences are that reward is now vector-valued, and optimal policies may now be stochastic and must now be welfare-optimal.

4 Algorithms for Fair Planning and Learning

We now present algorithms for fair planning and learning in our multi-beneficiary MDP setting. We first demonstrate how to plan in an MDP to maximize concave welfare objectives in section 4.1. We then introduce the *Equitable Explicit Explore Exploit* (E^4) adversarial fair MDP policy agent in section 4.2. Finally, we show that E^4 is a Kwik-Af learner in section 4.3.

4.1 On Welfare-Optimal Planning

For a given start-state distribution vector $\boldsymbol{p} \in \mathcal{P}(\mathcal{S})$, let $\boldsymbol{d}^\pi \in \mathbb{R}_{\geq 0}^{\mathcal{S} \times \mathcal{A}}$ be the geometrically-discounted state-action occupancy measure of the policy π, defined as

$$\boldsymbol{d}^\pi_{s,a} \doteq \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(s_t) \\ s_0 \sim \boldsymbol{p} \\ s_{t+1} \sim \mathbf{P}(s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}_s(s_t) \mathbb{1}_a(a_t) \right] = \pi(s,a) \left(\boldsymbol{p}_s + \gamma \sum_{\substack{s' \in \mathcal{S} \\ a' \in \mathcal{A}}} \mathbf{P}_s(s', a') \boldsymbol{d}^\pi_{s',a'} \right) . \tag{2}$$

Algorithm 1 Equitable Explicit Explore Exploit (E^4)

procedure AgentInit($\mathcal{S}, \mathcal{A}, \gamma, \mathrm{R}_{\max}, \varepsilon, \delta$)
  $T \leftarrow \max\left(1, \left\lceil \log_{\frac{1}{\gamma}}\left(\frac{6\lambda \mathrm{R}_{\max}}{\varepsilon(1-\gamma)}\right)\right\rceil\right)$; $t \leftarrow T$ ▷ Set escape time T and timer t
  $\alpha \leftarrow \frac{2\varepsilon(1-\gamma)^2}{3\lambda\left(2\sqrt{\mathrm{R}_{\max}}+\gamma\mathrm{R}_{\max}+6T(1-\gamma)\mathrm{R}_{\max}\right)}$ ▷ Set transition $\|\cdot\|_1$ error tolerance α
  $E \leftarrow 2\alpha T$ ▷ Set escape probability threshold E
  $\mathrm{m}_{\mathrm{knw}} \leftarrow \left\lceil \frac{1}{2\alpha^2} \ln\left(\frac{2|\mathcal{S}||\mathcal{A}|(2^{|\mathcal{S}|}-2+2g)}{\delta}\right)\right\rceil$ ▷ Compute sufficient per-state-action pair sample size
  $\forall s \in \mathcal{S}, a \in \mathcal{A} : m_{s,a} \leftarrow 0$ ▷ Initialize per-(s,a) visitation counters
  $\mathcal{S}_{\mathrm{unk}} \leftarrow \mathcal{S}$; $\mathcal{S}_{\mathrm{out}} \leftarrow \emptyset$; $\mathcal{S}_{\mathrm{inn}} \leftarrow \emptyset$ ▷ Initialize all states to unknown
  $\hat{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \hat{\mathbf{P}}, \hat{\mathbf{R}}, \gamma\rangle \leftarrow \langle \mathcal{S}, \mathcal{A}, (s,a) \mapsto \mathbb{1}_s, (s,a) \mapsto \mathbf{0}, \gamma\rangle$ ▷ Initialize empirical MDP to $\mathbf{0}$-reward recurrent
procedure AgentStep($s, \mathrm{W}(\cdot)$)
  case $s \in \mathcal{S}_{\mathrm{unk}}$ ▷ Successful escape attempt has reached $\mathcal{S}_{\mathrm{unk}}$
    $t \leftarrow T$ ▷ Stop escape timer
    return $a_{\mathrm{xpr}} \leftarrow \operatorname{argmin}_{a \in \mathcal{A}} m_{s,a}$ ▷ Select explore action a_{xpr} using balanced wandering
  case $t < T$ ▷ Ongoing attempt to escape to $\mathcal{S}_{\mathrm{unk}}$
    $t \leftarrow t+1$ ▷ Increment escape timer
    return $a_{\mathrm{xpr}} \leftarrow \pi_{\mathrm{esc}}(s,t)$ ▷ Explore action a_{xpr} from escape policy π_{esc}
  case $s \in \mathcal{S}_{\mathrm{inn}}$ ▷ Return exploit policy
    return $\pi_{\mathrm{xpt}} \leftarrow \operatorname{argmax}_{\pi \in \Pi_{\hat{\mathcal{M}}}} \mathrm{W}\left(\hat{\mathbf{V}}^{\pi}(s)\right)$ ▷ Exploit policy π_{xpt} computed from $\hat{\mathbf{V}}$ and $\hat{\mathcal{M}}$
  case $s \in \mathcal{S}_{\mathrm{out}}$ ▷ Begin escape attempt
    $t \leftarrow 0$ ▷ Start escape timer
    return $a_{\mathrm{xpr}} \leftarrow \pi_{\mathrm{esc}}(s,t)$
procedure AgentSarsUpdate($s, a, \boldsymbol{r}, s'$)
  if $s \in \mathcal{S}_{\mathrm{unk}}$ **then**
    $m_{s,a} \leftarrow m_{s,a} + 1$ ▷ Increment visitation count
    $\boldsymbol{X}^{\mathbf{P}}_{s,a,m_{s,a}} \leftarrow s'$; $\boldsymbol{X}^{\mathbf{R}}_{s,a,m_{s,a}} \leftarrow \boldsymbol{r}$ ▷ Append to experience buffers for transitions $\boldsymbol{X}^{\mathbf{P}}$ and rewards $\boldsymbol{X}^{\mathbf{R}}$
    if $\min_{a \in \mathcal{A}} m_{s,a} = \mathrm{m}_{\mathrm{knw}}$ **then** ▷ State s is now known
      $\forall a \in \mathcal{A}, s' \in \mathcal{S} : \ \hat{\mathbf{P}}_{s'}(s,a) \leftarrow \frac{1}{\mathrm{m}_{\mathrm{knw}}} \sum_{i=1}^{\mathrm{m}_{\mathrm{knw}}} \mathbb{1}_{s'}(\boldsymbol{X}^{\mathbf{P}}_{s,a,i})$ ▷ Empirical transition model $\hat{\mathbf{P}}$
      $\forall a : \hat{\mathbf{R}}(s,a) \leftarrow \frac{1}{\mathrm{m}_{\mathrm{knw}}} \sum_{i=1}^{\mathrm{m}_{\mathrm{knw}}} \boldsymbol{X}^{\mathbf{R}}_{s,a,i}$ ▷ Empirical reward function $\hat{\mathbf{R}}$
      $\pi_{\mathrm{esc}} \leftarrow \operatorname{argmax}_{\pi \in \Pi_T} \sum_{s \in \mathcal{S}} \mathbb{P}_{s_{t+1} \sim \hat{\mathbf{P}}(s_t, \pi(s_t,t))} \left(\bigvee_{i=0}^{T} s_i \in \mathcal{S}_{\mathrm{unk}} \,\middle|\, s_0 = s\right)$ ▷ T-step deterministic escape policy in $\hat{\mathbf{P}}$
      $\mathcal{S}_{\mathrm{unk}} \leftarrow \mathcal{S}_{\mathrm{unk}} \setminus \{s\}$ ▷ Remove s from the unknown set
      $\mathcal{S}_{\mathrm{out}} \leftarrow \left\{ s \in \left(\mathcal{S} \setminus \mathcal{S}_{\mathrm{unk}}\right) \,\middle|\, \mathbb{P}_{s_{t+1} \sim \hat{\mathbf{P}}(s_t, \pi_{\mathrm{esc}}(s_t,t))} \left(\bigvee_{i=0}^{T} s_i \in \mathcal{S}_{\mathrm{unk}} \,\middle|\, s_0 = s\right) \geq E \right\}$ ▷ Known states where T-step escape is E-likely
      $\mathcal{S}_{\mathrm{inn}} \leftarrow \mathcal{S} \setminus (\mathcal{S}_{\mathrm{unk}} \cup \mathcal{S}_{\mathrm{out}})$ ▷ Known states where T-step escape is not E-likely

Wang et al. (2008) show that the state-action occupancy measure gives rise to linear programs that efficiently plan (optimize value) in scalar-valued MDPs, and Zahavy et al. (2021) extend this idea to minimize arbitrary convex objectives of the state-action occupancy measure. We now apply this idea to address welfare-optimal planning in multi-beneficiary MDPs.

Proposition 4.1 (Welfare-Optimal Planning)**.** *Suppose start-state distribution $\boldsymbol{p}$ and λ-Lipschitz concave welfare function $\mathrm{W}(\cdot)$. Then the welfare-optimal policy $\pi^* = \operatorname{argmax}_{\pi \in \Pi_{\mathcal{M}}} \mathrm{W}\left(\mathbb{E}_{s \sim \boldsymbol{p}}\left[\mathbf{V}^{\pi}(s)\right]\right)$ from $\boldsymbol{p}$ can be identified by first solving for*

$$\boldsymbol{d}^* = \operatorname*{argmax}_{\boldsymbol{d} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}_{\geq 0}} \mathrm{W}\left(\sum_{s \in \mathcal{S}, a \in \mathcal{A}} \boldsymbol{d}_{s,a}\mathbf{R}_1(s,a), \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \boldsymbol{d}_{s,a}\mathbf{R}_2(s,a), \ldots, \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \boldsymbol{d}_{s,a}\mathbf{R}_g(s,a)\right) \tag{3}$$
$$\text{such that } \forall s \in \mathcal{S} : \sum_{a \in \mathcal{A}} \boldsymbol{d}_{s,a} = \boldsymbol{p}_s + \gamma \sum_{s' \in \mathcal{S}, a' \in \mathcal{A}} \mathbf{P}_s(s',a')\boldsymbol{d}_{s',a'} \ ,$$

and then setting $\pi^*(s,\cdot) \propto \boldsymbol{d}^*_{s,\cdot}$ for all $s \in \mathcal{S}$. Moreover, an ε-optimal policy can be computed using standard convex optimization methods in $\mathrm{Poly}\left(|\mathcal{S}|, |\mathcal{A}|, \frac{1}{1-\gamma}, \mathrm{R}_{\max}, g, \frac{1}{\varepsilon}, \lambda\right)$ *time.*

4.2 The E^4 Algorithm

We now describe E^4, for which we give pseudocode in algorithm 1. The key to E^4 is that the state space $\mathcal{S}$ is partitioned into three sets: The unknown set $\mathcal{S}_{\mathrm{unk}}$, the outer-known set $\mathcal{S}_{\mathrm{out}}$, and the

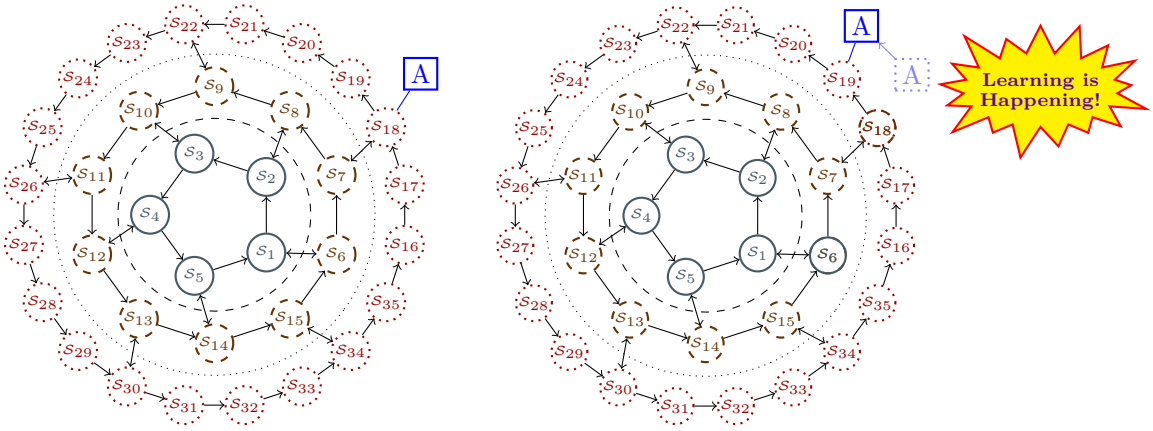


Figure 3: Depiction of an E^4 agent A learning, showing the inner-known $\mathcal{S}_{\text{inn}}$ (solid), outer-known $\mathcal{S}_{\text{out}}$ (dashed), and unknown $\mathcal{S}_{\text{unk}}$ (dotted) sets. All actions self-loop with probability $\approx \sqrt[T/2]{1-E}$, and arrows denote 1-step reachability via some action with probability $\approx 1 - \sqrt[T/2]{1-E}$. As the agent acts (explores) to reach s_{19} from s_{18}, s_{18} enters $\mathcal{S}_{\text{out}}$, which cascades to s_6 entering $\mathcal{S}_{\text{inn}}$.

inner-known set $\mathcal{S}_{\text{inn}}$. Initially, all states are unexplored in $\mathcal{S}_{\text{unk}}$ (line 6). After visiting a state $s \in \mathcal{S}_{\text{unk}}$ and taking all actions enough times (line 25) using balanced wandering (line 12), s becomes *known*, entering either $\mathcal{S}_{\text{inn}}$ or $\mathcal{S}_{\text{out}}$. We then construct an *empirical MDP* $\hat{\mathcal{M}}$ using the empirical transition frequencies and mean rewards from each known state and each action (lines 26 and 27), and self-loop probability 1 and reward $\mathbf{0}$ from unknown states (line 8). Then, for each known state s, if with nonnegligible probability (at least E) in $\hat{\mathcal{M}}$ it is possible to reach $\mathcal{S}_{\text{unk}}$ from s within T steps, we place s into $\mathcal{S}_{\text{out}}$ (line 30), otherwise we place s into $\mathcal{S}_{\text{inn}}$ (line 31). This process is graphically illustrated in figure 3. Note that E and T are set so as to ensure E^4 is KWIK-AF (lines 2 and 4).

As in the classic E^3 algorithm, within $\mathcal{S}_{\text{inn}}$, if all tail bounds hold, then the value functions of $\hat{\mathcal{M}}$ approximate the value functions of $\mathcal{M}$. Furthermore, under λ-Lipschitz continuity of welfare, optimizing welfare in $\hat{\mathcal{M}}$ ε-optimizes welfare in $\mathcal{M}$. Therefore, at each step, if the agent is in $\mathcal{S}_{\text{inn}}$, it outputs a ε-optimal policy (line 16). Otherwise, if the agent is in $\mathcal{S}_{\text{out}}$, it begins an *escape attempt* (line 18), which follows a T-step temporal policy (i.e., a policy $\pi \in \Pi_T$, where the action $\pi(s,t)$ depends on the current state and timestep) that maximizes the probability of reaching $\mathcal{S}_{\text{unk}}$ in $\hat{\mathcal{M}}$ (line 28). The escape attempt either proceeds for T steps (line 13), or until $\mathcal{S}_{\text{unk}}$ is reached (line 10). The main concrete difference between the E^3 and E^{4}algorithms is that E^4 has higher sample complexity, due both to vector-valued reward and to the nonlinearity of the welfare function. Furthermore, our analysis is more complex, as we show that E^4 KWIK-AF learns $\mathcal{M}$, which requires robustness against adversarial state selection even after an arbitrary number of exploitation steps, whereas the classical E^3 analysis only guarantees a single ε-optimal exploitation policy is output.

4.3 Theoretical Analysis

We are now ready to theoretically analyze E^4 in the KWIK-AF framework. We begin by defining (α, β)-uniform approximations of MDPs and computing the per-state sample complexity of attaining such approximations. All claims stated here are all proved in appendix A.

Definition 4.2 (Uniform Approximation MDPs)**.** *Let* $\text{TVD}(x, y)$ *denote the* total variation distance *between probability distributions* x *and* y*. An* (α, β)*-uniform approximation* $\mathcal{M}' = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}', \mathbf{R}', \gamma \rangle$ *of a vector-reward MDP* $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{R}, \gamma \rangle$ *is an MDP that, for all* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$*, satisfies*

1. $\text{TVD}\left(\mathbf{P}'(s,a), \mathbf{P}(s,a)\right) \leq \alpha$; *and* 2. $\|\mathbf{R}'(s,a) - \mathbf{R}(s,a)\|_\infty \leq \beta$.

Lemma 4.3 (Per-State Sample Complexity)**.** *Suppose MDP* $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{R}, \gamma \rangle$*, and let*

$$\mathrm{m}_{\text{knw}} \doteq \left\lceil \max\left(\frac{1}{2\alpha^2}, \frac{R_{\max}}{2\beta^2}\right) \ln\left(\frac{|\mathcal{S}||\mathcal{A}|(2^{|\mathcal{S}|} - 2 + 2g)}{\delta}\right) \right\rceil \quad . \tag{4}$$

If $\hat{\mathcal{M}}$ is estimated as the mean over $\mathrm{m}_{\mathrm{knw}}$ samples of the reward and destination state from each state-action pair, then with probability at least $1-\delta$, $\hat{\mathcal{M}}$ is an (α,β)-uniform approximation of $\mathcal{M}$.

When combined with Lipschitz continuity and a simulation lemma (Strehl et al., 2009), lemma 4.3 bounds the number of times we must take each action from each state before we can ε-δ maximize Lipschitz welfare functions. This gives us some sense of how E^4 operates (see line 5), although it says nothing of exploration and the learning process. The following result completes the story for the E^4 algorithm, analyzing its sample complexity in the KWIK-AF framework.

Theorem 4.4 (E^4 is a KWIK-AF Learner)**.** *Algorithm 1 is a KWIK-AF learner w.r.t. the class of all λ-$\|\cdot\|_\infty$ Lipschitz-continuous welfare functions, with sample complexity*

$$\begin{aligned}\mathrm{m}\left(|\mathcal{S}|,|\mathcal{A}|,\gamma,\mathrm{R}_{\max},g,\varepsilon,\delta\right) &\in \mathcal{O}\left(|\mathcal{S}|^2|\mathcal{A}|\left(\tfrac{\lambda \mathrm{R}_{\max}}{\varepsilon(1-\gamma)}\log\tfrac{1}{\gamma}\left(\tfrac{\lambda \mathrm{R}_{\max}}{\varepsilon(1-\gamma)}\right)\right)^3\log\tfrac{|\mathcal{S}||\mathcal{A}|g}{\delta}\right)\\ &\subseteq \mathrm{Poly}\left(|\mathcal{S}|,|\mathcal{A}|,\tfrac{1}{1-\gamma},\mathrm{R}_{\max},\log g,\tfrac{1}{\varepsilon},\log\tfrac{1}{\delta},\lambda\right)\ .\end{aligned}$$

Proof Sketch. Proof of this result is rather involved, and relies on several technical lemmata shown in appendix A. However, the crucial observation is that, subject to all tail bounds of lemma 4.3 holding over the course of algorithm 1 (which holds by union bound with probability at least $1-\frac{\delta}{2}$), then the E^4 agent is able to act in accordance with the duties of a KWIK-AF learner (definition 3.6).

Indeed, the *explore-exploit lemma* (A.4) shows that from anywhere in $\mathcal{S}_{\mathrm{inn}}$, *exploitation* is possible, and from within $\mathcal{S}_{\mathrm{out}}$, *exploration* is possible (i.e., escape attempts are worthwhile), which we use to satisfy both conditions of definition 3.6. In particular, from any inner-known state $s\in\mathcal{S}_{\mathrm{inn}}$, $\mathbf{V}(s)$ is approximately preserved between $\mathcal{M}$ and $\hat{\mathcal{M}}$, and welfare-optimal planning in $\hat{\mathcal{M}}$ is ε-optimal in $\mathcal{M}$. Finally, from any outer-known state $s\in\mathcal{S}_{\mathrm{out}}$, each escape attempt terminates within T steps, and reaches $\mathcal{S}_{\mathrm{unk}}$ with probability at least $\frac{E}{2}$, thus the expected number of exploration actions is bounded above by $\frac{2(T+1)}{E}|\mathcal{S}||\mathcal{A}|\,\mathrm{m}_{\mathrm{knw}}$, and standard binomial tail bounds yield the result. □

5 Conclusion

This work motivates and defines a formal model of welfare-centric fair reinforcement learning. We find that naïve approaches, like planning via policy iteration (example 2.1), and independent per-beneficiary exploration (example 2.2) do not yield fair RL agents. Defining fair RL and quantifying a learner's efficiency are challenging problems (section 3), as we must consider stochastic policies, and thus can not evaluate learners in terms of the *regret* or *mistakes* of *individual actions*. We thus define the *Adversarial Fair MDP Policy Agent* (definition 3.5) and the KWIK-AF learner (definition 3.6) to model fair RL and codify efficient learning in this domain. We then show (section 4) that it is possible to KWIK-AF learn the class of Lipschitz-continuous welfare functions in finite MDPs.

In practice, the decision to learn a policy *de novo* is quite radical, and many suboptimal actions will likely be taken while learning. This is a general issue for RL in sensitive settings: In medical contexts, Thomas et al. (2019) start from a *reference policy*, and seek to improve the policy while ensuring no reduction in performance. While reasonable in high-risk settings, when *fairness among groups* is a concern, it is inherently a conservative approach (as comparison to a reference policy centers the status quo), whereas starting *ex-nihilo* solely depends on the *structure of the MDP* and the *learning algorithm*, rather than existing societal biases, which may be encoded in the reference policy.

Finally, we note that suboptimal exploration actions could adversely affect some groups unfairly, and this should be monitored and controlled for. We note also that the number of suboptimal actions taken (as bounded by theorem 4.4) can be further reduced with more careful analysis; for instance the sample complexity of learning transition functions is much smaller when they are *sparse*, admit a *factoring*, or destination distributions are *far from uniform*, and the sample complexity of learning rewards may be much smaller when the variance of rewards is small. We are hopeful that future work will lead to KWIK-AF learners that explore more efficiently under various RL settings of interest.

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.

A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun. *Reinforcement Learning: Theory and Algorithms*. 2022.

K. Alikhademi, E. Drobina, D. Prioleau, B. Richardson, D. Purves, and J. E. Gilbert. A review of predictive policing from the perspective of fairness. *Artificial Intelligence and Law*, pages 1–17, 2021.

R. J. Arneson. Luck egalitarianism and prioritarianism. *Ethics*, 110(2):339–349, 2000.

P. Auer, T. Jaksch, and R. Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

J. Bentham. An introduction to the principles of morals and legislation. *University of London: the Athlone Press*, 1789.

J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

L. Buşoniu, R. Babuška, and B. De Schutter. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pages 183–221, 2010.

J. G. Cavazos, P. J. Phillips, C. D. Castillo, and A. J. O'Toole. Accuracy comparison across face recognition algorithms: Where are we on measuring race bias? *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2020.

Y. Chen, A. Cuellar, H. Luo, J. Modi, H. Nemlekar, and S. Nikolaidis. Fair contextual multi-armed bandits: Theory and experiments. In *Conference on Uncertainty in Artificial Intelligence*, pages 181–190. PMLR, 2020.

A. Chhabra, K. Masalkovaitė, and P. Mohapatra. An overview of fairness in clustering. *IEEE Access*, 9: 130698–130720, 2021.

F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. Fair clustering through fairlets. *Advances in neural information processing systems*, 30, 2017.

C. M. Cook, J. J. Howard, Y. B. Sirotin, J. L. Tipton, and A. R. Vemury. Demographic effects in facial recognition and their dependence on image acquisition: An evaluation of eleven commercial systems. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(1):32–41, 2019.

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.

C. Cousins. Algorithms and analysis for optimizing robust objectives in fair machine learning. In *Columbia Workshop on Fairness in Operations and AI*. Columbia University, 2023a.

C. Cousins. Revisiting fair-PAC learning and the axioms of cardinal welfare. In *Artificial Intelligence and Statistics (AISTATS)*, 2023b.

C. Cousins, K. Asadi, and M. L. Littman. Fair E^3: Efficient welfare-centric fair reinforcement learning. In *5th Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, 2022.

C. Cousins, I. E. Kumar, and S. Venkatasubramanian. To pool or not to pool: Analyzing the regularizing effects of group-fair training on shared models. In *Artificial Intelligence and Statistics (AISTATS)*, 2024.

H. Dalton. The measurement of the inequality of incomes. *The Economic Journal*, 30(119):348–361, 1920.

G. Debreu. Topological methods in cardinal utility theory. *Cowles Foundation Discussion Papers*, 76, 1959.

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.

Z. Fan, N. Peng, M. Tian, and B. Fain. Welfare and fairness in multi-objective reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1991–1999, 2023.

T. Gajdos and J. A. Weymark. Multidimensional generalized Gini indices. *Economic Theory*, 26(3):471–496, 2005.

W. M. Gorman. The structure of utility functions. *The Review of Economic Studies*, 35(4):367–390, 1968.

S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. Fairness in reinforcement learning. In *International conference on machine learning*, pages 1617–1626. PMLR, 2017.

S. M. Kakade. *On the sample complexity of reinforcement learning.* University of London, University College London (United Kingdom), 2003.

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.

L. Li, M. L. Littman, T. J. Walsh, and A. L. Strehl. Knows what it knows: A framework for self-aware learning. *Machine learning*, 82(3):399–443, 2011.

D. J. Lizotte, M. Bowling, and S. A. Murphy. Linear fitted-q iteration with multiple reward functions. *The Journal of Machine Learning Research*, 13(1):3253–3295, 2012.

B. Metevier, S. Giguere, S. Brockman, A. Kobren, Y. Brun, E. Brunskill, and P. S. Thomas. Offline contextual bandits with high probability fairness guarantees. *Advances in neural information processing systems*, 32, 2019.

J. S. Mill. *Utilitarianism*. Parker, Son, and Bourn, London, 1863.

D. Parfit. Equality and priority. *Ratio (Oxford)*, 10(3):202–221, 1997.

A. C. Pigou. *Wealth and welfare*. Macmillan and Company, limited, 1912.

L. Prashanth and M. Ghavamzadeh. Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning*, 105:367–417, 2016.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. USA, 1st edition, 1994. ISBN 0471619779.

M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481, 2020.

J. Rawls. *A theory of justice*. Harvard University Press, 1971.

J. Rawls. *Justice as fairness: A restatement*. Harvard University Press, 2001.

D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

H. Satija, P. S. Thomas, J. Pineau, and R. Laroche. Multi-objective SPIBB: Seldonian offline policy improvement with safety constraints in finite MDPs. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

H. Satija, A. Lazaric, M. Pirotta, and J. Pineau. Group fairness in reinforcement learning. *Transactions on Machine Learning Research*, 2022.

U. Siddique, P. Weng, and M. Zimmer. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *International Conference on Machine Learning*, pages 8905–8915. PMLR, 2020.

A. L. Strehl, L. Li, and M. L. Littman. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10(11), 2009.

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.

K. Van Moffaert and A. Nowé. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

T. Wang, D. Lizotte, M. Bowling, and D. Schuurmans. Dual representations for dynamic programming. *Journal of Machine Learning Research*, 1:1–29, 01 2008.

Y. Wang and M. P. Chapman. Risk-averse autonomous systems: A brief history and recent developments from the perspective of optimal control. *Artificial Intelligence*, 311:103743, 2022.

M. Wen, O. Bastani, and U. Topcu. Algorithms for fairness in sequential decision making. In *International Conference on Artificial Intelligence and Statistics*, pages 1144–1152. PMLR, 2021.

J. A. Weymark. Generalized Gini inequality indices. *Mathematical Social Sciences*, 1(4):409–430, 1981.

G. Yu, U. Siddique, and P. Weng. Fair deep reinforcement learning with generalized Gini welfare functions. 2023.

T. Zahavy, B. O'Donoghue, G. Desjardins, and S. Singh. Reward is enough for convex MDPs. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.