Fairness in Reinforcement Learning [*]

Shahin Jabbari   Matthew Joseph   Michael Kearns   Jamie Morgenstern   Aaron Roth [1]

Abstract

We initiate the study of *fairness* in reinforcement learning, where the actions of a learning algorithm may affect its environment and future rewards. Our fairness constraint requires that an algorithm never prefers one action over another if the long-term (discounted) reward of choosing the latter action is higher. Our first result is negative: despite the fact that fairness is consistent with the optimal policy, any learning algorithm satisfying fairness must take time exponential in the number of states to achieve non-trivial approximation to the optimal policy. We then provide a provably fair polynomial time algorithm under an approximate notion of fairness, thus establishing an exponential gap between exact and approximate fairness.

1. Introduction

The growing use of machine learning for automated decision-making has raised concerns about the potential for unfairness in learning algorithms and models. In settings as diverse as policing [22], hiring [19], lending [4], and criminal sentencing [2], mounting empirical evidence suggests these concerns are not merely hypothetical [1; 25].

We initiate the study of fairness in reinforcement learning, where an algorithm's choices may influence the state of the world and future rewards. In contrast, previous work on fair machine learning has focused on myopic settings where such influence is absent, e.g. in i.i.d. or no-regret models [5; 6; 9; 10]. The resulting fairness definitions therefore do not generalize well to a reinforcement learning setting, as they do not reason about the effects of short-term actions on long-term rewards. This is relevant for the settings where historical context can have a distinct influence on the future. For concreteness, we consider the specific example of hiring (though other settings such as college admission or lending decisions can be embedded into this framework). Consider a firm aiming to hire employees for a number of positions. The firm might consider a variety of hiring practices, ranging from targeting and hiring applicants from well-understood parts of the applicant pool (which might be a reasonable policy for short-term productivity of its workforce), to exploring a broader class of applicants whose backgrounds might differ from the current set of employees at the company (which might incur short-term productivity and learning costs but eventually lead to a richer and stronger overall applicant pool).

We focus on the standard model of reinforcement learning, in which an algorithm seeks to maximize its discounted sum of rewards in a Markovian decision process (MDP). Throughout, the reader should interpret the *actions* available to a learning algorithm as corresponding to choices or policies affecting individuals (e.g. which applicants to target and hire). The *reward* for each action should be viewed as the short-term payoff of making the corresponding decision (e.g. the short-term influence on the firm's productivity after hiring any particular candidate). The actions taken by the algorithm affect the underlying state of the system (e.g. the company's demographics as well as the available applicant pool) and therefore in turn will affect the actions and rewards available to the algorithm in the future.

Informally, our definition of fairness requires that (with high probability) in state s, an algorithm never chooses an available action a with probability higher than another action a' unless $Q^*(s, a) > Q^*(s, a')$, i.e. the long-term reward of a is greater than that of a'. This definition, adapted from Joseph et al. (2016), is *weakly meritocratic*: facing some set of actions, an algorithm must pick a distribution over actions with (weakly) heavier weight on the better actions (in terms of their discounted long-term reward). Correspondingly, a hiring process satisfying our fairness definition cannot probabilistically target one population over another if hiring from either population will have similar long-term benefit to the firm's productivity.

Unfortunately, our first result shows an exponential separation in expected performance between the best unfair algo-

[1] University of Pennsylvania, Philadelphia, PA, USA. Correspondence to: Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Aaron Roth <jabbari, majos, mkearns, jamiemor, aaroth@cis.upenn.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[*] The full technical version of this paper is available at https://arxiv.org/pdf/1611.03071.pdf.

rithm and any algorithm satisfying fairness. This motivates our study of a natural relaxation of (exact) fairness, for which we provide a polynomial time learning algorithm, thus establishing an exponential separation between exact and approximately fair learning in MDPs.

Our Results Throughout, we use *(exact)* fairness to refer to the adaptation of Joseph et al. (2016)'s definition defining an action's quality as its potential long-term discounted reward. We also consider two natural relaxations. The first, *approximate-choice fairness*, requires that an algorithm never chooses a worse action with *probability* substantially higher than better actions. The second, *approximate-action fairness*, requires that an algorithm never favors an action of substantially lower *quality* than that of a better action.

The contributions of this paper can be divided into two parts. First, in Section 3, we give a lower bound on the time required for a learning algorithm to achieve near-optimality subject to (exact) fairness or approximate-choice fairness.

Theorem (Informal statement of Theorems 3, 4, and 5)**.** *For constant ϵ, to achieve ϵ-optimality,* (i) *any fair or approximate-choice fair algorithm takes a number of rounds exponential in the number of MDP states and* (ii) *any approximate-action fair algorithm takes a number of rounds exponential in $1/(1-\gamma)$, for discount factor γ.*

Second, we present an approximate-action fair algorithm (**Fair-E**^{3}) in Section 4 and prove a polynomial upper bound on the time it requires to achieve near-optimality.

Theorem (Informal statement of Theorem 6)**.** *For constant ϵ and any MDP satisfying standard assumptions, **Fair-E**^{3} is an approximate-action fair algorithm achieving ϵ-optimality in a number of rounds that is (necessarily) exponential in $1/(1-\gamma)$ and polynomial in other parameters.*

The exponential dependence of **Fair-E**^{3} on $1/(1-\gamma)$ is tight: it matches our lower bound on the time complexity of any approximate-action fair algorithm. Furthermore, our results establish rigorous trade-offs between fairness and performance facing reinforcement learning algorithms.

1.1. Related Work

The most relevant parts of the large body of literature on reinforcement learning focus on constructing learning algorithms with provable performance guarantees. **E**^{3} [13] was the first learning algorithm with a polynomial learning rate, and subsequent work improved this rate (see Szita and Szepesvári (2010) and references within). The study of *robust* MDPs [16; 18; 20] examines MDPs with high parameter uncertainty but generally uses "optimistic" learning strategies that ignore (and often conflict with) fairness and so do not directly apply to this work.

Our work also belongs to a growing literature studying the problem of fairness in machine learning. Early work in data mining [8; 11; 12; 17; 21; 29] considered the question from a primarily empirical standpoint, often using *statistical parity* as a fairness goal. Dwork et al. (2012) explicated several drawbacks of statistical parity and instead proposed one of the first broad definitions of algorithmic fairness, formalizing the idea that "similar individuals should be treated similarly". Recent papers have proven several impossibility results for satisfying different fairness requirements simultaneously [7; 15]. More recently, Hardt et al. (2016) proposed new notions of fairness and showed how to achieve these notions via post-processing of a black-box classifier. Woodworth et al. (2017) and Zafar et al. (2017) further studied these notion theoretically and empirically.

1.2. Strengths and Limitations of Our Models

In recognition of the duration and consequence of choices made by a learning algorithm during its learning process – e.g. job applicants not hired – our work departs from previous work and aims to guarantee the fairness of *the learning process itself*. To this end, we adapt the fairness definition of Joseph et al. (2016), who studied fairness in the bandit framework and defined fairness with respect to one-step rewards. To capture the desired interaction and evolution of the reinforcement learning setting, we modify this myopic definition and define fairness with respect to long-term rewards: a fair learning algorithm may only choose action a over action a' if a has true long-term reward at least as high as a'. Our contributions thus depart from previous work in reinforcement learning by incorporating a fairness requirement (ruling out existing algorithms which commonly make heavy use of "optimistic" strategies that violates fairness) and depart from previous work in fair learning by requiring "online" fairness in a previously unconsidered reinforcement learning context.

First note that our definition is *weakly meritocratic*: an algorithm satisfying our fairness definition can *never* probabilistically favor a worse option but is not *required* to favor a better option. This confers both strengths and limitations. Our fairness notion still permits a type of "conditional discrimination" in which a fair algorithm favors group A over group B by selecting choices from A when they are superior and randomizing between A and B when choices from B are superior. In this sense, our fairness requirement is relatively minimal, encoding a necessary variant of fairness rather than a sufficient one. This makes our lower bounds and impossibility results (Section 3) relatively stronger and upper bounds (Section 4) relatively weaker.

Next, our fairness requirement holds (with high probability) across *all* decisions that a fair algorithm makes. We view this strong constraint as worthy of serious consideration, since "forgiving" unfairness during the learning may badly mistreat the training population, especially if the learning process is lengthy or even continual. Additionally, it is unclear how to relax this requirement, even for a small fraction of the algorithm's decisions, without enabling discrimination against a correspondingly small population.

Instead, aiming to preserve the "minimal" spirit of our definition, we consider a relaxation that only prevents an algorithm from favoring a *significantly* worse option over a better option (Section 2.1). Hence, approximate-action fairness should be viewed as a weaker constraint: rather than safeguarding against every violation of "fairness", it instead restricts how egregious these violations can be. We discuss further relaxations of our definition in Section 5.

2. Preliminaries

In this paper we study reinforcement learning in Markov Decision Processes (MDPs). An MDP is a tuple $M = (\mathcal{S}_M, \mathcal{A}_M, P_M, R_M, T, \gamma)$ where $\mathcal{S}_M$ is a set of n *states*, $\mathcal{A}_M$ is a set of k *actions*, T is a *horizon* of a (possibly infinite) number of rounds of activity in M, and γ is a *discount factor*. $P_M : \mathcal{S}_M \times \mathcal{A}_M \to \mathcal{S}_M$ and $R_M : \mathcal{S}_M \to [0,1]$ denote the *transition probability distribution* and *reward distribution*, respectively. We use $\bar{R}_M$ to denote the mean of R_M.[2] A policy π is a mapping from a history h (the sequence of triples (state, action, reward) observed so far) to a distribution over actions. The discounted state and state-action value functions are denoted by V^π and Q^π, and $V^\pi(s, T)$ represents expected discounted reward of following π from s for T steps. The highest values functions are achieved by the *optimal* policy π^* and are denoted by V^* and Q^* [24]. We use μ^π to denote the stationary distribution of π. Throughout we make the following assumption.

Assumption 1 (Unichain Assumption)**.** *The stationary distribution of any policy in M is independent of its start state.*

We denote the ϵ*-mixing time* of π by T^π_ϵ. Lemma 1 relates the ϵ-mixing time of any policy π to the number of rounds until the V^π_M values of the visited states by π are close to the expected V^π_M values (under the stationary distribution μ^π). We defer all the omitted proofs to the Appendix.

Lemma 1. *Fix $\epsilon > 0$. For any state s, following π for $T \geq T^\pi_\epsilon$ steps from s satisfies*

$$\mathbb{E}_{s\sim\mu^\pi}\left[V^\pi_M(s)\right] - \mathbb{E}\left[\tfrac{1}{T}\sum_{t=1}^{T} V^\pi_M(s_t)\right] \leq \tfrac{\epsilon}{1-\gamma},$$

where s_t is the state visited at time t when following π from s and the expectation in the second term is over the transition function and the randomization of π.[3]

The *horizon time* $H^\gamma_\epsilon := \log\left(\epsilon(1-\gamma)\right)/\log(\gamma)$ of an MDP captures the number of steps an approximately optimal policy must optimize over. The expected discounted reward of any policy after H^γ_ϵ steps approaches the expected asymptotic discounted reward (Kearns and Singh (2002), Lemma 2). A learning algorithm $\mathcal{L}$ is a non-stationary policy that at each round takes the entire history and outputs a distribution over actions. We now define a performance measure for learning algorithms.

Definition 1 (ϵ-optimality)**.** *Let $\epsilon > 0$ and $\delta \in (0, 1/2)$. $\mathcal{L}$ achieves* ϵ-optimality *in $\mathcal{T}$ steps if for any $T \geq \mathcal{T}$*

$$\mathbb{E}_{s\sim\mu^*}\left[V^*_M(s)\right] - \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} V^*_M(s_t)\right] \leq \frac{2\epsilon}{1-\gamma}, \qquad (1)$$

with probability at least $1-\delta$, for s_t the state $\mathcal{L}$ reaches at time t, where the expectation is taken over the transitions and the randomization of $\mathcal{L}$, for any MDP M.

We thus ask that a learning algorithm, after sufficiently many steps, visits states whose values are arbitrarily close to the values of the states visited by the optimal policy. Note that this is stronger than the "hand-raising" notion in Kearns and Singh (2002),[4] which only asked that the learning algorithm stop in a state from which discounted return is near-optimal, permitting termination in a state from which the optimal discounted return is poor. In Definition 1, if there are states with poor optimal discounted reward that the optimal policy eventually leaves for better states, so must our algorithms. We also note the following connection between the average V^π_M values of states visited under the stationary distribution of π (and in particular an optimal policy) and the average undiscounted rewards achieved under the stationary distribution of that policy.

Lemma 2 (Singh (2016))**.** *Let $\bar{\boldsymbol{R}}_M$ be the vector of mean rewards in states of M and $\boldsymbol{V}^\pi_M$ the vector of discounted rewards in states under π. Then $\mu^\pi \cdot \bar{\boldsymbol{R}}_M = (1-\gamma)\mu^\pi \cdot \boldsymbol{V}^\pi_M$.*

We design an algorithm which quickly achieves ϵ-optimality and we bound the number of steps $\mathcal{T}$ before this happens by a polynomial in the parameters of M.

2.1. Notions of Fairness

We now turn to formal notions of fairness. Translated to our setting, Joseph et al. (2016) define action a's quality as the expected immediate reward for choosing a from state s and then require that an algorithm not probabilistically favor a over a' if a has lower expected immediate reward.

However, this naive translation does not adequately capture the structural differences between bandit and MDP set-

[2] Note that $\bar{R}_M \leq 1$ and $\mathrm{Var}(R_M) \leq 1$ for all states. The bounded reward assumption can be relaxed (see e.g. [13]). Also assuming rewards in $[0,1]$ can be made w.l.o.g. up to scaling.

[3] Lemma 1 can be stated for a weaker notion of mixing time called the ϵ*-reward mixing time* which is always linearly bounded by the ϵ-mixing time but can be much smaller in certain cases (see Kearns and Singh (2002) for a discussion).

[4] We suspect unfair $\mathbf{E}^3$ also satisfies this stronger notion.

tings since present rewards may depend on past choices in MDPs. In particular, defining fairness in terms of immediate rewards would prohibit any policy sacrificing short-term rewards in favor of long-term rewards. This is undesirable, since it is the long-term rewards that matter in reinforcement learning, and optimizing for long-term rewards often necessitates short-term sacrifices. Moreover, the long-term impact of a decision should be considered when arguing about its relative fairness. We will therefore define fairness using the state-action value function Q^*_M.

Definition 2 (Fairness)**.** *$\mathcal{L}$ is* fair *if for all input $\delta > 0$, all M, all rounds t, all states s and all actions a, a'*

$$Q^*_M(s,a) \geq Q^*_M(s,a') \Rightarrow \mathcal{L}(s,a,h_{t-1}) \geq \mathcal{L}(s,a',h_{t-1})$$

with probability at least $1-\delta$ over histories h_{t-1}. [5]

Fairness requires that an algorithm *never* probabilistically favors an action with lower long-term reward over an action with higher long-term reward. In hiring, this means that an algorithm cannot target one applicant population over another unless the targeted population has a higher quality.

In Section 3, we show that fairness can be extremely restrictive. Intuitively, $\mathcal{L}$ must play uniformly at random until it has high confidence about the Q^*_M values, in some cases taking exponential time to achieve near-optimality. This motivates relaxing Definition 2. We first relax the *probabilistic* requirement and require only that an algorithm not *substantially* favor a worse action over a better one.

Definition 3 (Approximate-choice Fairness)**.** *$\mathcal{L}$ is α*-choice fair *if for all inputs $\delta > 0$ and $\alpha > 0$: for all M, all rounds t, all states s and actions a, a':*

$$Q^*_M(s,a) \geq Q^*_M(s,a') \Rightarrow \mathcal{L}(s,a,h_{t-1}) \geq \mathcal{L}(s,a',h_{t-1}) - \alpha,$$

with probability of at least $1-\delta$ over histories h_{t-1}. If $\mathcal{L}$ is α-choice fair for any input $\alpha > 0$, we call $\mathcal{L}$ approximate-choice fair.

A slight modification of the lower bound for (exact) fairness shows that algorithms satisfying approximate-choice fairness can also require exponential time to achieve near-optimality. We therefore propose an alternative relaxation, where we relax the *quality* requirement. As described in Section 1.1, the resulting notion of approximate-action fairness is in some sense the most fitting relaxation of fairness, and is a particularly attractive one because it allows us to give algorithms circumventing the exponential hardness proved for fairness and approximate-choice fairness.

Definition 4 (Approximate-action Fairness)**.** *$\mathcal{L}$ is α*-action fair *if for all inputs $\delta > 0$ and $\alpha > 0$, for all M, all rounds t, all states s and actions a, a':*

$$Q^*_M(s,a) > Q^*_M(s,a') + \alpha \Rightarrow \mathcal{L}(s,a,h_{t-1}) \geq \mathcal{L}(s,a'h_{t-1})$$

with probability of at least $1-\delta$ over histories h_{t-1}. If $\mathcal{L}$ is α-action fair for any input $\alpha > 0$, we call $\mathcal{L}$ approximate-action fair.

Approximate-choice fairness prevents equally good actions from being chosen at very different rates, while approximate-action fairness prevents substantially worse actions from being chosen over better ones. In hiring, an approximately-action fair firm can only (probabilistically) target one population over another if the targeted population is not substantially worse. While this is a weaker guarantee, it at least forces an approximately-action fair algorithm to learn different populations to statistical confidence. This is a step forward from current practices, in which companies have much higher degrees of uncertainty about the quality (and impact) of hiring individuals from under-represented populations. For this reason and the computational benefits mentioned above, our upper bounds will primarily focus on approximate-action fairness.

We now state several useful observations regarding fairness. We defer all the formal statements and their proofs to the Appendix. We note that there always exists a (possibly randomized) optimal policy which is fair (Observation 1); moreover, *any* optimal policy (deterministic or randomized) is approximate-action fair (Observation 2), as is the uniformly random policy (Observation 3).

Finally, we consider a restriction of the actions in an MDP M to nearly-optimal actions (as measured by Q^*_M values).

Definition 5 (Restricted MDP)**.** *The α*-restricted *MDP of M, denoted by M^α, is identical to M except that in each state s, the set of available actions are restricted to $\{a : Q^*_M(s,a) \geq \max_{a' \in \mathcal{A}_M} Q^*_M(s,a') - \alpha \mid a \in \mathcal{A}_M\}$.*

M^α has the following two properties: (i) any policy in M^α is α-action fair in M (Observation 4) and (ii) the optimal policy in M^α is also optimal in M (Observation 5). Observations 4 and 5 aid our design of an approximate-action fair algorithm: we construct M^α from estimates of the Q^*_M values (see Section 4.3 for more details).

3. Lower Bounds

We now demonstrate a stark separation between the performance of learning algorithms with and without fairness. First, we show that neither fair nor approximate-choice fair algorithms achieve near-optimality unless the number of time steps $\mathcal{T}$ is at least $\Omega(k^n)$, exponential in the size of the state space. We then show that any approximate-action fair algorithm requires a number of time steps $\mathcal{T}$ that is at least $\Omega(k^{\frac{1}{1-\gamma}})$ to achieve near-optimality. We start by proving a lower bound for fair algorithms.

Theorem 3. *If $\delta < \frac{1}{4}$, $\gamma > \frac{1}{2}$ and $\epsilon < \frac{1}{8}$, no fair algorithm*

[5] $\mathcal{L}(s,a,h)$ denotes the probability $\mathcal{L}$ chooses a from s given history h.

can be ϵ-optimal in $\mathcal{T} = O(k^n)$ steps.[6]

Standard reinforcement learning algorithms (absent a fairness constraint) learn an ϵ-optimal policy in a number of steps polynomial in n and $\frac{1}{\epsilon}$; Theorem 3 therefore shows a steep cost of imposing fairness. We outline the idea for proof of Theorem 3. For intuition, first consider the special case when the number of actions $k = 2$. We introduce the MDPs witnessing the claim in Theorem 3 for this case.

Definition 6 (Lower Bound Example)**.** *For $\mathcal{A}_M = \{L, R\}$, let $M(x) = (\mathcal{S}_M, \mathcal{A}_M, \mathcal{P}_M, \mathcal{R}_M, T, \gamma, x)$ be an MDP with*

- *for all $i \in [n]$, $P_M(s_i, L, s_1) = P_M(s_i, R, s_j) = 1$ where $j = \min\{i+1, n\}$ and is 0 otherwise.*
- *for $i \in [n-1]$, $R_M(s_i) = 0.5$, and $R_M(s_n) = x$.*

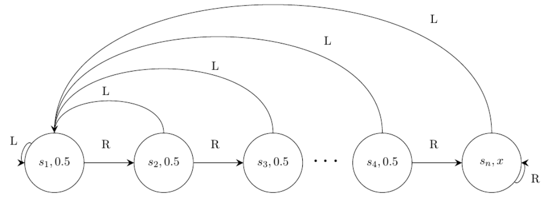

Figure 1. MDP(x): Circles represent states (labels denote the state name and deterministic reward). Arrows represent actions.

Figure 1 illustrates the MDP from Definition 6. All the transitions and rewards in M are deterministic, but the reward at state s_n can be either 1 or $\frac{1}{2}$, and so no algorithm (fair or otherwise) can determine whether the Q^*_M values of all the states are the same or not until it reaches s_n and observes its reward. Until then, fairness requires that the algorithm play all the actions uniformly at random (if the reward at s_n is $\frac{1}{2}$, any fair algorithm must play uniformly at random forever). Thus, *any* fair algorithm will take exponential time in the number of states to reach s_n. This can be easily modified for $k > 2$: from each state s_i, $k-1$ of the actions from state s_i (deterministically) return to state s_1 and only one action (deterministically) reaches any other state $s_{\min\{i+1,n\}}$. It will take k^n steps before any fair algorithm reaches s_n and can stop playing uniformly at random (which is necessary for near-optimality). The same example, with a slightly modified analysis, also provides a lower bound of $\Omega((k/(1+k\alpha))^n)$ time steps for approximate-choice fair algorithms as stated in Theorem 4.

Theorem 4. *If $\delta < \frac{1}{4}, \alpha < \frac{1}{4}, \gamma > \frac{1}{2}$ and $\epsilon < \frac{1}{8}$, no α-choice fair algorithm is ϵ-optimal for $\mathcal{T} = O((\frac{k}{1+k\alpha})^n)$ steps.*

Fairness and approximate-choice fairness are both extremely costly, ruling out polynomial time learning rates. Hence, we focus on approximate-action fairness. Before moving to positive results, we mention that the time complexity of approximate-action fair algorithms will still suffer from an exponential dependence on $\frac{1}{1-\gamma}$.

Theorem 5. *For $\delta < \frac{1}{4}$, $\alpha < \frac{1}{8}$, $\gamma > \max(0.9, c)$, $c \in (\frac{1}{2}, 1)$ and $\epsilon < \frac{1-e^{c-1}}{16}$, no α-action fair algorithm is ϵ-optimal for $\mathcal{T} = O((k^{\frac{1}{1-\gamma}})^c)$ steps.*

The MDP in Figure 1 also witnesses the claim of Theorem 5 when $n = \lceil \frac{\log(1/(2\alpha))}{1-\gamma} \rceil$. The discount factor γ is generally taken as a constant, so in most interesting cases $\frac{1}{1-\gamma} \ll n$: this lower bound is substantially less stringent than the lower bounds proven for fairness and approximate-choice fairness. Hence, from now on, we focus on designing algorithms satisfying approximate-action fairness with learning rates polynomial in every parameter but $\frac{1}{1-\gamma}$, and with tight dependence on $\frac{1}{1-\gamma}$.

4. A Fair and Efficient Learning Algorithm

We now present an approximate-action fair algorithm, **Fair-E**^{3} with the performance guarantees stated below.

Theorem 6. *Given $\epsilon > 0$, $\alpha > 0$, $\delta \in \left(0, \frac{1}{2}\right)$ and $\gamma \in [0, 1)$ as inputs, **Fair-E**^{3} is an α-action fair algorithm which achieves ϵ-optimality after*

$$\mathcal{T} = \tilde{O}\left(\frac{n^5 T^*_\epsilon k^{\frac{1}{1-\gamma}+5}}{\min\{\alpha^4, \epsilon^4\}\epsilon^2 (1-\gamma)^{12}}\right) \tag{2}$$

steps where $\tilde{O}$ hides poly-logarithmic terms.

The running time of **Fair-E**^{3} (which we have not attempted to optimize) is polynomial in all the parameters of the MDP except $\frac{1}{1-\gamma}$; Theorem 5 implies that this exponential dependence on $\frac{1}{1-\gamma}$ is necessary.

Several more recent algorithms (e.g. R-MAX [3]) have improved upon the performance of **E**^{3}. We adapted **E**^{3} primarily for its simplicity. While the machinery required to properly balance fairness and performance is somewhat involved, the basic ideas of our adaptation are intuitive. We further note that subsequent algorithms improving on **E**^{3} tend to heavily leverage the principle of "optimism in face of uncertainty": such behavior often violates fairness, which generally requires *uniformity* in the face of uncertainty. Thus, adapting these algorithms to satisfy fairness is more difficult. This in particular suggests **E**^{3} as an apt starting point for designing a fair planning algorithm.

The remainder of this section will explain **Fair-E**^{3}, beginning with a high-level description in Section 4.1. We then define the "known" states **Fair-E**^{3} uses to plan in Section 4.2, explain this planning process in Section 4.3, and

[6] We have not optimized the constants upper-bounding parameters in the statement of Theorems 3, 4 and 5. The values presented here are only chosen for convenience.

bring this all together to prove **Fair-E**^{3}'s fairness and performance guarantees in Section 4.4.

4.1. Informal Description of Fair-E^3

Fair-E^3 relies on the notion of "known" states. A state s is defined to be *known* after all actions have been chosen from s enough times to confidently estimate relevant reward distributions, transition probabilities, and Q^{π}_{M} values for each action. At each time t, **Fair-E**^{3} then uses known states to reason about the MDP as follows:

- If in an unknown state, take a uniformly random trajectory of length H^{γ}_{ϵ}.
- If in a known state, compute *(i)* an exploration policy which escapes to an unknown state quickly and p, the probability that this policy reaches an unknown state within $2T^{*}_{\epsilon}$ steps, and *(ii)* an exploitation policy which is near-optimal in the known states of M.
  - If p is large enough, follow the exploration policy; otherwise, follow the exploitation policy.

Fair-E^3 thus relies on known states to balance exploration and exploitation in a reliable way. While **Fair-E**^{3} and **E**^{3} share this general idea, fairness forces **Fair-E**^{3} to more delicately balance exploration and exploitation. For example, while both algorithms explore until states become "known", the definition of a known state must be much stronger in **Fair-E**^{3} than in **E**^{3} because **Fair-E**^{3} additionally requires accurate estimates of actions' Q^{π}_{M} values in order to make decisions without violating fairness. For this reason, **Fair-E**^{3} replaces the deterministic exploratory actions of **E**^{3} with random trajectories of actions from unknown states. These random trajectories are then used to estimate the necessary Q^{π}_{M} values.

In a similar vein, **Fair-E**^{3} requires particular care in computing exploration and exploitation policies, and must restrict the set of such policies to fair exploration and fair exploitation policies. Correctly formulating this restriction process to balance fairness and performance relies heavily on the observations about the relationship between fairness and performance provided in Section 2.1.

4.2. Known States in Fair-E^3

We now formally define the notion of known states for **Fair-E**^{3}. We say a state s becomes known when one can compute good estimates of *(i)* $R_M(s)$ and $P_M(s,a)$ for all a, and *(ii)* $Q^{*}_{M}(s,a)$ for all a.

Definition 7 (Known State)**.** *Let*

$$m_1 = O\left(k^{H^{\gamma}_{\epsilon}+3} n \left(\frac{1}{(1-\gamma)\,\alpha}\right)^2 \log\left(\frac{k}{\delta}\right)\right) \text{ and}$$

$$m_2 = O\left(\left(\frac{n}{\min\{\epsilon,\alpha\}}\right)^4 {H^{\gamma}_{\epsilon}}^8 \log\left(\frac{1}{\delta}\right)\right).$$

A state s becomes known *after taking*

$$m_Q := k \cdot \max\{m_1, m_2\} \tag{3}$$

length-H^{γ}_{ϵ} random trajectories from s.

It remains to show that motivating conditions *(i)* and *(ii)* indeed hold for our formal definition of a known state. Informally, m_1 random trajectories suffice to ensure that we have accurate estimates of all $Q^{*}_{M}(s,a)$ values, and m_2 random trajectories suffice to ensure accurate estimates of the transition probabilities and rewards.

To formalize condition *(i)*, we rely on Theorem 7, connecting the number of random trajectories taken from s to the accuracy of the empirical V^{π}_{M} estimates.

Theorem 7 (Theorem 5.5, Kearns et al. (2000))**.** *For any state s and $\alpha > 0$, after*

$$m = O\left(k^{H^{\gamma}_{\epsilon}+3} \left(\frac{1}{(1-\gamma)\,\alpha}\right)^2 \log\left(\frac{|\Pi|}{\delta}\right)\right)$$

random trajectories of length H^{γ}_{ϵ} from s, with probability of at least $1-\delta$, we can compute estimates $\hat{V}^{\pi}_{M}$ such that $|V^{\pi}_{M}(s) - \hat{V}^{\pi}_{M}(s)| \leq \alpha$, simultaneously for all $\pi \in \Pi$.

Theorem 7 enables us to translate between the number of trajectories taken from a state and the uncertainty about its V^{π}_{M} values for all policies (including π^* and hence V^{*}_{M}). Since $|\Pi| = k^n$, we substitute $\log(|\Pi|) = n\log(k)$. To estimate $Q^{*}_{M}(s,a)$ values using the $V^{*}_{M}(s)$ values we increase the number of necessary length-H^{γ}_{ϵ} random trajectories by a factor of k.

For condition *(ii)*, we adapt the analysis of **E**^{3} [13], which states that if each action in a state s is taken m_2 times, then the transition probabilities and reward in state s can be estimated accurately (see Section 4.4).

4.3. Planning in Fair-E^3

We now formalize the planning steps in **Fair-E**^{3} from known states. For the remainder of our exposition, we make Assumption 2 for convenience (and show how to remove this assumption in the Appendix).

Assumption 2. *T^{*}_{ϵ} is known.*

Fair-E^3 constructs two ancillary MDPs for planning: M_Γ is the *exploitation* MDP, in which the unknown states of M are condensed into a single absorbing state s_0 with no reward. In the known states Γ, transitions are kept intact and the rewards are deterministically set to their mean value. M_Γ thus incentivizes exploitation by giving reward only

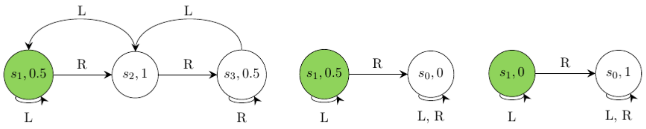

Figure 2. Left: An MDP M with two actions (L and R) and deterministic transition functions and rewards. Green denotes the set of known states Γ. Middle: M_Γ. Right: $M_{[n]\setminus\Gamma}$.

for staying within known states. In contrast, $M_{[n]\setminus\Gamma}$ is the *exploration* MDP, identical to M_Γ except for the rewards. The rewards in the known states Γ are set to 0 and the reward in s_0 is set to 1. $M_{[n]\setminus\Gamma}$ then incentivizes exploration by giving reward only for escaping to unknown states. See the middle (right) panel of Figure 2 for an illustration of M_Γ ($M_{[n]\setminus\Gamma}$), and Appendix for formal definitions.

Fair-E^3 uses these constructed MDPs to plan according to the following natural idea: when in a known state, **Fair-E**^{3} constructs $\hat{M}_\Gamma$ and $\hat{M}_{[n]\setminus\Gamma}$ based on the estimated transition and rewards observed so far (see the Appendix for formal definitions), and then uses these to compute additional restricted MDPs $\hat{M}^\alpha_\Gamma$ and $\hat{M}^\alpha_{[n]\setminus\Gamma}$ for approximate-action fairness. **Fair-E**^{3} then uses these restricted MDPs to choose between exploration and exploitation.

More formally, if the optimal policy in $\hat{M}^\alpha_{[n]\setminus\Gamma}$ escapes to the absorbing state of M_Γ with high enough probability within $2T^*_\epsilon$ steps, then **Fair-E**^{3} explores by following that policy. Otherwise, **Fair-E**^{3} exploits by following the optimal policy in $\hat{M}^\alpha_\Gamma$ for T^*_ϵ steps. While following either of these policies, whenever **Fair-E**^{3} encounters an unknown state, it stops following the policy and proceeds by taking a length-H^γ_ϵ random trajectory.

4.4. Analysis of Fair-E^3

In this section we formally analyze **Fair-E**^{3} and prove Theorem 6. We begin by proving that M^α_Γ is useful in the following sense: M^α_Γ has at least one of an exploitation policy achieving high reward or an exploration policy that quickly reaches an unknown state in M.

Lemma 8 (Exploit or Explore Lemma)**.** *For any state $s \in \Gamma$, $\beta \in (0, 1)$ and any $T > 0$ at least one of the statements below holds:*

- *there exists an* exploitation policy *π in M^α_Γ such that*

  $$\max_{\bar{\pi}\in\Pi} \mathbb{E}\sum_{t=1}^{T} V^{\bar{\pi}}_M\left(\bar{\pi}^t(s), T\right) - \mathbb{E}\sum_{t=1}^{T} V^{\pi}_{M_\Gamma}\left(\pi^t(s), T\right) \leq \beta T$$

  where the random variables $\pi^t(s)$ and $\bar{\pi}^t(s)$ denote the states reached from s after following π and $\bar{\pi}$ for t steps, respectively.
- *there exists an* exploration policy *π in M^α_Γ such that the probability that a walk of $2T$ steps from s following π will terminate in s_0 exceeds β/T.*

We can use this fact to reason about exploration as follows. First, since Observation 2 tells us that the optimal policy in M is approximate-action fair, if the optimal policy stays in the set of M's known states M_Γ, then following the optimal policy in M^α_Γ is both optimal and approximate-action fair.

However, if instead the optimal policy in M quickly escapes to an unknown state in M, the optimal policy in M^α_Γ may not be able to compete with the optimal policy in M. Ignoring fairness, one natural way of computing an escape policy to "keep up" with the optimal policy is to compute the optimal policy in $M_{[n]\setminus\Gamma}$. Unfortunately, following this escape policy might violate approximate-action fairness – high-quality actions might be ignored in lieu of low-quality exploratory actions that quickly reach the unknown states of M. Instead, we compute an escape policy in $M^\alpha_{[n]\setminus\Gamma}$ and show that if no near-optimal exploitation policy exists in M_Γ, then the optimal policy in $M^\alpha_{[n]\setminus\Gamma}$ (which is fair by construction) quickly escapes to the unknown states of M.

Next, in order for **Fair-E**^{3} to check whether the optimal policy in $M^\alpha_{[n]\setminus\Gamma}$ quickly reaches the absorbing state of M_Γ with significant probability, **Fair-E**^{3} simulates the execution of the optimal policy of $M^\alpha_{[n]\setminus\Gamma}$ for $2T^*_\epsilon$ steps from the known state s in M^α_Γ several times, counting the ratio of the runs ending in s_0, and applying a Chernoff bound; this is where Assumption 2 is used.

Having discussed exploration, it remains to show that the exploitation policy described in Lemma 8 satisfies ϵ-optimality as defined in Definition 1. By setting $T \geq T^*_\epsilon$ in Lemma 8 and applying Lemmas 1 and 10, we can prove Corollary 9 regarding this exploitation policy.

Corollary 9. *For any state $s \in \Gamma$ and $T \geq T^*_\epsilon$ if there exists an exploitation policy π in M^α_Γ then*

$$\left|\frac{1}{T}\mathbb{E}\sum_{t=1}^{T} V^{\pi}_M\left(\pi^t(s), T\right) - \mathbb{E}_{s\sim\mu^*}V^*_M\left(s\right)\right| \leq \frac{\epsilon}{1-\gamma}.$$

Finally, we have so far elided the fact that **Fair-E**^{3} only has access to the *empirically estimated* MDPs $\hat{M}^\alpha_\Gamma$ and $\hat{M}^\alpha_{[n]\setminus\Gamma}$ (see the Appendix for formal definitions). We remedy this issue by showing that the behavior of any policy π in $\hat{M}^\alpha_\Gamma$ (and $\hat{M}^\alpha_{[n]\setminus\Gamma}$) is similar to the behavior of π in M^α_Γ (and $M^\alpha_{[n]\setminus\Gamma}$). To do so, we prove a stronger claim: the behavior of any π in $\hat{M}_\Gamma$ (and $\hat{M}_{[n]\setminus\Gamma}$) is similar to the behavior of π in M_Γ (and $M_{[n]\setminus\Gamma}$).

Lemma 10. *Let Γ be the set of known states and $\hat{M}_\Gamma$ the approximation to M_Γ. Then for any state $s \in \Gamma$, any action a and any policy π, with probability at least $1 - \delta$:*

1. $V^{\pi}_{M_\Gamma}(s) - \min\{\alpha/2, \epsilon\} \leq V^{\pi}_{\hat{M}_\Gamma}(s) \leq V^{\pi}_{M_\Gamma}(s) + \min\{\alpha/2, \epsilon\}$,
2. $Q^{\pi}_{M_\Gamma}(s,a) - \min\{\alpha/2, \epsilon\} \leq Q^{\pi}_{\hat{M}_\Gamma}(s,a) \leq Q^{\pi}_{M_\Gamma}(s,a) + \min\{\alpha/2, \epsilon\}$.

We now have the necessary results to prove Theorem 6.

Proof of Theorem 6. We divide the analysis into separate parts: the performance guarantee of **Fair-E**^{3} and its approximate-action fairness. We defer the analysis of the probability of failure of **Fair-E**^{3} to the Appendix.

We start with the performance guarantee and show that when **Fair-E**^{3} follows the exploitation policy the average V^*_M values of the visited states is close to $\mathbb{E}_{s\sim\mu^*}V^*_M(s)$. However, when following an exploration policy or taking random trajectories, visited states' V^*_M values can be small. To bound the performance of **Fair-E**^{3}, we bound the number of these exploratory steps by the MDP parameters so they only have a small effect on overall performance.

Note that in each T^*_ϵ-step exploitation phase of **Fair-E**^{3}, the expectation of the average V^*_M values of the visited states is at least $\mathbb{E}_{s\sim\mu^*}V^*_M(s) - \epsilon/(1-\gamma) - \epsilon/2$ by Lemmas 1, 8 and Observation 5. By a Chernoff bound, the probability that the actual average V^*_M values of the visited states is less than $\mathbb{E}_{s\sim\mu^*}V^*_M(s) - \epsilon/(1-\gamma) - 3\epsilon/4$ is less than $\delta/4$ if there are at least $\frac{\log(\frac{1}{\delta})}{\epsilon^2}$ exploitation phases.

We now bound the total number of exploratory steps of **Fair-E**^{3} by

$$T_1 = O\left(nm_Q H^\gamma_\epsilon + nm_Q \frac{T^*_\epsilon}{\epsilon}\log\left(\frac{n}{\delta}\right)\right),$$

where m_Q is defined in Equation 3 of Definition 7. The two components of this term bound the number of rounds in which **Fair-E**^{3} plays non-exploitatively: the first bounds the number of steps taken when **Fair-E**^{3} follows random trajectories, and the second bounds how many steps are taken following explicit exploration policies. The former bound follows from the facts that each random trajectory has length H^γ_ϵ; that in each state, m_Q trajectories are sufficient for the state to become known; and that random trajectories are taken only before all n states are known. The latter bound follows from the fact that **Fair-E**^{3} follows an exploration policy for $2T^*_\epsilon$ steps; and an exploration policy needs to be followed only $O(\frac{T^*_\epsilon}{\epsilon}\log(\frac{n}{\delta}))$ times before reaching an unknown state (since any exploration policy will end up in an unknown state with probability of at least $\frac{\epsilon}{T^*_\epsilon}$ according to Lemma 8, and applying a Chernoff bound); that an unknown state becomes known after it is visited m_Q times; and that exploration policies are only followed before all states are known.

Finally, to make up for the potentially poor performance in exploration, the number of $2T^*_\epsilon$ steps exploitation phases needed is at least

$$T_2 = O\left(\frac{T_1(1-\gamma)}{\epsilon}\right).$$

Therefore, after $\mathcal{T} = T_1 + T_2$ steps we have

$$\mathbb{E}_{s\sim\mu^*}V^*_M(s) - \frac{1}{\mathcal{T}}\mathbb{E}\sum_{t=1}^{\mathcal{T}} V^*_M(s_t) \leq \frac{2\epsilon}{1-\gamma},$$

as claimed in Equation 2. The running time of **Fair-E**^{3} is $O(\frac{n\mathcal{T}^3}{\epsilon})$: the additional $\frac{n\mathcal{T}^2}{\epsilon}$ factor comes from offline computation of the optimal policies in $\hat{M}^\alpha_\Gamma$ and $\hat{M}^\alpha_{[n]\setminus\Gamma}$.

We wrap up by proving **Fair-E**^{3} satisfies approximate-action fairness in every round. The actions taken during random trajectories are fair (and hence approximate-action fair) by Observation 3. Moreover, **Fair-E**^{3} computes policies in $\hat{M}^\alpha_\Gamma$ and $\hat{M}^\alpha_{[n]\setminus\Gamma}$. By Lemma 10 with probability at least $1-\delta$ any Q^* or V^* value estimated in $\hat{M}^\alpha_\Gamma$ or $\hat{M}^\alpha_{[n]\setminus\Gamma}$ is within $\alpha/2$ of its corresponding true value in M^α_Γ or $M^\alpha_{[n]\setminus\Gamma}$. As a result, $\hat{M}^\alpha_\Gamma$ and $\hat{M}^\alpha_{[n]\setminus\Gamma}$ *(i)* contain all the optimal policies and *(ii)* only contain actions with Q^* values within α of the optimal actions. It follows that any policy followed in $\hat{M}^\alpha_\Gamma$ and $\hat{M}^\alpha_{[n]\setminus\Gamma}$ is α-action fair, so both the exploration and exploitation policies followed by **Fair-E**^{3} satisfy α-action fairness, and **Fair-E**^{3} is therefore α-action fair. □

5. Discussion and Future Work

Our work leaves open several interesting questions. For example, we give an algorithm that has an undesirable exponential dependence on $1/(1-\gamma)$, but we show that this dependence is unavoidable for any approximate-action fair algorithm. Without fairness, near-optimality in learning can be achieved in time that is polynomial in *all* of the parameters of the underlying MDP. So, we can ask: does there exist a *meaningful* fairness notion that enables reinforcement learning in time polynomial in all parameters?

Moreover, our fairness definitions remain open to further modulation. It remains unclear whether one can *strengthen* our fairness guarantee to bind across time rather than simply across actions available at the moment without large performance tradeoffs. Similarly, it is not obvious whether one can gain performance by *relaxing* the every-step nature of our fairness guarantee in a way that still forbids discrimination. These and other considerations suggest many questions for further study; we therefore position our work as a first cut for incorporating fairness into a reinforcement learning setting.

References

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *Propublica*, 2016.

Anna Barry-Jester, Ben Casselman, and Dana Goldstein. The new science of sentencing. *The Marshall Project*, August 8 2015. URL https://www.themarshallproject.org/2015/08/04/the-new-science-of-sentencing/. Retrieved 4/28/2016.

Ronen Brafman and Moshe Tennenholtz. R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

Nanette Byrnes. Artificial intolerance. *MIT Technology Review*, March 28 2016. URL https://www.technologyreview.com/s/600996/artificial-intolerance/. Retrieved 4/28/2016.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science*, pages 214–226, 2012.

Michael Feldman, Sorelle Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268, 2015.

Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im)possibility of fairness. *CoRR*, abs/1609.07236, 2016.

Sara Hajian and Josep Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1445–1459, 2013.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 3315–3323, 2016.

Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 325–333, 2016.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *Proceedings of the 12th IEEE International Conference on Data Mining*, pages 924–929, 2012.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50, 2012.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Michael Kearns, Yishay Mansour, and Andrew Ng. Approximate planning in large POMDPs via reusable trajectories. In *Proceedings of the 13th Annual Conference on Neural Information Processing Systems*, pages 1001–1007, 2000.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of the 7th Conference on Innovations in Theoretical Computer Science*, 2017.

Shiau Hong Lim, Huan Xu, and Shie Mannor. Reinforcement learning in robust markov decision processes. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 701–709, 2013.

Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. k-NN as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 502–510, 2011.

Shie Mannor, Ofir Mebel, and Huan Xu. Lightning does not strike twice: Robust MDPs with coupled uncertainty. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Clair Miller. Can an algorithm hire better than a human? *The New York Times*, June 25 2015. URL http://www.nytimes.com/2015/06/26/upshot/can-an-algorithm-hire-better-than-a-human.html/. Retrieved 4/28/2016.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568. ACM, 2008.

Cynthia Rudin. Predictive policing using machine learning to detect patterns of crime. *Wired Magazine*, August 2013. URL http://www.wired.com/insights/2013/08/predictive-policing-using-machine-learning-to-detect-patterns-of-crime/. Retrieved 4/28/2016.

Satinder Singh. Personal Communication, June 2016.

Richard Sutton and Andrew Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

Latanya Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 56(5):44–54, 2013.

Istv'an Szita and Csaba Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1031–1038, 2010.

Blake Woodworth, Suriya Gunasekar, Mesrob Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Proceedings of the 30th Conference on Learning Theory*, 2017.

Muhammad Bilal Zafar, Isabel Valera, Gomez-Rodriguez Manuel, and Krishna P. Gummadi. Fairness beyond disparate treatment and disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International World Wide Web Conference*, 2017.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning*, pages 325–333, 2013.

A. Omitted Proofs

A.1. Omitted Proofs for Section 2

Proof of Lemma 1. Let $\hat{\mu}^\pi_T$ denote the distribution of π on states of M after following π for T steps starting from s. Then we know

$$\begin{aligned}
\mathbb{E}_{s\sim\mu^\pi} V^\pi_M(s) \; - \frac{1}{T}\mathbb{E}\sum_{t=1}^{T} V^\pi_M(s_t) \\
&= \sum_{i=1}^{n} \left(\mu^\pi(s_i) - \hat{\mu}^\pi_T(s_i)\right) V^\pi_M(s_i) \\
&\leq \sum_{i=1}^{n} \left|\mu^\pi(s_i) - \hat{\mu}^\pi_T(s_i)\right| V^\pi_M(s_i) \\
&\leq \frac{\epsilon}{1-\gamma}.
\end{aligned}$$

The last inequality is due to the following observations: (i) $V^\pi_M(s_i) \leq \frac{1}{1-\gamma}$ as rewards are in $[0,1]$ and (ii) $\Sigma_{i=1}^n \left|\mu^\pi(s_i) - \hat{\mu}^\pi_T(s_i)\right| \leq \epsilon$ since T is at least the ϵ-mixing time of π. □

A.2. Omitted Proofs for Section 3

We first state the following useful Lemma about M.

Lemma 11. *Let M be the MDP in Definition 6. Then for any $i \in \{1,\ldots,n\}$, $V^*_M(s_i) < \frac{1+2\gamma^{n-i+1}}{2(1-\gamma)}$.*

Proof.

$$\begin{aligned}
V^*_M(s_i) &= \text{discounted reward before reaching state } n \\
&\quad + \text{discounted reward from staying at state } n \\
&< \left[\sum_{t=1}^{n-i-1} \frac{\gamma^t}{2}\right] + \frac{\gamma^{n-i+1}}{1-\gamma} \\
&= \left[\frac{1}{2}\left(\frac{1}{1-\gamma} - \frac{\gamma^{n-i}}{1-\gamma}\right)\right] + \frac{\gamma^{n-i+1}}{1-\gamma} \\
&= \frac{1-\gamma^{n-i}}{2(1-\gamma)} + \frac{\gamma^{n-i+1}}{1-\gamma} \\
&= \frac{1+\gamma^{n-i}(2\gamma-1)}{2(1-\gamma)} \\
&< \frac{1+2\gamma^{n-i+1}}{2(1-\gamma)},
\end{aligned}$$

via two applications of the summation formula for geometric series. □

Proof of Theorem 3. We prove Theorem 3 for the special case of $k=2$ first. Consider coupling the run of a fair algorithm $\mathcal{L}$ on both $M(0.5)$ and $M(1)$. To achieve this, we can fix the randomness of $\mathcal{L}$ up front, and use the same randomness on both MDPs. The set of observations and hence the actions taken on both MDPs are identical until $\mathcal{L}$ reaches state s_n. Until then, with probability at least $1-\delta$, $\mathcal{L}$ must play L and R with equal probability in order to satisfy fairness (since, for $M(0.5)$, the only fair policy is to play both actions with equal probability at each time step). We will upper-bound the optimality of uniform play and lower-bound the number of rounds before which s_n is visited by uniformly random play.

Let $f_\gamma = \lceil \frac{1}{1-\sqrt[3]{\gamma}} \rceil$ and $\mathcal{T} = 2^{n-2f_\gamma}$ for $n \geq 100(f_\gamma)^2$. First observe that the probability of reaching a fixed state s_i for any $i \geq n-f_\gamma$ from a random walk of length $\mathcal{T}$ is upper bounded by the probability that the random walk takes $i \geq n-f_\gamma$ consecutive steps to the right in the first $\mathcal{T}$ steps. This probability is at most $p = 2^{n-2f_\gamma}(\frac{1}{2})^{n-f_\gamma} = 2^{-f_\gamma}$ for any fixed i. Since reaching any state $i > i'$ requires reaching state i', the probability that the $\mathcal{T}$ step random walk arrives in any state s_i for $i \geq n-f_\gamma$ is also upper bounded by p.

Next, we observe that $V^*_M(s_i)$ is a nondecreasing function of i for both MDPs. Then the average V^*_M values of the visited states of *any* fair policy can be broken into two pieces: the average conditioned on (the probability at least $1-\delta$ event) that the algorithm plays uniformly at random before reaching state s_n *and* never reaching a state beyond s_{n-f_γ}, and the average conditioned on (the probability at most δ event) that the algorithm does not make uniformly random choices *or* the uniform random walk of length $\mathcal{T}$ reaches a state beyond s_{n-f_γ}. So, we have that

$$\begin{aligned}
\frac{1}{\mathcal{T}}\mathbb{E}\sum_{t=1}^{\mathcal{T}} V^*_M(s_t) &\leq (1-p-\delta)\, V^*_M(s_{n-f_\gamma}) + (p+\delta)\,\frac{1}{1-\gamma} \\
&\leq (1-p-\delta)\,\frac{1+2\gamma^{f_\gamma+1}}{2(1-\gamma)} + (p+\delta)\,\frac{1}{1-\gamma}.
\end{aligned}$$

The first inequality follows from the fact that $V^*_M(s_i) \leq \frac{1}{1-\gamma}$ for all i, and the second from Lemma 11 along with V^*_M values being nondecreasing in i. Putting it all together,

$$\begin{aligned}
&\mathbb{E}_{s\sim\mu^*}\; V^*_M(s) - \frac{1}{\mathcal{T}}\mathbb{E}\sum_{t=1}^{\mathcal{T}} V^*_M(s_t) \\
&\quad\geq \frac{1}{1-\gamma} - \left[(1-p-\delta)\,\frac{1+2\gamma^{f_\gamma+1}}{2(1-\gamma)} + (p+\delta)\,\frac{1}{1-\gamma}\right] \\
&\quad= \frac{1-p-\delta}{1-\gamma}\left[1 - \frac{1+2\gamma^{f_\gamma+1}}{2}\right].
\end{aligned}$$

So ϵ-optimality requires

$$\frac{2\epsilon}{1-\gamma} \geq \frac{1-p-\delta}{1-\gamma}\left[1 - \frac{1+2\gamma^{f_\gamma+1}}{2}\right]. \tag{4}$$

However, if $\epsilon < \frac{1}{8}$ we get

$$\begin{aligned}\frac{2\epsilon}{1-\gamma} &< \frac{1-0.04-1/4}{1-\gamma}\left[1-\frac{1+2\times e^{-3}}{2}\right] \\ &< \frac{1-2^{-f_\gamma}-\delta}{1-\gamma}\left[1-\frac{1+2\gamma^{f_\gamma+1}}{2}\right],\end{aligned}$$

where the third inequality follows when $\delta < \frac{1}{4}$ and $\gamma > \frac{1}{2}$. This means $\epsilon < \frac{1}{8}$ makes ϵ-optimality impossible, as desired.

Throughout we considered the special case of $k = 2$ and proved a lower bound of $\Omega(2^n)$ time steps for any fair algorithm satisfying the ϵ-optimality condition. However, it is easy to see that MDP M in Definition 6 can be easily modified in a way that $k-1$ of the actions from state s_i reach state s_1 and only one action in each state s_i reaches states $s_{\min\{i+1,n\}}$. Hence, a lower bound of $\Omega(k^n)$ time steps can be similarly proved. □

Proof of Theorem 4. We mimic the argument used to prove Theorem 3 with the difference that, until visiting s_n, $\mathcal{L}$ may not play R with probability more than $\frac{1}{2}+\alpha$ (as opposed to $\frac{1}{2}$ in Theorem 3). Let $f_\gamma = \lceil\frac{1}{1-\sqrt[3]{\gamma}}\rceil$ and $\mathcal{T} = (\frac{2}{1+2\alpha})^{n-2f_\gamma}$ for $n \geq 100(f_\gamma)^2$. By a similar process as in Theorem 3, the probability of reaching state s_i for any $i \geq n - f_\gamma$ from a random walk of length $\mathcal{T}$ is bounded by $p = (\frac{2}{1+2\alpha})^{-f_\gamma}$, and so the probability that the $\mathcal{T}$ steps random walk arrives in any state s_i for $i \geq n - f_\gamma$ is bounded by p. Carrying out the same process used to prove Theorem 3 then once more implies that ϵ-optimality requires Equation 4 to hold when $\delta < \frac{1}{4}$, $\alpha < \frac{1}{4}$ and $\gamma > \frac{1}{2}$. Hence, $\epsilon < \frac{1}{8}$ violates this condition as desired.

Finally, throughout we considered the special case of $k = 2$. The same trick as in the proof of Theorem 3 can be used to prove the lower bound of $\Omega((\frac{k}{1+k\alpha})^n)$ time steps for any fair algorithm satisfying the ϵ-optimality condition. □

Proof of Theorem 5. We also prove Theorem 5 for the special case of $k = 2$ first, again considering the MDP in Definition 6. We set the size of the state space in M to be $n = \lceil\frac{\log(\frac{1}{2\alpha})}{1-\gamma}\rceil$. Then given the parameter ranges, for any i, $Q^*_M(s_i, R) - Q^*_M(s_i, L) > \alpha$ in M(1). Therefore, any approximate-action fair algorithm should play actions R and L with equal probability.

Let $\mathcal{T} = 2^{cn} = \Omega((2^{1/(1-\gamma)})^c)$. First observe that the probability of reaching a fixed state s_i for any $i \geq (c+1)n/2$ from a random walk of length $\mathcal{T}$ is upper bounded by the probability that the random walk takes $i \geq (c+1)n/2$ consecutive steps to the right in the first $\mathcal{T}$ steps. This probability is at most $p = 2^{cn}2^{-(c+1)n/2} = 2^{(c-1)n/2}$ for any fixed i. Then the probability that the $\mathcal{T}$ steps random walk arrives in any state s_i for $i \geq (c+1)n/2$ is also upper bounded by p.

Next, we observe that $V^*_M(s_i)$ is a nondecreasing function of i, for both MDPs. Then the average V^*_M values of the visited states of *any* fair policy can be broken into two pieces: the average conditioned on the $1-\delta$ fairness *and* never reaching a state beyond $s_{(c+1)n/2}$, and the average when fairness might be violated *or* the uniform random walk of length $\mathcal{T}$ reaches a state beyond $s_{(c+1)n/2}$. So, we have that

$$\begin{aligned}\frac{1}{\mathcal{T}}\mathbb{E}\sum_{t=1}^{\mathcal{T}} V^*_M(s_t) &\leq (1-p-\delta)\, V^*_M(s_{(c+1)n/2}) \\ &\quad + (p+\delta)\,\frac{1}{1-\gamma} \\ &\leq (1-p-\delta)\,\frac{1+(2\gamma-1)\gamma^{\frac{(1-c)n}{2}}}{2(1-\gamma)} \\ &= (p+\delta)\,\frac{1}{1-\gamma}.\end{aligned}$$

The first inequality follows from the fact that $V^*_M(s_i) \leq \frac{1}{1-\gamma}$ for all i, and the second from (the line before the last in) Lemma 11 along with V^*_M values being nondecreasing in i. Putting it all together,

$$\begin{aligned}&\mathbb{E}_{s\sim\mu^*}\, V^*_M(s) - \frac{1}{\mathcal{T}}\mathbb{E}\sum_{t=1}^{\mathcal{T}} V^*_M(s_t) \\ &\geq \frac{1}{1-\gamma} - (1-p-\delta)\,\frac{1+(2\gamma-1)\gamma^{\frac{(1-c)n}{2}}}{2(1-\gamma)} \\ &- (p+\delta)\,\frac{1}{1-\gamma} \\ &= \frac{1-p-\delta}{1-\gamma}\left[1-\frac{1+(2\gamma-1)\gamma^{\frac{(1-c)n}{2}}}{2}\right] \\ &= \frac{1-p-\delta}{1-\gamma}\left[\frac{1}{2}-\frac{(2\gamma-1)\gamma^{\frac{(1-c)n}{2}}}{2}\right].\end{aligned}$$

So ϵ-optimality requires

$$\frac{2\epsilon}{1-\gamma} \geq \frac{1-p-\delta}{1-\gamma}\left[\frac{1}{2}-\frac{(2\gamma-1)\gamma^{\frac{(1-c)n}{2}}}{2}\right].$$

Rearranging and using $\delta < \frac{1}{4}$, we get that ϵ-optimality requires

$$4\epsilon \geq \left[0.75 - 2^{\frac{(c-1)n}{2}}\right]\left[1-(2\gamma-1)\gamma^{\frac{(1-c)n}{2}}\right]$$

and expand n to get

$$\begin{aligned}\epsilon \geq \frac{1}{4}&\left[0.75 - 2^{\frac{(c-1)\log(\frac{1}{2\alpha})}{2(1-\gamma)}}\right]\times \\ &\left[1-(2\gamma-1)\gamma^{\frac{(1-c)\log(\frac{1}{2\alpha})}{2(1-\gamma)}}\right] \equiv \frac{xy}{4}.\end{aligned}$$

Noting that x is minimized when $2^{\frac{(c-1)\log(\frac{1}{2\alpha})}{2(1-\gamma)}}$ is maximized, and that this quantity is maximized when $\frac{\log(\frac{1}{2\alpha})}{2(1-\gamma)}$ is minimized (as $c-1$ is negative), we get that ϵ-optimality requires

$$\epsilon \geq \frac{\left[0.75 - 2^{\frac{c-1}{1-\gamma}}\right] y}{4}$$

from $\alpha < \frac{1}{8}$. Similarly, $\alpha < \frac{1}{8}$ implies that ϵ-optimality requires

$$\epsilon \geq \frac{\left[0.75 - 2^{\frac{c-1}{1-\gamma}}\right]\left[1-(2\gamma-1)\gamma^{\frac{1-c}{1-\gamma}}\right]}{4}.$$

Note that $0.75 - 2^{\frac{c-1}{1-\gamma}}$ is minimized when γ is small, so $\gamma > c$ implies that ϵ-optimality requires

$$\begin{aligned}\epsilon &\geq \frac{\left[0.75 - 2^{-1}\right]\left[1-(2\gamma-1)\gamma^{\frac{1-c}{1-\gamma}}\right]}{4} \\ &\geq \frac{1}{16}\left[1-(2\gamma-1)\gamma^{\frac{1-c}{2(1-\gamma)}}\right].\end{aligned}$$

Conversely, $1-(2\gamma-1)\gamma^{\frac{1-c}{1-\gamma}}$ is minimized when γ is large, so as

$$\lim_{\gamma\to 1}\left(2\gamma-1\right)\gamma^{\frac{1-c}{1-\gamma}} = e^{c-1}$$

we get that ϵ-optimality requires

$$\epsilon \geq \frac{1}{16}\left(1-e^{c-1}\right).$$

Finally, the same trick as in the proof of Theorem 3 can be used to prove the $\Omega((k^{1/(1-\gamma)})^c)$ lower bound for $k > 2$ actions. □

A.3. Omitted Proofs for Section 4

Proof of Lemma 8. We first show that either

- there exists an *exploitation policy* π in M_Γ such that

$$\frac{1}{T}\max_{\bar{\pi}\in\Pi}\mathbb{E}\sum_{t=1}^{T}V_M^{\bar{\pi}}\left(\bar{\pi}^t(s),T\right)-\frac{1}{T}\mathbb{E}\sum_{t=1}^{T}V_{M_\Gamma}^{\pi}\left(\pi^t(s),T\right)\leq\beta$$

  where the random variables $\pi^t(s)$ and $\bar{\pi}^t(s)$ denote the states reached from s after following π and $\bar{\pi}$ for t steps, respectively, or
- there exists an *exploration policy* π in M_Γ such that the probability that a walk of $2T$ steps from s following π will terminate in s_0 exceeds $\frac{\beta}{T}$.

Let π be a policy in M satisfying

$$\frac{1}{T}\mathbb{E}\sum_{t=1}^{T}V_M^{\pi}(\pi^t(s),T) = \frac{1}{T}\max_{\bar{\pi}\in\Pi}\mathbb{E}\sum_{t=1}^{T}V_M^{\pi'}(\bar{\pi}^t(s),T) := \tilde{V}.$$

For any state s', let $p(s')$ denote all the paths of length T in M that start in s', $q(s')$ denote all the paths of length T in M that start in s' such that all the states in every path of length T in $q(s')$ are in Γ and $r(s')$ all the paths of length T in M that start in s' such that at least one state in every path of length T in $r(s')$ is not in Γ. Suppose

$$\frac{1}{T}\mathbb{E}\sum_{t=1}^{T}V_{M_\Gamma}^{\pi}(\pi^t(s)) < \tilde{V}-\beta.$$

Otherwise, π already witnesses the claim. We show that a walk of $2T$ steps from s following π will terminate in s_0 with probability of at least $\frac{\beta}{T}$. First,

$$\begin{aligned}\mathbb{E}\sum_{t=1}^{T}V_M^{\pi}(\pi^t(s),T) &= E\sum_{t=1}^{T}\sum_{p(\pi^t(s))}\mathbb{P}[p(\pi^t(s))]V_M(p(\pi^t(s))) \\ &= \mathbb{E}\sum_{t=1}^{T}\sum_{q(\pi^t(s))}\mathbb{P}[q(\pi^t(s))]V_M(q(\pi^t(s))) \\ &+ \mathbb{E}\sum_{t=1}^{T}\sum_{r(\pi^t(s))}\mathbb{P}[r(\pi^t(s))]V_M(r(\pi^t(s)))\end{aligned}$$

since $p(\pi^t(s)) = q(\pi^t(s)) \cup r(\pi^t(s))$, which is a disjoint union. Next,

$$\begin{aligned}&\mathbb{E}\sum_{t=1}^{T}\sum_{q(\pi^t(s))}\mathbb{P}[q(\pi^t(s))]V_M(q(\pi^t(s))) \\ &= \mathbb{E}\sum_{t=1}^{T}\sum_{q(\pi^t(s))}\mathbb{P}_{M_\Gamma}^{\pi}[q(\pi^t(s))]V_{M_\Gamma}(q(\pi^t(s))) \\ &\leq \mathbb{E}\sum_{t=1}^{T}V_{M_\Gamma}^{\pi}(\pi^t(s),T),\end{aligned}$$

where the equality is due to Definition 9 and the definition of q, and the inequality follows because $V_{M_\Gamma}^{\pi}(\pi^t(s),T)$ is the sum over all the T-paths in M_Γ, not just those that avoid the absorbing state s_0. Therefore by our original assumption on π,

$$\begin{aligned}&\mathbb{E}\sum_{t=1}^{T}\sum_{q(\pi^t(s))}\mathbb{P}[q(\pi^t(s))]V_M(q(\pi^t(s))) \\ &\leq \mathbb{E}\sum_{t=1}^{T}V_{M_\Gamma}^{\pi}(\pi^t(s),T) < T\tilde{V}-T\beta.\end{aligned}$$

This implies

$$
\begin{aligned}
&\mathbb{E}\sum_{t=1}^{T}\sum_{r(\pi^t(s))}\mathbb{P}[r(\pi^t(s))]V_M(r(\pi^t(s)))\\
&=\mathbb{E}\sum_{t=1}^{T}V_M^{\pi}(\pi^t(s),T)\\
&-\mathbb{E}\sum_{t=1}^{T}\sum_{q(\pi^t(s))}\mathbb{P}[q(\pi^t(s))]V_M(q(\pi^t(s)))\\
&=T\tilde{V}-\mathbb{E}\sum_{t=1}^{T}\sum_{q(\pi^t(s))}\mathbb{P}[q(\pi^t(s))]V_M(q(\pi^t(s)))\geq T\beta,
\end{aligned}
$$

where the last step is the result of applying the previous inequality. However,

$$
\begin{aligned}
&\mathbb{E}\sum_{t=1}^{T}\sum_{r(\pi^t(s))}\mathbb{P}[r(\pi^t(s))]V_M(r(\pi^t(s)))\\
&\leq T\mathbb{E}\sum_{t=1}^{T}\sum_{r(\pi^t(s))}\mathbb{P}[r(\pi^t(s))],
\end{aligned}
$$

because it is immediate that $V_M(r(\pi^t(s)))\leq T$ for all $\pi^t(s)$. So $T\beta\leq T\mathbb{E}\sum_{t=1}^{T}\sum_{r(\pi^t(s))}\mathbb{P}[r(\pi^t(s))]$. Finally, if we let $\mathbb{P}_{2T}^{\pi}$ denote the probability that a walk of $2T$ steps following π terminates in s_0, i.e. the probability that π escapes to an unknown state within $2T$ steps, then for each $t\in[T]$, $\mathbb{E}\sum_{r(\pi^t(s))}\leq T\mathbb{P}_{2T}^{\pi}$. It follows that

$$T\beta\leq T^2\mathbb{P}_{2T}^{\pi}$$

and rearranging yields $\mathbb{P}_{2T}^{\pi}\geq\frac{\beta}{T}$ as desired.

Next, note that the exploitation policy (if it exists) can be derived by computing the optimal policy in M_Γ. Moreover, the exploration policy (if it exists) in the exploitation MDP M_Γ can indeed be derived by computing the optimal policy in the exploration MDP $M_{[n]\setminus\Gamma}$ as observed by (Kearns and Singh, 2002). Finally, by Observation 5, any optimal policy in $\hat{M}_\Gamma^\alpha$ ($\hat{M}_{[n]\setminus\Gamma}^\alpha$) is an optimal policy in $\hat{M}_\Gamma$ ($\hat{M}_{[n]\setminus\Gamma}$) □

To prove Lemma 10, we need some useful background adapted from Kearns and Singh (2002).

Definition 8 (Definition 7, Kearns and Singh (2002)). *Let M and $\hat{M}$ be two MDPs with the same set of states and actions. We say $\hat{M}$ is a β-approximation of M if*

- *For any state s,*

$$\bar{R}_M(s)-\beta\leq\bar{R}_{\hat{M}}(s)\leq\bar{R}_M(s)+\beta.$$

- *For any states s and s' and action a,*

$$P_M(s,a,s')-\beta\leq P_{\hat{M}}(s,a,s')\leq P_M(s,a,s')+\beta.$$

Lemma 12 (Lemma 5, Kearns and Singh (2002)). *Let M be an MDP and Γ the set of known states of M. For any $s,s'\in\Gamma$ and action $a\in A$, let $\hat{P}_M(s,a,s')$ denote the empirical probability transition estimates obtained from the visits to s. Moreover, for any state $s\in\Gamma$ let $\bar{\hat{R}}(s)$ denote the empirical estimates of the average reward obtained from visits to s. Then with probability at least $1-\delta$,*

$$|\hat{P}_M(s,a,s')-P_M(s,a,s')|=O\left(\frac{\min\{\epsilon,\alpha\}^2}{n^2{H_\epsilon^{\gamma}}^4}\right),$$

and

$$|\bar{\hat{R}}_M(s)-\bar{R}_M(s)|=O\left(\frac{\min\{\epsilon,\alpha\}^2}{n^2{H_\epsilon^{\gamma}}^4}\right).$$

Lemma 12 shows that $\hat{M}_\Gamma$ and $\hat{M}_{[n]\setminus\Gamma}$ are $O(\frac{\min\{\epsilon,\alpha\}^2}{n^2{H_\epsilon^{\gamma}}^4})$-approximation MDPs for M_Γ and $M_{[n]\setminus\Gamma}$, respectively.

Lemma 13 (Lemma 4, Kearns and Singh (2002)). *Let M be an MDP and $\hat{M}$ its $O(\frac{\min\{\epsilon,\alpha\}^2}{n^2{H_\epsilon^{\gamma}}^4})$-approximation. Then for any policy $\pi\in\Pi$ and any state s and action a*

$$V_M^\pi(s)-\min\{\epsilon,\alpha\}\leq V_{\hat{M}}^\pi(s)\leq V_M^\pi(s)+\min\{\epsilon,\frac{\alpha}{4}\},$$

and

$$
\begin{aligned}
Q_M^\pi(s,a)-\min\{\frac{\alpha}{4},\epsilon\}&\leq Q_{\hat{M}}^\pi(s,a)\\
&\leq Q_M^\pi(s,a)+\min\{\frac{\alpha}{4},\epsilon\}.
\end{aligned}
$$

Proof of Lemma 10. By Definition 7 and Lemma 12, $\hat{M}_\Gamma$ is a $O(\frac{\min\{\epsilon,\alpha\}^2}{n^2{H_\epsilon^{\gamma}}^4})$-approximation of M_Γ. Then the statement directly follows by applying Lemma 13. □

Rest of the Proof of Theorem 6. The only remaining part of the proof of Theorem 6 is the analysis of the probability of failure of **Fair-E**^{3}. To do so, we break down the probability of failure of **Fair-E**^{3} by considering the following (exhaustive) list of possible failures:

1. At some known state the algorithm has a poor approximation of the next step, causing $\hat{M}_\Gamma$ to not be a $O(\frac{\min\{\epsilon,\alpha\}^2}{n^2{H_\epsilon^{\gamma}}^4})$-approximation of M_Γ.
2. At some known state the algorithm has a poor approximation of the Q_M^* values for one of the actions.
3. Following the exploration policy for $2T_\epsilon^*$ steps fails to yield enough visits to unknown states.
4. At some known state, the approximation value of that state in $\hat{M}_\Gamma$ is not an accurate estimate for the value of the state in M_Γ.

We allocate $\frac{\delta}{4}$ of our total probability of failure to each of these sources:

1. Set $\delta' = \frac{\delta}{4n}$ in Lemma 10.
2. Set $\delta' = \frac{\delta}{4nk}$ in Theorem 7.
3. By Lemma 8, each attempted exploration is a Bernoulli trial with probability of success of at least $\frac{\epsilon}{4T_\epsilon^*}$. In the worst case we might need to make every state known before exploiting, leading to the nm_Q trajectories (m_Q as Equation 3 in Definition 7) of length H_ϵ^γ. Therefore, the probability of taking fewer than nm_Q trajectories of length H_ϵ^γ would be bounded by $\frac{\delta}{4}$ if the number of $2T_\epsilon^*$ steps explorations is at least
$$m_{\text{exp}} = O\left(\frac{T_\epsilon^* nm_Q}{\epsilon}\log\left(\frac{n}{\delta}\right)\right). \tag{5}$$
4. Set $\delta' = \frac{\delta}{4m_{\text{exp}}}$ (m_{exp} as defined in Equation 5) in Lemma 10, as **Fair-E**^{3} might make $2T_\epsilon^*$ steps explorations up to m_{exp} times.

□

A.4. Relaxing Assumption 2

Throughout Sections 4.3 and 4.4 we assumed that T_ϵ^*, the ϵ-mixing time of the optimal policy π^*, was known (see Assumption 2). Although **Fair-E**^{3} uses the knowledge of T_ϵ^* to decide whether to follow the exploration or exploitation policy, Lemma 8 continues to hold even without this assumption. Note that **Fair-E**^{3} is parameterized by T_ϵ^* and for any input T_ϵ^* runs in time **poly**(T_ϵ^*). Thus if T_ϵ^* is unknown, we can simply run **Fair-E**^{3} for $T_\epsilon^* = 1, 2, \ldots$ sequentially and the running time and sample complexity will still be **poly**(T_ϵ^*). Similar to the analysis of **Fair-E**^{3} when T_ϵ^* is known we have to run the new algorithm for sufficiently many steps so that the possibly low V_M^* values of the visited states in the early stages are dominated by the near-optimal V_M^* values of the visited states for large enough guessed values of T_ϵ^*.

B. Observations on Optimality and Fairness

Observation 1. *For any MDP M, there exists an optimal policy π^* such that π^* is fair.*

Proof. In time t, let state s_t denote the state from which π chooses an action. Let $a^* = \text{argmax}_a\, Q_M^*(s_t, a)$ and $A^*(s_t) = \{a \in A \mid Q_M^*(s_t, a) = Q_M^*(s_t, a^*)\}$. The policy of playing an action uniformly at random from $A^*(s_t)$ in state s_t for all t, is fair and optimal. □

Approximate-action fairness, conversely, can be satisfied by *any* optimal policy, even a deterministic one.

Observation 2. *Let π^* be an optimal policy in MDP M. Then π^* is approximate-action fair.*

Proof. Assume that π^* is not approximate-action fair. Given state s, the action that π^* takes from s is uniquely determined since π^* is deterministic we may denote it by a^*. Then there exists a time step in which π^* is in state s and chooses action $a^*(s)$ such that there exists another action a with
$$Q_M^*(s, a) > Q_M^*(s, a^*(s)) + \alpha,$$
a contradiction of the optimality of π^*. □

Observations 1 and 2 state that policies with optimal performance are fair; we now state that playing an action uniformly at random is also fair.

Observation 3. *An algorithm that, in every state, plays each action uniformly at random (regardless of the history) is fair.*

Proof. Let $\mathcal{L}$ denote an algorithm that in every state plays uniformly at random between all available actions. Then $\mathcal{L}(s, h_{t-1})_a = \mathcal{L}(s, h_{t-1})_{a'}$ regardless of state s, (available) action a, or history h_{t-1}. $Q_M^*(s, a) > Q_M^*(s, a') + \alpha \Rightarrow \mathcal{L}(s, h_{t-1})_a \geq \mathcal{L}(s, h_{t-1})_{a'}$ then follows immediately, which guarantees both fairness and approximate-action fairness. □

Observation 4. *Let M be an MDP and M^α the α-restricted MDP of M. Let π be a policy in M^α. Then π is α-action fair.*

Proof. Assume π is not α-action fair. Then there must exist round t, state s, and action a such that $Q_M^*(s, a) > Q_M^*(s, a') + \alpha$ and $\mathcal{L}(s, h_{t-1})_a < \mathcal{L}(s, h_{t-1})_{a'}$. Therefore $\mathcal{L}(s, h_{t-1})_{a'} > 0$, so M^α must include action a' from state s. But this is a contradiction, as in state s M^α only includes actions a' such that $Q_M^*(s, a') + \alpha \geq Q_M^*(s, a)$. π is therefore α-action fair. □

Observation 5. *Let M be an MDP and M^α the α-restricted MDP of M. Let π^* be an optimal policy in M^α. Then π^* is also optimal in M.*

Proof. If π^* is not optimal in M, then there exists a state s and action a such that $Q_M^*(s, a) > \mathbb{E}_{a^*(s) \sim \pi^*(s)} Q_M^*(s, a^*(s))$ where $a^*(s)$ is drawn from $\pi^*(s)$ and the expectation is taken over choices of $a^*(s)$. This is a contradiction because action a is available from state s in M^α by Definition 5. □

C. Omitted Details of Fair-E^3

We first formally define the exploitation MDP M_Γ and the exploration MDP $M_{[n]\setminus\Gamma}$:

Definition 9 (Definition 9, Kearns and Singh (2002)). *Let $M = (\mathcal{S}_M, \mathcal{A}_M, P_M, R_M, T, \gamma)$ be an MDP with state space $\mathcal{S}_M$ and let $\Gamma \subset \mathcal{S}_M$. We define the* exploration MDP $M_\Gamma = (\mathcal{S}_{M_\Gamma}, \mathcal{A}_M, P_{M_\Gamma}, R_{M_\Gamma}, T, \gamma)$ *on Γ where*

- $\mathcal{S}_{M_\Gamma} = \Gamma \cup \{s_0\}$.
- *For any state $s \in \Gamma$, $\bar{R}_{M_\Gamma}(s) = \bar{R}_M(s)$, rewards in M_Γ are deterministic, and $\bar{R}_{M_\Gamma}(s_0) = 0$.*
- *For any action a, $P_{M_\Gamma}(s_0, a, s_0) = 1$. Hence, s_0 is an absorbing state.*
- *For any states $s_1, s_2 \in \Gamma$ and any action a, $P_{M_\Gamma}(s_1, a, s_2) = P_M(s_1, a, s_2)$, i.e. transitions between states in Γ are preserved in M_Γ.*
- *For any state $s_1 \in \Gamma$ and any action a, $P_{M_\Gamma}(s_1, a, s_0) = \Sigma_{s_2 \notin \Gamma} P_M(s_1, a, s_2)$. Therefore, all the transitions between a state in Γ and states not in Γ are directed to s_0 in M_Γ.*

Definition 10 (Implicit, Kearns and Singh (2002)). *Given MDP M and set of known states Γ, the* exploration MDP *$M_{[n]\setminus\Gamma}$ on Γ is identical to the exploitation MDP M_Γ except for its reward function. Specifically, rewards in $M_{[n]\setminus\Gamma}$ are deterministic as in M_Γ, but for any state $s \in \Gamma$, $\bar{R}_{M_{[n]\setminus\Gamma}}(s) = 0$, and $\bar{R}_{M_{[n]\setminus\Gamma}}(s_0) = 1$.*

We next define the approximation MDPs $\hat{M}_\Gamma$ and $\hat{M}_{[n]\setminus\Gamma}$ which are defined over the same set of states and actions as in M_Γ and $M_{[n]\setminus\Gamma}$, respectively.

Let M be an MDP and Γ the set of known states of M. For any $s, s' \in \Gamma$ and action $a \in A$, let $\hat{P}_{M_\Gamma}(s, a, s')$ denote the empirical probability transition estimates obtained from the visits to s. Moreover, for any state $s \in \Gamma$ let $\hat{\bar{R}}_{M_\Gamma}(s)$ denote the empirical estimates of the average reward obtained from visits to s. Then $\hat{M}_\Gamma$ is identical to M_Γ except that:

- in any known state $s \in \Gamma$, $\hat{R}_{\hat{M}_\Gamma}(s) = \hat{\bar{R}}_{M_\Gamma}(s)$.
- for any $s, s' \in \Gamma$ and action $a \in A$, $P_{\hat{M}_\Gamma}(s, a, s') = \hat{P}_{M_\Gamma}(s, a, s')$.

Also $\hat{M}_{[n]\setminus\Gamma}$ is identical to $M_{[n]\setminus\Gamma}$ except that:

- for any $s, s' \in \Gamma$ and action $a \in A$, $P_{\hat{M}_{[n]\setminus\Gamma}}(s, a, s') = \hat{P}_{M_{[n]\setminus\Gamma}}(s, a, s')$.