A Snapshot of the Frontiers of Fairness in Reinforcement Learning

Devin Ma[1] and Ann-Katrin Reuel[1,2]

[1] *University of Pennsylvania*
[2] *Stanford University*
{*devinma, akreuel*}*@seas.upenn.edu*

March 2022

1 Introduction

Reinforcement learning (RL) focuses on how autonomous agents interact in an environment to maximize a pre-specified cumulative reward. Since Jabbari et al. (2017) initiated the study of fairness in RL, there has been growing interest in examining the theory and applications of fairness metrics in this area. In our literature review, we will look at the current state of the research and compare different approaches as well as show potential future research directions.

Typically, RL maximizes for the long-term discounted reward of an agent where the reward is tailored to one specific objective. This can be overall output, points in a game, or stock market gains, for example. While this ensures that the reward with respect to the objective is maximized, it does not involve considerations of group or individual fairness (Weng, 2019). Since the perceived fairness of a system can impact human performance or trust, this can lead to an overall sub-optimal performance (Elmalaki, 2021).

Ignoring fairness in certain RL systems comes with a variety of negative ramifications. First, in many developed countries, explicit mandates exists to guarantee fairness. Developers and/or companies using RL-based technologies need to ensure that their algorithms don't discriminate on employment status, education, race, and religion by law. Second, if certain groups are underrepresented, user-centric systems might be abandoned altogether, both by the unfairly treated groups and other users. For example, if an RL-algorithm pushes white content creators on a platform like YouTube more than minority creators, the latter might leave the platform due to inadequate exposure. On the other hand, the

content on the platform will become more homogeneous, leading to less satisfied users. Both can lead to a decrease in income for the company (Liu et al., 2020). Third, in addition to business implications, using RL-algorithms without fairness guarantees can result in potentially discriminatory outcomes that are unethical at best and harmful at worst.

While fairness in RL as a research area has gained traction in recent years, it remains an understudied field. Hence, we aim to make two contributions to the current research with our work. To the best of our knowledge, we put together the first literature review on fairness in RL which builds a basis for future discussions in the field. Secondly, we critically discuss current RL fairness approaches to identify optimization potential and future research directions.

The rest of the paper is organized as follows. In Section 2, we introduce fairness definitions used in RL systems. We then show potential application domains for fair RL systems in Section 3. Section 4 follows with a description of the methods used to introduce fairness notions in RL models, before we describe trade-offs in Section 5. We conclude our work with an overview of potential future research directions in Section 6.

2 Definitions of Fairness

One of the difficulties with studying fairness in RL comes from the numerous definitions of fairness one can adopt. Different notions of fairness have been debated in fields such as philosophy and economics long before the area of fairness in machine learning emerged.

There have been different definitions of fairness proposed in a variety of contexts and applications. Below, we report some of the most prevalent definitions of fairness, including Q-value-based definitions, welfare economics definitions, weighted proportional fairness, the coefficient of variation, and α-fair utility.

2.1 Welfare Economics Definitions

Siddique et al. (2020) and Zimmer et al. (2021) consider fairness in RL and attempt to encode fairness requirements as social welfare functions in the objective function. Siddique et al. (2020) research learning fair policies in *single-agent* multi-objective RL, while Zimmer et al. (2021) further extend the same definitions of fairness to *multi-agent* RL.

In both papers, the definition of fairness consists of three components: impartiality, equity, and efficiency. These definitions are encoded in social welfare functions (SWF), which measure how good a utility vector is with respect to

social good. Concretely, a SWF is a function $\phi : \mathbb{R}^D \to \mathbb{R}$, where D is the number of objectives in Siddique et al. (2020) and the number of users[1] in Zimmer et al. (2021).

At the heart of this fairness notion are definitions of impartiality, equity, and efficiency. First, impartiality means that all agents are identical. In utility vectors, this means that permutations of a utility vector are equivalent solutions. In terms of SWFs, this means that $\phi(u) = \phi(u')$, where u and u' are two vectors of any permutation.

Secondly, equity is based on the *Pigou-Dalton principle*, which states that a transfer of reward from a better-off agent to a worse-off agent yields a more desirable utility profile given the total reward/utility of all agents remains the same. Formally, lets assume two utility vectors $u = (u_1, u_2, \cdots, u_n) \in \mathbb{R}^D$. If $u_j - u_i > \epsilon > 0$, meaning that agent j has more utility than agent i, then we can construct a new utility profile $u' = u + \epsilon e_i - \epsilon e_j$, where e_j is a vector with 1 at index j and zero everywhere else, and e_i is a vector with 1 at index i and zero everywhere else. In terms of SWFs, the Pigou-Dalton principle implies that ϕ is Schur-concave (Zimmer et al., 2021).

Thirdly, efficiency is the idea that between two potential solutions, if one solution is preferred by all agents, weakly or strictly, then that should be chosen over the other (Zimmer et al., 2021). This efficiency requirement is necessary because it prevents a situation where giving no reward to all users is being treated the same as giving non-zero rewards to all users, even though both satisfy the impartiality and equity requirements. In terms of SWFs, the efficiency requirement is captured by the idea that if u dominates u', then $\phi(u) > \phi(u')$.

Despite having the same three components in the definition of fairness, it is important to realize the differences in the definitions. Siddique et al. (2020) consider learning fair policies in *single-agent* deep reinforcement learning. Because of the single-agent setting, the notion of "fairness" is not across multiple users or agents, but rather among multiple objectives or criteria. Informally, fairness in this single-agent context refers to how the solution "balances" the different objectives. In the multi-agent setting in Zimmer et al. (2021), the critical assumption is that all users are identical. This implies, with regards to the impartiality requirement, that users should be treated at least similarly.

2.2 Weighted Proportional Fairness Definition

Liu et al. (2020) use weighted proportional fairness as their target fairness metric in the context of interactive recommender systems (IRS). They first define an

[1] Zimmer et al. (2021) formulated the problem in terms of "users", not agents, because a user can represent an individual or a group of individuals, leading to a more general representation.

allocation vector x_t^i representing the allocation proportion of a group i up to time t:

$$x_t^i = \frac{\sum_{k=1}^{t} y_{a_k} \mathbb{I}_{\mathcal{A}c_i}(a_k)}{\sum_{i'=1}^{l} \sum_{k=1}^{t} y_{a_k} \mathbb{I}_{\mathcal{A}c_i}(a_k)} \tag{1}$$

where A denotes a group of items with an attribute value c and i refers to the ith group. $\mathbb{I}_A(x)$ is 1 if $x \in A$ and 0 otherwise. We further have y_{a_k} as a user's feedback on a recommended item a_k. Weighted proportional fairness is subsequently defined as a generalized Nash solution across multiple groups. The weighted proportionally fair allocation is hence the solution to the following optimization problem:

$$\max x_t \sum_{i=1}^{l} w_i \log\left(x_t^i\right), \quad \text{s.t.} \sum_{i=1}^{l} x_t^i = 1, x_t^i \geq 0, i = 1, \ldots, l \tag{2}$$

where w_i is a parameter used to weigh the importance of each group. Solving this optimization problem by applying Lagrangian multiplier methods yields

$$x_*^i = \frac{w_i}{\sum_{i'=1}^{l} w_{i'}} \tag{3}$$

2.3 Coefficient of Variation

In multi-agent systems where we want to ensure a fair distribution of resources or rewards among agents, the coefficient of variation (cv) is often used to measure fairness among agents (Jiang and Lu, 2019; Elmalaki, 2021). The metric is defined as follows:

$$cv = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \frac{(u_i - \bar{u})^2}{\bar{u}^2}} \tag{4}$$

where u_i is the utility of agent i, n is the number of agents, and $\bar{u}$ is the average utility of all agents. This measure captures the sum of individual differences from the mean. The lower the cv value, the fairer the system.

Jiang and Lu (2019) note that in multi-agent sequential decision-making, it's often hard to optimize the coefficient of variation of individual agents because its value depends on the joint policy of all agents. In these cases, each agent i's contribution to cv is usually approximated by $(u^i - \bar{u})^2/\bar{u}^2$. The coefficient of variation is then often built into the reward function of each agent so that each

agent is punished for getting too much or too little resources or utility (Jiang and Lu, 2019; Elmalaki, 2021).

2.4 Q-Value Based Definitions

A natural idea of fairness is based on Q-values. Recall that a Markov Decision Process (MDP) consists of a set of states S, an initial state s_0, a set of actions $\textsc{Actions}(s)$ of each state $s \in S$, a transition model $P(s'|s,a)$, and a reward function $R(s,a,s)$. In this setting, the Q-function $Q(s,a)$ is the expected utility of taking a given action at a given state. In this context, $Q^*(s,a)$ is the expected utility starting out having taken action a from state s and (thereafter) acting optimally. Jabbari et al. (2017) introduce the idea of exact fairness, and its two relaxations: approximate-choice fairness and approximate-action fairness.

Exact fairness requires that in any state s, the algorithm never chooses an action a with a higher probability than another action a' unless $Q^*(s,a) > Q^*(s,a')$, i.e., in cases where the long term (discounted) reward of choosing a is higher than that of choosing a'.

Approximate-choice fairness requires that the algorithm never chooses a worse action with substantially higher probability than better actions. An algorithm $\mathcal{L}$ satisfies approximate-choice fairness if for all inputs $\delta > 0$ and $\alpha > 0$, for all MDP tuples 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 at least $1-\delta$ over histories h_{t-1}.

Approximate-action fairness, on the other hand, requires that an algorithm never favors an action of substantially lower quality than that of a better action. An algorithm $\mathcal{L}$ satisfies approximate-action fairness if for all inputs $\delta > 0$ and $\alpha > 0$, for all MDP tuples 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 at least $1-\delta$ over histories h_{t-1}.

Both exact fairness and approximate-choice fairness require exponential time learning algorithms to approach optimality. Further relaxation to the probabilistic requirement results in a weaker definition of fairness, but ensures a polynomial-time learning algorithm.

2.5 α-Fair Utility

In computer networking, fairness is often considered as fairly allocating network resources (i.e. bandwidth) to different data flows. In this context, Chen et al. (2021) use an α-fair utility function to capture fairness notions. For $\alpha \geq 0$, the

α-fair utility is defined as

$$U(x) = \begin{cases} x^{1-\alpha}/(1-\alpha) & \text{for } \alpha \neq 1 \\ \log(x) & \text{for } \alpha = 1 \end{cases} \tag{5}$$

Note that this notion of fairness is different from the previous notions of fairness in that this measure is one that can be "toggled" or controlled to achieve different levels of fairness: Setting $\alpha = 0$, for example, leads to throughput maximization, $\alpha = 1$ to proportional fairness, and $\alpha \to \infty$ to max-min fairness (Chen et al., 2021). The designer of the model can set the desired fairness parameter α, whereas definitions like the coefficient of variation are measures of fairness that are embedded into the RL problem itself via the reward function. In these cases, the reward function can account for fairness, even though the programmer cannot exactly control how fair the outcomes are.

2.6 Observations on Fairness Definitions

In general, we notice that definitions of fairness in RL are highly inconsistent, some of them are even mutually exclusive, which has been previously discussed in the literature (Berk et al., 2017; Friedler et al., 2016). This may be due to several reasons.

First, the majority of the existing literature is focused on incorporating fairness in RL in one specific application domain. For example, the work of Elmalaki (2021) focuses on incorporating fairness in RL for IoT devices, while Chen et al. (2021) study fairness in RL for network utility optimization. This results in many fairness approaches being highly specialized to certain application domains that may not work well in general settings.

Second, many notions of fairness originated from a diverse set of fields outside of computer science. For example, the notion of α-fairness was originally studied in the context of networks and welfare economics definitions have their origin in social welfare and game theory. This causes a discrepancy in definitions adopted in RL, since there has not been an agreement on a general fairness definition in the field, which is slowing down research of fair cross-domain RL approaches.

3 Application Domains

Fairness in RL is most relevant in two areas that have experienced notable adoption: decision support systems (DSS) and autonomous systems (AS). A DSS helps make better decisions in complex settings, while an AS is taken autonomous decisions in a pre-defined environment. Note that in both categories

and the specific application domains we discuss below, each system is either deployed to be used by multiple stakeholders, or that its decisions will impact many users. Ensuring fairness, or at least the attempt to ensure fairness under generally accepted definitions, is hence crucial for the users' acceptance of these systems.

For example, Liu et al. (2020) developed a model to balance fairness and accuracy in **interactive recommendation systems** (IRS). These systems have been used to recommend items of interest (for example news, movies, or articles (Steck et al., 2015)) to individual users. The recommendations are updated in an online setting based on user feedback, which is often expressed as taking a desired action or not, such as buying an article after seeing a recommendation for it. The latter is also known as conversion rate. Only using the conversion rate can lead to an imbalance across different demographic groups, which can lead to minorities being ignored in recommendations.

Claure et al. (2019), on the other hand, looked at fair RL in the context of **resource distribution in human-robot teams**. They specifically focused on a fair candidate selection, because in unconstrained RL, the agent would learn the individual worker performances first and then assign as many tasks as possible to the highest performing candidate. In real life, this can lead to a two-folded issue: Firstly, when a human worker is allocated significantly more resources to process than his team members, he might burn out quickly, which is a factor that is not being taken into account by the agent by default. Secondly, favoring one employee might cause a negative impact on all team members' motivation. Perceived inequalities have been shown to motivate people to act against their personal self-interest to eliminate the inequality (Camerer, 2003), sometimes including actions to retaliate (Skarlicki and Folger, 1997). It further undermines their trust in the system due to a perceived unfairness. In addition, not including human preferences in the allocation process has been shown to decrease willingness to work among human workers. All these aspects can negatively impact performance.

Chen et al. (2021), on the other hand, look at fair RL solutions in the context of **wireless network scheduling** and **Quality-of-Experience (QoE) optimization** in video streaming. The former is concerned with how to schedule the transmission and reception of sequences of network packages across a network of users. In this case, fairness considerations are necessary to prevent the discrimination of certain parties (and hence the potential restriction on the access to the wireless network of those users). Fairness with regards to QoE, on the other hand, means that all users are having a similar video streaming experience, based on metrics to capture the (perceived) video quality by users.

Siddique et al. (2020) and Zimmer et al. (2021) consider applications to traffic light control and data center control. In the **traffic light control problem**,

without fairness constraints, the problem seeks to minimize the total wait time across all lanes. By incorporating fairness considerations, the agent is then tasked to learn how to optimize expected wait time per road. In the **data center control setting**, the goal is to minimize the queue length of each switch in the network, which needs to be fair, too.

In the area of **Internet-of-Things** (IoT), fairness considerations could enable IoT devices to better mitigate the challenges brought by intra-human, inter-human, and multi-human variability. Take smart thermometers, for example. The same user's temperature preference might change over time (intra-human variability), each user has different temperature preferences (inter-human variability), and multiple people in the same room can have a wide range of temperature preferences (multi-human variability). Elmalaki (2021) show that using their developed "FaiR-IoT" fairness-aware human-in-the-loop framework will improve the user experience and improve fairness (measured by the coefficient of variation, discussed in Section 2.3).

4 Methodology

Current literature on fairness in RL can be categorized into research that focuses on single-agent or on multi-agent setups. We will discuss these approaches in the following chapters.

4.1 Single-agent RL

Several papers consider MDP-based RL settings and examine how to incorporate fairness into these problem formulations (Weng, 2019; Siddique et al., 2020; Zimmer et al., 2021). In model-based RL, the problem is formulated as an MDP, where the agent uses a transition model of the environment to decide how to act in each state. The model can be initially either known (like chess) or unknown (in which case the agent will learn the empirical model). The traditional sequential MDP model can be extended to multi-objective sequential decision making, where the scalar reward is replaced by a vector whose components represent objectives, also called MOMDPs. Each objective can be interpreted as a criterion that represents the welfare or utility of an agent or user, which is a natural fit to fairness considerations. Formally, this means that in MOMDPs, we have reward functions $R(s, a, s') \in \mathbb{R}^D$, where D is the number of objectives.

In the single-agent multi-objective setting pursued by **Siddique et al. (2020)**, the authors make use of welfare functions that satisfy the three-part fairness definition (impartiality, equity, and efficiency; described in Section 2.1). They specifically formulate the fair RL problem by **integrating the generalized**

***Gini* social welfare function (GGF) into MOMDPs**. The GGF is defined as

$$\mathrm{GGF}_w(v) = \sum_{i=1}^{D} w_i v_i^{\uparrow} \tag{6}$$

where $v \in \mathbb{R}^D$ is the utility vector, $w \in \mathbb{R}^D$ is the fixed positive weight vectors whose components are strictly decreasing, and $v^{\uparrow}$ is the vector with the components of vector v sorted in ascending order (Zimmer et al., 2021).

The problem of solving for an optimal policy that ensures fairness then becomes

$$\arg\max_{\pi} \mathrm{GGF}_w(J(\pi)) \tag{7}$$

where $J(\pi)$ can be defined as the discounted reward. Note that Equation 7 is a non-linear convex optimization problem (Siddique et al., 2020).

The algorithms to solve this problem could involve modifications to the Deep Q Network (DQN) or use policy gradient methods. To use DQN, the usual DQN is modified to take on values in $\mathbb{R}^{|\mathcal{A}| \times D}$ (where $\mathcal{A}$ is the set of actions in the MDP) instead of $\mathbb{R}^{|\mathcal{A}|}$. Meanwhile, policy gradient methods can be advantageous because they directly optimize for the desired objective function and can also learn stochastic policies instead of just deterministic policies.

Claure et al. (2019), on the other hand, consider a stochastic multi-armed bandits (MAB) framework together with an unconstrained Upper Confidence Bound (UCB) algorithm in their work. The aim in MAB for an agent is to maximize cumulative rewards by pulling bandits' arms based on previous information the agent obtained. In the UCB algorithm, the agent does this by using information on the number of times an arm was pulled and the average empirical rewards the agent received to estimate the expected reward of each arm. However, the authors find that an unconstrained UCB algorithm does not ensure fairness, since under-performing arms will not be used by the agent after a certain number of time steps. To prevent this, Claure et al. (2019) propose two **adjusted UCB algorithms**: a **strict-rate-constrained** and a **stochastic-rate-constrained** version.

The former guarantees[2] that there is a *minimum pull rate* for each lever because arms are pre-scheduled to be pulled in fixed time slots. In all other time slots, the agent will follow the standard UCB algorithm, i.e. it will choose the arm that's best according to the benchmark strategy. The stochastic-rate-constrained UCB algorithm, on the other hand, only guarantees[3] that the *expected pulling rate* is at least the minimum pull rate at any time. In comparison with the strict-rate-constrained UCB algorithm, randomness is introduced and it is ensured

[2]See Claure et al. (2019) for formalized theoretical guarantees.
[3]See Claure et al. (2019) for formalized theoretical guarantees.

that the probability that an arm will be pulled at time t is equivalent to the minimum pull rate. In this case, with a probability of $1 - Kv$, where K is the arm set and v is the minimum pull rate, the benchmark UCB policy is followed.

Liu et al. (2020) propose the RL-based ***FairRec* framework** to ensure a balanced long-term trade off between accuracy and fairness in a single-agent setup in IRS. The authors formulate the problem of IRS as an MDP. They use the weighted proportional fairness notion described in Section 2.2. Liu et al. (2020) specifically build an actor-critic architecture (see Figure 1) where the actor network is responsible for dynamic recommendations depending on the fairness status and the user preferences. The critic network, on the other hand, encourages or discourages a recommended item based on its estimate of the value of the actor's output.

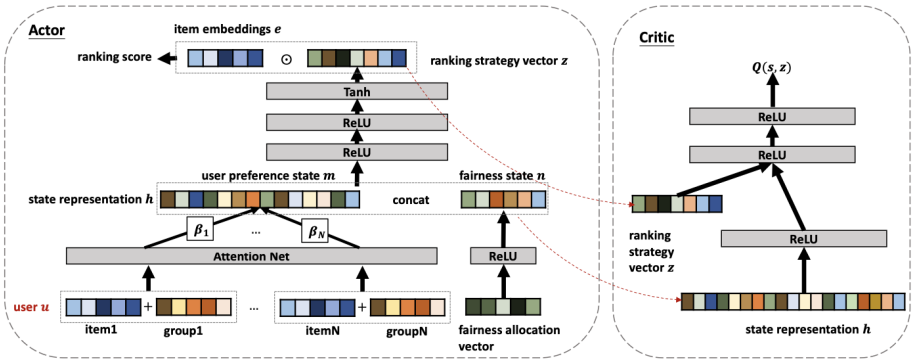

Figure 1: FairRec Architecture (Source: Liu et al. (2020))

To accomplish a focus on both accuracy and fairness, Liu et al. (2020) design a two-fold reward based on a personalized fairness-aware state representation. They consider whether a user performed a desired activity on a recommended item before the fairness gain of this activity is being evaluated.

4.2 Multi-agent RL

Chen et al. (2021) take the idea by Liu et al. (2020) to the multi-agent setting and aim to **optimize general fairness utility functions in actor-critic RL**. They specifically developed a method to adjust the standard RL rewards by a multiplicative weight that takes into account the history of rewards as well as the shape of the fairness utility. The multiplicative adjustment is defined using a uniformly-continuous function $\Phi(h_{\pi,t})$ that is dependent on a statistic $h_{\pi,t}$ capturing past rewards. The adjusted rewards are calculated by

$$\hat{r}_{k,t} = r_{k,t} \cdot \phi\left(h_{\pi,t}\right) \tag{8}$$

The authors further employ the algorithm in Figure 2, where r, s, a are the rewards, states, and actions, respectively, and A is the advantage function of a policy defined as the difference between the relative state- and action-value functions.

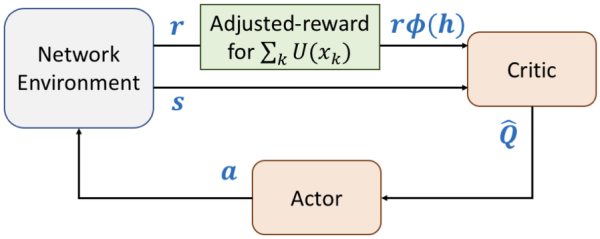

Figure 2: Rewards-adjusted actor-critic architecture (Source: Chen et al. (2021))

Chen et al. (2021) decided on this approach to address the issue that a non-linear α-fair utility ($\forall \alpha > 0$) doesn't satisfy the Markovian property which would be necessary to formulate α-fair network utility optimization as an MDP and guarantee convergence under the Policy Gradient Theorem. Instead, this approach guarantees that, given a proper choice of Φ and h, it converges to "at least a stationary point of the α-fair utility optimization" (Chen et al., 2021). Another advantage to this approach is that it builds on actor-critic architectures. This means that the optimization is converging quicker because of variance reduction as well as because the proposed algorithm doesn't rely on the Monte Carlo method, especially when optimizing in large state/action spaces.

In the multi-agent setup researched by **Zimmer et al. (2021)**, the authors formulated the problem of (deep) cooperative multi-agent RL as

$$\max_{\theta} \phi(J(\theta)) \tag{9}$$

where θ is the joint policy of all the agents and $J_k(\theta) = E_\theta[\sum_k \gamma^t r_{k,t}]$ is the expected sum of discounted rewards of user k.

The algorithmic solution for this optimization problem includes a policy gradient approach implemented in an actor-critic architecture. Zimmer et al. (2021) propose **Self-Oriented Team-Oriented (SOTO) networks** updated by dedicated policy gradient. On a high level, the network includes a self-oriented and a team-oriented policy. The former optimizes for an individual policy, whereas the team-oriented policy optimizes for the SWF $\phi(J(\theta))$, i.e. the two sub-networks

focus on efficiency and equity respectively. The details of this architecture can be found in Figure 3. An advantage of the approach of Zimmer et al. (2021) is that it is not domain-specific and allows for the adoption of a variety of fairness notions, since it only demands a (sub-)differentiable welfare function (Zimmer et al., 2021).

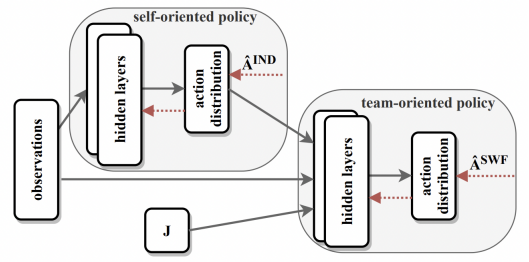

Figure 3: Rewards-adjusted actor-critic architecture (Source: Zimmer et al. (2021))

Jiang and Lu (2019), on the other hand, introduce the **Fair-Efficient Network (FEN)**, an RL-based model that makes use of a "fair-efficient reward" (Jiang and Lu, 2019) to address multi-agent RL settings. This reward is learned by each agent to optimize its own policy. Additionally, an average consensus among agents as part of the fair-efficient rewards allows for the coordination between agents' policies. Specifically, the authors consider a setting with n agents and limited, commonly-accessible resources. Each agent's fair-efficient reward at time t is

$$\hat{r}_t^i = \frac{\bar{u}_t / c}{\epsilon + \left| u_t^i / \bar{u}_t - 1 \right|} \tag{10}$$

where the utility of agent i at time step t is the average reward over time it received:

$$u_t^i = \frac{1}{t} \sum_{j=0}^{t} r_j^i \tag{11}$$

Furthermore, c is the maximum environmental reward an agent obtains at a time step and $\bar{u}_t$ is the mean utility across all agents. Hence, $\bar{u}_t / c$ can be seen as a proxy for system efficiency, since it's representing the system-wide resource allocation. On the other hand, $\left| u_t^i / \bar{u}_t - 1 \right|$ represents an agent's utility deviation from the average, which is taken as a proxy for fairness based on the idea of the coefficient of variation (see Section 2), and ϵ represents a small constant that prevents a division by zero. This reward is used by each agent to learn its

own policy $F_i = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \hat{r}_t^i\right]$ with a discount γ.

Potential multi-objective conflicts are circumvented by using a hierarchical RL-model with a controller that maximizes the fair-efficient reward by changing between multiple sub-policies. These sub-policies are designed with respect to different goals: to maximize an environmental reward and to explore different fair behaviours. The aim of this approach is to enable agents to learn efficiency and fairness simultaneously.

5 Trade-Offs

Imposing fairness requirements to RL algorithms often result in worse running time compared to problems without fairness constraints. Jabbari et al. (2017) show the trade-offs between efficiency and fairness. In particular, both fair and approximate-choice fair requirements impose an exponential time step $\mathcal{T} = \Omega(k^n)$, and approximate-action fairness requires number of time $\mathcal{T} = \Omega(k^{1/(1-\gamma)})$, where n is the size of the state space and γ is the discount factor. Note that without fairness constraints, standard RL algorithms learn an ϵ-optimal policy in a number of steps polynomial in n, $1/\epsilon$, and all parameters of the MDP. This shows that imposing fairness requirements comes a cost with regards to run time efficiency.

Imposing fairness requirements further (potentially) decreases performance. Liu et al. (2020) discussed this trade-off in their paper. Specifically, the authors use a two-fold reward that combines accuracy and fairness metrics in the context of IRS. The authors find that their *FairRec* framework does increase fairness in recommendation systems while maintaining a good recommendation quality but that, compared to non-fair baselines, there are still performance losses. The trade-off between fairness and accuracy is a general issue in designing ethical AI since any fair solution will introduce additional constraints and/or objectives that will, compared to an unconstrained, non-fair problem formulation, result in a decreased performance quality (Berk et al., 2017). This trade-off is an ongoing challenge in RL systems, too.

6 Future Research Directions

The study of fairness in machine learning has been and will continue to be an interdisciplinary field that involves economics, mathematics, computer science, and even fields like philosophy and political science. Based on the papers that we discussed in this literature review, we see the following research directions

as key in making progress in the field.

Unified Fairness Definition

As discussed in Section 2, the field of fair RL is currently shaped by a variety of fairness definitions. In most works that we've studied, these were derived from domain-specific settings. To advance fair RL algorithms, we suggest to investigate the merits and drawbacks of each of these definitions in order to find a unified definition that can be applied across domains. To do so, one could, for example, expand the approach by Zimmer et al. (2021), which accepts different (sub-)differentiable notions of fairness, and test the effects of different fairness definitions on the performance of subsequent models.

Cross-domain Fair RL Approach

While developing domain-specific fair RL algorithms ensures a high degree of applicability to the respective domain, we recommend investigating more general models that can be applied across domains. The approaches taken by Zimmer et al. (2021), Liu et al. (2020), and Chen et al. (2021) are promising but should be tested for effectiveness in a more diverse set of domains.

Fairness in Sequential Decision Making

The differences between ensuring fairness at a given point in time vs. ensuring fairness in long-term decision making settings need to be studied in more detail to find general RL algorithms that are fair in both short- and longterm setups. Specifically, fairness should not only be guaranteed at the final time step T but also in upstream time steps.

Additional Application Domains

While we advocate for the focus on general cross-domain fair RL algorithms, we further encourage research in application-specific settings to find optimal, situation-specific models where general models might not perform as well. Potential domains to investigate include routing, traffic light control systems, and cloud computing (Jiang and Lu, 2019).

7 Conclusion

Given the increased popularity and adoption of RL in society, ensuring – or even guaranteeing – fairness in these algorithms is an important aspect. In this literature review, we discussed nine different research pieces that focus on the development of fair RL approaches. We compared different definitions of fairness, showcased the methodologies pursued in the different papers, and proposed future research directions. With our literature review, we hope to build a discussion base and encourage further research in the field.

References

Berk, R., Heidari, H., Jabbari, S., Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. (2017). A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*.

Camerer, C. F. (2003). Behavioral game theory: Plausible formal models that predict accurately. *Behavioral and Brain Sciences*, 26(2):157–158.

Chen, J., Wang, Y., and Lan, T. (2021). Bringing fairness to actor-critic reinforcement learning for network utility optimization. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE.

Claure, H., Chen, Y., Modi, J., Jung, M., and Nikolaidis, S. (2019). Reinforcement learning with fairness constraints for resource distribution in human-robot teams. *arXiv preprint arXiv:1907.00313*.

Elmalaki, S. (2021). Fair-iot: Fairness-aware human-in-the-loop reinforcement learning for harnessing human variability in personalized iot. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, pages 119–132.

Friedler, S. A., Scheidegger, C., and Venkatasubramanian, S. (2016). On the (im) possibility of fairness. *arXiv preprint arXiv:1609.07236*.

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

Jiang, J. and Lu, Z. (2019). Learning fairness in multi-agent systems. *Advances in Neural Information Processing Systems*, 32.

Liu, W., Liu, F., Tang, R., Liao, B., Chen, G., and Heng, P. A. (2020). Balancing between accuracy and fairness for interactive recommendation with reinforcement learning. *Advances in Knowledge Discovery and Data Mining*, 12084:155.

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

Skarlicki, D. P. and Folger, R. (1997). Retaliation in the workplace: The roles of distributive, procedural, and interactional justice. *Journal of applied Psychology*, 82(3):434.

Steck, H., van Zwol, R., and Johnson, C. (2015). Interactive recommender systems: Tutorial. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 359–360.

Weng, P. (2019). Fairness in reinforcement learning. *arXiv preprint arXiv:1907.10323*.

Zimmer, M., Glanois, C., Siddique, U., and Weng, P. (2021). Learning fair policies in decentralized cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 12967–12978. PMLR.