Fair Deep Reinforcement Learning with Preferential Treatment

Guanbao Yu[1] , **Umer Siddique**[2] and **Paul Weng**[1]
[1]Shanghai Jiao Tong University, Shanghai, China
[2]University of Texas, San Antonio, USA
{gbyu66, paul.weng}@sjtu.edu.cn, umersiddique297@gmail.com

Abstract

Learning fair policies in reinforcement learning (RL) is important when the RL agent may impact many users. We investigate a variant of this problem where equity is still desired, but some users may be entitled to a preferential treatment. In this paper, we formalize this more sophisticated fair optimization problem in deep RL using *generalized fair social welfare functions* (SWF), provide a theoretical discussion to justify our approach, explain how deep RL algorithms can be adapted to tackle it, and empirically validate our propositions on several domains. Our contributions are both theoretical and algorithmic, notably: (1) We obtain a general bound on the suboptimality gap in terms of SWF-optimality using average reward of a policy SWF-optimal for the discounted reward, which notably justifies using standard deep RL algorithms, even for the average reward; (2) Our algorithmic innovations include a state-augmented DQN-based method for learning either deterministic or stochastic policies, which also applies to the usual fair optimization setting without any preferential treatment.

1 Introduction

We consider autonomous systems based on deep reinforcement learning (RL). When they operate in real applications (e.g., traffic lights, software-defined networking, data centers), they may interact and impact many users. Hence, for these systems to be accepted by end-users when deployed, fairness needs to be taken into account in their design.

Fairness is rooted in the *equal treatment of equals* principle, which informally speaking means that individuals with similar characteristics should be treated in a similar way. Previous work [Siddique *et al.*, 2020; Chen *et al.*, 2021; Zimmer *et al.*, 2021] in learning fair policies in RL focuses on such notion with the additional assumption that all individuals are equal, which may not be suitable for all applications. For instance, it is customary for service providers (in e.g., software-defined networking, data centers) to provide different levels of QoS (quality of service) to different user tiers. In such cases, although the principle of "equal treatment of equals" is still a desired objective, higher-paying users should arguably be entitled to higher priority or better services.

In our work, we relax the assumption of equal individuals, i.e., different users may have different rights. We investigate this more sophisticated fair optimization problem in deep RL, where efficient policies should be learned such that while some users may receive preferential treatment, users with similar rights are still equitably treated.

Contributions We formalize this novel problem in deep RL as a fair optimization problem using *generalized fair social welfare functions* (SWF) (Section 4). We discuss its difficulties and provide some theoretical results to justify our algorithms (Section 4). We notably extend a performance bound, which now holds for a general class of fair SWFs. Based on this discussion, we propose several adaptations of deep RL algorithms to solve our problem (Section 5). In particular, we design a novel state-augmented DQN-based method for learning fair stochastic policies. Finally, we experimentally validate our propositions (Section 6).

2 Related Work

Fairness has recently become an important and active research direction [Dwork *et al.*, 2012; Zafar *et al.*, 2017; Sharifi-Malvajerdi *et al.*, 2019; Singh and Joachims, 2019; Chierichetti *et al.*, 2017; Busa-Fekete *et al.*, 2017; Agarwal *et al.*, 2018; Nabi *et al.*, 2019; Zhang and Liu, 2021]. While the majority of this literature in machine learning focuses on the *equal treatment of equals* principle, other aspects of fairness have been considered in AI, e.g., proportionality [Sun *et al.*, 2021; Bei *et al.*, 2022] or envy-freeness [Chevaleyre *et al.*, 2006] and its multiple variants (e.g., [Beynier *et al.*, 2019; Chakraborty *et al.*, 2021]). In contrast, our work is based on studies in distributive justice [Rawls, 1971; Brams and Taylor, 1996a; Moulin, 2004]. We aim at optimizing a social welfare function that encodes fairness. This principled approach has also been recently advocated in several recent papers [Heidari *et al.*, 2018; Speicher *et al.*, 2018; Weng, 2019; Cousins, 2021; Do and Usunier, 2022].

In mathematical optimization, such an approach is called *fair optimization* [Ogryczak *et al.*, 2014]. Many continuous and combinatorial optimization problems in various application domains [Amaldi *et al.*, 2013; Shi *et al.*, 2014; Neidhardt *et al.*, 2008; Nguyen and Weng, 2017; Ogryczak *et al.*, 2013]

have been extended to optimize for fairness. In this direction, the closest work [Ogryczak *et al.*, 2013] regards fair optimization in Markov decision processes. However, the methods proposed in this direction typically assume that the model is known and therefore, they do not require learning.

Fairness also starts to be considered in RL, e.g., fairness constraint to reduce discrimination [Wen *et al.*, 2021], fairness with respect to state visitation [Jabbari *et al.*, 2017; Ghalme *et al.*, 2022], the usual case of fairness with respect to agents [Jiang and Lu, 2019], or the more general case of fairness with respect to users [Siddique *et al.*, 2020; Chen *et al.*, 2021; Zimmer *et al.*, 2021; Mandal and Gan, 2022]. This last direction can be understood as an extension of fair optimization to (deep) RL. Our work follows this principled approach, but investigates a more general setting. While previous work assumes all users to be equal, we relax this assumption.

State augmentation (used in our DQN variants) has been exploited in various previous work, e.g., in MDPs [Liu and Koenig, 2005] or more recently in safe RL [Sootla *et al.*, 2022], risk-sensitive RL [Chow and Ghavamzadeh, 2014a], RL with delays [Nath *et al.*, 2021], and partially observable path planning [Nardi and Stachniss, 2019]. However, to the best of our knowledge, this technique has not been applied in fair optimization. Moreover, our technique to learn stochastic policies in DQN is also novel.

3 Background

We introduce our notations and recall the necessary background in sequential decision making and fairness modeling.

Notations Matrices are denoted in uppercase and vectors in lowercase. Both are written in bold. For any integer $D > 0$, the $D-1$ simplex is denoted by $\Delta_D = \{\boldsymbol{w} \in [0,1]^D \mid \sum_i \boldsymbol{w}_i = 1\}$. We denote $\mathbb{S}_D$ the symmetric group of degree D (i.e., set of permutations over $\{1,\dots,D\}$). For any permutation $\sigma \in \mathbb{S}_D$ and vector $\boldsymbol{v} \in \mathbb{R}^D$, vector $\boldsymbol{v}_\sigma$ denotes $(\boldsymbol{v}_{\sigma(1)}, \dots, \boldsymbol{v}_{\sigma(D)})$. For any vector $\boldsymbol{v} \in \mathbb{R}^D$, $\boldsymbol{v}^\uparrow$ corresponds to the vector with the components of vector $\boldsymbol{v}$ sorted in an increasing order (i.e., $\boldsymbol{v}_1^\uparrow \leq \ldots \leq \boldsymbol{v}_D^\uparrow$).

3.1 Sequential Decision-Making

Markov Decision Process (MDP) We consider sequential decision-making problems that can be modeled as an MDP [Puterman, 1994], which is is defined by the following elements: a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, $\boldsymbol{P}$ is a transition model, $\boldsymbol{r}$ is a reward function, $\boldsymbol{d}_0$ is a probability distribution over initial states, and $\gamma \in [0,1)$ is a discount factor. The usual goal of this model is to learn a policy π that maximizes some performance measure, e.g., the *expected discounted reward* criterion or the *expected average reward* criterion.

A *(stationary and Markov) policy* π selects which action to take in any state s: it can be deterministic ($\pi(s) = a \in \mathcal{A}$) or stochastic ($\pi(a \mid s) = Pr(a \mid s)$). With the discounted reward, the *(state) value function* of a policy π is defined by:

$$\boldsymbol{v}_\pi(s) = \mathbb{E}_{\boldsymbol{P},\pi}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \boldsymbol{r}_t \mid S_0 = s\right]. \tag{1}$$

Similarly, its *action-value function* $Q_\pi(s,a)$ is given by $\boldsymbol{Q}_\pi(s,a) = \mathbb{E}_{P,\pi}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \boldsymbol{r}_t \mid S_0 = s, A_0 = a\right]$. The problem for the discounted reward can be formally defined as follows: $\operatorname{argmax}_\pi \sum_{s\in\mathcal{S}} \boldsymbol{d}_0(s)\boldsymbol{v}_\pi(s)$. The action-value function of a solution to this problem is called *optimal Q-function*.

With the average reward criterion, the (state) value function of a policy π is usually referred to as *gain* denoting by $\boldsymbol{g}_\pi$:

$$\boldsymbol{g}_\pi(s) = \lim_{h\to\infty} \frac{1}{h} \mathbb{E}_{\boldsymbol{P},\pi}\left[\sum_{t=1}^{h} \boldsymbol{r}_t \mid S_0 = s\right]. \tag{2}$$

The problem here is given by $\operatorname{argmax}_\pi \boldsymbol{\mu}_\pi$, where $\boldsymbol{\mu}_\pi = \sum_{s\in\mathcal{S}} \boldsymbol{d}_0(s)\boldsymbol{g}_\pi(s)$.

We assume that the MDP is *weakly communicating*[1]. Recall that for such MDPs, the optimal gain is constant (state-independent). Note that without such assumption, which is weaker than ergodicity, learning is hopeless. In the theoretical discussion, we also assume for simplicity that $\mathcal{S}$ and $\mathcal{A}$ are finite. In that case, all the functions in this model can be seen as vectors or matrices, which explains our bold notations.

Multiobjective MDP (MOMDP) Since in our setting, an RL agent's actions can impact several users, we consider MOMDP [Roijers *et al.*, 2013], an extension of MDP, in which the goal is to optimize multiple objectives (i.e., utilities of users) instead of a single one. Thus, the rewards in MOMDPs are vectors where each component can be interpreted as a scalar reward allocated to one user in our setting. The reward function of a MOMDP is denoted $\boldsymbol{R}(s,a) \in \mathbb{R}^D$ where D is the number of objectives (i.e., users).

The value functions in MDPs can be naturally extended to MOMDPs, e.g., the (state) value function becomes:

$$\boldsymbol{V}_\pi(s) = \mathbb{E}_{\boldsymbol{P},\pi}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \boldsymbol{R}_t \mid S_0 = s\right], \tag{3}$$

where $\boldsymbol{R}_t \in \mathbb{R}^D$ is the vector reward obtained at time step t and all operations (addition, product) are component-wise. Vectors can be compared according to *Pareto dominance*[2]. Being a partial order, finding all Pareto-optimal solutions in MOMDPs is infeasible in general [Perny and Weng, 2010].

Deep RL Since we are interested in domains where the state/action spaces may be large or even continuous, we consider deep RL algorithms where value functions or policies are approximated by neural networks.

Deep Q-Network (DQN) [Mnih *et al.*, 2015] is an example of deep RL algorithm where the optimal Q-function is approximated by a neural network with parameter $\boldsymbol{\theta}$. This Q-network takes a state s as input and outputs an estimated $\hat{Q}_{\boldsymbol{\theta}}(s,a)$ for all actions. It is trained to minimize the following L_2 loss for transitions (s,a,r,s') sampled from a replay

[1] An MDP is *weakly communicating* if its states $\mathcal{S}$ can be partitioned into two classes $\mathcal{T}, \mathcal{C}$: set $\mathcal{T}$ in which all states are transient under every stationary policy, and $\mathcal{C}$ in which any two states can be reached from each other under some stationary policy.

[2] $\forall \boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^D$, $\boldsymbol{v}$ *weakly Pareto-dominates* $\boldsymbol{v}' \Leftrightarrow \forall i, \boldsymbol{v}_i \geq \boldsymbol{v}'_i$. Besides, $\boldsymbol{v}$ *Pareto-dominates* $\boldsymbol{v}' \Leftrightarrow \forall i, \boldsymbol{v}_i \geq \boldsymbol{v}'_i$ and $\exists j, \boldsymbol{v}_j > \boldsymbol{v}'_j$. A non Pareto-dominated solution is called *Pareto-optimal*.

buffer storing experiences generated from online interactions with the environment:

$$(r + \max_{a' \in \mathcal{A}} \gamma \hat{Q}_{\boldsymbol{\theta}'}(s', a') - \hat{Q}_{\boldsymbol{\theta}}(s, a))^2,$$

where $\boldsymbol{\theta}'$ parametrized a target network to enable stabler training. The term $r + \gamma \hat{Q}_{\boldsymbol{\theta}'}(s', a^*)$ is called *target Q-value*.

3.2 Fairness

Since the agent in an MOMDP can be seen as allocating rewards to users, it is natural to resort to a notion of fairness discussed in distributive justice [Moulin, 2004], which studies fair distribution of wealth. We recall this notion next and its extension to preferential treatment.

Fairness with Equal Users The notion of fairness that is adopted in fair optimization enforces three natural principles: *impartiality*, *equity*, and *efficiency*. Impartiality corresponds to the *"equal treatment of equals"* principle. Under the assumption that all users are equal, which is a common assumption in past work, this principle implies that reordering a reward distribution leads to an equivalent one. Equity is based on the *Pigou-Dalton principle* [Moulin, 2004], which states that a *Pigou-Dalton transfer* (i.e., small transfer of reward from a better-off user to a worse-off user) results in a fairer solution. This principle enforces that more balanced reward distributions are preferred. Efficiency requires a fair solution to be Pareto-optimal. It is natural because selecting a Pareto-dominated solution would be irrational.

These three principles provide some guidance about how to select among reward allocations: they induce a binary relation between vectors. For instance, assume that there are only two users ($D = 2$) and that the possible solutions are $(0, 5)$, $(5, 2)$, and $(3, 3)$. These principles imply that $(5, 2)$ is preferred to $(0, 5)$. Indeed, by impartiality, $(5, 2)$ is equivalent to $(2, 5)$. By efficiency, $(2, 5)$ is preferred to $(2, 3)$. By equity, $(2, 3) = (0, 5) + (+2, -2)$ is preferred to $(0, 5)$. However, using these principles alone, we cannot decide whether $(3, 3)$ or $(5, 2)$ should be preferred. Thus, although this relation refines Pareto dominance, it is still only a partial order.

Since using a partial order is not practical in autonomous decision-making, we rely on the concept of *social welfare function (SWF)* to enforce a total order. An SWF $\phi : \mathbb{R}^D \to \mathbb{R}$ evaluates how good a solution is for a group of users by aggregating their utilities. Interestingly, the three previous principles translate into three properties for an SWF that encodes this notion of fairness. Impartiality implies that ϕ is symmetric (i.e., independent of the order of its arguments). Equity means that ϕ is Schur-concave (i.e., monotonic with respect to Pigou-Dalton transfers). This implies that ϕ cannot be linear, which formally shows that the utilitarian approach (i.e., $\sum_i \boldsymbol{v}_i$) is not suitable for fairness. Efficiency enforces that ϕ is monotonic with respect to Pareto dominance.

In the literature, two main families of SWFs satisfying those three conditions have been considered. The first family is the *generalized Gini SWF* (GGF) [Weymark, 1981]:

$$GGF_{\boldsymbol{w}}(\boldsymbol{v}) = \sum_{i=1}^{D} \boldsymbol{w}_i \boldsymbol{v}_i^{\uparrow}\,, \tag{4}$$

where $\boldsymbol{v} \in \mathbb{R}^D$ and $\boldsymbol{w} \in \Delta_D$ is a fixed positive weight vector whose components are strictly decreasing (i.e., $\boldsymbol{w}_1 > \ldots > \boldsymbol{w}_D > 0$). The second family has the following form:

$$\phi_u(\boldsymbol{v}) = \sum_{i=1}^{D} u(\boldsymbol{v}_i)\,, \tag{5}$$

where $u : \mathbb{R} \to \mathbb{R}$ is a strictly concave function. Both families can encode various well-known specific types of fairness. For instance, $GGF_{\boldsymbol{w}}$ can represent lexicographic maxmin fairness [Rawls, 1971] with the components of $\boldsymbol{w}$ decreasing sufficiently fast and ϕ_u includes α-fairness ($\alpha > 0$) [Mo and Walrand, 2000] with $u_\alpha(x) = \frac{x^{1-\alpha}}{1-\alpha}$ if $\alpha \neq 1$ and $u_\alpha(x) = \log(x)$ otherwise. Note that both families define functions that are concave and sub-differentiable.

Fairness with Preferential Treatment Assuming that all users are equal is inadequate in some domains. For instance, in situations where an autonomous system manages the operations of a service provider, as discussed in the introduction, different tiers of users need to be taken into account. Another example is in the medical domain, where priority may be given to patient with specific needs (e.g., elderly or children). In those cases, the three fairness principles should be enforced, while allowing preferential treatment.

One typical approach to achieve this more general notion of fairness is via *user duplication* [Brams and Taylor, 1996b], i.e., if a user is more important, s/he should be counted more times (via importance weight) than other users. Since this weight can naturally be normalized, formally, if each user i receives some fractional entitlement $\boldsymbol{p}_i$ (i.e., importance weights), when two users are equally important, they would receive equal weights. In contrast, if a user is entitled to a preferential treatment, s/he would consequently receive a larger share of the total importance weight. This technique allows to naturally extend the previous families of SWFs.

Assume given some importance weights $\boldsymbol{p} \in \Delta_D$. The first family, which we call *generalized GGF* ($\mathrm{G^3F}$) [Ogryczak, 2009] is defined as follows: for any $\boldsymbol{v} \in \mathbb{R}^D$,

$$\mathrm{G^3F}_{\boldsymbol{p},\boldsymbol{w}}(\boldsymbol{v}) = \sum_i \boldsymbol{\omega}_i \boldsymbol{v}_i^{\uparrow}, \tag{6}$$

where $\boldsymbol{w}$ is defined as in (4) and the weight $\boldsymbol{\omega}_i$ is defined as:

$$\boldsymbol{\omega}_i = w^* \left(\sum_{k=1}^{i} \boldsymbol{p}_{\sigma(k)} \right) - w^* \left(\sum_{k=1}^{i-1} \boldsymbol{p}_{\sigma(k)} \right), \tag{7}$$

with w^* being a monotone increasing function that linearly interpolates the points $(i/D, \sum_{k=1}^{i} \boldsymbol{w}_k)$ together with the point $(0, 0)$, and σ the permutation sorting the components of vector $\boldsymbol{v}$ in increasing order, i.e., $\boldsymbol{v}_\sigma(i) = \boldsymbol{v}_i^{\uparrow}$ for all i. This formula amounts to averaging each portion (in terms of cumulated $\boldsymbol{p}$) of the i/D-th smallest values of $\boldsymbol{v}$ and apply the standard GGF to these D averages.

The second generalized family can be obtained more simply as follows: for a given fixed strictly concave $u : \mathbb{R} \to \mathbb{R}$,

$$\phi_{\boldsymbol{p},u}(\boldsymbol{v}) = \sum_{i=1}^{D} \boldsymbol{p}_i u(\boldsymbol{v}_i)\,. \tag{8}$$

We call a fair SWF with preferential treatment a *generalized fair SWF* and denote it ψ. For instance, ψ can be any instance of the two previous families. A specific SWF enforces how trade-offs are made between efficiency and equity via the choice of $\boldsymbol{w}$ or u. The choice of $\boldsymbol{p}$ depends on what entitlement to give to some users. Overall, the specific choice of a suitable SWF for a task is domain and problem dependent.

4 Problem Statement and Discussions

We formulate the novel problem of fair optimization in deep RL with preferential treatment, then recall its difficulties and present some theoretical discussion justifying our approach.

This fair optimization problem can be formally stated by optimizing *generalized fair SWFs* ψ in MOMDPs. It corresponds to determine a policy that generates a fair distribution of rewards subject to importance weights $\boldsymbol{p}$. Since we focus on deep RL, we directly write it with parametrized policy $\pi_{\boldsymbol{\theta}}$:

$$\underset{\pi_{\boldsymbol{\theta}}}{\operatorname{argmax}}\, \psi\left(\boldsymbol{J}(\pi_{\boldsymbol{\theta}})\right), \tag{9}$$

where $\boldsymbol{J}(\pi_{\boldsymbol{\theta}})$ corresponds to the vectorial version of the standard RL objective, e.g., the discounted or average reward. Note that this formulation is general, since it accepts any generalized fair SWFs (e.g., $\mathrm{G}^3\mathrm{F}_{\boldsymbol{p},\boldsymbol{w}}$ or generalized α-fairness mentioned in Section 3.2). A solution to this problem is called *ψ-γ-optimal* policy if the discounted reward is used, *ψ-average-optimal* policy if the average reward is used, or simply *ψ-optimal* policy to include both cases.

Standard multi-objective approaches aim at finding the set of Pareto optimal solutions (or an approximation), which may be infeasible in general. In contrast, our formulation directly aims for a Pareto-optimal ψ-fair policy. Moreover, the usual approach of optimizing a weighted sum of the objectives is insufficient, because such aggregation functions do not favor equitable solutions (i.e., violation of the Pigou-Dalton principle). In addition, note that directly applying ψ on immediate rewards (instead of their expectation as we do) provides no guarantee in terms of fairness, because this naive approach does not allow compensation over time and expectation to reach more equitable reward distribution.

Difficulties As an extension of the problem investigated by Siddique *et al.* [2020], Problem (9) also faces the same difficulties, notably *state-dependent optimality* and *optimality of stochastic policies*. We succinctly recall those two points below.

For the first point, in contrast to standard RL, a policy that is ψ-γ-optimal in an initial state may not be ψ-γ-optimal in another initial state. Fortunately, ψ-average-optimality is state-independent, which may be another argument for using this criterion. For the second point, searching for fair solutions among deterministic policies becomes insufficient. However, a ψ-optimal policy exists among stationary Markov stochastic policies, because intuitively, stochastic choices allows fairer compensation.

4.1 Theoretical Discussion

We provide some useful new results that justify our approach.

Bounds Although the average reward criterion may be more suitable, notably for a service provider, the discounted reward criterion may still be desirable because optimizing the latter is usually considered easier than the former in RL. An important question is then to bound how far a ψ-γ-optimal policy π^*_γ is to a ψ-average-optimal policy π^*_1 in terms of ψ-average-optimality. The next theorem, stated informally for legibility sake, provides such a bound, which extends a previous result [Siddique *et al.*, 2020] only valid for GGF to any continuous concave ψ.

Theorem 1. *For any weakly-communicating MOMDP and any γ close to 1, there exist constants C and K such that:*

$$\psi(\boldsymbol{\mu}_{\pi^*_\gamma}) \geq \psi(\boldsymbol{\mu}_{\pi^*_1}) - (1-\gamma)CK$$

Proof. The proof technique is similar to [Siddique *et al.*, 2020]. The more general result is achieved by resorting to the Fenchel dual of ψ. We also corrected a sign issue in the previous result. □

Constant C depends on both the reward function and ψ, while constant K depends on the transition function. This result provides a performance guarantee for using the discounted reward criterion instead of the average reward, thus motivating the use of the former in the algorithm design, even if one aims to optimize the latter.

Concavity of $\mathrm{G}^3\mathrm{F}$ Although Ogryczak and Śliwiński [2010] have proved the concavity of $\mathrm{G}^3\mathrm{F}$, here we provide another straightforward proof as an alternative proof technique.

Lemma 1. *For any $\boldsymbol{p} \in \Delta_D$, for any $\boldsymbol{w} \in \Delta_D$ such that its components are decreasing, function $\mathrm{G}^3\mathrm{F}_{\boldsymbol{p},\boldsymbol{w}}$ is concave.*

Proof. We prove that $\mathrm{G}^3\mathrm{F}_{\boldsymbol{p},\boldsymbol{w}}$ is a Choquet integral with respect to a super-modular capacity. By [Lovász, 1983], such integrals are concave. □

While $\phi_{\boldsymbol{p},u}$ in (8) clearly defines a concave function, it was not completely obvious for $\mathrm{G}^3\mathrm{F}$. Our lemma confirms that both generalized families yield concave functions. The concavity of ψ suggests that our fair optimization (9) should enjoy some nice properties. For instance, with a linear approximation scheme, the overall problem would be a convex optimization problem. In deep RL, the overall problem is not convex anymore, but from the point of view of the last layer of a neural network (usually linear, e.g., in DQN), the optimization problem is still convex. In addition, concavity is required to justify the DQN variants, as discussed next.

5 Proposed Algorithms

We explain how deep RL methods for the discounted reward can be adapted to solve Problem (9). For space reasons, we focus here on the adaptation of DQN [Mnih *et al.*, 2015], because the extensions of actor-critic (AC) methods (i.e., A2C [Mnih *et al.*, 2016] and PPO [Schulman *et al.*, 2017]), called ψ-A2C and ψ-PPO respectively, are more straightforward. Notably, the policy gradient for these methods can be easily obtained via the chain rule. We explain how the critics can be adapted and trained next. Other AC methods could be extended in a similar way.

Following Siddique *et al.* [2020], the architectures of these critics, but also DQN, are modified such that their outputs are vectorial (e.g., for DQN, $\mathbb{R}^{|\mathcal{A}|\times D}$ instead of $\mathbb{R}^{|\mathcal{A}|}$). Although the Bellman principle of optimality does not hold anymore due to the state-dependent ψ-optimality, vector value functions can still be temporally decomposed. We discuss next our three extensions of DQN. Note that critics are trained like in DQN but without maximizing over future actions.

ψ-DQN ψ-DQN is the direct extension of GGF-DQN proposed by Siddique *et al.* [2020] to optimize GGF. Since ψ-DQN aims to optimize ψ, the target Q-value is changed to:

$$\boldsymbol{r} + \gamma \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(s', a^*),$$

where $a^* = \operatorname{argmax}_{a' \in \mathcal{A}} \psi\big(\boldsymbol{r} + \gamma \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(s', a')\big)$. The best next action is chosen such that the immediate reward plus discounted future rewards (both vectorial) is fair. For execution in a state s, an action in $\operatorname{argmax}_{a \in \mathcal{A}} \psi\big(\hat{\boldsymbol{Q}}_{\boldsymbol{\theta}}(s, a)\big)$ is chosen.

The performance of ψ-DQN is limited for several reasons. First, it learns a stationary, Markov, and deterministic policy, which is known to be insufficient (see Section 4). Moreover, since ψ is assumed to be concave, by Jensen inequality, $\mathbb{E}_{s'}\big[\psi\big(\boldsymbol{r} + \gamma \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(s', a^*)\big)\big] \leq \psi\big(\mathbb{E}_{s'}\big[\boldsymbol{r} + \gamma\ \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(s', a^*)\big]\big)$. Therefore, ψ-DQN implicitly maximizes a lower bound of an approximation of the objective function in (9).

Since DQN is known to be more sample-efficient than AC methods, it is useful to design better algorithms than ψ-DQN for our problem. Next, we propose two such novel extensions of DQN that can achieve better performance.

ψ-CDQN One simple approach to improve the performance of ψ-DQN is to relax the assumption that the learned policy is Markov, which can be achieved via state augmentation. In ψ-CDQN, the agent observes the current state and the past cumulated vector reward. Intuitively, using this additional information, the agent can better balance the reward distribution over users.

Formally, an original state s_t is augmented as follows: $\bar{s}_t = \big(s_t, \frac{1}{\lambda}\boldsymbol{R}_{1:t}\big)$ where $\lambda = \sum_{\tau=1}^{t-1} \gamma^{\tau-1}$ acts as a scaling factor, $\boldsymbol{R}_{1:t} = \sum_{\tau=1}^{t-1} \gamma^{\tau-1} \boldsymbol{r}_\tau$ denotes the discounted total reward received so far , which is reset to zero at the beginning of an episode. The target Q-value is changed as follows:

$$\boldsymbol{r}_t + \gamma \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(\bar{s}_{t+1}, a^*),$$

where $a^* = \operatorname{argmax}_{a' \in \mathcal{A}} \psi\big(\hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(\bar{s}_{t+1}, a')\big)$. Here, the immediate reward $\boldsymbol{r}_t$ is removed from the optimal action computation since this signal is already included in the augmented state as part of the discounted total reward. For execution in a state s, an action in $\operatorname{argmax}_{a \in \mathcal{A}} \psi\big(\hat{\boldsymbol{Q}}_{\boldsymbol{\theta}}(\bar{s}, a)\big)$ is chosen.

ψ-CSDQN Since stochastic policies may dominate deterministic ones (Section 4), the performance of ψ-DQN (and possibly ψ-CDQN) can be improved by considering stochastic policies. We describe how to achieve this next.

With stochastic policies, the target Q-value is changed to:

$$\hat{\boldsymbol{Q}}_{\boldsymbol{\theta}}(s, a) = \boldsymbol{r} + \gamma \hat{\boldsymbol{Q}}^*_{\boldsymbol{\theta}'}(s', \cdot),$$

where $\hat{\boldsymbol{Q}}^*_{\boldsymbol{\theta}'}(s', \cdot) = \sum_{a' \in \mathcal{A}} \pi^*(a'|s') \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(s', a')$ denotes an estimated Q-value achieved at a next state by a policy π^*, which is defined as:

$$\pi^*(\cdot|s') = \operatorname*{argmax}_{\pi} \psi(\boldsymbol{r} + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}'}(s', a')) \quad (10)$$

This reformulation assumes that in the next state, the best stochastic policy is applied (in contrast to a deterministic greedy policy in DQN or ψ-DQN). For execution in a state s, an action is sampled from $\pi^*(\cdot|s)$ in $\operatorname{argmax}_{\pi} \psi(\sum_{a' \in \mathcal{A}} \pi(a'|s') \hat{\boldsymbol{Q}}_{\boldsymbol{\theta}}(s', a'))$.

Problem (10) is an easy optimization problem. Since ψ is concave, it is a convex optimization with $|\mathcal{A}|$ variables corresponding to $\pi(\cdot \mid s') \in \Delta_{|\mathcal{A}|}$. As a general approach, it can be solved by projected gradient ascent, which consists in repeatedly updating the current $\pi(\cdot \mid s')$ and projecting the updated variables to the simplex $\Delta_{|\mathcal{A}|}$. Recall that projection on a simplex can be done efficiently [Chen and Ye, 2011]. Interestingly, for specific ψ, more specialized algorithms can be used. For instance, when choosing $\mathrm{G}^3\mathrm{F}_{\boldsymbol{p},\boldsymbol{w}}$ as ψ, one can obtain the optimal stochastic policy by solving a linear program (see Appendix C.3).

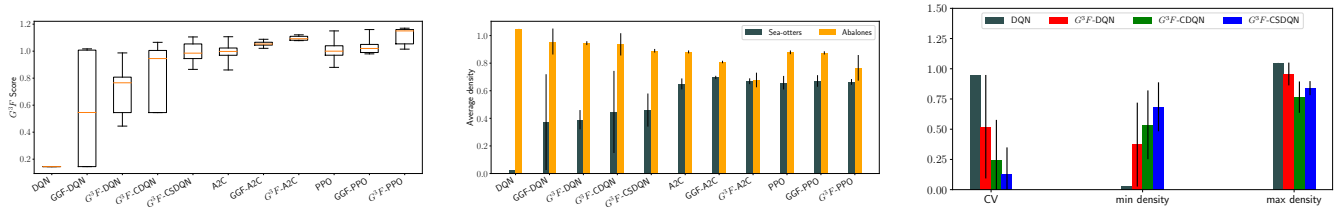
Finally, by augmenting states like in ψ-CDQN, we can formulate the last novel algorithm called ψ-*CSDQN*, which can learn fair stochastic policies with augmented states. Although one may expect a better performance from this new algorithm, ψ-CDQN may still be useful in domains (e.g., robotics) where deterministic policies are favored. Note that all these DQN variants follow the ϵ-greedy policy during training and with probability $1 - \epsilon$, the best action in state s is chosen in a way corresponding to the specific variant as explained above.

6 Experimental Results

Our proposed generic algorithms can be instantiated with different SWFs ψ. We have performed experiments with the two families of SWFs discussed in Section 5: $\mathrm{G}^3\mathrm{F}$ and the generalized α-fairness. Here we mainly discuss our results for $\mathrm{G}^3\mathrm{F}$ for two reasons. First, it is an extension of GGF, a well-studied SWF in economics [Weymark, 1981]. Second, it is a more general SWF than α-fairness, which only applies when rewards are positive. Moreover, we emphasize here our evaluation of the DQN variants, which constitute our main algorithmic contribution. Additional experimental results with α-fairness or the AC methods (A2C and PPO) can be found in the full version of this paper. All the results are averaged over 10 runs with different seeds.

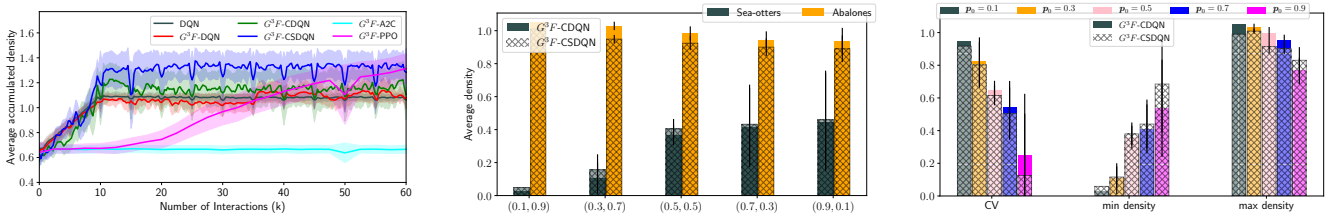
Our algorithms (with relevant baselines) are evaluated in the same three domains as in Siddique *et al.* [2020] to help with comparability. In roughly increasing problem sizes (i.e., number of users, state/action spaces), they are: (i) Species conservation (SC), (ii) Traffic light control (TL), and (iii) Data center control (DC). We briefly describe them next (see appendix of [Siddique *et al.*, 2020] for more details).

The first domain (SC) [Chadès *et al.*, 2012] simulates an ecological conservation problem in which two species—an endangered species (sea otters) and its prey (northern abalone)—interact with one another, potentially leading to the extinction of some species. An observed state includes the population levels of the two species. The size of the action set is 5. Fairness is expressed over the two species ($D = 2$) and



(a) G^3F score during the testing phase. (b) Population densities during testing. (c) CV, min, and max densities during testing.

Figure 1: Performances of DQN, A2C, PPO, and their GGF or G^3F counterparts in SC. Weight $\boldsymbol{p} = (0.9, 0.1)$ for the G^3F algorithms.



(a) Accumulated densities during training. (b) Population densities during testing. (c) CV, min and max densities during testing.

Figure 2: Additional experimental results in the SC domain: (Left) G^3F with $\boldsymbol{p} = (0.9, 0.1)$, (Center, Right) G^3F with varying $\boldsymbol{p}$.

can be understood as both species remaining alive and having a balanced population density. Because population densities may not be comparable directly, using equal weights for $\boldsymbol{p}$ may not be suitable. In that case, a fair SWF with importance weights may be beneficial.

The second domain (TL) is a traffic light control problem [Lopez *et al.*, 2018] in which an agent controls the traffic lights at a single intersection to optimize traffic flow. A state includes the waiting times and densities of cars waiting in each lane. An action amounts to selecting the next traffic-light phase among four phases: NSL, NSSR, EWL, and EWSR, where NSL (north-south left) represents the phase with the green light assigned to the left lanes of the roads approaching from the north and south, NSSR (north-south straight and right) represents the phase with the green light assigned to the straight and right lanes of roads approaching from the north and south, and so on. Here, fairness is defined over each direction at the intersection (i.e., D=4). Moreover, we assume that some lanes will be given preferential treatment (e.g., due to morning rush, traffic flows are unbalanced) and that the waiting times for cars in these lanes will be optimized with higher priorities, while other lanes with equal preferences will be treated fairly.

The third and last domain (DC) is a data center traffic control problem [Ruffy *et al.*, 2019] with a continuous action space and $D = 16$ users. Since the action space is continuous, DQN-like algorithms cannot be run. We refer the reader to the full version of this paper for more details about this domain and the experimental results with the AC methods (A2C and PPO). They are in line with those presented here.

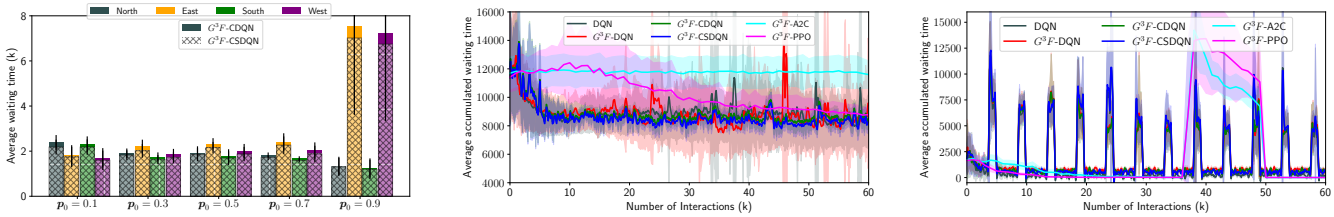
On these domains, we have run an extensive set of experiments to answer a series of questions.

Do our algorithms learn fairer (w.r.t. G^3F) solutions than their respective counterparts? This is a sanity check to verify that our methods perform as intended. All the algorithms are run in SC with weight $\boldsymbol{p}$ set to $(0.9, 0.1)$ where the first component corresponds to sea otters. The G^3F scores are obtained by applying G^3F on the empirical average vector returns of trajectories sampled with the learned policies. As expected, Figure 1a show all the G^3F algorithms reach higher scores than their GGF and original counterparts, indicating that fairness with priority set by $\boldsymbol{p}$ was better achieved.

Figure 1b shows the corresponding empirical average vector returns before aggregating with G^3F. Recall that optimizing GGF (i.e., G^3F with $\boldsymbol{p} = (0.5, 0.5)$) would lead to a much larger density of abalones [Siddique *et al.*, 2020]. However, optimizing G^3F with a higher priority given to sea otters achieves more balanced individual densities than their corresponding counterparts. A non-uniform $\boldsymbol{p}$ may help correct advantages conferred to some users by the environment.

How much control over solutions does $\boldsymbol{p}$ provide? To answer this question, we evaluate the G^3F algorithms with varying importance weights $\boldsymbol{p}$ in the SC and TL domains. Figure 2c shows the testing performance of CDQN and CSDQN in SC in terms of *Coefficient of Variation* (CV), minimum and maximum density. Recall that CV is a simple inequality measure defined as the ratio of the standard deviation to the mean. Lower CV values imply more balanced solutions. For experiments in SC, we increase the preference weight $\boldsymbol{p}_0$ of the first objective from 0.1 to 0.9 (i.e., $\boldsymbol{p}_1$ decreases from 0.9 to 0.1, correspondingly). As a result, the density of the sea otter increases, resulting in lower CV, higher minimum density, and lower maximum density.

In the TL domain, we vary weight $\boldsymbol{p}_0$ (assigned to North), while the remaining weight is assigned uniformly over the remaining three components (directions) of $\boldsymbol{p}$. Figure 3a shows that waiting times of cars coming from lanes with higher weights are shorter than those coming from lanes with lower



(a) Individual waiting times during testing.

(b) Waiting times in stationary env.

(c) Waiting times in non-stationary env.

Figure 3: Experimental results in the TL domain: (Left) Effects of using different weights for $\boldsymbol{p}$, (Center, Right) Waiting times during training.

weights. Interestingly, the waiting times of cars coming from north and south are close, although they are assigned different weights. This is because the agent's action can affect two lanes at the same time in this case.

We performed another sanity check to verify the impartiality principle when using G^3F with some non-uniform $\boldsymbol{p}$. Since the results in the TL domain suggests that the inherent problem structure may influence the set of achievable reward distributions, which seems natural, for this experiment, we chose the DC domain, where in contrast to TL, there is no dependence between the objectives. For setting $\boldsymbol{p}$, we assign a larger weight to one objective and distribute the remaining weight uniformly over the other ones. As expected, it can be observed experimentally that the *equal treatment of equals* principle holds.

The above results show that by appropriately adjusting weights $\boldsymbol{p}$, we can achieve desired control over multiple users, up to constraints imposed by the problem structure.

Does considering past discounted reward or learning a stochastic policy help in DQN-based algorithms? For this question, we compare all our DQN variants in the SC and TL domains to investigate the benefits of considering past discounted reward or stochastic policies. Figures 1a, 1c and 2a show the performances of those algorithms in the SC domain. Note that while Figure 2a plots the training curves within 60k interactions, the AC methods in all the domains are indeed trained with 600k interactions for convergence before testing. Moving from DQN, G^3F-DQN, G^3F-CDQN, to G^3F-CSDQN nearly always yields an increase in terms of average density (more efficient), a decrease in terms of CV (more equitable), an increase in terms of min density (more equitable), and an increase in terms of G^3F (fairer). This latter point experimentally confirms the theoretical discussion about the optimality of stochastic policies in Section 4.

Focusing on GGF (i.e., uniform $\boldsymbol{p}$), our proposed G^3F-CDQN (i.e., GGF-CDQN) can find better solutions than GGF-DQN as shown in additional experimental results. This shows that our novel DQN-based algorithms outperform the one proposed by Siddique *et al.* [2020].

Interestingly, G^3F-CDQN and G^3F-CSDQN outperform DQN in terms of average density, which is exactly what is optimized by DQN. This is explained by the fact that this domain is actually partially observable. In such situations, state augmentation and stochastic policies are known to be beneficial. Similar conclusions can be drawn for the TL domain as well.

When is it preferable to resort to our DQN-based variants? Figures 2a, 3b and 3c show the training performances. Note that the x-axis corresponds to the number of interactions, which may not correspond to the timesteps in an environment (e.g., A2C simultaneously use several environments to generate training data).

In the SC domain, Figure 2a shows that DQN-based methods enjoy much better sample efficiency than the AC methods. This is confirmed in the TL domain. Our experiments in that domain are usually run in a stationary environment (i.e., probabilities of cars entering in the intersection are fixed). We also performed some experiments in a non-stationary environment case (i.e., simulating different time periods during the day: morning/after rush hours or low traffic). In the stationary environment case, Figure 3b shows again that the DQN-based algorithms learn faster the AC methods in terms of number of interactions. The results in a non-stationary environment shown in Figure 3c further strengthen the case for the DQN-based algorithms: they can adapt faster to environmental changes than the AC methods.

In conclusion, if sample efficiency is important, one should choose CSDQN if learning stochastic policies is acceptable; otherwise CDQN should be preferred if deterministic policies are required (e.g., in robotics).

7 Conclusion

We investigated the fair optimization problem with preferential treatment in (deep) RL. For this novel problem, we presented both theoretical and algorithmic contributions. For the theory, we extended an existing bound to justify the use of the discounted reward instead of the average reward in the algorithm design. For the algorithms, we presented several extensions of deep RL algorithms and notably proposed a novel state-augmented DQN-based method, which can be adapted to learn either deterministic (CDQN) or stochastic policies (CSDQN). Extensive experimental results on several domains were provided for validation.

The novel algorithmic idea of CSQDN could be adapted to other RL problems with sophisticated objective functions (e.g., safe RL [Liu *et al.*, 2020] or risk-sensitive RL [Chow and Ghavamzadeh, 2014b]) or with constraints [Achiam *et al.*, 2017]. In contrast to existing work based on policy gradient, our technique could tackle those problems with a DQN-based method. In addition, our approach could also be extended to the fair multi-agent setting [Zimmer *et al.*, 2021]. We leave these directions to future work.

Ethical Statement

We hope that our work will lead to more research effort in fair sequential decision-making and will help in the future design autonomous systems with built-in ethical principles, such as fairness.

Acknowledgments

This work is supported in part by the program of the Shanghai NSF (No. 19ZR1426700).

References

J. Achiam, D. Held, and A. Tamar et al. Constrained policy optimization. In *ICML*, 2017.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *ICML*, 2018.

Edoardo Amaldi, Stefano Coniglio, Luca G. Gianoli, and Can Umut Ileri. On single-path network routing subject to max-min fair flow allocation. *Electronic Notes in Discrete Mathematics*, 41:543–550, June 2013.

Xiaohui Bei, Shengxin Liu, Chung Keung Poon, and Hongao Wang. Candidate selections with proportional fairness constraints. In *AAMAS*, 2022.

Aurélie Beynier, Yann Chevaleyre, Laurent Gourvès, Ararat Harutyunyan, Julien Lesca, Nicolas Maudet, and Anaëlle Wilczynski. Local envy-freeness in house allocation problems. *AAMAS*, 2019.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, March 1996.

Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, and Shie Mannor. Multi-objective bandits: Optimizing the generalized Gini index. In *ICML*, pages 625–634, 2017.

Iadine Chadès, Janelle MR Curtis, and Tara G Martin. Setting realistic recovery targets for two interacting endangered species, sea otter and northern abalone. *Conservation Biology*, 26(6):1016–1025, 2012.

Mithun Chakraborty, Ayumi Igarashi, Warut Suksompong, and Yair Zick. Weighted envy-freeness in indivisible item allocation. *ACM TEAC*, 9(3):1–39, 2021.

Yunmei Chen and Xiaojing Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011.

Jingdi Chen, Yimeng Wang, and Tian Lan. Bringing fairness to actor-critic reinforcement learning for network utility optimization. In *INFOCOM*, 2021.

Yann Chevaleyre, Paul E Dunne, Michel Lemaître, Nicolas Maudet, Julian Padget, Steve Phelps, and Juan A Rodríguez-aguilar. Issues in Multiagent Resource Allocation. *Computer*, 30:3–31, 2006.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *NeurIPS*, 2017.

Gustave Choquet. Theory of capacities. In *Annales de l'institut Fourier*, volume 5, pages 131–295, 1954.

Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for CVaR optimization in MDPs. 2014.

Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for CVaR optimization in MDPs. In *NIPS*, 2014.

Cyrus Cousins. An axiomatic theory of provably-fair welfare-centric machine learning. In *NeurIPS*, 2021.

Virginie Do and Nicolas Usunier. Optimizing generalized gini indices for fairness in rankings. In *SIGIR*, 2022.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conf.*, 2012.

Ganesh Ghalme, Vineet Nair, Vishakha Patil, and Yilun Zhou. Long-term resource allocation fairness in average markov decision process (amdp) environment. In *AAMAS*, 2022.

Hoda Heidari, Claudio Ferrari, Krishna Gummadi, and Andreas Krause. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *NeurIPS*, 2018.

Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in reinforcement learning. In *ICML*, 2017.

Jiechuan Jiang and Zongqing Lu. Learning fairness in multi-agent systems. 2019.

Bernard F. Lamond and Martin L. Puterman. Generalized Inverses in Discrete Time Markov Decision Processes. *SIAM J. on Matrix Analysis and Appl.*, 10(1):118–134, jan 1989.

Y. Liu and S. Koenig. Risk-sensitive planning with one-switch utility functions: Value iteration. In *AAAI*, pages 993–999. AAAI, 2005.

Y. Liu, J. Ding, and X. Liu. IPO: Interior-point policy optimization under constraints. *AAAI*, 2020.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using SUMO. In *IEEE ITSC*, 2018.

László Lovász. Submodular functions and convexity. In *Mathematical programming the state of the art*, pages 235–257. Springer, 1983.

Debmalya Mandal and Jiarui Gan. Socially fair reinforcement learning. *arXiv preprint arXiv:2208.12584*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2004.

Razieh Nabi, Daniel Malinsky, and Ilya Shpitser. Learning optimal fair policies. In *ICML*, 2019.

Lorenzo Nardi and Cyrill Stachniss. Uncertainty-aware path planning for navigation on road networks using augmented MDPs. In *ICRA*, 2019.

Somjit Nath, Mayank Baranwal, and Harshad Khadilkar. Revisiting state augmentation methods for reinforcement learning with stochastic delays. In *CIKM*, 2021.

Arnie Neidhardt, Hanan Luss, and K. R. Krishnan. Data fusion and optimal placement of fixed and mobile sensors. In *IEEE Sensors Applications Symposium*, 2008.

Viet Hung Nguyen and Paul Weng. An efficient primal-dual algorithm for fair combinatorial optimization problems. In *COCOA*, 2017.

Włodzimierz Ogryczak and Tomasz Śliwiński. On optimization of the importance weighted owa aggregation of multiple criteria. In *International Conference on Computational Science and Its Applications*, pages 804–817, 2007.

Włodzimierz Ogryczak and Tomasz Śliwiński. On solving optimization problems with ordered average criteria and constraints. *Fuzzy Optimization: Recent Advances and Applications*, pages 209–230, 2010.

Wlodzimierz Ogryczak, Patrice Perny, and Paul Weng. A compromise programming approach to multiobjective markov decision processes. *Intl. J. of Information Tech. & Decision Making*, 12(05):1021–1053, 2013.

Wlodzimierz Ogryczak, Hanan Luss, Michał Pióro, Dritan Nace, and Artur Tomaszewski. Fair optimization and networks: A survey. *J. of Applied Mathematics*, 2014, 2014.

W. Ogryczak. On principles of fair resource allocation for importance weighted agents. In *International Workshop on Social Informatics SOCINFO*, 2009.

Patrice Perny and Paul Weng. On finding compromise solutions in multiobjective Markov decision processes. In *ECAI (short paper)*, 2010.

M.L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley, 1994.

John Rawls. *The Theory of Justice*. Havard university press, 1971.

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

Fabian Ruffy, Michael Przystupa, and Ivan Beschastnikh. Iroko: A framework to prototype reinforcement learning for data center traffic control. In *Workshop on ML for Systems at NeurIPS*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. Average Individual Fairness: Algorithms, Generalization and Experiments. In *NeurIPS*, 2019.

Huaizhou Shi, R. Venkatesha Prasad, Ertan Onur, and I. G. M. M. Niemegeers. Fairness in wireless networks:issues, measures and challenges. *IEEE Communications Surveys & Tutorials*, 16(1):5–24, 2014.

Umer Siddique, Paul Weng, and Matthieu Zimmer. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *ICML*, 2020.

Ashudeep Singh and Thorsten Joachims. Policy Learning for Fairness in Ranking. In *NeurIPS*, 2019.

Aivar Sootla, Alexander I Cowen-Rivers, Taher Jafferjee, Ziyan Wang, David H Mguni, Jun Wang, and Haitham Ammar. Sauté rl: Almost surely safe reinforcement learning using state augmentation. In *ICML*, 2022.

Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. A unified approach to quantifying algorithmic unfairness: Measuring individual & group unfairness via inequality indices. In *KDD*, 2018.

Ankang Sun, Bo Chen, and Xuan Vinh Doan. Connections between fairness criteria and efficiency for allocating indivisible chores. *arXiv preprint arXiv:2101.07435*, 2021.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 2000.

Min Wen, Osbert Bastani, and Ufuk Topcu. Algorithms for fairness in sequential decision making. In *ICML*, 2021.

Paul Weng. Fairness in reinforcement learning. In *AI for Social Good Workshop at IJCAI*, 2019.

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

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, Krishna P. Gummadi, and Adrian Weller. From Parity to Preference-based Notions of Fairness in Classification. In *NeurIPS*, 2017.

Xueru Zhang and Mingyan Liu. Fairness in learning-based sequential decision algorithms: A survey. In *Handbook of Reinforcement Learning and Control*, pages 525–555. Springer, 2021.

Matthieu Zimmer, Claire Glanois, Umer Siddique, and Paul Weng. Learning fair policies in decentralized cooperative multi-agent reinforcement learning. In *ICML*, 2021.