Learning Fair Policies in Decentralized Cooperative Multi-Agent Reinforcement Learning

Matthieu Zimmer [* 1] **Claire Glanois** [* 1] **Umer Siddique** [1] **Paul Weng** [1 2]

Abstract

We consider the problem of learning fair policies in (deep) cooperative multi-agent reinforcement learning (MARL). We formalize it in a principled way as the problem of optimizing a welfare function that explicitly encodes two important aspects of fairness: efficiency and equity. We provide a theoretical analysis of the convergence of policy gradient for this problem. As a solution method, we propose a novel neural network architecture, which is composed of two sub-networks specifically designed for taking into account these two aspects of fairness. In experiments, we demonstrate the importance of the two sub-networks for fair optimization. Our overall approach is general as it can accommodate any (sub)differentiable welfare function. Therefore, it is compatible with various notions of fairness that have been proposed in the literature (e.g., lexicographic maximin, generalized Gini social welfare function, proportional fairness). Our method is generic and can be implemented in various MARL settings: centralized training and decentralized execution, or fully decentralized. Finally, we experimentally validate our approach in various domains and show that it can perform much better than previous methods, both in terms of efficiency and equity.

1. Introduction

Adaptive distributed control systems start to be considered in real applications, e.g., traffic light control (van der Pol & Oliehoek, 2016), multi-robot patrolling (Portugal & Rocha, 2013), or internet congestion (Jay et al., 2019). Although those systems generally may impact many end-users, the current main focus is on their performance with respect to the total (or average) of some per-user efficiency measure (e.g., waiting times of cars in traffic light control, safety of different sites in patrolling, or throughput of users in internet congestion). However, this approach is clearly unsatisfactory due to the users' conflicting interests. Thus, for such systems, fairness becomes a key factor to consider in their designs for their successful deployments and operations.

Fairness is a multifaceted concept (Section 2), which can refer to or include different aspects, e.g., impartiality, equity, Pareto-efficiency, envy-freeness, or proportionality among others. Given the importance of this notion, it has been investigated in various scientific disciplines, from philosophy to computer science, including economics and applied mathematics. In this work, fairness specifically refers to the combination of the first three aspects. Interestingly, this definition of fairness can be encoded in a *fair* social welfare function, which combines the users' utilities and can be used to evaluate and compare different solutions.

In this paper, we consider adaptive distributed control systems modeled as cooperative decentralized multi-agent reinforcement learning (MARL), and study the problem of learning fair distributed policies. This approach applies to situations where a system designer needs to implement a distributed system to solve a specific task (e.g., traffic regulation, patrolling, or congestion control) for many users in a fair way. Thanks to our definition of fairness, this problem can be expressed as a fair optimization problem, i.e., optimization of a fair social welfare function.

This formalization can then be tackled with standard multi-agent deep reinforcement learning techniques. Yet, as agents need to learn both efficiency and equity, two conflicting aspects of fairness, a naive approach is insufficient, as shown in our experiments. Thus, we propose a novel architecture specifically designed for fair optimization in multi-agent deep reinforcement learning (MADRL), which is shown to experimentally over-perform previous approaches.

Contributions We formulate a general and principled model for the problem of learning fair solutions in cooperative multi-agent reinforcement learning (Section 3). We propose a simple, scalable and efficient decentralized method to

[*] Equal contribution [1] UM-SJTU Joint Institute, Shanghai Jiao Tong University, China [2] Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Correspondence to: Paul Weng <paul.weng@sjtu.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

solve this problem (Section 4). We also provide a theoretical analysis of the convergence of policy gradient for this problem (Section 5). To validate our approach, we extensively compare it with previous approaches and evaluate it on a diverse set of domains (Section 6).

2. Related Work

The notion of fairness has been extensively studied in political philosophy (e.g., (Rawls, 1971)), political sciences (e.g., (Brams & Taylor, 1996)) and in economics (e.g., (Moulin, 2004)). This literature has led to the considerations of various aspects of fairness, e.g., equal treatment of equals, efficiency with respect to Pareto dominance, equal distribution (of goods, wealth, opportunities...), or envy-freeness, which have been exploited in more applied fields, such as operations research (OR), artificial intelligence (AI), or machine learning (ML). In this paper, we follow the approaches based on social welfare functions (Moulin, 2004). Various formulations have been considered, e.g., the utilitarian one that considers the users' total utility or the egalitarian approach that focuses on the lowest utility. In this paper, we investigate a family of fair social welfare functions that encodes impartiality, equity, and efficiency (see Section 3.2).

Such approaches, referred to as *fair optimization*, have been adopted before in OR and related fields (Ogryczak et al., 2014), and have many applications notably in networking (Amaldi et al., 2013; Shi et al., 2014). Various classic OR problems have been studied in the fair optimization setting, e.g., location (Neidhardt et al., 2008), allocation (Bertsimas et al., 2011), or Markov decision process (Ogryczak et al., 2013). As typical in OR, those works usually deal with a centralized and known environment setting. Our work can be seen as an extension of this literature to the decentralized and learning setting.

In AI, fairness has been considered in multi-agent systems with a large focus on resource allocation problems, notably with envy-freeness (Chevaleyre et al., 2006) and some works in non-cooperative games (de Jong et al., 2008; Hao & Leung, 2016). In contrast, we deal with more complex control problems, but in the **cooperative** setting. As such, our proposition is based on MARL instead of a game-theoretic formulation, which is more suitable for the non-cooperative setting. Besides, we formulate fairness with respect to users instead of agents, which is a more general framework.

Recently, fairness has started to become an important topic in ML. Indeed, as ML models are deployed in various applications (e.g., banking or law enforcement), the decisions made on their outputs may severely impact some users due to the presence of bias in data. Different ML tasks have been inspected in this regard, e.g., classification (Dwork et al., 2012; Zafar et al., 2017; Sharifi-Malvajerdi et al., 2019), ranking (Singh & Joachims, 2019), sequential-decision making (Busa-Fekete et al., 2017) or clustering (Chierichetti et al., 2017). Most of such work focuses on the impartiality aspect of fairness, expressed at the individual or group level, which leads to a constrained-based or penalty-based formulation. However, some recent work (Speicher et al., 2018; Heidari et al., 2018) advocates a more complete approach based on fair social welfare function that we also adopt in our work. Besides, such approach was recently investigated in single-agent deep reinforcement learning (RL) (Siddique et al., 2020).

Due to the recent successes of deep RL, research on MADRL has become very active (Hernandez-Leal et al., 2019). Different settings have been considered depending on whether training or execution is centralized or not, state observability is partial or not, and communication is allowed or not. Some recent work focuses on tackling problems related to decentralized training (Zhang et al., 2018), communication (Foerster et al., 2016; Sukhbaatar et al., 2016), coordination (van der Pol & Oliehoek, 2016), or agent modeling (Raileanu et al., 2018). In cooperative MADRL, the usual approach is based on a utilitarian formulation or a unique common reward signal.

However, fairness has been explicitly considered in multi-agent sequential decision-making in some few exceptions (Zhang & Shah, 2014; Jiang & Lu, 2019). Zhang & Shah (2014) consider a regularized maxmin egalitarian approach in order to find an equitable solution. Yet, this may be deficient as the solution without the worse-off agent may not be fair. Also, this work does not consider learning. In order to learn fair solutions, Jiang & Lu (2019) propose FEN, a decentralized method using two main ingredients. First, a gossip algorithm is used to estimate the average utility obtained by all agents. Second, the policy of each agent has a hierarchical architecture, where the high level decides to optimize its own utility or not, and the low level is composed of several sub-policies: the first one optimizes the individual reward gathered by the agent, while the others optimize their probability of being selected by the higher level. That work has several limitations. It implicitly assumes that agents have equal access to resources, which may not be true in practice (see Section 6). Besides, fairness is implicitly defined with the coefficient of variation (CV)[1], an inequality measure —measuring the dispersion of the utility— which does not guarantee efficiency.

3. Formalization

Notations For any natural integer n, $[n]$ denotes the set $\{1, \dots, n\}$. Vectors, which are column vectors, and matrices are denoted in bold and their components in normal

[1] The ratio of the standard deviation to the mean.

typeface with indices, e.g., $\boldsymbol{x} = (x_1, \dots, x_n)$. For any set X, $\Delta(X)$ denotes the set of probability measures over X.

3.1. Multi-Agent Reinforcement Learning

Recall that a decentralized partially observable Markov decision process can be defined with the following n-tuple $\big(\mathcal{S}, \mathcal{A} = (\mathcal{A}_i)_{i\in[N]}, (\mathcal{O}_i)_{i\in[N]}, P, (\Omega_i)_{i\in[N]}, r, \gamma\big)$ where N is the number of agents, $\mathcal{S}$ is the global state space, $\mathcal{A}_i$ is the action space of agent i, $\mathcal{O}_i$ is the observation space of agent i, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the joint transition function, $\Omega_i : \mathcal{S} \to \Delta(\mathcal{O}_i)$ is the observation function of agent i, r is a joint reward function, and $\gamma \in (0, 1)$ is a discount factor.

Since the operations of an agent may impact many different users, we extend the previous formulation by redefining the reward function to be vectorial: $\boldsymbol{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^D$ where D is the number of users. A user can represent an individual or a group of individuals. We denote $\boldsymbol{r} = (r_k)_{k\in[D]}$. As the system is distributed, the agents may not observe the whole reward vector. At a given time step, an agent i observes $\boldsymbol{r}_{I_i} = (r_k)_{k\in I_i}$ where $I_i \subseteq [D]$. Note that the partial observability of rewards does not imply reward independence between agents. The rewards depend on the state of the whole system and the actions of all agents. For ease of presentation of our solution method, we will assume that the set I_i is fixed for each agent i and the sets I_i's of all agents form a partition of $[D]$. Our approach can readily be extended to the more general case where the sets I_i and I_j of two agents may have a non-empty intersection. Note that our formulation is strictly more general than the usual approach where fairness is defined over agents. By setting $D = N$ and $I_i = \{i\}$, we can recover the usual formulation.

A joint policy can be written as follows $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{o}) = (\pi_1(a_1|o_1), \dots, \pi_N(a_N|o_N))$. The individual policy of the i^{th} agent is denoted $\pi_i : \mathcal{O}_i \to \Delta(\mathcal{A}_i)$ since an agent only perceives its local observation. Likewise, since each agent cannot access the whole reward vector nor the joint state, each agent learns an individual state value function $\hat{\boldsymbol{V}}_{I_i} : \mathcal{O}_i \to \mathbb{R}^{|I_i|}$ in order to approximate $\boldsymbol{V}_{I_i}(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \boldsymbol{r}_{I_i,t} \mid \boldsymbol{s}_0 = \boldsymbol{s}\right]$, which represents the utilities of users in I_i in state $\boldsymbol{s}$.

3.2. Fairness Formulation

The notion of fairness we focus on in this paper encompasses three important aspects (Adler, 2012): impartiality, equity, and efficiency. Impartiality corresponds to the "equal treatment of equals" principle, which is arguably one of the most important pillars of fairness. In this paper, we assume that all users are identical and should therefore be treated similarly. In terms of utility vectors, impartiality implies that permutations of a utility vector are equivalent solutions.

Equity is based on the *Pigou-Dalton principle* (Pigou, 1912; Dalton, 1920), which states that a reward transfer from a better-off user to a worse-off user yields a fairer solution. Formally, it is expressed as follows: for any utility vector $\boldsymbol{u} \in \mathbb{R}^D$, if $u_j - u_i > \varepsilon > 0$, then $\boldsymbol{u} + \varepsilon\boldsymbol{e}_i - \varepsilon\boldsymbol{e}_j$ is considered fairer than $\boldsymbol{u}$, where $\boldsymbol{e}_i \in \mathbb{R}^D$ (resp. $\boldsymbol{e}_j \in \mathbb{R}^D$) is the null vector except in component i (resp. j) where it is equal to 1. Such a transfer is called a *Pigou-Dalton transfer*. This principle formally expresses the notion of equal distribution of "wealth", which is the basis of the equity property that we want our fairness concept to satisfy. This principle is natural in our context where accumulated rewards vectors are interpreted as wealth distributions.

Efficiency states that between two feasible solutions, if one solution is (weakly or strictly) preferred by all users, then it should be preferred to the other one. This simply corresponds to Pareto dominance[2] in the space of users' utilities. Although efficiency is not always considered an integral part of fairness, one could argue that it would be unfair in the name of equity not to increase the rewards of all or some users while not decreasing the rewards of any other users, if that were possible. Without efficiency, giving no reward to all users would be as good as giving 100 to all users.

To make this notion of fairness operational, we adopt the approach based on social welfare functions. A *social welfare function* (SWF) is a function $\phi : \mathbb{R}^D \to \mathbb{R}$, which aggregates a utility vector and measures how good it is in terms of social good. Naturally, among all SWFs, we consider those that satisfy the notion of fairness we have just discussed.

Impartiality implies that an SWF ϕ should be symmetric, that is ϕ should be independent of the order of its arguments, i.e., $\phi(\boldsymbol{u}) = \phi(\boldsymbol{u}_\sigma)$ where σ is a permutation and $\boldsymbol{u}_\sigma$ is the vector obtained from vector $\boldsymbol{u}$ permuted by σ. Efficiency means that ϕ should be strictly monotonic with respect to Pareto dominance, i.e., $\boldsymbol{u} \succ \boldsymbol{u}' \Rightarrow \phi(\boldsymbol{u}) > \phi(\boldsymbol{u}')$. Finally, the Pigou-Dalton principle implies that ϕ should be strictly Schur-concave (i.e., strictly monotonic with respect to Pigou-Dalton transfers).

In this paper, an SWF will be called *fair SWF* if it satisfies the three previous properties. Many fair SWFs have been proposed in the literature. One may distinguish two main families. The first is the generalized Gini SWF (GGF), which is defined as follows:

$$G_{\boldsymbol{w}}(\boldsymbol{u}) = \sum_{k\in[D]} w_k u_k^{\uparrow} \tag{1}$$

where $\boldsymbol{w} \in [0, 1]^D$ is a fixed strictly decreasing weight vector (i.e., $w_1 > w_2 > \dots > w_D$) and $\boldsymbol{u}^{\uparrow}$ is the vector obtained from $\boldsymbol{u}$ by sorting its components in an increasing order. By choosing appropriately the weights $\boldsymbol{w}$ (and in

[2] For any $(\boldsymbol{u}, \boldsymbol{u}') \in \mathbb{R}^{D\times D}$, $\boldsymbol{u}$ Pareto-dominates $\boldsymbol{u}'$ (denoted $\boldsymbol{u} \succ \boldsymbol{u}'$) if $\forall i, u_i \geq u'_i$ and $\exists j, u_j > u'_j$.

some cases allowing them to be weakly decreasing), this family of SWF includes the maxmin egalitarian approach ($w_1 = 1, w_2 = \ldots = w_D = 0$), the regularized maxmin egalitarian approach ($w_1 = 1, w_2 = \ldots = w_D = \varepsilon$), the leximin egalitarian approach ($\forall i, w_i/w_{i+1} \to \infty$), or the utilitarian approach ($\forall i, w_i = 1$). However, requiring that weights $\boldsymbol{w}$ are strictly decreasing is important to ensure that the obtained solution is fair (i.e., Pareto optimal and equitable). Another family of SWF can be written as follows:

$$\phi(\boldsymbol{u}) = \sum_{k \in [D]} U(u_k)$$

where $U : \mathbb{R} \to \mathbb{R}$ is strictly increasing and strictly concave. Recall a function that is symmetric and strictly concave is strictly Schur-concave. This family is very general and includes proportional fairness (Pióro et al., 2002) when $U(x) = \log(x)$ and more generally α-fairness (Mo & Walrand, 2000) when $U_\alpha(x) = \frac{x^{1-\alpha}}{1-\alpha}$ if $\alpha \neq 1$ and $U_\alpha(x) = \log(x)$ otherwise, with parameter $\alpha > 0$ controlling the aversion to inequality. We denote the corresponding SWF ϕ_α. When $\alpha \to \infty$, it tends to the leximin egalitarian formulation. Even more broadly, this family includes SWFs derived from the generalized entropy index (Shorrocks, 1980).

The exact choice of an SWF depends on the specific problem one wants to solve. Intuitively, a fair SWF trades off between equity and efficiency: optimizing a given fair SWF amounts to selecting among Pareto-optimal solutions the one with the best trade-off. As we aim at designing a generic approach for fairness, we leave this choice unspecified.

3.3. Problem Statement

As usual, in order to tackle problems with large-sized or even continuous state/action spaces, we assume that the policy space is parameterized. Based on the notion of fair SWF, our problem can be simply formulated as follows:

$$\max_{\boldsymbol{\theta}} \phi(\boldsymbol{J}(\boldsymbol{\theta})) \tag{2}$$

where $\boldsymbol{\theta}$ is the parameters of the joint policy of all the agents and $J_k(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}[\sum_t \gamma^t r_{k,t}]$ is the expected sum of discounted rewards for user k. Interestingly, the two families of fair SWFs that we recalled correspond to concave functions, which implies that (2) is a convex optimization problem.

Since each user's utility only depends on one agent, our problem can be written as:

$$\max_{\boldsymbol{\theta}} \mathfrak{J}(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \phi(\boldsymbol{J}_{I_1}(\theta_1), \ldots, \boldsymbol{J}_{I_N}(\theta_N)) \tag{3}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ is the policies' parameters of $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ respectively and I_i corresponds to the set of indices of users whose utilities depend on agent i. In the next section, we propose an efficient MADRL method to solve this problem. Note that although $\boldsymbol{J}(\boldsymbol{\theta})$ is vectorial, this is a **single** objective optimization problem since $\phi : \mathbb{R}^D \to \mathbb{R}$. We leave for future work the case where the satisfaction of one user may depend on several agents.

This formulation, which may appear restrictive, is already a generalization of the usual setting where fairness is defined over agents. Moreover, it enjoys attractive advantages. It is simple, and transparent, openly presenting what is optimized. It is theoretically-founded as fair SWFs encode a clear and well-defined notion of fairness. This formulation and our solution method are generic, since it accepts any (sub)differentiable social welfare function (actually, even if it does not encode fairness).

4. Solution Method

To learn distributed fair policies, our solution is based on the optimization of SWFs combined with communication between agents. To efficiently optimize the SWF, we propose Self-Oriented Team-Oriented networks (SOTO) updated by dedicated policy gradients (Algorithm 1).

4.1. Policy Gradient

As the problem we want to solve can be expressed as a convex optimization problem, we adopt a policy gradient approach implemented in an actor-critic architecture for increased efficiency. In the context of decentralized policies, we can derive a direction to optimize the SWF for the i^{th} agent (see (3)):

$$\nabla_{\theta_i} \phi(\boldsymbol{J}(\boldsymbol{\theta})) = \nabla_{\boldsymbol{u}} \phi(\boldsymbol{J}(\boldsymbol{\theta}))^\intercal \cdot \nabla_{\theta_i} \boldsymbol{J}(\boldsymbol{\theta}), \tag{4}$$

where $\nabla_{\theta_i} \boldsymbol{J}(\boldsymbol{\theta})$ is a $D \times |\theta_i|$-matrix representing the usual stochastic policy gradient over the D different reward components and $\nabla_{\boldsymbol{u}} \phi(\boldsymbol{J}(\boldsymbol{\theta}))$ is a D-dimensional vector. For instance, with GGF, $\nabla_{\boldsymbol{u}} G_{\boldsymbol{w}}(\boldsymbol{J}(\boldsymbol{\theta})) = \boldsymbol{w}_\sigma$ where σ is a permutation that sorts $\boldsymbol{J}(\boldsymbol{\theta})$ in an increasing order. Similarly, for α-fairness, we have $\nabla_{\boldsymbol{u}} \phi_\alpha(\boldsymbol{J}(\boldsymbol{\theta})) = \boldsymbol{J}(\boldsymbol{\theta})^{-\alpha}$ where exponentiation is componentwise.

Let $\boldsymbol{A}(\boldsymbol{s}, \boldsymbol{a})$ denote a D-dimensional vector representing the joint advantage function of taking the joint action $\boldsymbol{a}$ in joint state $\boldsymbol{s}$ under the joint policy parameterized by $\boldsymbol{\theta}$. Using the policy gradient theorem (Sutton et al., 2000) and since the policies are independent, the gradient can be written as:

$$\begin{aligned} \nabla_{\theta_i} \boldsymbol{J}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{\theta}}\Big[\boldsymbol{A}(\boldsymbol{s}, \boldsymbol{a}) \cdot \nabla_{\theta_i} \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})^\intercal\Big] \\ &\approx \mathbb{E}_{\boldsymbol{\theta}}\Big[\Big(\boldsymbol{A}_{I_j}(\boldsymbol{s}, \boldsymbol{a})\Big)_{j \in [N]} \cdot \nabla_{\theta_i} \log \pi_{\theta_i}(a_i|o_i)^\intercal\Big], \end{aligned} \tag{5}$$

where a_i refers to the individual action taken by the i^{th} agent, o_i is the local observation of the i^{th} agent sampled from $\Omega_i(\boldsymbol{s})$ and $\nabla_{\theta_i} \log \pi_{\theta_i}(a_i|o_i)$ is a $|\theta_i|$-dimensional vector.

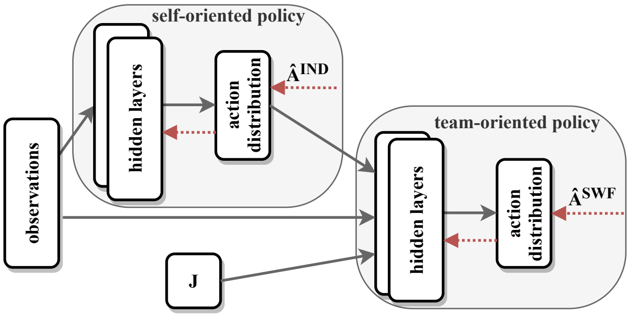

Figure 1: The SOTO architecture is composed of a self-oriented policy and a team-oriented policy. The self-oriented policy optimizes its individual utility $\boldsymbol{J}_{I_i}$ and recommends an action distribution to the team-oriented policy, which optimizes the SWF $\phi(\boldsymbol{J})$. Dashed arrows represent backpropagation flow.

The approximation is due to using decentralized policies with local observations.

However, in the decentralized multi-agent setting, computing $(\boldsymbol{A}_{I_j}(\boldsymbol{s},\boldsymbol{a}))_{j\in[N]}$ would usually require a centralized critic, thus computing the correct direction $\nabla_{\theta_i}\phi(\boldsymbol{J}(\boldsymbol{\theta}))$ is generally not possible. Instead, to approximate the aggregated advantages, we use the local critic of each agent (each critic ignores the effects of other agents):

$$\boldsymbol{A}_{I_i}(\boldsymbol{s},\boldsymbol{a}) \approx \hat{\boldsymbol{A}}_{I_i}(o_i,a_i) = \boldsymbol{r}_{I_i} + \gamma\hat{\boldsymbol{V}}_{I_i}(o_i') - \hat{\boldsymbol{V}}_{I_i}(o_i), \quad (6)$$

with $\boldsymbol{s}' \sim P(\cdot|\boldsymbol{s},\boldsymbol{a})$ and $o_i' \sim \Omega_i(\boldsymbol{s}')$. Hence, to approximate the aggregate advantage $(\boldsymbol{A}_{I_j}(\boldsymbol{s},\boldsymbol{a}))_{j\in[N]}$, the agents share their local advantages $\hat{\boldsymbol{A}}_{I_j}(o_j,a_j)$. We denote this approximated aggregate advantage by $\hat{\boldsymbol{A}}(\boldsymbol{o},\boldsymbol{a})$:

$$\Big(\boldsymbol{A}_{I_j}(\boldsymbol{s},\boldsymbol{a})\Big)_{j\in[N]} \approx \hat{\boldsymbol{A}}(\boldsymbol{o},\boldsymbol{a}) = \Big(\hat{\boldsymbol{A}}_{I_j}(o_j,a_j)\Big)_{j\in[N]} \quad (7)$$

In practice, instead of using the temporal difference (6) over one transition, TD(λ) can be used to reduce the bias of this estimation (Sutton & Barto, 2018; Schulman et al., 2016).

By combining (4), (5) and (7), the SWF policy gradient direction becomes:

$$\nabla_{\theta_i}\phi(\boldsymbol{J}(\boldsymbol{\theta})) \approx \mathbb{E}_{\boldsymbol{\theta}}\Big[\hat{\boldsymbol{A}}^{\text{SWF}} \cdot \nabla_{\theta_i}\log\pi_{\theta_i}(a_i|o_i)^\intercal\Big], \quad (8)$$

where $\hat{\boldsymbol{A}}^{\text{SWF}} = \nabla_{\boldsymbol{u}}\phi(\hat{\boldsymbol{J}}(\boldsymbol{\theta}))^\intercal \cdot \hat{\boldsymbol{A}}(\boldsymbol{o},\boldsymbol{a})$. As the policies are represented by neural networks, this gradient (8) is convenient to compute by simply backpropagating $\hat{\boldsymbol{A}}^{\text{SWF}}$ inside the policy network.

4.2. Neural Network Architecture

Since the agents do not have access to a centralized critic, they may receive conflicting information about the quality of their behaviors from $\hat{\boldsymbol{A}}^{\text{SWF}}$. This can prevent an agent i from knowing whether any good/bad performance with respect to its "individual utility" $\boldsymbol{J}_{I_i}$ (self-oriented performance) or with respect to the global social welfare (team-oriented performance) comes from themselves or from the behavior of others (credit assignment problem with non-stationarity).

To avoid this conflict and potential catastrophic forgetting of a good self-oriented behavior, we propose a neural network architecture where the policy optimizing the individual utility is no longer disturbed by the local critics of other agents (Figure 1).

In this architecture, the actor is composed of two subnetworks, which can be viewed as two different policies: one is self-oriented and the other team-oriented. The critic is designed in a similar fashion with two corresponding subnetworks, which take the same inputs as their respective policies, providing a critic to them. The self-oriented policy optimizes its individual utility given by its own critic without taking into account the shared advantages. The backpropagated advantages $\hat{\boldsymbol{A}}_i^{\text{IND}}$ for the self-oriented policy are defined as:

$$\hat{\boldsymbol{A}}_i^{\text{IND}} = \nabla_{\boldsymbol{u}_{\boldsymbol{I}_i}}\phi(\boldsymbol{J}(\boldsymbol{\theta}))^\intercal \cdot \hat{\boldsymbol{A}}_{I_i}(o_i,a_i). \quad (9)$$

Note that for the specific case of $|I_i| = 1$, $\hat{\boldsymbol{A}}_i^{\text{IND}} = \hat{\boldsymbol{A}}_{I_i}(o_i,a_i)$ is used directly as $\nabla_{\boldsymbol{u}_{\boldsymbol{I}_i}}\phi(\boldsymbol{J}(\boldsymbol{\theta}))$ reduces to scaling the learning rate. The team-oriented policy is updated by (8) with the aggregated advantages $\hat{\boldsymbol{A}}^{\text{SWF}}$.

The team-oriented policy takes as input the distribution proposed by the self-oriented one, the estimated $\boldsymbol{J}(\boldsymbol{\theta})$ of its neighbors, and the usual environmental observations. Observing $\boldsymbol{J}(\boldsymbol{\theta})$ is important because it is an essential information for making a fair decision. Without it, an agent cannot know whether all users are treated fairly or not, since it only observes the rewards of a subset of the users. For instance, in a resource collection task, an agent needs to know its own score and its neighbors' scores to decide if it wants to start sharing resources. Moreover, having access to the output of the self-oriented policy greatly simplifies decision-making when an agent's score is lower than its neighbors': it can simply mimic the self-oriented policy.

4.3. Training Schedule

Because the decentralized execution of independent policies causes non-stationarity in the gathered experience, like previous MADRL methods (Foerster et al., 2016; Jiang & Lu, 2019), we avoid off-policy learning. Therefore, to ensure on-policyness of transitions used to train our proposed architecture, the policy applied in the environment must be fixed during a period (we used the minibatch size). When an episode starts (see Alg. 1), each agent chooses with probability β if it applies its self-oriented policy (or its team-oriented policy otherwise). Since, an agent must already know how to exploit its own utility before being fair, β should be high

Algorithm 1 SOTO algorithm in CLDE scenario

```
Given $E$ the total number of episode
Initialize $\pi_i, \pi'_i, v_i, v'_i$, respectively the team-oriented/self-oriented policies, team-oriented/self-oriented critics.
for each episode e do
    $\beta = \max(1 - \frac{e}{0.5E}, 0)$
    for each agent i do
        Initialize $\boldsymbol{J}_{I_i} = \mathbf{0}$
        $(\mu_i, w_i) \leftarrow \begin{cases} (\pi'_i, v'_i) \text{ with probability } \beta \\ (\pi_i, v_i) \text{ otherwise} \end{cases}$
    end for
    while episode e is not completed do
        Collect $M$ a minibatch of transitions with $\boldsymbol{\mu}$ while updating and sharing $\boldsymbol{J}$ to the neighbors
        for each agent i do
            Update $w_i$ with TD($\lambda$) on $M$
            Compute $\hat{\boldsymbol{A}}_{I_i}(o_i, a_i)$ on $M$ with $w_i$ and TD($\lambda$) and send it to everyone    (6)
            if $\mu_i = \pi'_i$ then
                Update $\pi'_i$ with $\hat{\boldsymbol{A}}_i^{\text{IND}}$    (9)
            else
                Collect and form $\hat{\boldsymbol{A}}(\boldsymbol{o}, \boldsymbol{a})$    (7)
                Update $\pi_i$ with $\hat{\boldsymbol{A}}^{\text{SWF}}$    (8)
            end if
            $(\mu_i, w_i) \leftarrow \begin{cases} (\pi'_i, v'_i) \text{ with prob. } \beta \\ (\pi_i, v_i) \text{ otherwise} \end{cases}$
        end for
    end while
end for
```

at the beginning of training. However, since we ultimately want to optimize the SWF, β should decrease over time. When β reaches zero, which happens at half of the learning with linear annealing in our experiments, the weights of the self-oriented policy will not be updated anymore.

4.4. Communication

The presentation of our method corresponds to the Centralized Learning with Decentralized Execution (CLDE) scenario. We can also evaluate our approach in the Fully Decentralized (FD) scenario. Recall that for both scenarios, during the execution phase, the communication for an agent i is restricted to the sharing of its $\boldsymbol{J}_{I_i}(\boldsymbol{\theta})$ with its neighbors. During learning, while the agents in CLDE can communicate with all other agents, the agents in FD are allowed to communicate only with neighbors.

Note that our method never learns a centralized critic: it neither communicates full states nor actions, but only $\boldsymbol{J}_{I_i}(\boldsymbol{\theta})$ and advantages $\hat{\boldsymbol{A}}_{I_i}$. Thus, it scales well since the costliest operation depending on D is a matrix product of size $D \times m$ where m is the minibatch size to compute $\hat{\boldsymbol{A}}^{\text{SWF}}$.

When a complete minibatch is collected, the advantages are shared during the learning phase to form $\hat{\boldsymbol{A}}(\boldsymbol{o}, \boldsymbol{a})$. This is only possible in the CLDE scenario. In the FD scenario, several rows of the advantages inside $\hat{\boldsymbol{A}}(\boldsymbol{o}, \boldsymbol{a})$ might be set to zero for agents not being in the neighborhood. Instead of using (7), we have

$$\hat{\boldsymbol{A}}(\boldsymbol{o}, \boldsymbol{a}) = \left(\begin{cases} \hat{\boldsymbol{A}}_{I_j}(o_j, a_j), & \text{if } j \in \mathcal{N}(i) \\ \mathbf{0} & \text{otherwise} \end{cases} \right)_{j \in [N]},$$

where $\mathcal{N}(i)$ refers to the neighbors of the i^{th} agent.

In the following paragraph, we compare the number of messages (1 float) sent by an agent for different algorithms. Given k_i the number of neighbors of agent i, with $k_i \leq N-1$, at each time step the agent shares its $\boldsymbol{J}_{I_i}(\boldsymbol{\theta})$ with each neighbor. During the update phase, which happens when a minibatch of size M is full, they also share their estimated advantages for this minibatch and their estimated $\boldsymbol{J}_{I_i}$ to all agents. Thus, our method sends on average $(k_i + (1 + \frac{1}{M})(N-1))|I_i|$ messages per step for each agent in the CLDE scenario and $2k_i|I_i|$ messages in the FD scenario. As comparison, an agent in FEN sends on average $g\tilde{k}$ messages where g is the number of gossip rounds and $\tilde{k}$ is the number of random chosen agents to send the message. Generally, g needs to be greater than the diameter of the graph (so that the information can traverse the graph), which is upperbounded by the number of agents. Thus, assuming $\tilde{k} \approx k_i$, our method can be much more parsimonious than FEN in terms of communication when N becomes large. Likewise, a centralized critic would require even more communication by sending on average $\frac{N-1}{N}(\dim(\mathcal{A}_i) + \dim(\mathcal{O}_i) + |I_i| + 1) + k_i|I_i|$ messages every steps.

5. Theoretical Analysis

We analyze the convergence of a policy gradient method to solve Problem 3 under standard assumptions. The novelty of our analysis is two-fold. Contrary to previous work, we consider the partial observability of states and rewards, which is more realistic and fits better the decentralized setting. Besides, the overall objective is a non-linear concave function of the vector of the expected discounted rewards.

We can prove the following convergence result, which we state informally (see Appendix A for full details):

Theorem 5.1. *Under standard assumptions with a linear approximation scheme, the SWF objective $\mathfrak{J}(\boldsymbol{\theta}^k)$ converges almost surely and with a sub-linear convergence rate within a radius of convergence $\mathfrak{r}$ of the optimal value $\mathfrak{J}^*$ where $\mathfrak{r}$ depend on the approximation errors of (a) estimating $\boldsymbol{J}$, (b) estimating $\boldsymbol{A}(\boldsymbol{o}, \boldsymbol{a})$, and (c) ignoring the effects of one agent's action over other agents.*

Interestingly, this result implies a corollary, which provides a high-probability bound on the number of iteration steps before convergence. We provide all the details, further discussion, and the proofs in Appendix A. Besides, this theoretical analysis somewhat further justifies our architecture with a specific critic for the self-oriented policy, which helps reduce error (b) and therefore the radius of convergence.

6. Experiments

To test our algorithms, we carried out experiments in three different domains (detailed descriptions is available in the appendix): Matthew Effect (Jiang & Lu, 2019), distributed traffic light control (Lopez et al., 2018) and distributed data center control (Ruffy et al., 2019). We also evaluated our approach on the two other domains proposed by Jiang & Lu (2019), Job Scheduling and Plant Manufacturing. However, Job Scheduling being an easy artificial domain and Plant Manufacturing having an artificially-designed reward function, most of those results are presented in the appendix.

The first domain is Matthew Effect where 10 pac-men with different initial sizes have to collect resources which reappear randomly each time they are collected in a grid. The more resources an agent collects, the easier the task becomes for an agent because its size and speed increase.

The second domain adopted is a distributed traffic light control scenario. In this problem, we simulate a 3x3 intersection grid with Simulation of Urban Mobility (SUMO) where each of the nine agents controls the traffic light phase of one intersection. The global state is composed of the waiting time, density of cars, queue lengths, and current traffic-light phase of each intersection. For each agent, an action amounts to choosing the next traffic-light phase. The reward function of an agent is defined as the negative total waiting time at its intersection. Fairness can be understood as having low waiting times at every intersection. Note this domain is typically an example where there is no equal access to resources: some intersections will have naturally more traffic than others.

Our third domain is a data center control problem, where 16 hosts are connected with 20 switches in a fat-tree topology (see Figure 21 in the appendix). The network is shared by a certain number of hosts. The state is composed of information statistics about network features and the goal of each host/agent is to minimize the queue lengths in network switches. The continuous action corresponds to the allowed bandwidth for a host.

In all our experiments, we rely on the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017). The detailed hyperparameters are provided and in Appendix D.1 and available online[3]. To demonstrate the generality of our approach, we run our method with GGF (using $\boldsymbol{w}_i = \frac{1}{2^i}$) and α-fairness (using $\alpha = 0.9$). The different statistics are computed over the 50 last trajectories of 5 different runs.

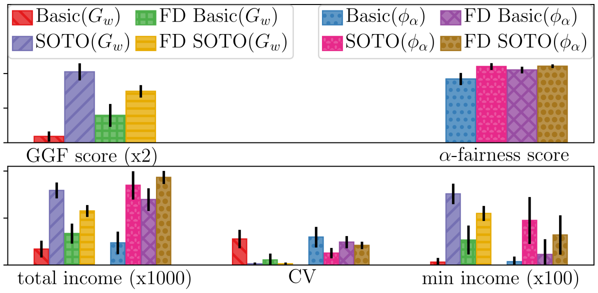

Figure 2: Comparison of SOTO and Basic in Matthew Effect in the CLDE and FD scenarios with GGF and α-fairness.

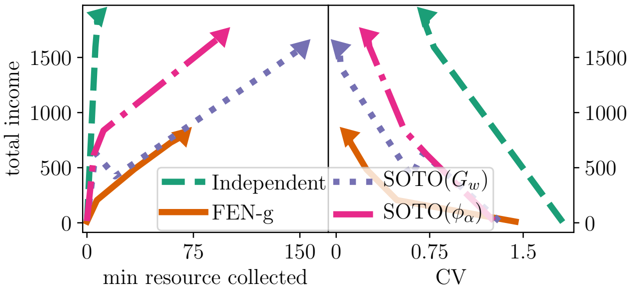

Figure 3: Trajectory of solutions reached by Independent, FEN, and SOTO on Matthew Effect in the CLDE scenario.

We name the methods that we evaluate as follows. Basic($G_{\boldsymbol{w}}$) and Basic(ϕ_α) refer to baselines where PPO optimizes the SWF directly without our proposed neural network architecture. It is equivalent to removing the self-oriented policy from our architecture and keeping the observation of neighbors $\boldsymbol{J}(\boldsymbol{\theta})$. SOTO($G_{\boldsymbol{w}}$) and SOTO($\phi_\alpha$) refer to instances of our proposed method. The prefix "FD" refers to the fully decentralized version. We also compare our methods with state-of-art algorithms such as FEN (Jiang & Lu, 2019), a centralized critic method COMA (Foerster et al., 2018) and value-based algorithm WQMIX (Rashid et al., 2020). FEN without gossip, labeled "FEN-g", assumes that the agents know the average utility (e.g., by exchanging all their utilities).

How does our architecture SOTO perform? We first discuss the experimental results in Matthew Effect and compare our architecture SOTO with several baselines. Similar observations can be made in other domains. Comparing to Basic, Figure 2 shows that SOTO provides a large improvement over the different criteria (for CV lower is better). Both in CLDE and FD scenarios, with GGF and α-fairness, our architecture Pareto-dominates the equivalent approach using the basic architecture without it.

[3] https://gitlab.com/AAAL/DFRL

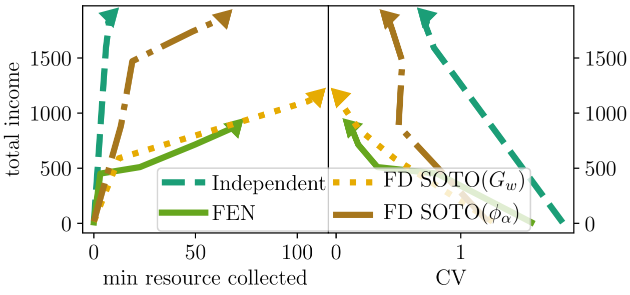

Figure 4: Trajectory of solutions reached by Independent, FEN, and SOTO on Matthew Effect in the FD scenario.

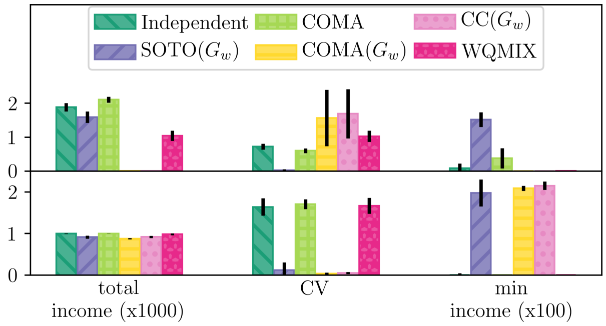

Figure 5: Comparison of SOTO, centralized critic, value based approaches and Independent in Matthew Effect (top) and Job Scheduling (bottom) in the CLDE scenario.

To compare with FEN, for better legibility, we plot the trajectories of the policies obtained during training in the space ”efficiency” vs ”equity” (Figure 3 and Figure 4 in the CLDE and the FD scenario respectively): total income vs min resource collected (higher in both dimensions is better) and total income vs CV (lower CV is better). As a sanity check, we include Independent where each agent optimizes its own utility, which in this domain is similar to optimizing the total income. This plot clearly shows that FEN converges to a worse policy than SOTO($G_{\boldsymbol{w}}$) both in terms of efficiency and equity. SOTO(ϕ_α) can also Pareto-dominate FEN in terms of min resource collected, but not in terms of CV. These plots illustrate that α-fairness provides a different trade-off between efficiency and equity compared to GGF. Our experiments also suggest that FEN may perform well in terms of CV due to its low efficiency.

As a sanity check, we also compare our method with state-of-the-art standard algorithms such as COMA and WQMIX. In addition, we consider two other variants: COMA($G_{\boldsymbol{w}}$), which is COMA extended to optimize $G_{\boldsymbol{w}}$; CC($G_{\boldsymbol{w}}$), which is the equivalent of Basic but with a centralized state-value function. Note that because of negative rewards, applying ϕ_α is not possible without tuning the reward function.

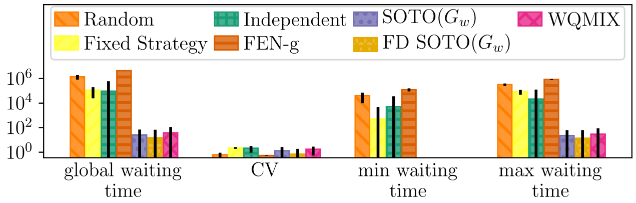

Figure 6: Algorithms' performances in the SUMO environment.

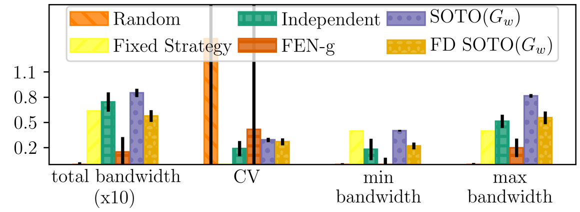

Figure 7: Algorithms' performances in data center control.

As expected, Figure 5 shows that COMA is better than Independent in terms of total income. In Job Scheduling, WQMIX performs well in terms of total income but it has the worst CV, while COMA($G_{\boldsymbol{w}}$) and CC($G_{\boldsymbol{w}}$) reach the lowest (better) CV. It shows that our proposition of optimizing a SWF to achieve fairness can easily be extended with a centralized critic. Due to the availability of $(\boldsymbol{o}, \boldsymbol{a})$, (7) can be computed with less bias, which explains why it can outperform SOTO. However, due to the centralized critic and value function, those variants of COMA and vanilla WQMIX are not able to scale in Matthew Effect where they perform poorly compared to SOTO($G_{\boldsymbol{w}}$).

Using the SUMO domain, we further demonstrate that our method can scale up to more complex control tasks, even with unequal access to resources. For this domain, we added two classic baselines. At each time step, ”Random” selects an action according to a uniform random distribution and ”Fixed Strategy” cycles between the traffic-light phases following an optimized period. Note that α-fairness can not be directly applied here because of negative rewards. Figure 6 shows that in terms of global waiting time, Independent works better than Random, but worse than Fixed Strategy, which means that being too selfish in this domain makes the task harder to solve globally. On the contrary, if the agents cooperate, the traffic flows more smoothly.

Our methods SOTO($G_{\boldsymbol{w}}$) and FD SOTO($G_{\boldsymbol{w}}$) are able to reach the lowest waiting times. The latter performs better than FEN and WQMIX on all other dimensions (global waiting time, CV, and max waiting time). FEN achieves a lower CV than Random but at the cost of the worst global waiting time. Note that FEN diverges in this environment.

Using the data center control problem, we show how well our method can perform on continuous action spaces. To do so, we extended FEN to continuous actions and we also added two classic baselines, ”Random” and ”Fixed Strategy”. ”Random Policy” selects an action according to a uniform random distribution and ”Fixed Strategy” always

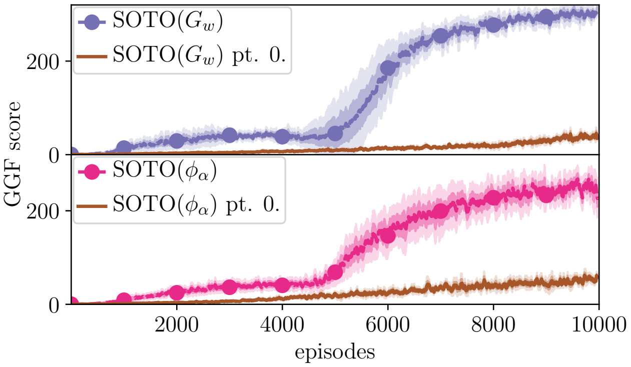

Figure 8: Comparison of SOTO and SOTO with a randomly initialized self-oriented policy on Matthew Effect.

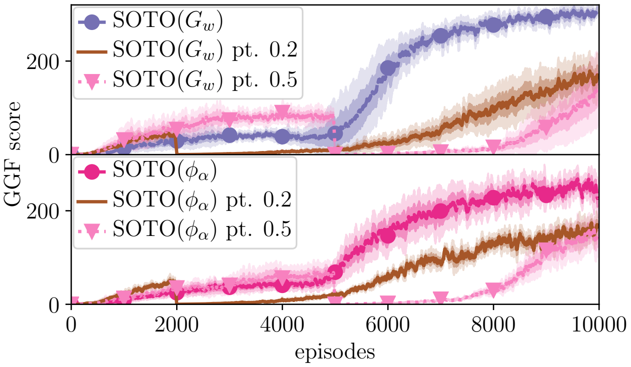

Figure 9: Comparison of SOTO($G_{\boldsymbol{w}}$) and SOTO(ϕ_α) with/without pretraining in Matthew Effect in the CLDE scenario.

chooses an optimized fixed bandwidth for each host. Note that α-fairness cannot also be applied directly here due to negative rewards. In Figure 7, as expected, the random policy performs worse as it has the lowest total bandwidth. The fixed policy performs better than random but worse than RL algorithms except FEN. Our method with GGF has a lower CV than current state-of-art FEN and the random policy. The fixed policy has the lowest CV as the same action is applied to all agents. In terms of total bandwidth our method performs very well as it maintains the maximum and minimum bandwidths.

Ablation Study We first check that the information contained in the self-oriented policy is really used by the team-oriented policy. One could argue that SOTO works better than Basic because of the additional inputs (the projection of the observation). To verify this hypothesis, we trained SOTO with a randomly initialized self-oriented policy (equivalent of using the pretraining baseline with $x = 0$). Therefore β is not used in this baseline. In Figure 8, it is clear that the information gathered in the self-oriented policy is important to optimize the SWF.

To justify the use of β, we compare our approach in Matthew Effect with two other baselines with pretraining, i.e., the self-oriented policy is trained first, then the team-oriented one is trained. Those baselines are labeled with the ”pt. x” tag where x refers to the ratio of the episode dedicated to the pretraining. Figure 9 clearly indicates that training incrementally with β by switching the policy used is more data efficient than using pretraining.

We refer the reader to Appendix B for additional experiments analyzing SOTO, FEN, and Basic.

7. Conclusion

We justified and formalized in a theoretically founded way the problem of fair policy optimization in the context of cooperative multi-agent reinforcement learning with independent policies. We proposed a simple, general and scalable method with a novel neural network architecture allowing an agent to learn to be first self-concerned in order to be able to reach a fair solution in a second step. We furthermore provided a theoretical convergence analysis of policy gradient for this fair optimization problem. We experimentally shown that each component of our proposed method is useful and that our approach achieves state-of-the-art results on various domains in two different training scenarios.

As future work, the relaxation of impartiality, the non independence of agents regarding users' utilities, or the simultaneous learning of the self-oriented and the team-oriented policies will be considered. Another interesting question is how to learn to communicate to achieve fairer solutions.

8. Acknowledgments

This work is supported in part by the program of National Natural Science Foundation of China (No. 61872238), the program of the Shanghai NSF (No. 19ZR1426700), and a Yahoo FREP grant. Half of the experiments presented in this paper were carried out using the Grid5000 testbed, supported by a scientific interest group hosted by INRIA and including CNRS, RENATER and several universities as well as other organizations (see https://www.grid5000.fr).

References

Adler, M. D. *Well-Being and Fair Distribution: Beyond Cost-Benefit Analysis*. Oxford University Press, 2012.

Amaldi, E., Coniglio, S., Gianoli, L. G., and Ileri, C. U. On single-path network routing subject to max-min fair flow allocation. *Electronic Notes in Discrete Mathematics*, 41: 543–550, June 2013.

Bertsimas, D., Farias, V. F., and Trichakis, N. The price of fairness. *Operations Research*, 2011.

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

Busa-Fekete, R., Szörényi, B., Weng, P., and Mannor, S. Multi-objective bandits: Optimizing the generalized Gini index. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 625–634. PMLR, 06–11 Aug 2017.

Chevaleyre, Y., Dunne, P. E., Lemaître, M., Maudet, N., Padget, J., Phelps, S., and Rodríguez-aguilar, J. A. Issues in Multiagent Resource Allocation. *Computer*, 30:3–31, 2006.

Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

de Jong, S., Tuyls, K., and Verbeeck, K. Fairness in multi-agent systems. *The Knowledge Engineering Review*, 23 (2):153–180, 2008.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 214–226, January 2012.

Foerster, J., Assael, I. A., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*, 2018.

Hao, J. and Leung, H.-F. *Fairness in Cooperative Multiagent Systems*, pp. 27–70. Springer, 2016.

Heidari, H., Ferrari, C., Gummadi, K., and Krause, A. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Hernandez-Leal, P., Kartal, B., and Taylor, M. E. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 2019.

Jay, N., Rotman, N., Godfrey, B., Schapira, M., and Tamar, A. A deep reinforcement learning perspective on internet congestion control. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3050–3059. PMLR, 09–15 Jun 2019.

Jiang, J. and Lu, Z. Learning fairness in multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Konda, V. and Tsitsiklis, J. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, 2000.

Kumar, H., Koppel, A., and Ribeiro, A. On the sample complexity of actor-critic method for reinforcement learning with function approximation. In *arXiv preprint: 1910.08412*, 2019.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

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

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

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

Ogryczak, W., Perny, P., and Weng, P. A compromise programming approach to multiobjective Markov decision processes. *International Journal of Information Technology & Decision Making*, 12:1021–1053, 2013.

Ogryczak, W., Luss, H., Pióro, M., Nace, D., and Tomaszewski, A. Fair optimization and networks: A survey. *Journal of Applied Mathematics*, 2014, 2014.

Papini, M., Binaghi, D., Canonaco, G., Pirotta, M., and Restelli, M. Stochastic variance-reduced policy gradient. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4026–4035. PMLR, 10–15 Jul 2018.

Pigou, A. *Wealth and Welfare*. Macmillan, 1912.

Pióro, M., Malicsko, G., and Fodor, G. Optimal link capacity dimensioning in proportionally fair networks. In *NETWORKING 2002: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications, Lecture Notes in Computer Science*, 2002.

Portugal, D. and Rocha, R. P. Distributed multi-robot patrol: A scalable and fault-tolerant framework. *Robotics and Autonomous Systems*, 61(12):1572–1587, December 2013. ISSN 0921-8890. doi: 10.1016/j.robot.2013.06.011.

Qiu, S., Yang, Z., Ye, J., and Wang, Z. On finite-time convergence of actor-critic algorithm. *IEEE Journal on Selected Areas in Information Theory*, pp. 1–1, 2021. doi: 10.1109/JSAIT.2021.3078754.

Raileanu, R., Denton, E., Szlam, A., and Fergus, R. Modeling others using oneself in multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4257–4266. PMLR, 10–15 Jul 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 10199–10210. Curran Associates, Inc., 2020.

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

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

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint: 1707.06347*, 2017.

Sharifi-Malvajerdi, S., Kearns, M., and Roth, A. Average Individual Fairness: Algorithms, Generalization and Experiments. In *Advances in Neural Information Processing Systems*, 2019.

Shi, H., Prasad, R. V., Onur, E., and Niemegeers, I. G. M. M. Fairness in Wireless Networks:Issues, Measures and Challenges. *IEEE Communications Surveys & Tutorials*, 16 (1):5–24, 2014.

Shorrocks, A. F. The class of additively decomposable inequality measures. *Econometrica*, 48(3):613–625, 1980.

Siddique, U., Weng, P., and Zimmer, M. Learning fair policies in multi-objective (Deep) reinforcement learning with average and discounted rewards. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8905–8915. PMLR, 13–18 Jul 2020.

Singh, A. and Joachims, T. Policy Learning for Fairness in Ranking. In *Advances in Neural Information Processing Systems*, 2019.

Speicher, T., Heidari, H., Grgic-Hlaca, N., Gummadi, K. P., Singla, A., Weller, A., and Zafar, M. B. A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pp. 2239–2248. ACM, 2018. doi: 10.1145/3219819.3220046.

Sukhbaatar, S., Szlam, A., and Fergus, R. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, 2016.

Sutton, R. and Barto, A. *Reinforcement learning: An introduction*. MIT Press, 1998.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2000.

van der Pol, E. and Oliehoek, F. A. Coordinated deep reinforcement learners for traffic light control. In *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems*, 2016.

Xu, P., Gao, F., and Gu, Q. An improved convergence analysis of stochastic variance-reduced policy gradient. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, 2020.

Zafar, M. B., Valera, I., Rodriguez, M. G., Gummadi, K. P., and Weller, A. From Parity to Preference-based Notions of Fairness in Classification. In *Advances in Neural Information Processing Systems*, 2017.

Zhang, C. and Shah, J. A. Fairness in multi-agent sequential decision-making. In *Advances in Neural Information Processing Systems*, 2014.

Zhang, K., Yang, Z., Liu, H., Zhang, T., and Basar, T. Fully decentralized multi-agent reinforcement learning with networked agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5872–5881. PMLR, 10–15 Jul 2018.

Zhang, K., Koppel, A., Zhu, H., and Baar, T. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2020. doi: 10.1137/19M1288012.

Zimmer, M. *Apprentissage par renforcement développemental*. PhD thesis, University of Lorraine, January 2018.