Towards Fair and Equitable Policy Learning in Cooperative Multi-Agent Reinforcement Learning

Umer Siddique
muhammadumer.siddique@my.utsa.edu
Department of Electrical and
Computer Engineering
The University of Texas at San Antonio

Peilang Li
peilang.li@my.utsa.edu
Department of Electrical and
Computer Engineering
The University of Texas at San Antonio

Yongcan Cao
yongcan.cao@utsa.edu
Department of Electrical and Computer Engineering
The University of Texas at San Antonio

Abstract

In this paper, we consider the problem of learning independent fair policies in cooperative multi-agent reinforcement learning (MARL). The objective is to design multiple policies simultaneously that can optimize a welfare function for fairness. To achieve this objective, we propose a novel Fairness-Aware multi-agent Proximal Policy Optimization (FAPPO) algorithm, which learns individual policies for all agents separately and optimizes a welfare function to ensure fairness among them, in contrast to optimizing the discounted rewards. The proposed approach is shown to learn fair policies in the independent learning setting, where each agent estimates its local value function. When inter-agent communication is allowed, we further introduce an attention-based variant of FAPPO (AT-FAPPO) by incorporating a self-attention mechanism for inter-agent communication. This variant enables agents to communicate and coordinate their actions, potentially leading to more fair solutions by leveraging the ability to share relevant information during training. To show the effectiveness of the proposed methods, we conduct experiments in two environments and show that our approach outperforms previous methods both in terms of efficiency and equity.

1 Introduction

Recent advances in reinforcement learning (RL) and multi-agent RL (MARL) have significantly improved the abilities of adaptive artificial agents to cooperate and solve complex tasks, including autonomous vehicles (Cao et al., 2012; Siddique et al., 2024), traffic light control (Wiering et al., 2000), data center control (Yao et al., 2022), and wireless networks (Naderializadeh et al., 2021). Despite the diverse applications of these systems, their primary focus has been mainly on optimizing a single performance metric, such as overall efficiency in autonomous cars, minimizing average waiting time in traffic light control, network congestion reduction in data centers, and throughput maximization in wireless networks. However, this singular focus on performance optimization often neglects the consideration of fairness, particularly in scenarios where these systems impact multiple end-users. For instance, in wireless networks, higher transmission powers to some users/agents may lead to interference issues for neighboring terminals. Hence, fairness becomes a key factor for the deployment and operation of such systems if we want the users to trust and use the systems.

Several studies have proposed to incorporate fairness into multi-agent systems. Notably, areas such as fair division (Beynier et al., 2019), fair mixing (Aziz et al., 2019), and fairness in non-cooperative games (de Jong et al., 2008; Hao & Leung, 2016) have been explored, mostly in static settings that do not require learning. Recent efforts have investigated fairness considerations in multi-agent sequential decision-making. Zhang & Shah (2014) proposed a regularized maxmin egalitarian approach to find equitable solutions. However, their approach may not guarantee an optimal solution, as solutions excluding the worse-off agent may still lack fairness. Fairness has also been considered in application-specific settings. For instance, Madani & Hooshyar (2014) developed a Markov game theory-RL method for optimal operation policies in multi-operator reservoir systems, emphasizing fairness and efficiency criteria. Jain et al. (2017) proposed cooperative MARL for co-optimization of cores, caches, and on-chip networks, highlighting the benefits of collaborative learning. Cui et al. (2019) introduced a distributed Q-learning-based method for fair resource allocation for UAV networks. Jiang & Lu (2019) introduced FEN, a decentralized method utilizing a gossip algorithm to estimate average utility and a hierarchical policy structure. Very recently, Ju et al. (2023) theoretically studied the fairness problem in multi-agent finite-horizon episodic MDPs and provided a probably approximately correct guarantee-based method. However, their work is limited to tabular settings. The closest to our work is Zimmer et al. (2021), which proposed SOTO, a method that learns self-oriented and team-oriented policies optimizing individual utility and welfare functions respectively. However, similar to FEN, SOTO also assumes that sharing individual agent utilities among neighboring agents is necessary to learn fair policies. Additionally, both of these methods rely on hierarchical structure or specialized network architecture where individual policies are first learned and then fairness is optimized via the leader or team policy, which may not always be practical, especially in systems where individual agent utility is sensitive or not allowed to share.

In contrast to existing methods, our approach does not rely on a specialized network architecture nor a hierarchical structure with the central controller and sub-policies. Instead, we propose a fairness-aware multi-agent Proximal Policy Optimization (FAPPO), an extension to the independent PPO (IPPO) (de Witt et al., 2020), that learns individual policies for all agents separately in the context of cooperative MARL. In particular, rather than optimizing the discounted sum of rewards, agents in our method learn to optimize a welfare function to ensure equitable rewards among them. When inter-agent communication is present, we also propose an attention-based variant of FAPPO (AT-FAPPO) by incorporating a self-attention mechanism (Bahdanau et al., 2014; Vaswani et al., 2017) for communication. Specifically, AT-FAPPO allows agents to learn what to share via an attention mechanism during the training phase. The advantage of the attention mechanism is that it enables communication, which allows agents to coordinate their actions and cooperate to achieve a common goal, potentially leading to more fair solutions. By doing so, we aim to address fairness in the cooperative MARL setting, which has been largely ignored in the literature.

The main contributions of this paper are summarized as follows:

1. We propose a novel algorithm, namely FAPPO, that learns independent PPO policies for each agent while simultaneously optimizing a welfare function to ensure fairness and equitable treatment across all agents, to address the challenge of fair optimization in cooperative MARL.
2. We introduce AT-FAPPO, an attention-based variant of FAPPO, where agents learn what information to share with other agents through a self-attention mechanism during the training phase. This attention-based approach enables communication and information sharing among agents, potentially improving the overall performance and fairness of the learned policies.
3. We validate our algorithms in two environments where fairness plays a crucial role. Through extensive experiments, we demonstrate the effectiveness of our approaches in achieving fair solutions in MARL settings while maintaining competitive performance compared to baseline methods.

2 Preliminaries

Dec-POMDPs. Cooperative multi-agent tasks are formalized as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek et al., 2016). The Dec-POMDP model can be defined with the following tuple $\langle \mathcal{S}, \mathcal{U}, \mathcal{P}, r, \mathcal{Z}, \mathcal{O}, N, \rho, \gamma \rangle$, where $s \in S$ describes the true state of the environment, $\mathcal{U}$ represents the action space (which can be discrete or continuous), $\mathcal{P}$ denotes the transition function, $N = \{1, \ldots, n\}$ denotes the set of n agents, ρ represents the initial state distribution, and $\gamma \in [0, 1)$ represents the discount factor that determines the importance of future rewards. In this model, at each time step t, each agent $a \in A \equiv \{1, \ldots, n\}$ chooses an action $u_t^a \in \mathcal{U}$, resulting in a joint action $\boldsymbol{u}_t = \{u_t^a\}_{a=1}^n$ in the environment. This causes a transition on the environment according to the state transition function $\mathcal{P}(s_t' \mid s_t, \boldsymbol{u}_t) : \mathcal{S} \times \boldsymbol{\mathcal{U}} \times \mathcal{S}$, and subsequently, each agent receives a team reward $r_t = r(s_t, \boldsymbol{u}_t) : \mathcal{S} \times \boldsymbol{\mathcal{U}} \to \mathbb{R}$ based on the current state and joint action.

We consider a partially observable scenario, which means that the agents have access only to partial observations of the environment $z_t \in \mathcal{Z}$ instead of the full state s_t, according to the observation function $\mathcal{O}(s_t, a) : \mathcal{S} \times A \to \mathcal{Z}$. The joint observation $\boldsymbol{z}_t = \{z_t^a\}_{a=1}^n$ represents the collective observations of all agents and can be referred to as the full state of the environment. Each agent has an action-observation history, which is denoted by $\tau_t^a \in T_t \equiv (\mathcal{Z} \times \mathcal{U})^t \times \mathcal{Z}$, where T_t is the set of all possible histories up to time t for each agent, and $\boldsymbol{\tau}_t = \{\tau_t^a\}_{a=1}^n$ is the set of all agents' histories. Each agent a selects its actions with a decentralized policy $u_t^a \sim \pi^a(\cdot \mid \tau_t^a)$ based only on its individual action-observation history. All agents in a team aim to learn a joint policy $\pi(\boldsymbol{u}_t|\boldsymbol{\tau}_t) \equiv \prod_{a=1}^n \pi^a(u_t^a|\tau_t^a)$ that maximizes some performance metric, such as the expected discounted return $J(\boldsymbol{\pi}) = \mathbb{E}[\sum_{t=0}^\infty \gamma^t r_t]$. The joint policy $\pi(\boldsymbol{u}_t|\boldsymbol{\tau}_t)$ induces a joint action-value function, which can be formally defined as $Q^\pi(s_t, \boldsymbol{\tau}_t, \boldsymbol{u}_t) = \mathbb{E}_\pi\left[\sum_{i=0}^\infty \gamma^i r_{t+i}\right]$, where γ is a discount factor and r_t is the random variable represents the team reward at timestep t.

Centralized Training and Decentralized Execution. We adopt the centralized training with decentralized execution (CTDE) learning paradigm (Oliehoek et al., 2016; Sunehag et al., 2017; Foerster et al., 2018). In CTDE, centralization is exploited during the training phase while maintaining decentralization during execution. In other words, during training, agents have access to the full environment state in addition to their local observation histories and they can also share policies and experiences with each other. However, during execution, agents must operate in a decentralized manner as agents may not have access to others' full state information. The CTDE paradigm allows the use of additional information during training to avoid non-stationarity issues and to facilitate the training of decentralized policies (Papoudakis et al., 2019). However, this can come up with great cost as even accessing the full state during training could be expensive.

Self-Attention. Self-attention is a mechanism used in deep learning models, particularly in transformer-based architectures, to capture the interdependencies between different input parts (Vaswani et al., 2017; Tay et al., 2021; Chen et al., 2021). The key idea behind self-attention is to allow the model to focus on the most relevant parts of the input when generating an output, rather than treating all parts of the input equally. Given the input, self-attention computes a set of queries, keys, and values simultaneously and packs them into query $\mathcal{Q}$, key $\mathcal{K}$, and value $\mathcal{V}$ matrices, which are used to compute the attention as

$$\text{Attention}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \text{softmax}\left(\frac{\mathcal{Q}\mathcal{K}^\top}{\sqrt{d_k}}\right)\mathcal{V},$$

where d_k is the dimensionality of the key vector $\mathcal{K}$. Additionally, we incorporate multi-head self-attention in this paper. Instead of computing a single attention function, multi-head self-attention linearly project the query $\mathcal{Q}$, key $\mathcal{K}$, and value $\mathcal{V}$ matrices h times with different parameter matrices $W^\mathcal{Q}, W^\mathcal{K}$ and $W^\mathcal{V}$, and performs the attention function in parallel with each of the projected version of query, key and value matrices. The output values of each attention function are then concatenated and transformed to the desired dimension by matrix W^O, given by

$$\text{MultiHead}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \text{Concat}\left(\text{head}_1, \cdots, \text{head}_h\right) W^O,$$

where $\text{head}_i = \text{Attention}\left(\mathcal{Q}W_i^\mathcal{Q}, \mathcal{K}W_i^\mathcal{K}, \mathcal{V}W_i^\mathcal{V}\right)$ and W^O is the parameter matrix of the output.

Fairness Formulation. We focus on the notion of fairness, which is rooted in distributive justice (Moulin, 2004), generally revolves around the equitable distribution of resources among different users. Previous studies in fair optimization within single-agent RL have primarily concentrated on learning a single fair policy (Weng, 2019; Siddique et al., 2020; Yu et al., 2023; Siddique et al., 2023; Yu et al., 2024). However, this paper extends its scope to encompass a more general setting wherein each user is viewed as an agent, and the objective is to treat all agents equally. This notion of fairness requires an optimal solution to satisfy three properties *efficiency*, *equity*, and *impartiality*. The efficiency property states that a solution should be Pareto-optimal. The equity property is based on the *Pigou-Dalton principle* (Moulin, 2004), which states that transferring utility from a more advantaged user to a less advantaged user results in a fairer solution. The impartiality property corresponds to the *"equal treatment of equals"* principle.

To operationalize this notion of fairness, we rely on welfare functions (d'Aspremont & Gevers, 2002). A welfare function, denoted as $\phi_{\boldsymbol{\omega}} : \mathbb{R}^n \to \mathbb{R}$, aggregates the utilities of all agents and measures how good it is in terms of social welfare with $\boldsymbol{\omega}$ representing the set of aggregation weights for all agents. While numerous welfare functions exist in the literature (Ogryczak et al., 2014), this paper focuses only on those that satisfy our definition of fairness. One such fair welfare function is the generalized Gini welfare function, which can be defined as

$$\phi_{\boldsymbol{\omega}}(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{\omega}_i \boldsymbol{x}_i^{\uparrow} \, , \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{\omega} \in \Delta_n$ is a fixed positive weight vector whose components are strictly decreasing (i.e., $\boldsymbol{\omega}_1 > \ldots > \boldsymbol{\omega}_n > 0$). Intuitively, by assigning larger weights to smaller utility values, this welfare function will yield larger scores when the utility distribution becomes more balanced. The chosen welfare function satisfies the above-mentioned three properties.

3 Proposed Method

We consider fully cooperative MARL tasks, where a set of agents cooperate to solve a given task. In such tasks, the impact of the final decision can impact multiple agents within the system. Therefore, it is crucial to consider fairness in the design of these systems to ensure their successful deployment. Our objective is to learn fair policies that optimize a welfare function, which can be formulated as

$$\max_{\boldsymbol{\pi_\theta}} \phi_{\boldsymbol{\omega}}(\boldsymbol{J}(\boldsymbol{\pi_\theta})), \tag{2}$$

where $\boldsymbol{\pi_\theta}$ represents the joint policy for all agents parameterized by $\boldsymbol{\theta}$, $\boldsymbol{J}(\boldsymbol{\pi_\theta}) = \mathbb{E}_{\boldsymbol{\pi_\theta}}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ denotes the joint expected discounted returns, and $\phi_{\boldsymbol{\omega}}$ is the welfare function. The objective is to maximize the welfare utility over the joint policy $\boldsymbol{\pi_\theta}$.

FAPPO. To solve the problem (2), we modify the Independent Proximal Policy Optimization (IPPO) (de Witt et al., 2020) algorithm to optimize the welfare function $\phi_{\boldsymbol{\omega}}$ and refer to it as Fairness-Aware multi-agent PPO (FAPPO), shown in Figure 1. In contrast to optimizing the discounted sum of rewards, agents in FAPPO learn to optimize a welfare function to ensure equitable rewards among them. This welfare optimization effectively addresses fairness because the weights $\boldsymbol{\omega}$ of $\phi_{\boldsymbol{\omega}}$ are chosen such that agents with lower utility values are assigned higher weights, ensuring fair treatment for all agents compared to scenarios where weights are assigned without considering utility values. In FAPPO, we employ PPO (Schulman et al., 2017) for learning individual policies for agents, with each agent learning its policy based solely on its local observations. Additionally, we normalize each agent's advantage separately to prevent the advantage scale from being dominated by other agents, and also clip the policy and critic loss separately for each agent. Since our method learns stochastic policies, we can optimize the welfare function $\phi_{\boldsymbol{\omega}}$ by computing gradients using a variant of the policy gradient theorem to update the policies as

$$\nabla_{\boldsymbol{\theta}} \phi_{\boldsymbol{\omega}}(\boldsymbol{J}(\boldsymbol{\pi_\theta})) = \nabla_{\boldsymbol{J}(\boldsymbol{\pi_\theta})} \phi_{\boldsymbol{\omega}}(\boldsymbol{J}(\boldsymbol{\pi_\theta}))^{\top} \cdot \nabla_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\pi_\theta}) = \boldsymbol{w}_\sigma^{\top} \nabla_{\boldsymbol{\theta}} \boldsymbol{J}(\pi_{\boldsymbol{\theta}}), \tag{3}$$

where $\nabla_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\pi_\theta})$ is a $n \times D$ matrix representing the joint policy gradient over the n agents, $\boldsymbol{w}_\sigma$ is a vector sorted based on the values of $\boldsymbol{J}(\boldsymbol{\pi_\theta})$, and D denotes the number of policy parameters.

Interestingly, in the independent learning setting, $\boldsymbol{J} = (J^1(\pi_\theta), \ldots J^n(\pi_\theta))$, where J^a is the utility of agent a. Thus, our optimization problem (2) can be expressed as

$$\max_{\boldsymbol{\pi_\theta}} \phi_{\boldsymbol{\omega}}(J^1(\pi_{\theta_1}), \ldots J^n(\pi_{\theta_n})),$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ is the policies parameters $\boldsymbol{\pi} = (\pi^1, \ldots, \pi^n)$ respectively. Using the policy gradient theorem (Sutton et al., 2000), the gradient of the utility function $J^a(\pi_\theta)$ for each agent a can be computed as

$$\nabla_\theta J^a(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[A^a_{\pi_\theta}(z^a, u^a)\nabla_\theta \log \pi_\theta(u^a \mid z^a)\right], \tag{4}$$

where A^a is the advantage estimation for the agent a. As we are in an independent learning setting, we estimate the advantage for each agent as $\sum_t (\gamma\lambda)^{t-1}\delta^a_t$, where $\delta^a_t = r_t(z^a_t, u^a_t) + \gamma V_\theta(z^a_{t+1}) - V_\theta(z^a_t)$. We use the team reward $r_t(s_t, \boldsymbol{u}_t)$ as the per-time-step reward $r_t(z^a_t, u^a_t)$ of agent a for approximation. $V_\theta(z^a_t)$ denotes the value function associated with agent a with local observation z_t, and λ represents the temporal difference (TD) estimation of the advantage function. Finally, for each agent a, the clipping objective becomes

$$\mathbb{E}_{z^a_t \sim \rho\pi, u^a_t \sim \pi\theta(\cdot|z^a_t)}\left[\min(\rho_\theta A^a_{\pi_\theta}(u^a_t|z^a_t), \bar{\rho}_\theta A^a \pi_\theta(u^a_t|z^a_t))\right],$$

where $\rho_\theta = \dfrac{\pi_\theta(u^a_t|z^a_t)}{\pi_{\theta_{\text{old}}}(u^a_t|z^a_t)}$, $\bar{\rho}_\theta = \text{clip}(\rho_\theta, 1-\epsilon, 1+\epsilon)$, $\pi_{\theta_{\text{old}}}$ represents the policy generating the transitions, and ϵ is a hyperparameter controlling the constraint.

AT-FAPPO. FAPPO leverages welfare functions to improve fairness in cooperative MARL. Another crucial factor that impacts fairness is inter-agent communication. In cooperative MARL, communication enables agents to coordinate their actions and cooperate to achieve a common goal, which is crucial for fairness. Since FAPPO learns individual policies, communication is not inherently present. Therefore, we propose AT-FAPPO, which incorporates a communication mechanism to further enhance fairness.

In AT-FAPPO, agents share information and collaborate during policy learning. We employ a multi-head self-attention mechanism, as illustrated in Figure 1. This mechanism allows each a to compute its query q^a, key k^a, and value v^a. For computational convenience, all agents' queries, keys, and values are packed together into matrices $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$, respectively to perform the attention function in parallel. In this paper, we use four parallel attention heads ($h = 4$), which allow agents to incorporate relevant information from their neighbors for better coordination and informed decisions.

Consider an example of agent 1 with a single attention head, in which the self-attention mechanism is implemented as follows: Agent 1 generates a query vector based on its current observation $q^1 = W^q z^1_t$. Along with the query q^1, all the neighboring agents and agent 1 compute their key vectors $k^a = W^k z^a_t$ and value vectors $v^a = W^v z^a_t$. The agent 1 then computes attention scores by taking the inner product of its query vector with the key vectors from neighboring agents and itself. These attention scores are then scaled and normalized using a softmax function to obtain attention weights $\{\alpha'_{1,1}, \alpha'_{1,2}, ..., \alpha'_{1,n}\}$. The attention weights are then multiplied with their corresponding value vectors, followed by summation to generate the output vector $b^1 = \sum_a \alpha'_{1,a} v^a$. Finally, all the output vectors $\{b^1, b^2, ..., b^n\}$ are packed into a matrix and fed into a linear layer. After applying the non-linear activation, the output is then fed into the learning agent policy.

In the self-attention mechanism, the inner product of the query vector with the corresponding key vector serves as a compatibility function, which is used to quantify the relationship or compatibility between different agents' observations. When two agents' observations are similar or relevant, the inner product of their query and key vectors is larger, which indicates that these agents are in proximity and should communicate more as their actions influence each other significantly. This communication resolves non-stationarity issues (Papoudakis et al., 2019) and mitigates unfairness among agents. By enabling communication and information sharing through the self-attention mechanism, agents can learn to coordinate their actions more effectively, yielding improved fairness and overall performance in multi-agent settings.

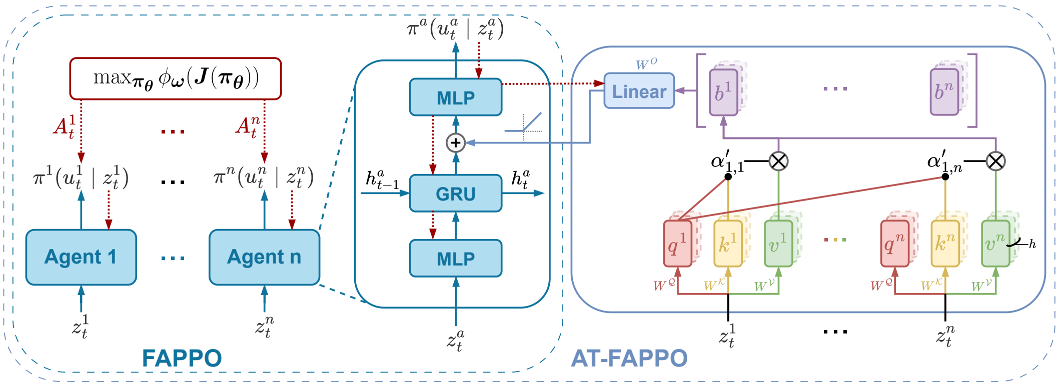


Figure 1: The FAPPO architecture is shown within the dotted box on the left. When combined with the multi-head self-attention mechanism on the right, it forms the AT-FAPPO architecture. Dashed arrows represent the backpropagation flow.

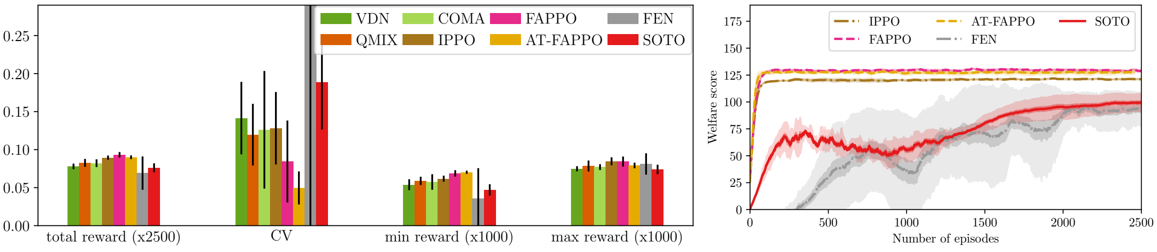
4 Experimental Results

To validate the efficacy of our proposed method, we performed experiments in two different environments. Each experiment showcases a unique scenario where fairness plays a crucial role. Specifically, our experiments are designed to show that cooperative MARL tasks can inherently exhibit unfairness, where a certain number of agents might be more advantageous than others, resulting in an unfair distribution of rewards. This behavior is akin to the Matthew effect, which describes how agents starting with an advantage tend to accumulate more of that advantage over time, while those starting with a disadvantage become further disadvantaged. To mitigate such behaviors, we proposed FAPPO and AT-FAPPO and conducted a rigorous evaluation, assessing their performance in achieving fairness objectives while maintaining desirable learning outcomes. To ensure the reproducibility of the results, we choose the welfare function weight vector as $\boldsymbol{w}_i = \frac{1}{2^i}, i = 0, ..., n-1$ and averaged results from 5 runs with different seeds. For all algorithms, we optimize the hyperparameters using grid search on two computers equipped with A100 GPUs.

For a comprehensive performance evaluation of our proposed method, we compared it against key multi-agent RL baselines, including value-based methods such as Value-Decomposition Networks (VDN) (Sunehag et al., 2017) Monotonic Value mixing network (QMIX) (Rashid et al., 2018), and policy-based methods such as Counterfactual Multi-Agent Policy Gradients (COMA) (Foerster et al., 2018). As an independent learning baseline, we also compare our method with Independent PPO (IPPO) (de Witt et al., 2020). Moreover, we also compared our method with state-of-the-art fairness baselines in MARL, including Fair Efficient Network (FEN) (Jiang & Lu, 2019) and Self-Oriented and Team-Oriented (SOTO) (Zimmer et al., 2021), which optimize the fair reward and welfare function, respectively, to achieve fairness.

4.1 Random MDP

The Random MDP environment is a simulated environment for evaluating the performance of RL methods. Here, we implemented a simple 5x2 grid-world-based multi-agent version of the Random MDP environment involving three agents. This dynamic environment presents the challenge of randomly distributed rewards among agents. The state representation in this environment encapsulates the current position of agents in the grid world using a one-hot encoding vector. To navigate this highly stochastic environment, each agent is equipped with five possible actions: move up, down, left, right, or take no action. The transition matrix governing state transitions is shared among all agents and is randomly generated using uniform distributions. To introduce an additional layer of complexity, each agent can only observe its local observation and receive its reward. This allows for the possibility of different rewards among agents, potentially leading to reward disparities. Since we



(a) CV, min reward, max reward, and welfare scores.

(b) Welfare scores.

Figure 2: Performances of VDN, QMIX, COMA, IPPO, FEN, SOTO, and our proposed methods in the multi-agent Random MDP environment.

are in a cooperative setting, all agent's rewards are then summed up for training the policies. The goal in this environment is twofold: to maximize the accumulation of rewards while simultaneously maintaining a balanced distribution of rewards among different agents. By creating this trade-off, agents aim to maximize overall efficiency and equity.

In this experiment, our primary goal is to assess the effectiveness of our method in optimizing the welfare function by achieving an equal distribution of rewards among agents and finding an efficient solution. To evaluate this, we conduct a comparative analysis of several state-of-the-art MARL algorithms and fairness baselines in MARL. Figure 2a illustrates the total rewards, the Coefficient of Variation (CV) among each agent's rewards, and maximum and minimum agent rewards. It can be seen that FEN performs the worst, with the lowest total reward and highest CV. SOTO outperforms FEN but underperforms other MARL algorithms, likely due to their reliance on neighbor impacts and communication, which aren't present in this environment. On the other hand, VDN, QMIX, and COMA learn better than FEN and SOTO both in terms of rewards and CV. Notably, independent learning baselines, including IPPO and our proposed methods, perform well, albeit with less hyperparameter tuning. Surprisingly, our proposed FAPPO and AT-FAPPO outperform all other methods both in terms of overall reward and CV. A lower CV indicates fewer variations in agent rewards, which is confirmed by the minimum reward, where only our proposed methods are capable of maximizing the minimum agent reward to establish a balanced distribution of rewards among all agents. In this environment, although learning independently is sufficient, AT-FAPPO still performs as well as FAPPO.

We also compare the algorithms in terms of their welfare score to assess if our proposed methods effectively optimize the welfare function $\phi_{\boldsymbol{\omega}}$. Figure 2b shows the welfare scores learning curves, focusing on IPPO, FEN, SOTO, and our proposed methods. Once again, our proposed FAPPO and AT-FAPPO demonstrate the highest welfare scores, showcasing their capability to identify fair solutions compared to other baselines. As the welfare score and CV might not show the full picture of objective balance, we present individual reward plots in Figure 3 as it is easy to show for this environment. Consistent with our previous findings, FAPPO and AT-FAPPO demonstrate their ability to deliver more balanced solutions.

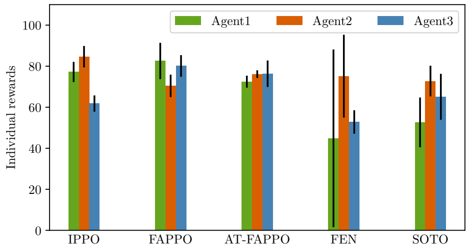
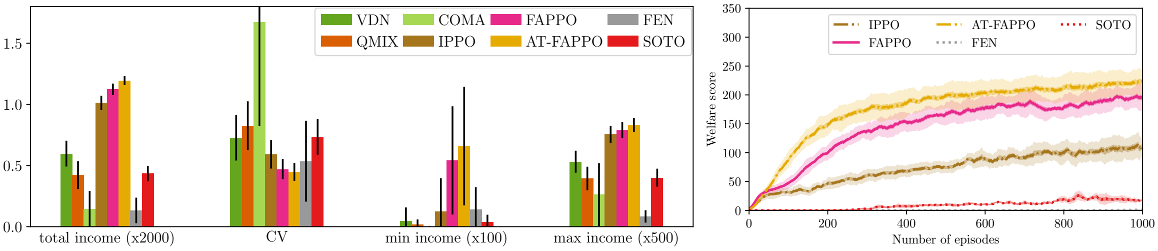


Figure 3: Individual rewards of IPPO, FAPPO, AT-FAPPO, FEN, and SOTO.

4.2 Matthew Effect

The Matthew effect refers to the phenomenon where the rich get richer and the poor get poorer. This phenomenon naturally extends to MARL systems, where agents starting with advantages or receiving more rewards during exploration tend to accumulate more rewards compared to those starting with fewer advantages. To address this challenge and incorporate fairness considerations,



(a) Total, CV, minimum, and max maximum income.

(b) Welfare scores.

Figure 4: Performances of VDN, QMIX, COMA, IPPO, FEN, SOTO and our proposed methods in the Matthew effect environment.

we adopt the Matthew effect scenario from (Jiang & Lu, 2019) and design our method to avoid it. In this scenario, we consider 10 agents (pac-men) initialized with different positions, sizes, and speeds, along with 3 stationary ghosts initialized at random locations. The state is defined by the position, size, and speed of each agent, as well as the position of the ghosts. To add complexity, each agent can observe the nearest three other agents and the nearest ghost. Agents have five available actions: move up, down, left, right, or take no action. The transition function governs state transitions by determining the distance between each agent (pac-men) and the ghost. If the distance between them is less than the size of the agent, the ghost is consumed, and a new ghost is generated at a random location. The reward function in this environment is sparse, meaning that agents receive a reward of 1 only when they consume a ghost. This environment exemplifies the Matthew effect, as when an agent consumes a ghost, its size, and speed increase correspondingly until reaching upper bounds. Consequently, agents who consume more ghosts become larger and faster, making it easier for them to consume ghosts. The goal in this environment is not to cooperatively consume more ghosts, but to ensure equal consumption of ghosts among all agents, thereby avoiding the Matthew effect and achieving an equal distribution of income among all agents.

To demonstrate the efficacy of our proposed approach in mitigating the Matthew effect, we conducted experiments and analyzed several metrics, including total income, CV, minimum and maximum agent income, and welfare scores. Figure 4a shows these metrics. As expected, FEN performs the worst in terms of overall income, although it achieves a lower CV than QMIX, VDN, and COMA. Interestingly, VDN performs better than QMIX in terms of reward but is outperformed by IPPO and our proposed FAPPO methods. FAPPO outperforms IPPO in terms of total income and CV as it maximizes the minimum agent income. AT-FAPPO surpasses FAPPO in all aspects, maximizing total and minimum agent income while achieving the lowest CV. Moreover, we also compare the algorithms in terms of their welfare score. Figure 4b shows welfare score curves during learning. As expected, both FAPPO and AT-FAPPO achieved the highest welfare scores, surpassing IPPO, FEN, and SOTO. This demonstrates the ability of our proposed methods to optimize the welfare function, creating a balanced distribution of income among all agents to avoid the Matthew effect.

5 Conclusions

In this paper, we addressed the unsolved problem of fair optimization in an independent learning setting. We proposed novel algorithms, fairness-aware PPO (FAPPO) and its attention-based variant (AT-FAPPO), designed to learn fair solutions that generate more equitable outcomes among multiple cooperative agents. We evaluated our algorithms in two different environments and demonstrated their practicality and potential applications in real-world scenarios where fairness considerations are imperative. Our results highlight the effectiveness of our approaches in achieving fairness objectives while maintaining efficiency. Potential future directions include studying other welfare functions and exploring additional policy gradient methods along with their convergence properties. Additionally, developing provably efficient fair value-based MARL algorithms could be a promising area for future research.

Acknowledgments

The work was supported in part by the Army Research Office under grants W911NF2110103 and W911NF2310363, and the Office of Naval Research under grant N000142412405.

References

Haris Aziz, Anna Bogomolnaia, and Hervé Moulin. Fair mixing: the case of dichotomous preferences. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 753–781, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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

Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1): 427–438, 2012.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Jingjing Cui, Yuanwei Liu, and Arumugam Nallanathan. Multi-agent reinforcement learning-based resource allocation for uav networks. *IEEE Transactions on Wireless Communications*, 19(2): 729–743, 2019.

Claude d'Aspremont and Louis Gevers. Social welfare functionals and interpersonal comparability. In *Handbook of Social Choice and Welfare*. Elsevier, 2002.

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

Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Jianye Hao and Ho-Fung Leung. *Fairness in Cooperative Multiagent Systems*, pp. 27–70. Springer, 2016. URL https://doi.org/10.1007/978-3-662-49470-7_3.

Rahul Jain, Preeti Ranjan Panda, and Sreenivas Subramoney. Cooperative multi-agent reinforcement learning-based co-optimization of cores, caches, and on-chip network. *ACM Transactions on Architecture and Code Optimization (TACO)*, 14(4):1–25, 2017.

Jiechuan Jiang and Zongqing Lu. Learning Fairness in Multi-Agent Systems. In *Advances in neural information processing systems*, 2019.

Peizhong Ju, Arnob Ghosh, and Ness Shroff. Achieving fairness in multi-agent mdp using reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2023.

Kaveh Madani and Milad Hooshyar. A game theory–reinforcement learning (gt–rl) method to develop optimal operation policies for multi-operator reservoir systems. *Journal of Hydrology*, 519: 732–742, 2014.

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

Navid Naderializadeh, Jaroslaw J Sydir, Meryem Simsek, and Hosein Nikopour. Resource management in wireless networks via multi-agent deep reinforcement learning. *IEEE Transactions on Wireless Communications*, 20(6):3507–3523, 2021.

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

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*, 2019.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

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

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

Umer Siddique, Abhinav Sinha, and Yongcan Cao. Fairness in preference-based reinforcement learning. *arXiv preprint arXiv:2306.09995*, 2023.

Umer Siddique, Abhinav Sinha, and Yongcan Cao. On deep reinforcement learning for target capture autonomous guidance. In *AIAA SCITECH 2024 Forum*, pp. 0957, 2024.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 2085–2087. Springer, 2017.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 2000.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *International conference on machine learning*, pp. 10183–10192. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Paul Weng. Fairness in reinforcement learning. In *AI for Social Good Workshop at International Joint Conference on Artificial Intelligence*, 2019.

Marco A Wiering et al. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pp. 1151–1158, 2000.

Zhiyuan Yao, Zihan Ding, and Thomas Clausen. Multi-agent reinforcement learning for network load balancing in data center. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 3594–3603, 2022.

Guanbao Yu, Umer Siddique, and Paul Weng. Fair deep reinforcement learning with preferential treatment. In *ECAI*, 2023.

Guanbao Yu, Umer Siddique, and Paul Weng. Fair deep reinforcement learning with generalized gini welfare functions. In *Autonomous Agents and Multiagent Systems. Best and Visionary Papers*, pp. 3–29, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-56255-6.

Chongjie Zhang and Julie A. Shah. Fairness in multi-agent sequential decision-making. In *Advances in neural information processing systems*, 2014.

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