Policy Optimization with Advantage Regularization for Long-Term Fairness in Decision Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Long-term fairness is an important factor of consideration for designing and deploying learning-based decision systems that do not discriminate against certain groups or populations in high-impact decision-making contexts, such as for admission and hiring, criminal justice, and resource allocation. Recent work has used the framework of Markov Decision Processes (MDPs) to formulate decision-making with long-term fairness requirements in dynamically changing environments, and demonstrated major challenges in directly deploying heuristic and rule-based policies that worked well in static environments. We will show that policy optimization methods from deep reinforcement learning can be used to find strictly better decision policies represented by neural networks, which can often achieve both higher overall utility and less violation of the fairness requirements, compared to previously known strategies. In particular, we propose a new method for imposing fairness requirements in policy optimization algorithms by regularizing the advantage evaluation in policy gradient, which makes it easy to impose fairness constraints without reward engineering or sacrificing training efficiency. We perform a detailed evaluation of the proposed methods in three established case studies, including attention allocation in incident monitoring, bank loan approval, and vaccine distribution for infectious diseases in population networks.

1 Introduction

Learning-based algorithmic tools are increasingly used in high-impact decision-making contexts, and whether they can ensure fairness and avoid discrimination has becomes a critical factor of consideration in their design and deployment [19]. A wide range of approaches have been developed to study and ensure fairness in algorithmic decision systems [19]. However, most of the literature studies fair decision-making in static environments, without explicitly dealing with how the decisions affect the statuses and behaviors of the targeted groups, which can in turn form a feedback loop that affects the effectiveness and fairness of the decision-making strategies. The long-term implications of such systems [12, 21, 29, 18, 10] have been recently studied through explicit modeling of the dynamics and feedback effects in the interactions between the decision-makers and the affected populations. In particular, the recent work of [12] has demonstrated, with concrete simulation examples, how the long-term fairness can not be analyzed in closed forms, but requires the use of more computational analysis tools based on simulations. Importantly, they proposed to formulate such long-term dynamics and the interaction between the decision-making and the environment in the framework of Markov Decision Processes (MDPs). This formulation and the corresponding simulation environments make it possible to take advantage of the recent advances in deep reinforcement learning (RL), for finding new decision-making policies that can potentially achieve both better overall utility and fairness in the dynamic setting, which is hard to handle with existing heuristic and rule-based strategies.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

One challenge of directly using RL-based methods for learning decision-making policies, however, is that the goal of decision systems in this critical context is inherently multi-objective. On the one hand, from a utilitarian perspective, an effective policy should maximize the overall utility of the decisions as much as possible for all target groups or populations. On the other hand, the fairness requirement should be explicitly enforced to prevent biased policies that negatively affect certain groups due to their temporary or historical disadvantageous situation. The standard RL framework for policy optimization only considers monolithic reward functions, and can only optimize the policies with respect to a single numeric metric. Consequently, users of RL methods in the decision-making context need to first explicitly define the trade-off between the utilitarian objective and the fairness objective, typically as simplistic weights in a linear combination. This requirement of *reward engineering* can make it hard to justify the policies obtained by the RL-algorithms, because one can question whether the predefined weights have introduced problematic assumptions and trade-offs between the objectives in the first place. The monolithic reward definitions may also incentivize the learning agent to perform *reward hacking* [22], reaching undesirable behaviors that exploit the wrong trade-off between the different objectives, such as conservatively accumulating incremental rewards without achieving the overall goal just to avoid the penalty of violating constraints. One approach for addressing such problems is the framework of Constrained Markov Decision Processes (CMDPs) [3], which allows one to explicitly declare rewards and constraints and use RL algorithms such as Constrained Policy Optimization (CPO) [1] to simultaneously maximize reward and minimize violation of the constraints over time. However, the CMDP formulation only requires the learning algorithms to lower the *expectation* of constraint violation *asymptotically*, i.e., achieving constraint-abiding policies in a probabilistic sense if training time is allowed to be infinitely long. In practice, in comparison to standard policy optimization methods such as Proximal Policy Optimization (PPO) [25], algorithms for CMDPs can take significantly longer to make progress, while the policies obtained after any finite training period may still have high violation rate of the constraints.

In this paper, we use policy optimization algorithms from deep RL for the design of decision-making strategies that balance overall utility and long-term fairness in dynamically changing environments. We utilize the formulation of such problems as MDPs following the work of [12], and propose the new technique of advantage regularization for imposing fairness requirements in the policy optimization process. The new method allows us to algorithmically optimize the trade-off between overall utility (reward functions) and fairness constraints (advantage regularizers), which can be used in standard policy optimization procedures [25, 23] and achieve more efficient learning and better performance than using the CMDP formulations. We evaluate our approaches in established case studies using the simulation environments designed in [12, 4], including incident monitoring, bank loan approval, and vaccine distribution for infectious diseases in population networks. We find that the proposed policy optimization with advantage regularization is able to find policies that are better than previously known strategies, both in achieving higher overall utility and less violation of fairness constraints across all the case study environments.

We note that our approach only guarantees optimality under the definitions of the objective and fairness constraint, and the optimal way to minimize this constraint does not necessarily correspond to the most socially acceptable way to solve long-term fairness problems. Thus, how *fair* our proposed method behaves is limited to the human design of the long-term fairness problem.

2 Related Work

Long-term Fairness in Algorithmic Decision-Making. The work in [12] is the first to formulate the long-term decision making and fairness problems as Markov Decision Processes (MDPs) and formalize simulations to evaluate on. These simulations allow us to consider the problem of decision-making systems as an agent design problem that is very similar to designing how to control robots. Similar works show that long-term fairness is nontrivial, and analyzing it in the context of a static scenario can be harmful because it contradicts fairness objectives optimized in static settings [16, 17, 20]. For example, [16, 20] find that providing a direct subsidy for a disadvantaged group with the purpose of improving some institutional utility actually widens the gap between advantaged and disadvantaged groups over time, which further shows that long-term fairness is difficult to achieve.

There have been a growing number of studies on fairness in the long-term [21, 29, 18, 10] with various algorithmic approaches. [18] proposes a graph-based algorithm to improve fairness in recommendations for items and suppliers. They relate fairness to breaking the perpetuation of bias

in the interactions between users and items. [10] proposes the use of causal directed acyclic graphs (DAGs) as a paradigm for studying fairness in dynamical systems. They argue that causal reasoning can help improve the fairness of off-policy learning, and if the underlying environment dynamics are known, causal DAGs can be used as simulators for the models in training. [29] provides a framework for studying long-term fairness and finds that static fairness constraints can either promote fairness or increase disparity between advantaged and disadvantaged groups in dynamical systems.

Several recent works have also considered fairness-like constraints in deep reinforcement learning in various different contexts. [6] designs fairness optimized actor-critic algorithms in deep reinforcement learning. They enforce fairness by multiplicatively adjusting the reward for fairness utility optimization in standard actor-critic reinforcement learning. The work of [26] studied multi-dimensional reward functions for MDPs motivated by fairness and equality constraints, and performed theoretical analysis on the approximation error with respect to the optimal average reward. Our focus is on proposing a practical algorithm for making fair decisions in the dynamic environments formulated in [12] and show that policy optimization through advantage regularization can find the neural network policies that significantly outperform previously known strategies in the dynamic setting.

Policy Optimization under Constraints. The most widely-adopted formulation of RL with a set of constraints is constrained Markov Decision Processes (CMDPs) [3, 28]. To encode the concept of safety, it augments the standard MDP framework with constraints over expectations of auxiliary costs. When models are known in discrete tabular settings, a CMDP is solvable using linear programming (LP) [3]. However, results are limited for model-free scenarios where model dynamics are unknown, and for large-scale or even continuous state action spaces [1, 8, 28]. More importantly, both objective and constraint in high-dimensional CMDP settings, where high-capacity function approximators are adopted, are non-convex. Recent methods in solving CMDPs in continuous spaces can be divided into two categories, in terms of ways to incorporate constraints. In soft constrained RL, it is a common practice to apply Lagrangian method with learnable Lagrangian multipliers and solve the converted unconstrained saddle-point optimization problem using policy-based methods [5, 7, 27]. Such Lagrangian methods achieve overall safety when policies converge asymptotically, nevertheless allowing possible violations during training. On the contrary, hard-constrained RL aims to learn safe policies throughout training. Representative works include Constrained Policy Optimization (CPO) based on trust region [1], surrogate algorithms with stepwise [11] and super-martingale [**?**] surrogate constraints, as well as Lyapunov-based approaches [8, 9].

3 Policy Optimization with Advantage Regularization

In long-term fairness studies [12], fairness is evaluated over a time horizon where the agent interacts with the environment, and the environment mutates in response to the interactions. Simulations following the MDP framework is one way to analyze fairness in a time horizon and systematically come up with strategies for maximizing fairness in the long-term rather than in a single step. The MDP naturally incorporates this idea that actions made in the present have small but accumulating consequences on the environment over time. Therefore, strategies that typically maximize fairness or minimize prediction errors in a static setting no longer optimize the correct objective in these dynamic scenarios. Long-term fairness is evaluated with metrics that describe the consequences made by an agent's policy on the different subgroups in an environment over time. These metrics are typically computed at each step of the MDP, and include data collected from the past time steps.

Policy Optimization in MDPs. Formally, an MDP is defined as $M = \langle \mathcal{S}, \mathcal{A}, f, r, \gamma \rangle$ with the following components. $\mathcal{S}$ denotes the state space, and $\mathcal{A}$ the action space. The transition function $f : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ determines the probability $f(s'|s, a)$ of transitioning into state s' from state s after taking action a. We consider general forms of reward functions $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ defined over transitions. We write π_θ to denote a stochastic policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$ parameterized by θ. The goal of policy optimization is to maximize the expected γ-discounted cumulative return

$$J(\phi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \right]. \tag{1}$$

Policy optimization methods estimate policy gradient and use stochastic gradient ascent to directly improve policy performance. A standard gradient estimator is to use $\nabla_\phi \log \pi_\phi(a_t|s_t)\hat{A}_t$, where π_ϕ

is a stochastic policy and $\hat{A}_t$ estimates the advantage that represents the difference between the Q value of an action compared with the expected value of a state, to indicate whether an action should be taken more frequently in the future. The gradient steps will move the distribution over actions in the right direction accordingly. The expectation is estimated by the empirical average over finite batch of samples. The proximal policy optimization algorithm (PPO) [25] applies clipping to the objective function to remove incentives for the policy to change dramatically, using:

$$J^{\mathrm{CLIP}}(\phi, \beta) \leftarrow \hat{\mathbb{E}}\left[\min\left(r(\phi)\hat{A}_\beta, \mathrm{clip}(r(\phi), 1-\epsilon, 1+\epsilon)\hat{A}_\beta\right)\right], \tag{2}$$

where $r_t(\phi) = \pi_\phi(a_t|s_t)/\pi_{old}(a_t|s_t)$ and ϵ is a hyperparameter. The clipping ensures the gradient steps do not overshoot in the policy parameter space in each policy update.

Algorithm 1 Policy Optimization with Constraint Advantage Regularization (POCAR)

Initialize policy (actor) network π_ϕ and the reward value function network V^r_η
Initialize replay buffer B as empty set
for episodes $= 1, \ldots, K$ **do**
  for $t = 1, \ldots, T$ **do**
    Sample $a_t \sim \pi_\phi(a_t|s_t)$
    Sample $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$
    Compute Δ_t
    $B \leftarrow B \cup (s_t, a_t, r_t, s_{t+1}, \Delta_t)$
  end for
  Sample mini-batches of size N from B
  $\theta \leftarrow \theta - \alpha_\theta \nabla_\theta R_{f,N,\rho,\Delta t}(\theta)$
  for each policy optimization step **do**
    Sample mini-batches of transitions from B
    $\delta_t \leftarrow r_t + \gamma V^r_\eta(s_{t+1}) - V^r_\eta(s_t)$ ▷ cf. [24]
    $\hat{A}(s_t, a_t) \leftarrow \delta_t + \gamma\delta_{t+1} + \cdots + \gamma^{T-t+1}\delta_{T-1}$
    $\hat{A}_\beta(s_t, a_t) \leftarrow \hat{A}(s_t, a_t) + \beta_1 \min(0, -\Delta_t + \omega) + \beta_2 \begin{cases} \min(0, \Delta_{t+1} - \Delta_t) & \text{if } \Delta_t > \omega \\ 0 & \text{otherwise} \end{cases}$
    $r(\phi) \leftarrow \pi_{\phi_{new}}(a|s)/\pi_\phi(a|s)$
    $J^{\mathrm{CLIP}}(\phi, \beta) \leftarrow \hat{\mathbb{E}}\left[\min\left(r(\phi)\hat{A}_\beta, \mathrm{clip}(r(\phi), 1-\epsilon, 1+\epsilon)\hat{A}_\beta\right)\right]$
    $\phi \leftarrow \phi + \alpha_\phi \nabla_\phi J^{\mathrm{CLIP}}(\phi, \beta)$
    $\eta \leftarrow \eta - \alpha_\eta \nabla_\eta \hat{\mathbb{E}}[(V^r_\eta(s_t) - G(s_t))^2]$ ▷ $G(s_t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$
  end for
end for

Advantage Regularization. We modify the PPO algorithm to be fairness-constrained during policy optimization in the following way. First, in addition to the original MDP with only the overall utility as reward function r, we declare a separate fairness metric $\Delta : S \to \mathbb{R}^{\geq}$ as a function of the state, where smaller Δ-value indicate better satisfaction of the fairness constraints. We then task the PPO algorithm to learn a decision policy to maximize the expected discounted future return J, while regulating the advantage term to reduce the change in Δ-value in each step. Suppose fairness at a particular time step can be modeled by the fairness constraint metric $\Delta(s_t)$. We minimize this term by modifying the advantage function in PPO to:

$$\hat{A}_\beta(s_t, a_t) = \hat{A}(s_t, a_t) + \beta_1 \min(0, -\Delta(s_t) + \omega) + \beta_2 \begin{cases} \min(0, \Delta(s_{t+1}) - \Delta(s_t)) & \text{if } \Delta(s_t) > \omega \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{A}(s_t, a_t)$ is the original PPO advantage. The two terms added to the original advantage estimator are designed as follows.

• The value-thresholding term $\min(0, -\Delta(s_t) + \omega)$ penalizes the advantage value for (s_t, a_t) if the fairness metric $\Delta(s_t)$ is too high. That is, if it is higher than some threshold ω, then this term evaluates to a negative value. On the other hand, if $\Delta(s_t)$ is already less than ω, then the term evaluates to zero, and does not affect the advantage value.

• The decrease-in-violation term $\min(0, \Delta(s_{t+1}) - \Delta(s_t))$ is activated when the value of $\Delta(s_t)$ is larger than the tolerable threshold ω. When that is the case, this term penalizes the overall advantage

if the degree of fairness violation does not decrease in the transition from s_t to s_{t+1}. If either the violation is decreasing or is less than ω, then this term does not affect the overall advantage.

For both of the advantage regularization terms, the hyperparameters ω, β_1, and β_2 are positive numbers that can be tuned. The design of these two terms uses the idea of Lyapunov stability [14], which is a powerful framework in control theory for stabilization problems such as reducing the error signal, which can be used to model the violation of the fairness requirement over time. In particular, the third term of decrease-in-violation imitates the requirement of negative-definiteness of Lie derivatives in the standard Lyapunov conditions, which measures the rate of convergence of a system to its equilibrium point. Here we use the the finite difference of $\Delta(s_{t+1}) - \Delta(s_t)$ to measure the dynamics of the fairness metric over time.

The full pseudocode for policy optimization with advantage regularization is shown in Algorithm 1. At each iteration of the policy update loop (Line 4-22), we sample trajectories from the system f_π, and compute the fairness constraint $\Delta(s_t)$ (Line 7) to save into buffer B alongside state s_t, action a_t, and next state s_{t+1} (Line 8). In each policy update step, we modify the advantage function to include the two additional fairness regularizer terms as described above.

4 Experiments

We evaluate the proposed approaches in three case studies as proposed in [12, 4], including attention allocation for incident monitoring, credit approval for lending, and disease control in population networks. In all environment, we compare the performance of policies that can be found by the proposed methods, in comparison with previous known heuristic, rule-based, and random strategies. We also compare different ways of imposing the fairness constraints in the policy optimization procedures, as well as the Constrained Policy Optimization (CPO) [1] approach, to evaluate the effectiveness of advantage regularization techniques.

4.1 Case Study: Attention Allocation for Incident Monitoring

We first consider the problem of attention allocation in a dynamic setting, formulated as an MDP in [12]. In this problem, an agent is tasked with discovering incidents across several sites, and does not have a large enough attention span to cover all incidents occurring at any given moment. The site incident rates increase or decrease proportional to the allocated site attention, making the environment dynamic. We find that training an RL agent in this environment outperforms the human-designed policy in both utility and fairness over time. Moreover, we modified the environment to be harder, where the previously known policies perform significantly worse. We show that policy optimization with advantage regularization continues to work well in the harder environments as well.

Environment. Let a_{kt} be the discrete attention allocated toward site k at time t, and R_{kt} be the incident rate. At each time step, the agent assigns N discrete units of attention across K sites. The number of incidents occurred at each site is sampled $y_{kt} \sim \text{Poisson}(R_{kt})$ and number of incidents discovered at each site is $\hat{y}_{kt} := \min(a_{kt}, y_{kt})$. The incident rates change proportional to the attention allocated to each site: if $a_{k,t} = 0$, then $R_{k,t+1} = R_{k,t} + d$, otherwise $R_{k,t+1} = R_{k,t} - d \cdot a_{k,t}$, where d is the parameter that controls how dynamic the environment is. The reward function promotes incident discovery and penalizes for incidents missed. It is defined $r(s_t) = \zeta_0 \sum_k \hat{y}_{kt} - \zeta_1 \sum_k (y_{kt} - \hat{y}_{kt})$, where ζ is a vector of reward weights.

Fairness Constraint. The fairness metric to be minimized is:

$$\Delta(s_t) := \max_{k,k'} \left| \frac{\sum_t \hat{y}_{kt}}{\sum_t y_{kt} + 1} - \frac{\sum_t \hat{y}_{k't}}{\sum_t y_{k't} + 1} \right|. \tag{4}$$

which measures the maximum difference in the ratios between total incidents discovered and total incidents occurred across all sites.

Baselines. The baselines include the purely greedy agent [12], which discovers the most incidents and is considered the most fair out of all agents as evaluated in [12]. This agent approximates each site's incident rate and sequentially allocates each unit of attention to the site that has the highest probability of inciting an incident to occur. We also compare with a neural network agent trained by CPO [2] to compare our approach of advantage regularization with the closest-related RL approach.

PPO Agents. The PPO agent observation is defined as the ξ-timestep history of the incidents discovered, incidents occurred, attention allocated, and fairness ratio at each site. The agent is able to observe incidents that occurred in the past, but not in the current time step. The action is a K-dimensional vector representing the discrete attention units a_{kt} to allocate to each site, where this vector sums to N.

We evaluate three variations of PPO-based policy optimization: Greedy PPO (G-PPO) greedily maximizes the objective without any fairness constraints, Reward-Only Fairness Constrained PPO (R-PPO) is fairness constrained at the reward level by adding some variation of $-\max(0, \Delta_t - \omega)$ to the reward function where ω thresholds the fairness constraint, and Advantage Regularization PPO (A-PPO) is fairness constrained at the policy gradient level by applying Equation 3 to PPO during training. Going forward, we will be referring to these three PPO variations and their definitions by their short-hand notations (G-PPO, R-PPO and A-PPO) for all experiments.

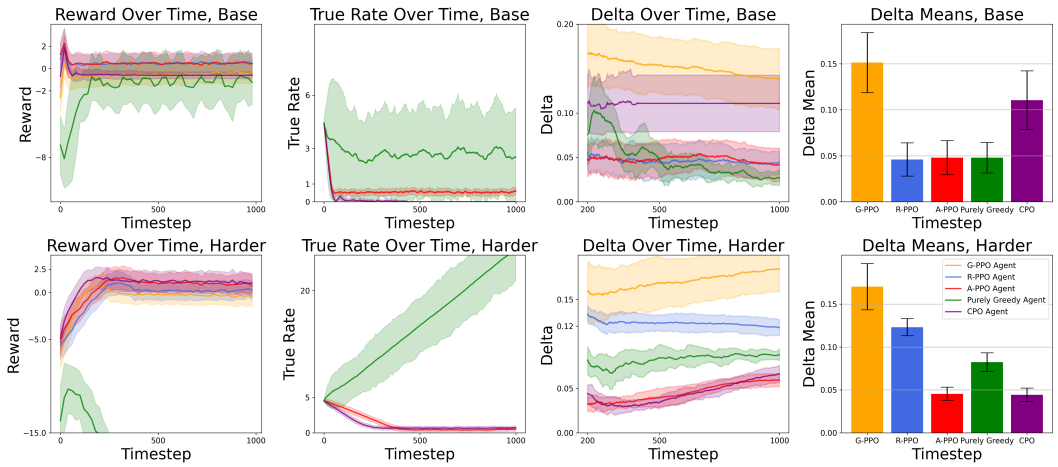

Figure 1: The agent reward over time, average incident rate across all sites over time, and delta summary evaluated over 10 trials for the base and harder attention allocation environments. All policy optimization algorithms, including the CPO agent, outperform the purely greedy agent in both reward and fairness. In the harder environment, the purely greedy policy collapses in performance, and we observe how advantage regularization can be a robust form of imposing fairness.

Results. We begin our experiments with first evaluating on the base attention allocation environment, where $d = 0.1$, $N = 6$, and $k = 5$. We observe more interesting results by evaluating on the harder attention allocation environment, i.e. changing $k = 10$.

In the base attention allocation environment in Figure 1, every PPO variation outperforms the baseline purely greedy agent in terms of reward over time. G-PPO does not perform as well as R-PPO or A-PPO, which is expected because the evaluation metric includes the fairness constraint term. Although A-PPO also does not have a fairness term in the objective, having advantage regularization in this setting allows A-PPO to perform just as well as R-PPO. In terms of fairness, we observe that G-PPO is the least fair agent as expected. Although the purely greedy agent does become the most fair agent at the end of the trajectory, R-PPO and A-PPO are more consistent in how fair they are on average over the entire trajectory. We observe that the CPO agent does not perform well on the base environment, because its incident rates get pushed to 0 early on in the trajectory in Figure 1.

To analyze why PPO outperforms the purely greedy agent in reward and fairness, we need to interpret their behaviors in Figure 1. Here, we observe that the average incident rate for the purely greedy agent is much higher than that of the PPO agent, so it will miss more incidents. The purely greedy agent's incident rate also has the highest standard deviation, which is consistent with the greedy agent's behavior of focusing all its attention on each site sequentially. By reducing the incident rate of a single site at a time but increasing the incident rates at all other sites, the purely greedy agent pushes the incident rates across sites to oscillate. After becoming well-trained, the PPO agents find that refining control on the incident rates across sites in this range allow the agents to maximize on the objective while being consistently fair.

Since any attention placed on a site means the site's incident rate goes down, it is possible to push all site incident rates to 0 if $K \geq N$. However, what happens if we modify the environment such that $N < K$ and the agent is unable to decrease the incident rates across all sites at once? After evaluating on this harder environment, our results become more interesting.

In Figure 1 A-PPO now performs significantly better than R-PPO, both of which achieve higher rewards over time than N-PPO. The CPO agent also does very well, and actually exceeds A-PPO in reward over time in the beginning of the trajectory. However, the purely greedy policy collapses in performance. To understand the cause of the baseline failing, we look at Figure 1. We see that the average incident rate in the purely greedy agent's evaluation shoots up uncontrollably. This is a stark contrast from its evaluation on the base environment, and reveals how this policy may be sensitive to slight changes in the problem setup. The PPO agents remain relatively consistent in their strategy, and still push the incident rates to a range in which they can exert a more refined control on the sites. In terms of fairness, incorporating advantage regularization allows A-PPO to be more fair than R-PPO as observed in Figure 1. Purely greedy agent, despite collapsing in performance, is more fair than the R-PPO and G-PPO. The CPO agent is also very fair, and actually matches performance with A-PPO. This tells us that our A-PPO is at least as good as the CPO agent, and can be a very favorable alternative considering how it takes longer to train the CPO agent.

We find that in this experiment, neural decision policies clearly outperform the human-designed policies in both reward and fairness. Incorporating advantage regularization for fairness performs better than only having a fairness term in the reward. Advantage regularization also appears to be more robust since in a harder environment, A-PPO is able to perform the best.

4.2 Case Study: Credit Approval for Lending

Next, we consider credit approval for lending in a dynamic setting [12]. In this environment, the agent plays the role of the bank and is tasked with deciding whether to accept or reject loan requests from a pool of applicants. These decisions affect the underlying population distribution for future time steps. In our experiments, we find that PPO agents are able to perform well without the oracle access to the underlying environment, with performance on par with the baseline strategies that need to make use of such ground-truth information. A-PPO is still the best policy optimization approach, though the gap between A-PPO and R-PPO is than the previous example because of the simpler task.

Environment. The agent is presented a stream of loan applicants, each with a discrete credit score $C \in \{1, 2, ..., C_{\text{max}}\}$ and some group membership variable $g \in \{1, 2\}$. Group 2 is considered disadvantaged compared to Group 1 with a lower mean of the initial credit score compared to Group 2. Applicants are uniformly sampled from these two groups. The environment dynamics change in response to the bank decision (accept or reject) and applicant decision (repay or default). If a loan request is accepted, the probability that an applicant pays the bank back is given by the underlying deterministic function of credit score $\pi(C)$. The higher the applicant credit score, the more likely they are to pay the bank back. The reward function encourages high bank profits and is defined $r(s_t) = \zeta_0(B_{t+1} - B_t)$.

Fairness Constraint. The fairness constraint is defined as $\Delta(s_t) = \max_{g,g'} |\text{TPR}_{gt} - \text{TPR}_{g't}|$, where TPR_g is the true positive rate $\frac{\text{TP}}{\text{TP+FN}}$ for group g. TP is the number of true positives corresponding to when the agent accepts a loan request and the loan is repaid, and FN is the number of false negatives corresponding to when the agent rejects a loan request and the applicant would have repaid.

Baselines. We have two baseline policies following the human-designed lending policies in [12]. Our first baseline agent is the greedy policy that attempts to maximize bank profits. The second baseline agent is the equality of opportunity (EO) agent [15], which maximizes bank profits with the constraint of equalizing the TPRs between the two groups. We note that to simplify the task, [12] gives the EO agent oracle access to the underlying distribution of credit scores amongst both groups and the exact repayment probabilities function π. We later show that RL is able to achieve similar, if not better, performance compared with the EO agent even without access to the underlying environment information. We also include the CPO agent as our secondary fairness-constrained deep RL baseline.

PPO Agents. The PPO agent observation is the applicant data consisting of their credit score C and group membership variable g. The action is a binary decision for whether to accept or reject an applicant's loan request. Like in the attention allocation environment, we evaluate the three previously defined PPO variations (G-PPO, R-PPO, A-PPO), but for added stability during training, we apply a min-max normalization to each term in Equation 3 for A-PPO before performing their weighted sum.

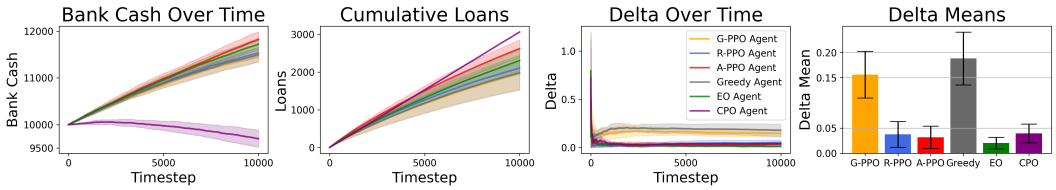

Figure 2: The bank cash over time, loans over time, and delta plots from evaluating on the lending experiment for 10 trials. We note that in this experiment, the PPO variations make more profits on average than the baseline algorithms. Although the EO policy is the most fair and makes a big profit over time, the A-PPO still out-earns the EO policy and is able to maintain a competitive level of fairness without having oracle access to underlying environment information. The CPO agent is one of the most fair agents, but fails to profit.

Results. We run experiments with $C_{\text{max}} = 7$. From Figure 2, it is interesting to see that all the fairness constrained agents except CPO make a bigger profit than the fairness unconstrained agents like G-PPO and greedy. To understand this trend, we look at the average number of loans made across groups by each agent in Figure 2. We see that the profit an agent makes roughly correlates to how many loans they give out on average. This makes sense since an agent cannot make money unless it gives out loans, and the agents that give out more loans can be interpret as more willing to take risks for higher rewards.

4.3 Case Study: Infectious Disease Control in Population Networks

In our third experiment, we analyze the precision disease control problem [4]. In this problem, the agent's task is to vaccinate individuals within a social network to minimize the spread of a disease. In this experiment, we observe that the PPO variations still outperform the human-designed policies in both utility and overall fairness. Integrating advantage regularization appears to have an equal effect as constraining fairness at the reward level only, which we analyze later. The CPO agent fails to perform well on this environment.

Environment. A social network N is composed of individuals V connected with edges E. Each individual has one of three health states defined by $\{S, I, R\}$ for susceptible, infected, and recovered. The health of the network is then characterized by vector $H \in \{S, I, R\}^{|V|}$. At the beginning of a trajectory, a random individual in N gets infected. For each time step, the infection spreads from an infected individual to a susceptible individual v under $p^v_{S\to I} = 1 - (1-\tau)^{\#I(v,N)}$, where $\tau \in [0, 1]$ is the probability of spreading and $\#I(v, N)$ is the number of infected neighbors of susceptible individual v in N. Then, probability of recovery is given by $p_{I\to R} = \rho$. At each time step, the agent is able to allocate one vaccine to give to an individual. Any individual can receive an unrestricted number of vaccinations, but vaccinating an individual will only have an effect if they are in the susceptible state. Vaccinations do not roll over over time steps, but the agent is able to choose not to vaccinate anyone at a time step. The reward function promotes the healthiness of a population and is given $r(s_t) = \zeta_0 \cdot (\sum_{i=1}^{|V|} \mathbb{1}(H_{it} \neq I))/|V|$.

We define the notion of a community in N by applying the Girvan-Newman algorithm [13] on the graph. This community detection algorithm progressively removes edges from the graph by betweenness, which is the number of shortest paths between two nodes that travel through an edge. We apply this algorithm once to obtain two communities (visualized in the Appendix).

Fairness Constraint. The fairness constraint is defined as

$$\Delta(s_t) = \max_{c,c'} \left| \frac{\sum_t \text{vaccinations given}_{ct}}{\sum_t \text{newly infected}_{ct} + 1} - \frac{\sum_t \text{vaccinations given}_{c't}}{\sum_t \text{newly infected}_{c't} + 1} \right|, \tag{5}$$

where c is a community. This fairness constraint is inspired by that from the attention allocation problem, and focuses on minimizing the differences in the ratio of vaccinations to newly infected individuals over time across communities.

Agents. The PPO agent observation is the health states H of the network. The action is a number $\{1, 2, ..., |V|\} \cup \{\varnothing\}$ where $\varnothing$ corresponds to not vaccinating anyone. We again use the three previously defined PPO agents (G-PPO, R-PPO, A-PPO). Similar to the previous case study, a min-max normalization is applied to each term in Equation 3 for A-PPO. We define two variations of the human-designed policies in [12]. First, the random baseline will randomly select a susceptible individual to vaccinate, and if there are no more susceptible individuals, then it will try to vaccinate someone randomly in the graph regardless of status. We also define the max neighbor (denoted as "max") baseline that will vaccinate susceptible individuals sorted by number of neighbors in the graph, and do the same for non-susceptible individuals if there are no more susceptible individuals in the graph.

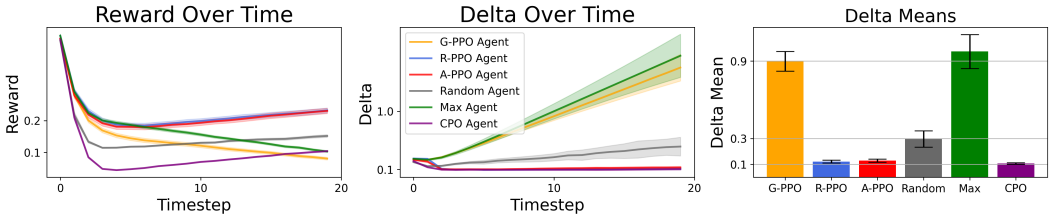

Figure 3: The reward over time and delta plots generated by evaluating on the precision disease control environment for 200 trials. We observe that all PPO policies outperform the baseline policies, with both A-PPO and R-PPO being equally the most fair.

Results. We run experiments on the Karate Club graph for 20 time steps per trajectory with $\tau = 0.5$ and $\rho = 0.005$. We set the initially infected individual to be the most connected node in the graph for the most interesting results, and impose a burn-in period where the infection is allowed to freely spread for one time step.

As seen in Figure 3, the PPO agents again outperform the baseline agents in reward and fairness. A-PPO and R-PPO both collect an equal amount of rewards, with R-PPO being slightly more favorable. The G-PPO and max agents' rewards fall off as they fail to satisfy the fairness evaluation objective over time. We observe that the random agent begins with a lower reward, but increases over time, which can be explained by the fairness analysis next. The CPO agent is observed to actually perform the worst in terms of utility. In Figure 3, A-PPO and R-PPO are also the most fair, as they both quickly drive the fairness constraint toward zero over time. The random agent is the next fairest agent, which can be explained by the fact that the random agent does not discriminate between groups when deciding who to vaccinate next. The G-PPO and max agents actually increase linearly in delta over time, which also makes sense since neither agent are constrained for fairness. Although CPO did not perform well in reward, it is actually one of the most fair agents evaluated here.

This experiment provides an additional insight into how RL agents can outperform human-designed policies. Here, we observe that A-PPO and R-PPO outperform all other agents in terms of reward and fairness. Although advantage regularization appears to not have any significant improvement over fairness constraining at the reward level, this experiment still serves as an example for how our approach can perform at least as well as reward-only fairness constraining, and how it can be more robust of a procedure than CPO.

5 Conclusion

We proposed new methods to design fairness-constrained decision-based neural control policies in long-term fairness problems. We augmented our approach with advantage regularization by formulating fairness as a stability property within policy optimization. We demonstrated the benefits of these methods in several dynamic environments for long-term fairness. Future work includes extending this framework to other policy optimization algorithms and investigating how our approach can be extended with stricter fairness-constrained methods.

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.

[2] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 22–31, 2017.

[3] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[4] James Atwood, Hansa Srinivasan, Yoni Halpern, and David Sculley. Fair treatment allocations in social networks. *CoRR*, abs/1911.05489, 2019.

[5] Steven Bohez, Abbas Abdolmaleki, Michael Neunert, Jonas Buchli, Nicolas Heess, and Raia Hadsell. Value constrained model-free continuous control. *arXiv preprint arXiv:1902.04623*, 2019.

[6] Jingdi Chen, Yimeng Wang, and Tian Lan. Bringing fairness to actor-critic reinforcement learning for network utility optimization. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[7] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

[8] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *Advances in neural information processing systems*, pages 8092–8101, 2018.

[9] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

[10] Elliot Creager, David Madras, Toniann Pitassi, and Richard Zemel. Causal modeling for fairness in dynamical systems. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[11] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

[12] Alexander D'Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, D. Sculley, and Yoni Halpern. Fairness is not static: deeper understanding of long term fairness via simulation studies. In Mireille Hildebrandt, Carlos Castillo, L. Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna, editors, *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, pages 525–534. ACM, 2020.

[13] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[14] Wassim Haddad and Vijaysekhar Chellaboina. Nonlinear dynamical systems and control: A lyapunov-based approach. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*, 01 2008.

[15] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[16] Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 259–268, New York, NY, USA, 2019. Association for Computing Machinery.

[17] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6196–6200. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[18] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. A graph-based approach for mitigating multi-sided exposure bias in recommender systems. *ACM Trans. Inf. Syst.*, 40(2), nov 2021.

[19] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2021.

[20] Smitha Milli, John Miller, Anca D. Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 230–239, New York, NY, USA, 2019. Association for Computing Machinery.

[21] Pegah Nokhiz, Aravinda Kanchana Ruwanpathirana, Neal Patwari, and Suresh Venkatasubramanian. *Precarity: Modeling the Long Term Effects of Compounded Decisions on Individual Instability*, page 199–208. Association for Computing Machinery, New York, NY, USA, 2021.

[22] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

[23] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML 2015*, pages 1889–1897, 2015.

[24] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[26] Umer Siddique, Paul Weng, and Matthieu Zimmer. Learning fair policies in multiobjective (deep) reinforcement learning with average and discounted rewards. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[27] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

[28] Ming Yu, Zhuoran Yang, Mladen Kolar, and Zhaoran Wang. Convergent policy optimization for safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3121–3133, 2019.

[29] Xueru Zhang, Ruibo Tu, Yang Liu, Mingyan Liu, Hedvig Kjellström, Kun Zhang, and Cheng Zhang. How do fair decisions fare in long-term qualification? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will release the code later once the paper is published.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]