Fair Contracts in Principal-Agent Games with Heterogeneous Types

Jakub Tłuczek
Université de Neuchâtel
jakub.tluczek@unine.ch

Victor Villin
Université de Neuchâtel
victor.villin@unine.ch

Christos Dimitrakakis
Université de Neuchâtel
christos.dimitrakakis@unine.ch

Abstract

Fairness is desirable yet challenging to achieve within multi-agent systems, especially when agents differ in latent traits that affect their abilities. This hidden heterogeneity often leads to unequal distributions of wealth, even when agents operate under the same rules. Motivated by real-world examples, we propose a framework based on repeated principal-agent games, where a principal, who also can be seen as a player of the game, learns to offer adaptive contracts to agents. By leveraging a simple yet powerful contract structure, we show that a fairness-aware principal can learn homogeneous linear contracts that equalize outcomes across agents in a sequential social dilemma. Importantly, this fairness does not come at the cost of efficiency: our results demonstrate that it is possible to promote equity and stability in the system while preserving overall performance.

1 Introduction

Modern economies, at both macro and micro levels, constitute complex multi-agent systems. Interactions between agents often involve contracts designed to align incentives between parties, fitting the principal-agent model: one party (the principal) offers a reward to another (the agent) in exchange for a specific outcome. This framework applies to a wide range of real-world scenarios, from employment agreements to government subsidies.

While usually parties in the principal-agent model are assumed to be greedy with respect to their own wealth, this assumption doesn't hold in real-life settings. It's been shown that it is often not the case [19], with fairness concerns about the other party being a decisive factor in contracting schemes. A boss might be inclined to offer bigger contracts to make sure the agent will stay motivated, while the agent might exercise more effort anticipating bigger contracts rewarding him in the future.

Unfortunately, fair contracts are hard to design. Typically, principals may seek to maximize their own benefits, overlooking their agents' wealth. From the agent's perspective, this can mean accepting unfavorable conditions or rejecting them altogether, resulting in either exploitation or stagnation. If contracts are not carefully constructed, inequality can also appear between agents. In retrospect, encouraging principals to account for the global health of the system in contract design should enhance social interactions and lead to more prosperous outcomes.

These insights extend naturally to Multi-Agent Reinforcement Learning (MARL), where principals and agents are all independently seeking to maximize their rewards through strategic decision-making. Within this setting, a principal can serve as a central planner (who still potentially takes part in the game), offering contracts that agents may choose to accept or reject. By shaping reward structures

18th European Workshop on Reinforcement Learning (EWRL 2025).

through contract design, the principal can steer agents toward more cooperative and equitable behavior, ultimately promoting system-wide efficiency and fairness.

Contributions. Leveraging from Contract Theory (CT) [6], which provides a principled way to incorporate incentive designs through contracts, we hypothesize that a fairness-aware principal can help the system converge to an equilibrium in which agents are rewarded more equally. Our contributions are as follows:

1. In Section 3, we formalize the principal-agents model through MARL, and state the objectives for the principal and agents in terms of reward maximization.
2. In Section 4, we interpret our setup through the lens of CT, establishing theoretical grounding for contract validity. We then study the learning of simple linear contracts, using policy gradient methods.
3. We propose two objectives for contract design (Section 5). The first adapts prior work [25] by incorporating the wealth of the agents into contract design. The second explicitly integrates fairness into the principal's objective. Both approaches are intentionally simple, relying solely on contractual information and drawing inspiration from the notion of reciprocity observed in real-world interactions.
4. We verify our methods on a multi-agent sequential social dilemma and explore the emerging behaviors adopted by agents and principals (Section 6). Our results demonstrate that fairness can lead to learning contracting strategies that yield a stable and fair principal-agent system.

2 Related Work

Multi Agent Reinforcement Learning is the subfield of reinforcement learning where multiple agents act simultaneously in a shared environment. The agents may pursue a common objective, or try to achieve their individual goals. The setting can either be centralized, or the agents may learn independently [24, 36]. The definition of optimality is problem-dependent in MARL and includes arriving at some type of equilibrium (Nash, SPE, Stackelberg), achieving Pareto optimality, maximizing welfare or maximizing fairness [1]. In this work, we focus on the problem of maximizing fairness, when agents are acting independently, but their behavior is mediated through a *mechanism*.

Incentive design is realized by providing external rewards to align agents with a particular goal. It is one of the fundamental challenges in the field of mechanism design [30]. In the scenarios with multiple agents, there exists a need to modify the rewards in order to incentivize agents to take desired actions [31] and when to induce the cooperative behavior among the population [5, 10, 34, 35, 37]. In our framework we are using identical contracts to as a mean of incentive design, to induce fairness in the multi-agent system.

Sequential social dilemmas (SSD) are repeated games where agents have to pick between cooperation and defection. Introduced by Leibo et al. [29], they build on matrix games and study the agents' behavior in a repeated setting. We test our solution on an SSD, Coin game [20], where agents don't have an explicit incentive to cooperate. Using simple contracts to get the heterogeneous population of agents to cooperate in SSD is one of the main contributions of this paper.

Contract theory is a well established subdomain in the field of economics [6, 23, 28, 32], which recently started getting attention in computer science [2, 4, 16, 17, 18, 22]. Principal-multiple agents model, has already been a topic of several groundbreaking works [9, 14, 15]. Informational assumptions vary and include bandit games [33], learning in repeated settings [22, 21, 12] and hidden rewards settings [11]. Reinforcement learning in the principal-agent framework has been introduced by [25], which we adapt for this work. However, our work differs in several key aspects. Specifically, we assume that both the principal's and all agents' policies are learned continuously. Additionally, agents are heterogeneous independently learning without sharing any parameters.

Typed contracts is a line of work in contract theory that considers contracting agents with hidden types. Two main approaches to the typed contracts are *menus of contracts* and *type-soliciting contracts* [16]. Menus of contracts [8] represent a tuple of contracts for each type. It is then up to the agent to select not only an action but also a type. In type-soliciting contracts [3] an agent is asked to declare his type, based on which the principal makes a contract offer. A contract is incentive-compatible if the agent not only declares his true type but also takes action to implement

the contract. Types might be multi- or single-dimensional, known or hidden [16]. Our framework falls in neither of the two traditional approaches since agents in our case have no control over their type, but at the same time, they are not being asked to disclose it to the principal. The difficulty in our case is to offer the same contract for each type, with agents never disclosing this information with the principal, which can only be elicitated by using historical data.

Fairness in multi-agent systems can be defined in several ways. In this work, we focus on equity, which aims to ensure that each agent obtains comparable benefits from the system. Equity-driven learning has been explored under the assumption that agents can observe each other's rewards [26]. An alternative approach is inspired by cake-cutting algorithms, where agents receive wealth allocations based on their private valuations [7]. Proportional fairness is one such concept [27], and states that any gain by one agent must result in proportional losses to others. Finally, individual fairness, formalized using the Lipschitz condition [13] requires similar agents to receive similar outcomes, according to some distance metric. In our setting, the agents' types and the principal's share are unknown. While individual fairness can be evaluated under perfect information, the principal can't ensure it without either approximating or having access to the agent type, making the equity a more practical objective.

3 Preliminaries

Markov Games. A finite horizon n-player Markov game can be formalized as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, T, s_o)$, where $\mathcal{S}$ is a set of states, $\mathcal{A} = A^1 \times ... \times A^n$ is a set of discrete actions for each player, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$ is a transition function, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \{1, \dots, n\} \to \mathbb{R}^+$ is a reward function, T is an horizon and s_0 an initial state.

At each time step t, all players observe the current state $s_t \in \mathcal{S}$, select actions $\mathbf{a}_t = (a_t^1, \dots, a_t^n) \in \mathcal{A}$, and receive individual rewards $\mathbf{r}_t = (r_t^1, \dots, r_t^n)$. The following time step, the game transitions to a new state $s_{t+1} \sim T(s_t, \mathbf{a}_t)$. This process repeats until the horizon T is reached. Each player i follows a policy π^i that maps their observations to a distribution over actions in A^i.

Heterogeneous Principal-Agent Markov Games. To introduce contracts into this framework, we define an heterogeneous principal-agent Markov game [25] as an $(n+1)$-player extension of an underlying n-player game $\mathcal{M}$. Formally, we denote it as $\mathcal{M}_{\text{pa}} = (\mathcal{M}, B, \mathcal{A}_{\text{a}}, \boldsymbol{\theta}, R_{\text{p}}, R_{\text{a}})$, where the first n players are *agents*, and the additional player indexed $p = n + 1$ acts as the *principal*. B is the action space of the principal, which is the set of contracts. The joint action space of agents is $\mathcal{A}_{\text{a}} = A_{\text{a}}^1 \times \cdots \times A_{\text{a}}^n$, where $A_{\text{a}}^i = A^i \cup \{\text{reject}\}$ augments the agent's action space to include a "reject contract" action.

To capture heterogeneity among agents, each agent i is endowed with a type $\theta^i \in \boldsymbol{\theta}$, not directly observable by the principal. Types may represent differences in skill, efficiency, or preferences. Following [37], we assume that an agent's type scales their effective contributions $\theta^i r_t^i$.

The contractual reward function for agents is defined as:

$$R_{\text{a}}(s_t, \mathbf{a}_t, s_{t+1}, i, b_t) = \big(b_t(\theta^i r_t^i) - c\big) \cdot \mathbb{1}[a_t^i \neq \text{reject}], \qquad i \neq p,$$

where b_t is a contract function that specifies payment based on the original reward, and c is a fixed cost incurred for choosing to act. Agents who accept the contract receive a reward determined by the contract and incur a cost c. Agents who reject the contract receive no reward and are treated as inactive during that step. In contrast, the principal's reward corresponds to the difference between the agents' raw contributions and their payments, e.g., the net surplus:

$$R_{\text{p}}(s_t, \mathbf{a}_t, s_{t+1}, b_t) = \sum_{i=1}^{n} \big(\theta^i r_t^i - b_t(\theta^i r_t^i)\big) \cdot \mathbb{1}[a_t^i \neq \text{reject}].$$

In this setting, the joint policy includes both the principal's policy and the agents' policies $\boldsymbol{\pi} = (\boldsymbol{\pi}_{\text{a}}, \pi_{\text{p}}) = (\pi_{\text{a}}^1, \dots, \pi_{\text{a}}^n, \pi_{\text{p}})$. The principal-agent interaction unfolds in three steps:

1. The principal proposes a unique and homogeneous contract to all agents upon observing state s, $b \sim \pi_{\text{p}}(\cdot \mid s)$.
2. The agents observe the common state s and respond to the proposed contract b by sampling an action $a^i \sim \pi_{\text{a}}^i(\cdot \mid s, b)$ from their policy.

3. A new state s' is generated from $\mathcal{M}_{\text{pa}}$, the agents pay their costs c, and obtain their contractual rewards $b(\theta^i r^i)$.

This framework allows the principal to shape the system dynamics by offering incentive-aligned contracts, while agents respond strategically based on their preferences and the contract terms.

Objectives. Within a principal-agent Markov game $\mathcal{M}_{\text{pa}}$, we define the value function of an agent i at time t in state s under contract b as the expected cumulative reward obtained from that point onward, under the joint policy $\boldsymbol{\pi}$:

$$V_t^i(s) \coloneqq \mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{M}_{\text{pa}}}\left[\sum_{k=t}^{T-1} R_{\text{a}}^i(s_t, \mathbf{a_t}, s_{t+1}, b_t, i)\middle| s_t = s\right], \quad i \neq p.$$

The value function can analogously be defined for the principal by replacing R_{a}^i with R_{p}. Each player (agents and the principal) is assumed to maximize their individual *wealth*, defined as their expected return from the initial state: $w^i \coloneqq V_0^i(s_0)$. We denote the set of wealth over all players as $\mathcal{W} = \{w^1, \ldots, w^n\}$, and define the system's *welfare* as $W \coloneqq \sum \mathcal{W}$.

4 Learning Contracts

To effectively learn principal policies for fair contract design, we must first formalize the core properties that define a valid CT setting, including the conditions under which agents should accept or reject contracts. In Section 4.1, we analyze CT literature to assess whether our framework satisfies the standard constraints. Then, in Section 4.2, we focus on a specific case where contracts are modeled as simple linear functions of the rewards perceived by agents.

4.1 Contract Theory

The principal-agent model [6, 17, 23, 32] is characterized by two parties aligning their interests with the help of contracts. The principal offers a contract, which offers particular payments conditioned on *outcomes*. The agent then decides whether to accept the contract or not: if the contract is accepted, the agent takes a hidden action, bearing its cost. While it is the principal who receives the reward for the outcome, the agent gets rewarded according to the contract agreed on beforehand. In the principal-agent Markov game defined in Section 3 we assume that outcomes are identical to the rewards collected by the agents.

To prevent trivial or degenerate solutions, contract theory imposes three standard constraints [16]:

1. *Incentive Compatibility* (IC): Agents are rational and aim to maximize their own wealth. A contract is IC if the agent's best response, i.e., the action that maximizes wealth under the contract, is aligned with the behavior desired by the principal. Actions that satisfy this condition are called *implementable*.
2. *Limited liability* (LL): Contractual payments must always flow from the principal to the agent. Agents must never have to pay the principal.
3. *Individual Rationality* (IR): Agents should reject contracts that yield negative expected returns. Formally, agent i accepts a contract b_t at state s_t if and only if there exists an action $a^i \in A_i$ such that:

$$\mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{M}_{\text{pa}}}\left[\left(b_t(\theta^i r_t^i) - c\right) + V_{t+1}^i(s_{t+1})\middle| s_t, b_t, a_t^i = a^i\right] \geq 0.$$

We address the LL constraint first. While the principal determines the contract terms and initiates all transfers, this does not automatically guarantee that resulting payments are non-negative. However, as shown in the next section, linear contracts allow us to enforce LL by appropriately constraining the principal's contract space. From this point onward, we assume that LL is satisfied by design.

When agents act optimally, as assumed in Section 3, the IC and IR constraints are also by definition satisfied. Agents following optimal policies act only when doing so is more profitable than rejecting contracts (IC). Since all transfers are non-negative, optimal policies implicitly avoid negatively valued states, thereby further respecting IR.

Algorithm 1 Principal-Agent Policy Gradients with Linear Contracts

Input: $(n+1)$-player principal-agent Markov game $\mathcal{M}_{\text{pa}}$; learning rates η_{p} (principal), η_{a} (agents)
Randomly initialize policy parameters $\phi_0^1, \ldots, \phi_0^n, \phi_0^p$
for $k = 0, \ldots$ **do**
  for episode $e = 1, \ldots, N$ **do**
    Reset game $\mathcal{M}_{\text{pa}}$ to initial state s_0
    Initialize episodic wealth $\hat{w}_e^i \leftarrow 0$ for all players
    for timestep $t = 1, \ldots, T$ **do**
      Principal selects contract $\alpha_t \sim \pi_{\text{p}}(\cdot \mid s_t; \phi_k^p)$
      Each agent $i \neq p$ selects action $a_t^i \sim \pi_{\text{a}}^i(\cdot \mid s_t, \alpha_t; \phi_k^i)$
      Observe rewards $\mathbf{r}_t$ and next state s_{t+1}
      for agent $i = 1, \ldots, n$ **do**
        if $a_t^i \neq$ reject **then**
          Pay agent $\hat{w}_e^i \leftarrow \hat{w}_e^i + \left(\alpha_t \theta^i r_t^i - c\right)$
          Accumulate principal wealth $\hat{w}_e^p \leftarrow (1-\alpha_t)\theta^i r_t^i$
  Estimate wealth $\hat{w}^i \leftarrow \frac{1}{N}\sum_{e=1}^N \hat{w}_e^i$ for each player i
  // Gradients can be approximated through any policy-gradient methods (e.g. PPO)
  Update principal's policy: $\phi_{k+1}^p \leftarrow \phi_k^p + \eta_{\text{p}} \nabla_{\phi_p} \hat{w}^p$
  Update each agent's policy : $\phi_{k+1}^i \leftarrow \phi_k^i + \eta_{\text{a}} \nabla_{\phi_i} \hat{w}^i,\ i \neq p$

However, this reasoning assumes that agents know the environment and the strategies of others. In practice, agents often operate under partial knowledge of the game and must learn through interaction. This uncertainty introduces a key challenge: value estimation errors may lead agents to make suboptimal decisions, inadvertently violating the IR or IC constraints. Let $\hat{V}^i$ denote the agent i estimate of its value function. Suppose that, at some state s_t, the agent is offered a contract b_t, and selects an action $a^i \in A_i$ based on the belief that it will yield a higher expected return than rejecting:

$$\mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{M}_{\text{pa}}}\left[\alpha_t \theta^i r_t^i - c + \hat{V}^i_{t+1}(s_{t+1}) \middle| \alpha_t, a_t^i = a^i\right] \geq \mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{M}_{\text{pa}}}\left[\hat{V}^i_{t+1}(s_{t+1}) \middle| \alpha_t, a_t^i = \text{reject}\right].$$

From the agent's perspective, accepting the contract appears rational and IR-compliant. Yet due to inaccurate value estimates, the actual return could still be negative, thereby violating IR in practice, even if not in intent. This illustrates a fundamental tension in learning-based principal-agent Markov games: agents may act optimally relative to their beliefs, while still accepting unreasonable contracts due to their poor value estimates. Addressing this challenge requires contract mechanisms that are robust to agent learning errors, e.g., by explicitly incorporating uncertainty for the contract designer itself.

4.2 Linear Adaptive Contracts

A particularly interpretable and practical class of contracts is *linear contracts*, in which agents receive a fixed proportion of the rewards they help generate, rather than a fixed payment. In this setup, a contract is parameterized by a single scalar $\alpha \in [0, 1]$, representing the agent's share of the reward. Linear contracts are not only easier to analyze and interpret, but they also admit efficient geometric solutions for optimal design [17]. Additionally, it ensures that LL always holds, as the principal always pays to agents and not the opposite ($\alpha \geq 0$). For these reasons, we restrict the principal to issuing linear contracts, meaning the contract space is $B = [0, 1]$. Under this formulation, the objectives of the principal and agents become:

$$\max_{\pi_{\text{p}}} \mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{M}_{\text{pa}}}\left[\sum_{t=0}^{T-1}(1-\alpha_t)\sum_{i=1}^{n}\theta^i r_t^i \middle| s_0\right], \qquad \max_{\pi_{\text{a}}^i} \mathbb{E}^{\boldsymbol{\pi}}_{\mathcal{M}_{\text{pa}}}\left[\sum_{t=0}^{T-1}\alpha_t \theta^i r_t^i \middle| s_0\right]. \tag{1}$$

We address this learning problem using *policy gradient* methods, treating both the principal and agents as learners. In particular, we model the principal's policy as a Gaussian distribution: contracts at state s are sampled as $\alpha \sim \mathcal{N}(\mu_s, \sigma_s)$, where μ_s and σ_s are learned state-dependent parameters.

This stochasticity adds uncertainty to contract offers, preventing agents from precisely predicting and exploiting the contract dynamics. In contrast, a deterministic contract policy may encourage agents to repeatedly reject suboptimal offers, effectively coercing the principal into making increasingly generous proposals.

Algorithm 1 outlines the learning procedure in this setting. After collecting sufficient experience to estimate policy gradients, both the principal and agents update their policies. However, looking at the principal's objective in Equation (1), an imbalance in learning speeds can be problematic. If agents have fixed policies or learn significantly more slowly than the principal, the principal can trivially exploit this by reducing contract values α_t, thereby maximizing its own profit before agents have the chance to adapt and start rejecting unfair offers. To mitigate this issue, it makes sense to set a smaller learning rate for the principal ($\eta_{\text{p}} \ll \eta_{\text{a}}$).

Even under balanced learning dynamics, the principal's policy may still converge to offering contracts that only marginally satisfy the IR constraint, i.e., contracts that agents are barely willing to accept. While this behavior is optimal from the principal's wealth-maximizing perspective, it is undesirable from a fairness standpoint. This highlights the need to regularize the principal's objective, either by promoting agent welfare directly or by penalizing large disparities in wealth across players.

5 Regularization for Fair Contracts

Rather than solely maximizing its own wealth, the principal can incorporate fairness considerations into its objective. To guide the principal's policy toward fair treatment of agents despite their unknown types, the reward signal can be modified to reflect this secondary goal. Since promoting fairness requires some degree of altruism from the principal, this inclination must be quantified. While agents have type θ^i, we can similarly characterize the principal by an altruism parameter λ, which captures the extent to which it values the overall welfare, or fairness of the system, alongside maintaining a reasonable share of the profits.

In Section 5.1, we introduce a welfare-based regularization approach, inspired by the formulation in [25], as a baseline. Then, in Section 5.2, we present an alternative regularization method that penalizes disparities in the wealth accumulated by agents, using variance as a fairness metric.

5.1 Welfare-based regularization

It has been shown that directly tying the principal's objective to agent wealth can encourage more cooperative behavior [25]. Following this insight, we incorporate system welfare into the principal's reward by augmenting it with the agents' collected rewards. This leads to a modified reward function:

$$\begin{aligned} R_{\text{p}}^{\text{welfare}}(s_t, \mathbf{a}_t, s_{t+1}, b_t) &= \sum_{i=1}^{n} \big((1-\alpha_t)\theta^i r_t^i \cdot \mathbb{1}[a_t^i \neq \text{reject}]\big) + \lambda \sum_{i=1}^{n} \theta^i r_t^i \cdot \mathbb{1}[a_t^i \neq \text{reject}] \\ &= \sum_{i=1}^{n} (1-\alpha_t+\lambda)\theta^i r_t^i \cdot \mathbb{1}[a_t^i \neq \text{reject}]. \end{aligned}$$

Here, the principal observes the rewards $\theta^i r_t^i$ for each agent but cannot infer the agent's underlying type. The coefficient λ plays a central role by modulating the degree of altruism in the principal's behavior. When $\lambda \to 0$, the principal becomes greedy, which in turn leads to a situation where the contracts offered exploit agents and let them learn to collect amounts just barely enough to offset the costs. As λ grows, the principal becomes altruistic to the point of becoming a central planner, who disregards its own wealth and seeks to only maximize the welfare of agents interacting with the environment. The result is thus sensitive to the choice of λ, having risks of the system either converging to unfair, exploitative contracts (low λ) or to overly altruistic behavior that sacrifices the principal's wealth (high λ). This highlights the need for a more nuanced regularization method that embeds fairness more explicitly and robustly into the contract design objective.

5.2 Fairness-based regularization

An alternative approach is to regularize the principal's objective based on the fairness over parties' wealth, rather than overall welfare. Specifically, we suggest penalizing the principal for creating

Table 1: Comparison of training metrics. Standard deviations are provided in the Appendix. *For NoP, metrics were computed without the principal.

	NoP*	Greedy	Fix	Regularized					
				Welfare			Wealth Variance		
λ				1	9	12	0.75	1	1.25
1 - Gini	0.95	0.64	0.95	0.73	0.84	0.87	0.96	**0.99**	0.97
Welfare	**45.7**	8.6	44.9	25.5	32.3	44.3	44.9	45.3	44.3
Rawlsian	**18.3**	-0.3	11.0	1.6	6.8	7.5	12.2	14.7	13.8
AIE	43.4	5.6	42.5	18.7	29.2	38.8	43.1	**45.0**	43.5

inequalities in the accumulated wealth across all parties, thereby encouraging contract offers that promote a more balanced distribution over an episode.

Let $\mathcal{W}_t$ represent the cumulative wealth of all parties up to time t, and let $F(\mathcal{W}_t)$ denote a fairness metric applied to that distribution. The principal fairness-aware rewards become:

$$R_{\text{p}}^{\text{fairness}}(s_t, \mathbf{a}_t, s_{t+1}, b_t) = \sum_{i=1}^{n} \big((1-\alpha_t)\theta^i r_t^i \cdot \mathbb{1}[a_t^i \neq \text{reject}]\big) + \lambda F(\mathcal{W}_t),$$

where the principal's wealth $w^p \in \mathcal{W}_t$ is calculated with its non-regularized rewards R_{p}. The simplest form of fairness endorsing equity would be to just use negative variance in wealth: $F(\mathcal{W}) = -\text{Var}[\mathcal{W}]$. This formulation is simple and computationally tractable but has a notable limitation: variance-based fairness does not account for heterogeneity in agent types. As a result, it may fail to ensure proportional fairness in settings where differences in ability or effort should be reflected in wealth. Similar behavior can be observed when using either the negative Jain's or Gini index as F since both of them also measure equity only.

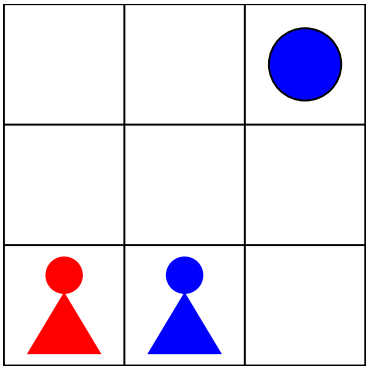
6 Experiments in the Coin Game

The main goal of our experiments is to see if fairness-aware principals can achieve better fairness in heterogeneous agent populations. We inspect, whether a linear homogenous contract, conditioned only on the current observation, can arrive at a stable state with equal wealth allocation. To do so, we compare wealth Variance Regularization (VR) against 4 baselines:

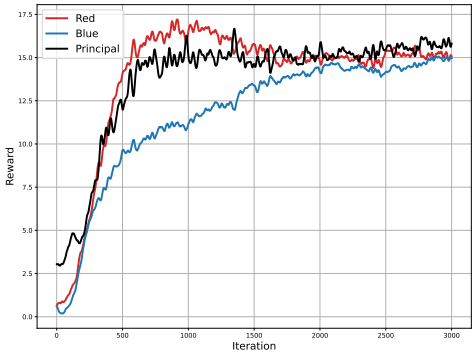
1. No Principal (NoP): equivalent to a principal with fixed linear contracts $\alpha = 1$.
2. Greedy Principal (Greedy): the principal has no form of regularization.
3. Fixed contracts (Fix): We fix linear contracts to an arbitrary constant value.
4. Welfare Regularization (WR).

Additional experimental results and details are provided in the Appendix.

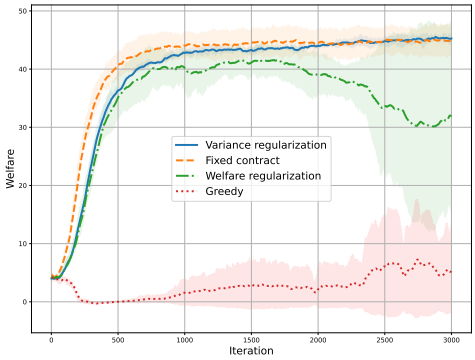
Coin Game. We consider a modified version of the Coin Game [20], a sequential social dilemma that considers two agents, red and blue, who move around a square grid (see Figure 1a). Their goal is to collect a coin, which can also be red or blue. The coin is always present on one of the unoccupied grid spots. Upon collection, a new coin is generated randomly on the grid. An agent (excluding its type) is rewarded with one point if they collect a coin matching their color. If they collect a coin not matching their color, they are still rewarded with 0.2 points. Any ties are broken at random. Selfish agents will give no attention to the wealth of another agent, seeking to maximize their own. It is detrimental to the welfare of the entire system, with less skilled agents having fewer incentives to move and stagnate. For the Coin Game to be a principal-agent game, we introduce an additional third player who doesn't move on the board and assumes the role of the principal, as described in Section 3, and allow agents to refuse contracts as an additional action. If a player refuses a contract, it does not move in the grid.



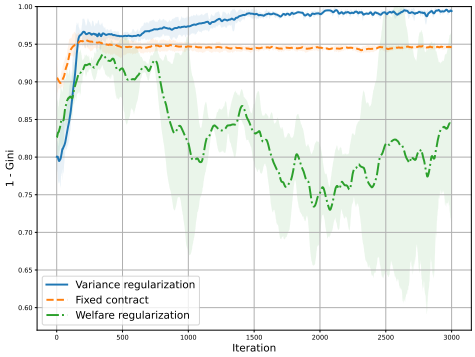
(a) Example grid of the coin game.



(b) Example wealth learning curve over parties under wealth variance regularization ($\lambda = 1$).



(c) Welfare learning curve.



(d) $1 -$ Gini fairness index learning curve.

Figure 1: Results on the coin game. Standard deviations are computed over three runs and given in shaded color.

Experimental Setup. Due to reward sparsity on larger grids, we use a 3×3 board large enough to avoid agents moving randomly and collecting a coin that is almost always on the adjacent cell, yet small enough to arrive at equilibrium policies in reasonable time. Aligning with Section 4.2, principals always learn linear contracts with shares within $[0, 1]$. We fix agents had fixed types $(\theta_{\text{red}}, \theta_{\text{blue}}) = (1.25, 0.75)$. In all of our experiments and aligning with Section 4.2, principals always learn linear contracts with shares within $[0, 1]$. The cost for acting is set to $c = 0.01$. For the Fix baseline, we select a constant contract share of $2/3$ based on both preliminary experiments and the intuitive assumption that the principal keeps roughly one-third of the reward, distributing the remainder between the two agents. All policies are learned using Proximal Policy Optimization (PPO).

We validate each formulation on the following metrics over three separate seeds:

1. The $(1 -$ Gini$)$ index, which maps equality in wealth to the range $[0, 1]$, from most unequal to most equal. Denoting μ the mean of the parties' wealth, and n the number of players in the game:

$$1 - \text{Gini} = 1 - \frac{1}{2n^2\mu} \sum_{i}^{n} \sum_{j}^{n} |w_i - w_j|,$$

2. Welfare, or the total wealth sum over parties: $\sum \mathcal{W}$.
3. The Rawlsian index, or the wealth of the poorest agent : $F(\mathcal{W}) = \min \mathcal{W}$.
4. The product of the $(1 -$ Gini$)$ index and welfare, a metric used in the AI economist [37], which we call AIE.

Results. Results are presented in Table 1 and Figure 1. VR outperforms all other benchmarks, achieving consistently near maximal $1 -$ Gini score, while achieving very high welfare at the same time. Moreover, the Rawlsian metric is the highest amongst the benchmarks with three players (NoP consists of two agents only). While WR achieves good results, it is quite unstable, not presenting the convergent behavior Fix and VR have.

The learning curves of selected algorithms (Greedy, Fix, WR with $\lambda = 9$, VR with $\lambda = 1$) can be seen in Figure 1. The wealth plot of VR (Figure 1b) shows how the principal manages to achieve a fair result, even though in the beginning the welfare was spread unequally. The welfare plot in Figure 1c presents welfare performance during training. Both Fix and VR manage to achieve similar welfare, while WR gets unstable in the late stages of training. Greedy achieves the lowest welfare by far, confirming our hypothesis that selfish principals create an inefficient system. $1 -$ Gini plot in Figure 1d showcases how the particular algorithms learn equality. VR achieves a near-maximal score consistently across all runs. Fix arrives at an equilibrium quite fast, but maxes out in terms of fairness at 0.95, while WR stands out from both of them, presenting the same unstable behavior as in the Welfare plot. We have omitted the $1 -$ Gini score for Greedy due to its erratic behavior.

7 Conclusion and Future Work

We have proposed a framework for designing fair contracts in SSD games with heterogeneous agents and verified the results empirically. Crucially, we have shown that simple linear fairness-aware contracts are able to induce a cooperative behavior between agents, ensuring the fairness of the entire system, without sacrificing welfare in comparison to the baselines.

Limitations. Our experiments are limited to a simple domain, which may not capture the complexities of real-world settings. Additionally, while linear contracts offer simplicity and interpretability, it is not necessarily the case for learned contract policies, which remain difficult to analyze.

Future Work. A natural extension of this work involves exploring more expressive contract forms, where payments can be conditioned on richer outcome spaces. While potentially harder to interpret, such contracts may offer greater flexibility and effectiveness in steering the system toward desired outcomes. Another promising direction is to consider deceiving agents who might try to game the contracts offered by the principal, and how such behavior could be counteracted. Finally, it would be interesting to explore how adaptive contracts could be leveraged when the principal's goal is to allocate wealth according to an arbitrary target distribution rather than ensuring equity.

Acknowledgments

This research was partially supported by Swiss National Science Foundation (grant no. 219499).

References

[1] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-agent reinforcement learning: foundations and modern approaches*. The MIT Press, Cambridge, Massachusetts, 2024.

[2] Tal Alon, Paul Dütting, Yingkai Li, and Inbal Talgam-Cohen. Bayesian Analysis of Linear Contracts, July 2023. arXiv:2211.06850 [cs, econ].

[3] Tal Alon, Paul Dütting, and Inbal Talgam-Cohen. Contracts with Private Cost per Unit-of-Effort, November 2021. arXiv:2111.09179 [cs].

[4] Moshe Babaioff, Michal Feldman, and Noam Nisan. Combinatorial agency. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 18–28, Ann Arbor Michigan USA, June 2006. ACM.

[5] Tobias Baumann, Thore Graepel, and John Shawe-Taylor. Adaptive Mechanism Design: Learning to Promote Cooperation, November 2019. arXiv:1806.04067 [cs].

[6] Patrick Bolton and Mathias Dewatripont. *Contract theory*. MIT Press, Cambridge, MA Londin, England, 2005.

[7] Felix Brandt. *Handbook of Computational Social Choice*. Cambridge University Press, Cambridge, 1st ed edition, 2016.

[8] Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Designing Menus of Contracts Efficiently: The Power of Randomization, August 2022. arXiv:2202.10966 [cs].

[9] Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Multi-Agent Contract Design: How to Commission Multiple Agents with Individual Outcomes. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pages 412–448, London United Kingdom, July 2023. ACM.

[10] Panayiotis Danassis, Aris Filos-Ratsikas, Haipeng Chen, Milind Tambe, and Boi Faltings. AI-driven Prices for Externalities and Sustainability in Production Markets, January 2023. arXiv:2106.06060 [cs].

[11] Ilgin Dogan, Zuo-Jun Max Shen, and Anil Aswani. Estimating and Incentivizing Imperfect-Knowledge Agents with Hidden Rewards, August 2023. arXiv:2308.06717 [cs, stat].

[12] Ilgin Dogan, Zuo-Jun Max Shen, and Anil Aswani. Repeated Principal-Agent Games with Unobserved Agent Rewards and Perfect-Knowledge Agents, May 2023. arXiv:2304.07407 [cs, stat].

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness Through Awareness, November 2011. arXiv:1104.3913 [cs].

[14] Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. Multi-Agent Contracts, November 2022. arXiv:2211.05434 [cs].

[15] Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. Multi-Agent Combinatorial Contracts, May 2024. arXiv:2405.08260 [cs].

[16] Paul Dütting, Michal Feldman, and Inbal Talgam-Cohen. Algorithmic Contract Theory: A Survey, December 2024. arXiv:2412.16384 [cs].

[17] Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. Simple versus Optimal Contracts. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 369–387, Phoenix AZ USA, June 2019. ACM.

[18] Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. The Complexity of Contracts, February 2020. arXiv:2002.12034 [cs].

[19] Ernst Fehr, Alexander Klein, and Klaus M Schmidt. Fairness and Contract Design. *Econometrica*, 75(1):121–154, January 2007.

[20] Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with Opponent-Learning Awareness, September 2018. arXiv:1709.04326 [cs].

[21] Guru Guruganesh, Yoav Kolumbus, Jon Schneider, Inbal Talgam-Cohen, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Joshua R. Wang, and S. Matthew Weinberg. Contracting with a Learning Agent, January 2024. arXiv:2401.16198 [cs, econ].

[22] Guru Guruganesh, Jon Schneider, and Joshua R. Wang. Contracts under Moral Hazard and Adverse Selection. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 563–582, Budapest Hungary, July 2021. ACM.

[23] Bengt Holmstrom and Paul Milgrom. Aggregation and Linearity in the Provision of Intertemporal Incentives. *Econometrica*, 55(2):303, March 1987.

[24] Dom Huh and Prasant Mohapatra. Multi-agent Reinforcement Learning: A Comprehensive Survey, July 2024. arXiv:2312.10256 [cs].

[25] Dima Ivanov, Paul Dütting, Inbal Talgam-Cohen, Tonghan Wang, and David C. Parkes. Principal-Agent Reinforcement Learning, July 2024. arXiv:2407.18074 [cs].

[26] Jiechuan Jiang and Zongqing Lu. Learning Fairness in Multi-Agent Systems, October 2019. arXiv:1910.14472 [cs, stat].

[27] Peizhong Ju, Arnob Ghosh, and Ness B. Shroff. Achieving Fairness in Multi-Agent Markov Decision Processes Using Reinforcement Learning, June 2023. arXiv:2306.00324 [cs].

[28] Jean-Jacques Laffont and David Martimort. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press, December 2009.

[29] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent Reinforcement Learning in Sequential Social Dilemmas, February 2017. arXiv:1702.03037 [cs].

[30] Noam Nisan, editor. *Algorithmic game theory*. Cambridge University Press, Cambridge ; New York, 2007. OCLC: ocn122526907.

[31] Dario Paccagnan, Rahul Chandan, and Jason R. Marden. Utility and mechanism design in multi-agent systems: An overview. *Annual Reviews in Control*, 53:315–328, 2022.

[32] Bernard Salanié. *The economics of contracts: a primer*. The MIT Press, Cambridge, Massachusetts London, England, second edition edition, 2005.

[33] Antoine Scheid, Daniil Tiapkin, Etienne Boursier, Aymeric Capitaine, El Mahdi El Mhamdi, Eric Moulines, Michael I. Jordan, and Alain Durmus. Incentivized Learning in Principal-Agent Bandit Games, March 2024. arXiv:2403.03811 [cs, stat].

[34] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. Learning to Incentivize Other Learning Agents, October 2020. arXiv:2006.06051 [cs].

[35] Jiachen Yang, Ethan Wang, Rakshit Trivedi, Tuo Zhao, and Hongyuan Zha. Adaptive Incentive Design with Multi-Agent Meta-Gradient Reinforcement Learning, December 2021. arXiv:2112.10859 [cs].

[36] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms, April 2021. arXiv:1911.10635 [cs].

[37] Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C. Parkes, and Richard Socher. The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies, April 2020. arXiv:2004.13332 [econ].

A Additional results

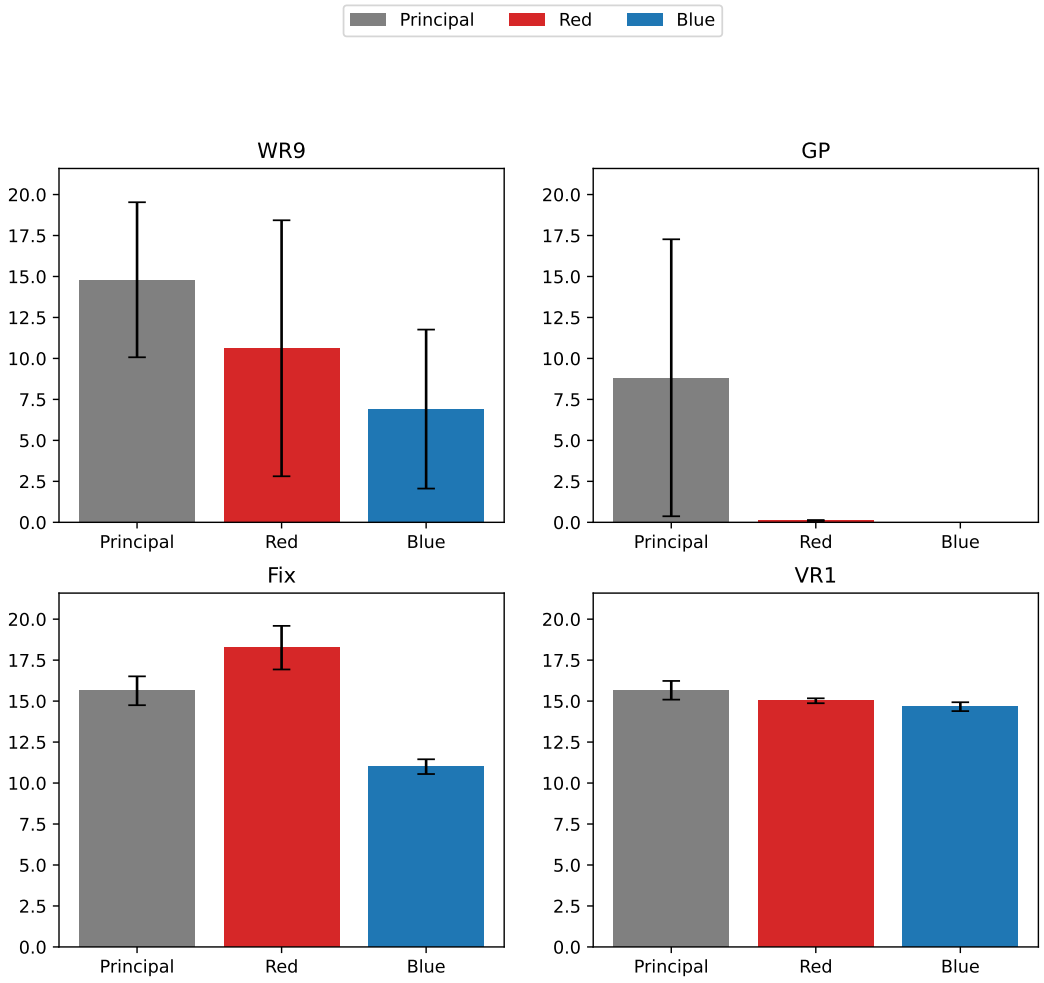
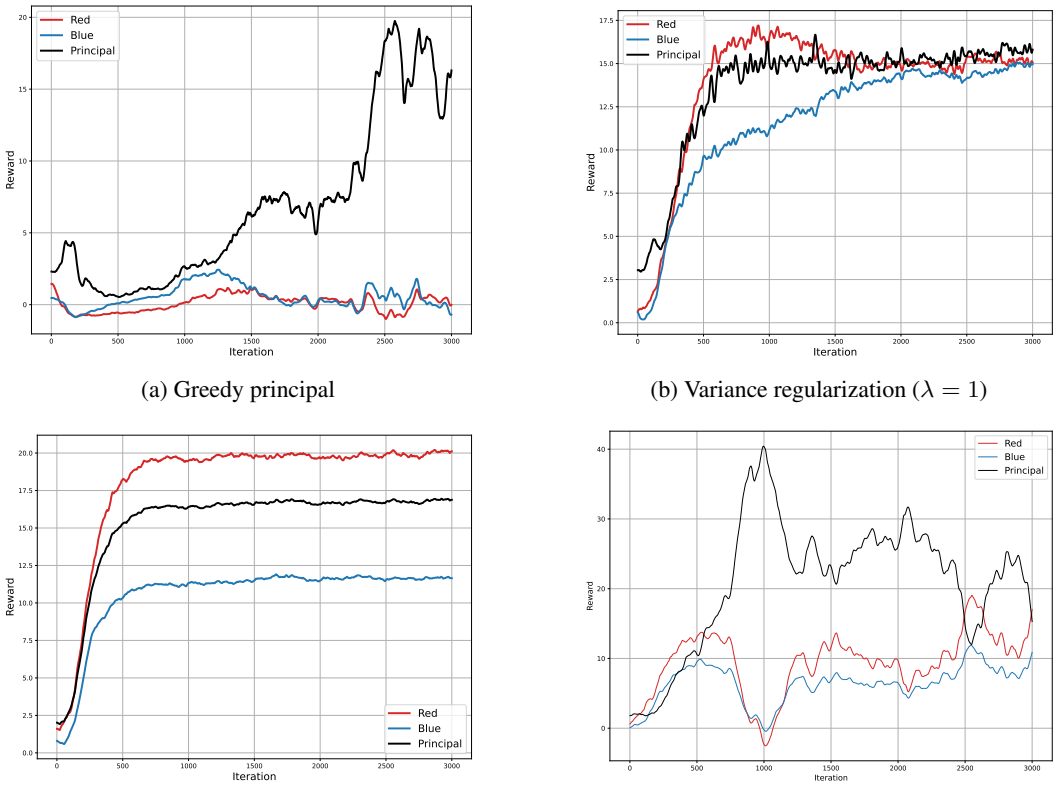


Figure 2: The mean final spread of welfare among principal and agents, with wiskers indicating the standard deviation. Greedy principal exploits agents, while at the same time achieving suboptimal wealth. Welfare based regularization and fixed contracts result in disproportions between agents. Variance based regularization results in almost perfect split of wealth.



(a) Greedy principal

(b) Variance regularization ($\lambda = 1$)

(c) Fixed contract $2/3$

(d) Welfare regularization ($\lambda = 9$)

Figure 3: Comparison of mean wealth achieved by players over the course of the training

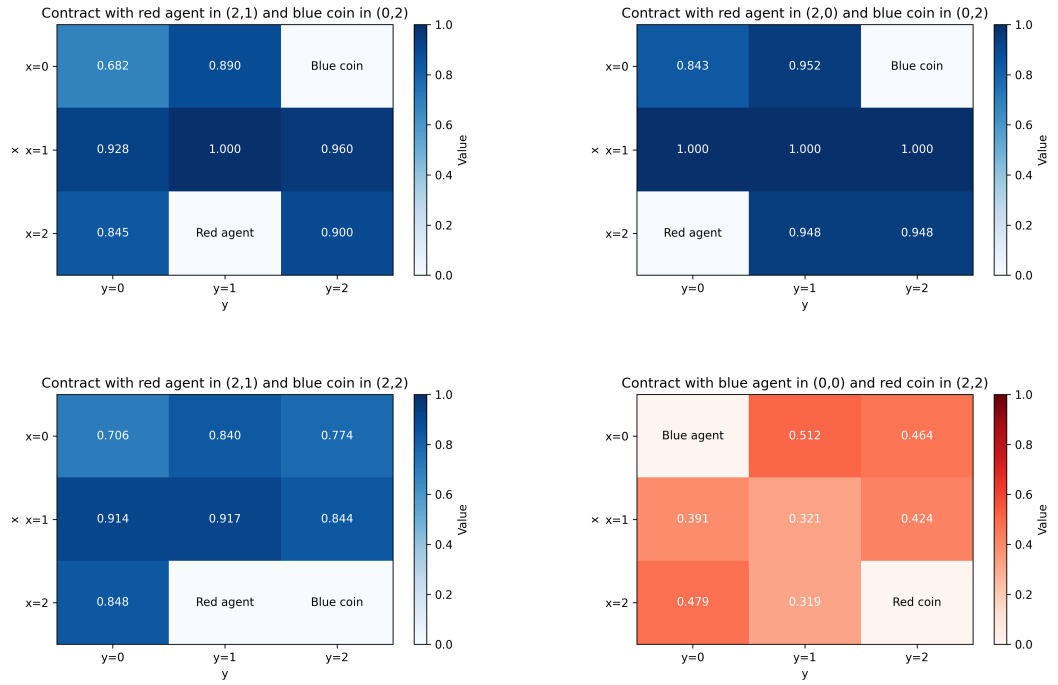


Figure 4: Means of contracts of the policy learned by the principal with wealth variance regularization and $\lambda = 1$.

Table 2: Detailed comparison of training metrics

	NoP	Greedy	Fix	Regularized					
				Welfare			Wealth Variance		
λ				1	9	12	0.75	1	1.25
1 - Gini	$0.95_{\pm 0.02}$	$0.64_{\pm 0.01}$	$0.95_{\pm 0.01}$	$0.73_{\pm 0.06}$	$0.84_{\pm 0.13}$	$0.87_{\pm 0.02}$	$0.96_{\pm 0.02}$	**0.99**$_{\pm 0.01}$	$0.97_{\pm 0.01}$
Welfare	**45.7**$_{\pm 2.8}$	$8.6_{\pm 8.1}$	$44.9_{\pm 2.7}$	$25.5_{\pm 2.0}$	$32.3_{\pm 16.1}$	$44.3_{\pm 2.5}$	$44.9_{\pm 2.9}$	$45.3_{\pm 0.9}$	$44.3_{\pm 4.2}$
Rawlsian	**18.3**$_{\pm 0.8}$	$-0.3_{\pm 0.3}$	$11.0_{\pm 0.4}$	$1.6_{\pm 1.7}$	$6.8_{\pm 5.0}$	$7.5_{\pm 2.4}$	$12.2_{\pm 0.9}$	$14.7_{\pm 0.3}$	$13.8_{\pm 1.8}$
AIE	$43.4_{\pm 1.9}$	$5.6_{\pm 5.2}$	$42.5_{\pm 2.4}$	$18.7_{\pm 3.0}$	$29.2_{\pm 16.2}$	$38.8_{\pm 4.2}$	$43.1_{\pm 1.4}$	**45.0**$_{\pm 0.8}$	$43.5_{\pm 4.6}$

B Training details

Table 3: PPO training parameters

Parameter	Principal	Agent
Episode length	100	
Grid size	3×3	
Batch size	1600	
Learning rate	2×10^{-4}	5×10^{-4}
Entropy cost	10^{-5}	10^{-3}
GAE λ	0.95	
KL β_0	1	0
KL target d_{targ}	10^{-2}	
Clipping ϵ	0.2	
Baseline coefficient	0.5	

Compute. All the experiments were performed on a PC with 32GB RAM memory, AMD Ryzen 5 2600X CPU with 6 cores and 3.6 GHz frequency, and NVIDIA GeForce RTX 3060 GPU with 12GB VRAM. Rollouts were parallelized on CPU using the open source Python package Ray, while PPO has been implemented from scratch, using Tensorflow 2.19 framework for deep learning. Parameters were unchaged across all experiments, and are listed in the Table 3. Single experiment took on average 3 hours to complete.