Learning to Participate through Trading of Reward Shares

Kyrill Schmid
LMU Munich
Munich, Germany
kyrill.schmid@ifi.lmu.de

Michael Kölle
LMU Munich
Munich, Germany
michael.koelle@ifi.lmu.de

Tim Matheis
LMU Munich
Munich, Germany
tim.matheis@campus.lmu.de

ABSTRACT

Enabling autonomous agents to act cooperatively is an important step to integrate artificial intelligence in our daily lives. While some methods seek to stimulate cooperation by letting agents give rewards to others, in this paper we propose a method where agents have the opportunity to participate in other agents' returns by acquiring shares. Intuitively, an agent may learn to act according to the common interest when being directly affected by the other agents' rewards. The empirical results of the tested general-sum Markov games show that this mechanism promotes cooperative policies among independently trained agents in social dilemma situations. Moreover, as demonstrated in a temporally and spatially extended domain, participation can lead to the development of roles and the division of subtasks between the agents.

KEYWORDS

Multi-Agent Systems, Reinforcement Learning, Social Dilemma

1 INTRODUCTION

The field of cooperative AI seeks to explore methods which establish cooperative behavior among independent and autonomous agents [1]. The ability to act cooperatively is a mandatory step in order to integrate artificial intelligence in our daily lives especially in applications where different decision makers interact like autonomous driving. Various breakthroughs in the field of single agent domains [12, 20] have also led to the successful application of reinforcement learning in the field of multi-agent systems [8, 15, 22]. However, while purely cooperative scenarios, where all agents receive the same reward and thus pursue the same goal, can be addressed with centralized training techniques, this is not the case if agents have individual rewards and goals. Moreover, if agents share resources it is likely that undesired behaviors are learned especially when resources are getting scarce [8]. Independent optimization may lead to sub-optimal outcomes such as in the Prisoner's dilemma or public good games. In more complex games, the agents' risk-aversion as well as information asymmetry additionally deteriorate the likelihood of a desired outcome [19].

In recent years, various approaches have been proposed to promote cooperation among independent agents, such as learning proven game theoretic strategies like tit-for-tat [9], the possibility for agents to incentivize each other to be more cooperative [10, 16, 24], or the integration of markets to let agents trade for increased overall welfare [18]. In this work, we adopt the market concept in order to generate increased cooperation between independent decision makers. More specifically, we propose a method that allows agents to trade shares of their own rewards. In the presence of such a participation market, we argue that a better equilibrium can be reached by letting agents directly participate in other agents' rewards. A socially optimal equilibrium may be established due to the direct incorporation of all global rewards instead of only incorporating individual rewards. Enabling agents to trade their shares at a fair price, while at the same time creating a trading path that in fact leads to a beneficial distribution of shares, is the most challenging implementation in this model.

	C	D
C	(-1, -1)	(-3, 0)
D	(-3, 0)	(-2, -2)

(a) PD without particpation

	C	D
C	(-1, -1)	(-1.5,-1.5)
D	(-1.5,-1.5)	(-2, -2)

(b) PD with 50% particpation

Figure 1: By enabling agents to trade shares in their payoffs, a socially optimal outcome may be achieved.

To motivate the effectiveness of participation, consider the Prisoner's dilemma (PD) as depicted in Figure 1. In the standard version of the Prisoner's dilemma, each agent receives its individual reward. For two rational decision makers, the only Nash equilibrium lies in mutual defection, which is socially undesirable. However, if each agent holds shares in the other agent's return (say 50% participation), then the overall dynamic changes and mutual cooperation becomes a dominant strategy. In order to let agents learn to participate, here we apply different methods of reinforcement learning, so that the agents autonomously find strategies. Thus, participation in other agents' returns is realized over the course of many training episodes. To that end, different variants of the *market for participation* are tested in this work, which differ in the way the participation mechanism is implemented. More specifically, the following contributions are made:

- A theoretical motivation is given why market participation is beneficial.
- The participation mechanism is empirically evaluated in the Prisoner's dilemma as well as in a complex multi-agent scenario, called the clean-up game [24].

All code for the experiments can be found here [1].

2 RELATED WORK

Despite the increasing success of reinforcement learning on an expanding set of tasks, most effort has been devoted to single-agent environments as well as fully cooperative multi-agent environments [12, 13, 20, 22]. However, with multiple agents involved their goals are often not (perfectly) aligned, which renders centralized

Proc. of the Adaptive and Learning Agents Workshop (ALA 2022), Cruz, Hayes, da Silva, Santos (eds.), May 9-10, 2022, Online, https://ala2022.github.io/. 2022.

[1] https://github.com/TimMatheis/Learning-to-participate

training techniques in general unfeasible. The drawback of fully decentralized models is that agents focus only on their individual rewards, which therefore might result in undesirable collective performance especially in situations of social dilemmas or with common pool resources [8, 14].

Previously, one way of tackling this problem has been to give independent agents intrinsic rewards [2, 5, 23]. The concept of intrinsic rewards draws from concepts in behavioral economics such as altruistic behavior, reciprocity, inequity aversion, social influence, and peer evaluations. These intrinsic rewards are usually either predefined or they evolve based on the other agents' performance over the game. Other works suggest that reward mechanisms [24] or penalty mechanisms [17] may lead to cooperation in sequential social dilemmas. The literature distinguishes between selective incentives and sanctioning mechanisms which incentivize cooperative behavior in social dilemmas [6]. Selective incentives describe methods that attempt to positively promote cooperation. For instance, this could occur by giving monetary rewards to reduce the consumption of common pool goods, such as water or electricity [11]. Contrarily, penalties could be a method to reduce defective behavior. In fact, experiments with humans suggest that penalties are effective in reducing defective behavior [7].

In *LIO* [24], a reward-giver's incentive function is learned on the same timescale as policy learning. Adding an incentive function is a deviation from classical reinforcement learning, where the reward function is the exclusive property of the environment, and is only altered by external factors. As shown by empirical research [10], augmenting an agent's action space with a "give-reward" action can improve cooperation during certain training phases. Through opponent shaping, an agent can influence the learning update of other agents for its own benefit. A different attempt at opponent shaping is to account for the impact of one agent's policy on the anticipated parameter update of the other agents [4]. Through this additional learning component in *LOLA*, strategies like tit-for-tat can emerge in the iterated Prisoner's dilemma, whereby cooperation can be maintained.

Other work in the field suggests markets as vehicles for cooperativeness [16, 18]. Usually, agents only receive their individual rewards. As they are not affected by the other agents' individual rewards, they only act in their own interest. However, by only receiving individual rewards, the agents are exposed to substantial risk. According to economic theory, it is usually beneficial to be diversified. In portfolio theory, there is a common agreement that diversification increases expected returns. Although people do not seem to diversify enough, which is called the *diversification puzzle* [21], a rational agent should be perfectly diversified and should only hold a combination of the market portfolio and a risk-free asset (such as a safe government bond). When applying this to games such as the Prisoner's dilemma, the agents should be interested in receiving a combination of their individual rewards and the other agents' individual rewards to minimize their risk exposure.

3 LEARNING TO PARTICIPATE

Multi-agent systems consist of multiple agents that share a common environment. An agent is an autonomous entity with two main capabilities: perceiving and acting. The perception of the current state of the environment allows the agent to choose an appropriate action out of an action set. The chosen action depends on an agent's policy. Reinforcement learning methods are often applied to teach an agent a good policy in multi-agent reinforcement learning.

Since cooperative strategies can be difficult to find and maintain, we suggest a participation mechanism. The idea is to let agents participate in other agents' environmental rewards directly in order to align their formerly conflicting goals. In the following, we use the trading of participation shares as an instrument to make cooperation possible. If the agents are willing to hold a significant amount of shares in all agents' rewards, they may act in the "society's interest". Namely, the difference of the individual interest of a single agent and the common interest of the collective of all agents could vanish. In our implementation, two agents only trade shares whenever both choose to increase or decrease the amount of their own shares. In the initial state, agents hold 100% of their own shares. If the agents want to be perfectly diversified, each agent could have $\frac{100\%}{n}$ shares, with n denoting the number of agents in the environment, of every agent's rewards after some trading steps.

3.1 Theoretic motivation of the participation market

To motivate why the trading of participation shares should enable cooperation among multiple agents, we adapt the proof for the Prisoner's dilemma with two agents from *LIO* ([24] Appendix B).

In a two-agent Prisoner's dilemma such as in Figure 1, both agents initially own 100% of their own rewards. At that point, they would only consider their own rewards when choosing actions. However, if there is a market that enables them to trade their shares and trading improves their payoffs, we can assume that the agents would trade their shares. If we think about a symmetric framework with homogeneous agents, it seems likely that the trading of shares could lead to a steady state in which both agents own 50% of the shares in both payoffs. This would lead to a direct consideration of their own actions on all agents in the model.

The goal in the Prisoner's dilemma is to maximize the resulting overall rewards r. Since the rewards are not deterministic (they also depend on the other agent's actions), p is used as a probability vector, whereby θ^1 represents the probability that agent 1 cooperates and $(1-\theta^1)$ that agent 1 defects. This applies symmetrically to agent 2. For instance, $(1-\theta^1)(\theta^2)$ is the probability that agent 1 defects and agent 2 cooperates. Additionally, rewards are increasingly discounted via the factor $0 < \gamma < 1$. The discounted expected rewards are then summed up. The geometric sum is applied for simplifying the term:

$$V^i(\theta^1, \theta^2) = \sum_{t=0}^{\infty} \gamma^t p^T r^i = \frac{1}{1-\gamma} p^T r^i$$

where $p = \left[\theta^1\theta^2, \theta^1(1-\theta^2), (1-\theta^1)\theta^2, (1-\theta^1)(1-\theta^2)\right]$. In the following, the two agents may flatten their rewards and decrease the volatility of the received rewards by exchanging their own shares for shares of the other agent. Here, m denotes the share in agent 1's reward (1 sells a share in its"business", 2 buys the share). Symmetrically, n denotes the share in agent 2's reward (2 sells a share in its "business", 1 buys the share). Including the shares, the following total reward vectors reflect the rewards per step. The

rewards are again vectors, that are connected to the actions of the agents. The order of the reward vector entries is analogue to the probabilities: CC, CD, CD, DD, where D stands for defection and C for cooperation. For instance, agent 1 receives the reward $-2(1-m)+(-2)n$ if both agents defect. A reward consists of two components – the reward from the own shares, and the reward from the other agent's shares. For example, if both agents cooperate, agent 1 receives a reward of $-1(1-m)$ due to its own shares and a reward of $-1n$ due to its shares from the other agent.

$$\begin{aligned} r^1 &= [-1(1-m)+(-1)n, -3(1-m)+0n, \\ &\quad 0(1-m)+(-3)n, -2(1-m)+(-2)n] \\ &= [-1+m-n, -3+3m, -3n, -2+2m-2n] \end{aligned}$$

$$\begin{aligned} r^2 &= [-1(1-n)+(-1)m, 0(1-n)+(-3)m, \\ &\quad -3(1-n)+0m, -2(1-n)+(-2)m] \\ &= [-1-m+n, -3m, -3+3n, -2-2m+2n] \end{aligned}$$

Both agents iteratively update their policy after one or multiple periods. The updating of agent 2's policy can be described by the following equation. α is the learning rate determining how much the policy is updated.

$$\begin{aligned} \hat{\theta}^2 &= \theta^2 + \alpha \nabla_{\theta^2} V^2(\theta^1, \theta^2) \\ &= \theta^2 + \frac{\alpha}{1-\gamma} \nabla_{\theta^2} [\theta^1\theta^2(-1-m+n)+ \\ &\quad + \theta^1(1-\theta^2)(-3m) + (1-\theta^1)\theta^2(-3+3n) \\ &\quad + (1-\theta^1)(1-\theta^2)(-2-2m+2n)] \\ &= \theta^2 + \frac{\alpha}{1-\gamma} \left[\theta^1(2m+n-1) + (1-\theta^1)(2m+n-1)\right] \\ &= \theta^2 + \frac{\alpha}{1-\gamma} [2m+n-1] \end{aligned}$$

Due to symmetry, the policy update for agent 1 is analogue.

$$\hat{\theta}^1 = \theta^1 + \frac{\alpha}{1-\gamma} [2n+m-1]$$

Agent 1 and 2 update their shares via the following equation. $\hat{p}$ is the joint action probability under updated policies $\hat{\theta}^1$ and $\hat{\theta}^2$.

$$m \leftarrow min \Big\{ \underbrace{m + \beta \nabla_m \frac{1}{1-\gamma} \hat{p}^T r^1}_{\text{agent 1 willing to sell } m}; \quad \underbrace{m + \beta \nabla_m \frac{1}{1-\gamma} \hat{p}^T r^2}_{\text{agent 2 willing to buy } m} \Big\}$$

$$n \leftarrow min \Big\{ \underbrace{n + \beta \nabla_n \frac{1}{1-\gamma} \hat{p}^T r^1}_{\text{agent 1 willing to buy } n}; \quad \underbrace{n + \beta \nabla_n \frac{1}{1-\gamma} \hat{p}^T r^2}_{\text{agent 2 willing to sell } n} \Big\}$$

Importantly, in this theoretical model, we assume that agent 1 can only sell m and buy n (symmetric for agent 2). This is only necessarily true in the initial state ($m = n = 0$). But we want the agents to only sell their own shares and not buy them back as well as buy the other agent's share and not sell them afterwards. For creating a situation in which trading shares to the other party is beneficial (expected rewards must increase), we need a price setting mechanism. Otherwise, in the initial state ($m = n = 0$), the agents would not trade their shares. Moreover, their dominant strategies do not lead to an optimal equilibrium. However, if share prices are used to equally distribute the shares, a new "socially optimal" equilibrium may be reached. When the equilibrium is reached, the price setting mechanism may not be needed anymore. If the shares are evenly distributed, the agents act in the complete market's interest since their rewards co-move to 100% with the market.

As a starting point, the initial state, in which all agents only have their own shares ($m = n = 0$), needs to be analyzed.

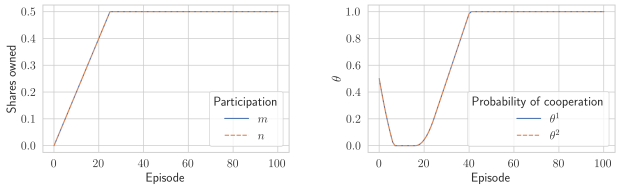
$$\begin{aligned} & m \leftarrow min \left\{ m + \beta \nabla_m \frac{1}{1-\gamma} \hat{p}^T r^1; \quad m + \beta \nabla_m \frac{1}{1-\gamma} \hat{p}^T r^2 \right\} \\ \Rightarrow\ & m \leftarrow min \left\{ 0 + \beta \nabla_m \frac{1}{1-\gamma} \hat{p}^T r^1; \quad 0 + \beta \nabla_m \frac{1}{1-\gamma} \hat{p}^T r^2 \right\} \\ \Rightarrow\ & m \leftarrow min \Big\{ \beta \frac{1}{1-\gamma} [\theta^1\theta^2 + \theta^1(1-\theta^2)3 + (1-\theta^1)(1-\theta^2)2]; \\ & \qquad \beta \frac{1}{1-\gamma} \left(-[\theta^1\theta^2 + \theta^1(1-\theta^2)3 + (1-\theta^1)(1-\theta^2)2] \right) \Big\} \end{aligned}$$

Thus, agent 2 would not even accept an additional (first) share m if it was free of cost ($-[\theta^1\theta^2 + \theta^1(1-\theta^2)3 + (1-\theta^1)(1-\theta^2)2] < 0$). This makes sense because a share in agent 1's rewards only leads to participation in negative rewards. On the other side, agent 1 would be more than happy to sell shares of its own rewards ($[\theta^1\theta^2 + \theta^1(1-\theta^2)3 + (1-\theta^1)(1-\theta^2)2] > 0$).

A broker (let us assume without any profit incentives and perfect information) could set the price $p_m = -\frac{1}{1-\gamma}[\theta^1\theta^2 + \theta^1(1-\theta^2)3 + (1-\theta^1)(1-\theta^2)2] < 0$. Under the assumption that agents trade if they are indifferent, they trade Δm. Then, a new price is set such that the agents are again indifferent and trade again.

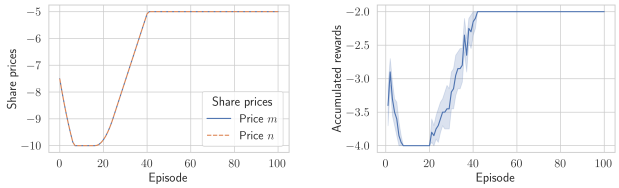
$$\begin{aligned} & m \leftarrow min \left\{ \beta \nabla_m \left(\frac{1}{1-\gamma} \hat{p}^T r^1 + \Delta m * p_m \right); \right. \\ & \qquad \left. \beta \nabla_m \left(\frac{1}{1-\gamma} \hat{p}^T r^2 - \Delta m * p_m \right) \right\} \\ & m \leftarrow min \Big\{ \beta \Big(\frac{1}{1-\gamma} [\Delta m * p_m + \theta^1\theta^2 \\ & \qquad + \theta^1(1-\theta^2)3 + (1-\theta^1)(1-\theta^2)2] + p_m \Big); \\ & \qquad \beta \Big(\frac{1}{1-\gamma} \Big(-[\theta^1\theta^2 + \theta^1(1-\theta^2)3 + \\ & \qquad (1-\theta^1)(1-\theta^2)2] \Big) - p_m \Big) \Big\} \end{aligned}$$

The numerical simulations in Figure 2 test the previous equations. They indicate that the participation share market can indeed be a cooperation enabler. Over 100 episodes, a stable cooperative equilibrium is reached. For the plotting of the accumulative rewards – the sum of both agents' rewards – we use 20 runs. Importantly, instead of directly applying reinforcement learning, we only test the interplay of the previous equations. However, a smart reinforcement learner works very similar. Hence, the successful cooperation in the numerical runs implies that a reinforcement learning model that makes use of participation and the share market, should also enable cooperation.



(a) The participation in the other agent's rewards increases to the capped proportion of 0.5.

(b) Once the participation is high enough, the likelihood of participation rises to 100%.



(c) The price of the shares is directly determined by the probability of cooperation.

(d) With sufficient participation, the sum of total reward converge to the optimum.

Figure 2: Simulations of the Iterated Prisoner's dilemma with participation.

4 EXPERIMENTS

In this section, we examine our theoretical considerations in the iterated prisoner's dilemma and the clean-up game experimentally.

4.1 The Iterated Prisoner's Dilemma

The prisoner's dilemma can be thought of as the action of two burglars. When they get caught, they can decide between $a_0 = admitting$ and $a_1 = notadmitting$ a crime. If both admit the crime, they receive a high punishment. If none of them admits the crime, they only receive a low punishment. The dilemma evolves from the case in which only one of them admits the crime. Then, the admitter is not punished due to its status as a principal witness, whereas the denier receives a very high punishment. Admitting is the defective (D) action and not admitting is the cooperative (C) action in Figure 1. By definition, none of the agents can put itself in a better position by changing its strategy in a Nash equilibrium. Agent 1 knows that agent 2 can defect and cooperate. When agent 2 defects, agent 1 is better off by defecting as well. If agent 2 cooperates instead, agent 1 is again better off by defecting. Thus, agent 1 should defect in any case, which makes defecting a dominant strategy in a one-shot game. By symmetry, agent 2 faces the same problem and should also defect. The tragedy is that this outcome is not desirable, as mutual cooperation leads to a better payoff for both agents. When the game is iterated multiple times, defecting is in theory not necessarily dominant. Although the agents could defect in every iteration based on backward induction, the agents could develop strategies to incentivize cooperation.

4.2 Participation in the Iterated Prisoner's Dilemma

For testing different implementations of the iterated prisoner's dilemma, we use an actor-critic method. We adapt Algorithm 1 from *LIO* ([24] page 5) by replacing the incentive function and its parameter η with a greater action space during the whole episode or a preliminary trade action. Hence, either the actions specify an environment action as well as a trade action during the whole episode, or there is one additional step at the start of each episode in which the participation is determined, or both.

We use the same fully-connected neural network for function approximation as *LIO*. However, we only use the policy network, as we do not make use of the incentive function, for which *LIO* uses another neural network. The policy network has a softmax output for discrete actions in all environments. For all experiments, we use the same neural architecture. The same hyperparameters were used for all experiments: $\beta = 0.1$, $\epsilon_{\text{start}} = 1.0$, $\epsilon_{\text{end}} = 0.01$, $\alpha_\theta = 1.00E{-}03$. As the agents do not participate in the other agent's rewards at the start, *max steps* is set to 40 instead of 5 in the implementations with trading. Hence, they have enough steps for trading. We test the following implementations of the iterated Prisoner's dilemma with two agents.

(i) No participation: The possible environment actions of each agent are *cooperation* and *defection*. In the implementation, these two actions are encoded as 0 and 1. There are four possible states: (cooperation, cooperation), (cooperation, defection), (defection, cooperation), (defection, defection). If the agents chose their actions randomly, the average individual reward would be $(-1{+}0{-}3{-}2)/4 = -1.5$. If instead the agents learned to maintain a cooperative equilibrium in which both choose *cooperation*, the individual rewards would converge to -1. But according to theory, *defection* is a dominant action in the "classical" Prisoner's dilemma. In that case, the individual rewards would converge to -2. Indeed, in the implementation without participation shares, a stable equilibrium evolves in which both agents *defect*. The accumulated reward converges to $-2 + (-2) = -4$ (Figure 3a).

(ii) Equal distribution of individual rewards: In this implementation, the agents always receive the average reward of all individual shares. Mathematically speaking, this means that all individual rewards are aggregated and then divided by the number of agents: $\frac{1}{n} \times \sum_i^n reward_i$ for n agents. The action space and everything else stays the same. An agent cannot choose between sharing the rewards or receiving the individual reward. Each agent learns to *cooperate*, which leads to an accumulated reward of $-1 + (-1) = -2$. In this implementation, cooperating is a dominant action. This implementation demonstrates that the sharing of rewards can lead to cooperation. However, it does not demonstrate whether the agents can actively find and maintain such an equilibrium, when they can freely trade (Figure 3a).

(iii) Choosing whether to share rewards: The agents can decide to share their individual rewards. The action space is extended. In addition to the two environment actions, an agent can choose between sharing the rewards or receiving the individual reward. The action space is defined by an action tuple: the environment action and the trade action. Hence, there are now $2 \times 2 = 4$

possible actions per agent. Importantly, an individual agent cannot determine whether the combined rewards are evenly divided between both agents. Instead, this is only the case if both agents decide to share their rewards. The state space is extended to eight states: *sharing* $\times$ *env. action 1* $\times$ *env. action 2* $= 2 \times 2 \times 2 = 8$. Both agents learn to *defect*, which leads to an accumulated reward of $-2 + (-2) = -4$ (Figure 3a). In this implementation, defecting without sharing is a dominant action. Hence, the sole opportunity to share rewards is not enough.

(iv) Trading 50% shares: Initially, both agents do not hold participation shares of the other agent. In every step, they can choose between six actions. An action can be regarded as a 3-tuple: the environment action, whether to increase the shares of the own rewards, and whether to increase the shares of the other agent's rewards. When an agent decides to cooperate, there are three possible actions: (cooperate, not buy own shares, not buy other agent's shares), (cooperate, buy own shares, not buy other agent's shares), (cooperate, not buy own shares, buy other agent's shares). Symmetrically, there are three possible actions for defection. Again, a trade of shares is only executed if both agents intend to do so. For instance, if both choose to buy own shares and they hold 50% in each agent's rewards, they exchange shares and now hold 100% of their own shares. This would mean that there is no participation anymore. However, if they now choose to buy own shares again, this trade cannot occur, as they already hold all of their own shares. As a consequence, only the environment action has an effect. The implication is that all shares are valued at the same price, they are just exchanged in the same proportions, and the sum of shares per agent is always 100%. There are 36 states: *env. actions* $\times$ *portion own shares* $\times$ *trade* $= 4 \times 3 \times 3$. The portion of own shares can be 0, 0.5, or 1. The "trade" variable represents whether there is no trade, a trade which leads to an increasing amount of own shares, or a trade which leads to a decreasing amount of own shares. This implementation is successful in establishing cooperation. The accumulated rewards converge to $-1 + (-1) = -2$ (Figure 3b). However, the learning process takes rather long. Additionally, the actions and rewards first drift off to the previous inefficient equilibrium.

(v) Trading 10% shares: Compared to the trading of 50% shares, the state space increases to $4 \times 11 \times 3 = 132$ because the proportion of own shares can now be 0, 0.1, 0.2, ..., 1. Again, a socially optimal equilibrium can be established. A proportion of around 40% own shares and 60% other shares seems to be efficient to make the agents not deviate to defection (Figure 3c).

4.3 The clean-up game

In the clean-up game, multiple agents simultaneously attempt to collect apples in the same environment. An agent gets rewarded +1 for each apple that they collect. The apples spawn randomly on the right side of a quadratic *7x7* or *10x10* map. On the left side of the map, there is a river that gets increasingly polluted with waste. As the waste level increases and approaches a depletion threshold, the apple spawn rate decreases linearly to zero. To avoid a quick end of the game, an agent can fire a *cleaning beam* to clear waste. But an agent can only do so when being in the river. The cleaning beam then clears all the waste upwards from the agent. Each agent's observation is an egocentric RGB image of the whole map. The dilemma is that clearing waste by staying in the river and firing cleaning beams is less attractive than receiving rewards by collecting apples. However, if all agents focus on only collecting apples, the game is quickly over and the total reward of the agents remains very low. Hence, the game is an *intertemporal* social dilemma, in which there is a trade-off between short-term individual incentives and long-term collective interest [5]. This domain is especially interesting as models based on behavioral economics can only explain cooperation in simple, unrealistic, stateless matrix games. In contrast, the cleanup game is a temporally and spatially extended Markov game.

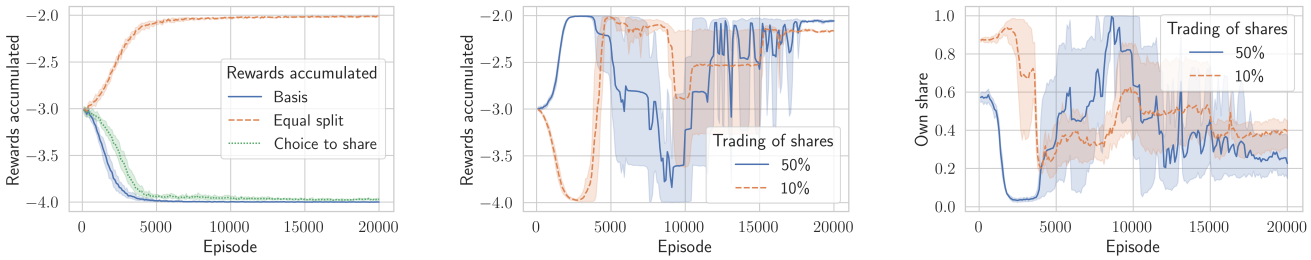
4.4 Participation in the clean-up game with two agents

The smaller *7x7* map is used as described in [24]. The initial state of the game is displayed in Figure 4. The blue cells on the left side of the map represent the river, the green cells indicate the area where apples are randomly spawned, and the brown cells represent the waste. Agent 1 gets spawned in the river, which enables it to clear the waste. Agent 2 gets spawned close to the green area on the right where apples are randomly spawned. This already gives a hint at one possible strategy, which consists of agent 1 clearing the waste, and agent 2 collecting the apples. However, there is no justifying reason for agent 1 to clear waste as long as it does not get the chance to collect apples or profits from agent 2's rewards. Agent 2 is in a similar dilemma. If it attempts to collect apples by being in the green area, it cannot fire the cleaning beam. Without a division of work between the agents, the game must stop early, as the blue and green area are too far away. The possible environment actions of each agent are *move left*, *move right*, *move up*, *move down*, *no operation*, and *cleaning*. In all implementations of the clean-up domain, an actor-critic method is used. The optimization is decentralized.

We use the same convolutional networks to process image observations in Cleanup as *LIO*. The policy network has a softmax output for discrete actions in all environments. For all experiments, We use the same neural architecture. We test the following implementations of the clean-up game with two agents.

(i) No participation: Without any additional incentive structures, the social dilemma seems unsolvable. As depicted in Figure 5a, the accumulated reward remains at a very low level. Cooperation is not attractive for any of the agents. It is difficult to learn that clearing waste is creating value since the apples are spawned far away from the river. If the agent moves back from the river to the apples, the game is likely to stop before it even collects an apple.

(ii) LIO: The baseline scenario is augmented with the possibility for each agent to give rewards to the other agent as an additional channel for cooperation. Importantly, the additional reward payments $r^j_{\eta^i}$ from agent i to agent j is learned via direct gradient ascent on the agent's own extrinsic objective, involving its effect on all other agents' policies. Hence, the payments are not part of the reinforcement learning, leaving the action space of the reinforcement learner unaffected (instead of augmenting it). The agents successfully divide their tasks. The one that is closer to the river and waste becomes the cleaner, whereas the one that is closer to the apples specializes in collecting apples. See [24] for implementation details.



(a) Rewards **(b) Rewards with trading of 10% or 50% shares** **(c) Shares in own reward**

Figure 3: Results from the iterated Prisoner's dilemma with and without participation.

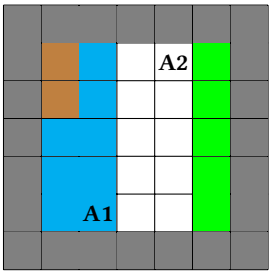


Figure 4: Small *7x7* map with two agents. Agent 1 gets spawned by the river, whereas agent 2 gets spawned close to the apples. If they do not divide their tasks, the game is likely to stop soon, as the waste expands downwards if it is not cleared by the agents.

(iii) Equal distribution of individual rewards: In this scenario, the baseline scenario is augmented with the equal distribution of the joint rewards between the agents after each step. This additional computation does not affect the state or action space. Similar to *LIO*, the one that is closer to the river turns into the cleaner, whereas the one that is closer to the apples specializes in collecting apples. Both profit from this task division, as both are evenly rewarded for any apple being collected by anyone. In comparison with *LIO*, the learning process looks less volatile.

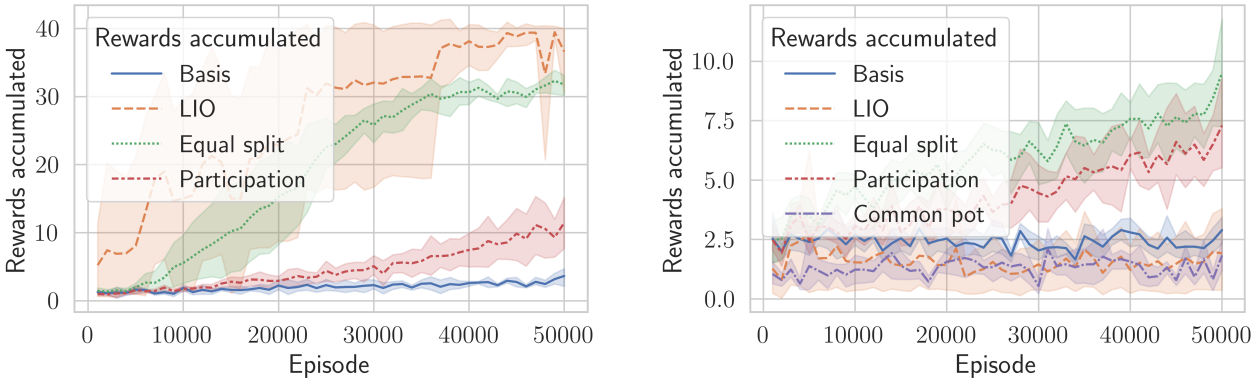
(iv) Pre-trade of participation rights: In this scenario, there is an additional first step added to each episode of the baseline scenario. In this first step, both agents can choose between six actions (0-5), representing 0%, 20%, 40%, 60%, 80%, and 100%. The maximum of both agents' chosen number determines the participation in their own rewards over the episode. The remaining portion is the participation in the other agent's individual rewards. For instance, if agent 1 chooses 40%, and agent 2 prefers 80%, they will receive 80% of their individual rewards and 20% of the other agent's individual rewards. The additional first step is part of the reinforcement learning, but no rewards are distributed in this step. The idea behind the additional trade step is that both agents can avoid cooperation by choosing 100%. By taking the maximum, the more conservative action is executed. Again, the agents successfully divide their tasks. The amount of waste cleared by agent 1 moves in parallel with the accumulated rewards. However, the magnitude of the joint rewards only reaches around 11, and it is unclear whether this is a stable level.

4.5 Participation in the clean-up game with three agents

For all of the following implementations with three agents, the bigger *10x10* map is used (Figure 7). Agent 1 gets spawned by the river, agent 3 gets spawned close to the green area on the right where apples are spawned, and agent 2 gets spawned in the middle of the map. The distribution of the agents on the map already gives a hint at one possible strategy, which consists of agent 1 clearing the waste, and agent 3 collecting the apples. However, it is unclear whether agent 2 should clear waste or collect apples. Like in the case with two agents, there is no justifying reason for any of the agents to clear waste as long as it does not get the chance to collect apples or profits from the other agents' rewards. We test the following implementations of the clean-up game with three agents.

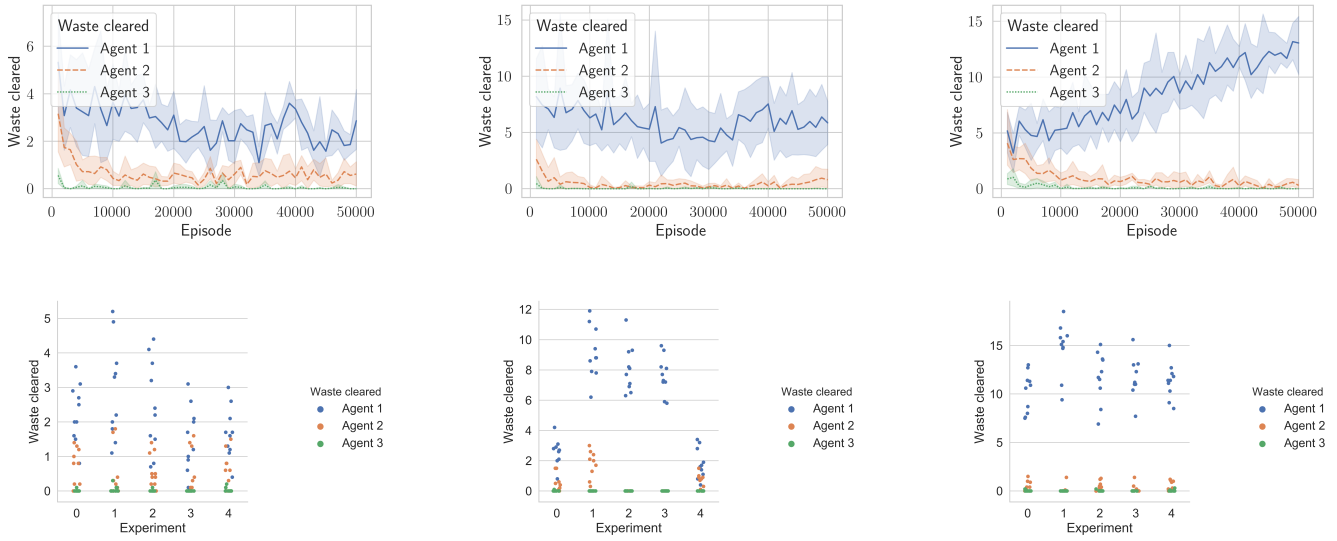
(i) No participation: As in the case with two agents, the social dilemma seems unsolvable without any additional incentive structures. As depicted in Figure 5b, the accumulated reward remains at a very low level. Cooperation does not seem to be attractive for any of the agents. Agent 1 does not learn to clear more waste because it is not rewarded for doing so. In fact, agent 1 learns to clear less waste over the course of the episodes. Similarly, agent 2 and agent 3 seem to focus on collecting apples. It is not easy to learn that clearing waste is creating value since the apples are spawned far away from the river. In the big map, it becomes increasingly difficult for an agent to move back from the river to the apples, as the distance between the river and the green area is larger. In the last episodes (40,000-50,000), the agents still do not act according to a clear strategy, as shown in Figure 6a. As the dots are spread within the experiments, there is no convergence to a certain strategy. Moreover, the dots differ between the experiments, showing that there is randomness in the behavior that the agents learn. More importantly, all three agents clear some waste and the differing amount of waste cleared seems to stem from the starting position. Hence, we assume that the agents do not learn to cooperate.

(ii) LIO: The baseline scenario is augmented with the possibility for each agent to give rewards to any of the other agents as an additional channel for cooperation. Except for the additional agent and the bigger map, nothing changed in the setting compared with



(a) Two agents **(b) Three agents**

Figure 5: Results from the clean-up game



(a) In the basic implementation, none of the agents learns to clear waste.

(b) In the LIO implementation, no clear strategy evolves.

(c) With participation, one of the agents learns to clear waste.

Figure 6: Waste cleared in clean-up with three agents for the basic implementation, LIO, and participation.

the *LIO* case with two agents. But the agent that is closest to the river and waste (agent 1) does not specialize in clearing the waste anymore (Figure 6b). In fact, none of the agents learns to specialize in clearing the waste. As a result of this, the joint reward does not go up over the episodes. This is a puzzling and important result, as it questions the applicability of *LIO* to scenarios with more than two agents or more complicated maps.

(iii) Equal distribution of individual rewards: The baseline scenario is augmented with the equal distribution of the joint rewards between the agents after each step. Hence, every agent always receives one third of the sum of all individual rewards. The agents successfully divide their tasks. Due to the starting position, agent 3 that is close to the apples turns into the harvester. Interestingly, agent 1 that is closest to the river only learns to clear waste in four out of five experiments. In the remaining one, it learns to collect apples. Agent 2 that is spawned in the middle of the map, learns to collect apples in three out of the five experiments, and becomes specialized in clearing waste in the remaining two experiments. Hence, there is substantial variation in the strategy learned between the experiments, but the strategies are stable within the experiments.

(iv) Pre-trade of participation rights: Similar to the case with two agents, the maximum of the three agents' chosen number determines the participation in their own rewards over the episode.

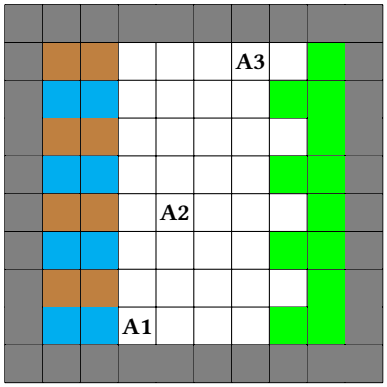


Figure 7: Big *10x10* map with three agents. Agent 1 gets spawned by the river, whereas agent 2 gets spawned between the river and the apples, and agent 3 starts close to the apples.

The remaining portion is the participation in the other agents' individual rewards. Again, the agents successfully divide their tasks. The amount of waste cleared by agent 1 moves in parallel with the accumulated rewards. However, the magnitude of the joint rewards only reaches around 7, and it is unclear whether this is a stable level. When looking at the share in own rewards, that is determined by the additional first step, there is no convergence recognizable. The evolution over the periods looks like a random walk. Either the agents need more episodes to figure out an optimal participation strategy, there is no optimal share as long as it is in a certain range, or the agents are not able to find a suitable participation strategy. Interestingly, the role strategies – namely becoming a clearer or harvester – is stable within and across the experiments (Figure 6c).

(v) Participation through common reward pool: Through an additional first step at the beginning of each episode, each agent can decide whether to participate in a common reward pool or not. In this first step, both agents can choose between six actions (0-5), representing *no participation* for actions 0-2 and *participation* for actions 3-5. When they participate, their individual rewards are put in the collective pool. After each step, the sum of all rewards of the participating agents is divided by the number of participants and distributed evenly among them. The common reward pool analyses a potential free-riding dilemma. For instance, two of the agents could participate, making one of them clear waste. But then the third agent would not need to participate and could just collect apples. By applying this strategy, the non-participating agent could end up with higher individual rewards than the participating ones. The results show that none of the agents learn to clear waste.

5 CONCLUSION

By introducing the idea of participating in other agents' rewards, we suggest a new method for coordination and cooperation in shared multi-agent environments. An agent learns that direct participation in other agents' rewards can motivate to act in the interest of all agents. Through this mechanism, no additional extrinsic incentive structures are needed such as in *LIO* [24]. Other previous works focused on intrinsic incentives [2, 5, 23] which would not be needed either. Especially the simplistic and graspable extension of standard models makes the participation appealing. In the two tested social dilemma problems, the Iterated Prisoner's dilemma and Cleanup, the opportunity to participate via shares is used by the agents to discover cooperative behaviors. In fact, the division of labor in Cleanup is effectively enabled. Without the participation, the agents cannot learn to divide their task into subtasks, as clearing waste does not directly lead to rewards. But through the participation, the positive impact of clearing waste is directly observed through the other agents' rewards from collecting apples. Importantly, the other agent can only collect apples thanks to the clearing of the waste beforehand. The introduced method of participation can achieve optimal collective performance in the Prisoner's dilemma. In Cleanup, the improvement in the collective performance through participation is clearly visible. Although it is not clear whether the agents exhaustively learn the perfect participation allocation, the agents manage to coordinate and enhance their performance over time.

Our approach attempts to answer some open questions on the path of ensuring cooperation in a decentralized multi-agent population. Firstly, although the agents must simultaneously learn how to participate via shares in addition to learning their environment actions, learning to keep shares of other agents may be a simpler task than punishing other agents whenever they act only in their own interest. Secondly, punishments or other incentivizing rewards should be earned beforehand. When working with shares, they can be directly traded without costs. Thirdly, a share market can be implemented in various ways. We are certain that a suitable market structure can be found in most social dilemmas. Furthermore, *LIO* assumes that recipients cannot reject an incentive, but an agent may accept only some incentives. In the case of shares, this is not a problem as the respective rewards are just distributed according to the owned shares. Another benefit of the participation approach is that there is no clear strategy for the agents to misuse the additional market feature to exploit the other agents. There is an emerging literature on reward tampering [3], and a participation market could be a step in the right direction of deploying safe applications.

Our work contributes to the aim of ensuring the common good in environments with independent agents. Although the participation approach works well in the tested social dilemmas, it remains unclear whether this is also the case in other environments. Another open question is if the agents can always find a stable allocation of shares. We suggest experiments with a broker that sets prices in combination with a limit order book for matching demand and supply. Additionally, participation needs to be tested in other environments with more agents as well as other game structures.

REFERENCES

[1] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. 2020. Open Problems in Cooperative AI. *CoRR* abs/2012.08630 (2020). arXiv:2012.08630 https://arxiv.org/abs/2012.08630

[2] Tom Eccles, Edward Hughes, János Kramár, Steven Wheelwright, and Joel Z. Leibo. 2019. Learning Reciprocity in Complex Sequential Social Dilemmas. *arXiv:1903.08082 [cs]* (March 2019). http://arxiv.org/abs/1903.08082 arXiv:1903.08082.

[3] Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. 2021. Reward tampering problems and solutions in reinforcement learning: a causal influence diagram perspective. *Synthese* 198, 27 (Nov. 2021), 6435–6467. https://doi.org/10.1007/s11229-021-03141-4

[4] Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with Opponent-Learning Awareness. *arXiv:1709.04326 [cs]* (Sept. 2018). http://arxiv.org/abs/1709.04326 arXiv: 1709.04326.

[5] Edward Hughes, Joel Z. Leibo, Matthew G. Phillips, Karl Tuyls, Edgar A. Duéñez-Guzmán, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin R. McKee, Raphael Koster, Heather Roff, and Thore Graepel. 2018. Inequity aversion improves cooperation in intertemporal social dilemmas. *arXiv:1803.08884 [cs, q-bio]* (Sept. 2018). http://arxiv.org/abs/1803.08884 arXiv: 1803.08884.

[6] Peter Kollock. 1998. Social Dilemmas: The Anatomy of Cooperation. *Annual Review of Sociology* 24, 1 (1998), 183–214. https://doi.org/10.1146/annurev.soc.24.1.183 _eprint: https://doi.org/10.1146/annurev.soc.24.1.183.

[7] Samuel S. Komorita. 1987. Cooperative Choice in Decomposed Social Dilemmas. *Personality and Social Psychology Bulletin* 13, 1 (March 1987), 53–63. https://doi.org/10.1177/0146167287131005 Publisher: SAGE Publications Inc.

[8] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. *arXiv:1702.03037 [cs]* (Feb. 2017). http://arxiv.org/abs/1702.03037 arXiv: 1702.03037.

[9] Adam Lerer and Alexander Peysakhovich. 2018. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv:1707.01068 [cs]* (March 2018). http://arxiv.org/abs/1707.01068 arXiv: 1707.01068.

[10] Andrei Lupu and Doina Precup. 2020. Gifting in Multi-Agent Reinforcement Learning. *New Zealand* (2020), 9.

[11] Judith E Maki, Donnie M Hoffman, and Richard A Berk. 1978. A time series analysis of the impact of a water conservation campaign. *Evaluation Quarterly* 2, 1 (1978), 107–118. Publisher: Sage Publications Sage CA: Thousand Oaks, CA.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (Feb. 2015), 529–533. https://doi.org/10.1038/nature14236 Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7540 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computer science Subject_term_id: computer-science.

[13] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv:1912.06680 [cs, stat]* (Dec. 2019). http://arxiv.org/abs/1912.06680 arXiv: 1912.06680.

[14] Julien Perolat, Joel Z. Leibo, Vinicius Zambaldi, Charles Beattie, Karl Tuyls, and Thore Graepel. 2017. A multi-agent reinforcement learning model of common-pool resource appropriation. *arXiv:1707.06600 [cs, q-bio]* (Sept. 2017). http://arxiv.org/abs/1707.06600 arXiv: 1707.06600.

[15] Thomy Phan, Lenz Belzner, Thomas Gabor, and Kyrill Schmid. 2018. Leveraging Statistical Multi-Agent Online Planning with Emergent Value Function Approximation. *arXiv:1804.06311 [cs]* (April 2018). http://arxiv.org/abs/1804.06311 arXiv: 1804.06311.

[16] Kyrill Schmid, Lenz Belzner, Thomas Gabor, and Thomy Phan. 2018. Action Markets in Deep Multi-Agent Reinforcement Learning. In *Artificial Neural Networks and Machine Learning – ICANN 2018 (Lecture Notes in Computer Science)*, Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis (Eds.). Springer International Publishing, Cham, 240–249. https://doi.org/10.1007/978-3-030-01421-6_24

[17] Kyrill Schmid, Lenz Belzner, and Claudia Linnhoff-Popien. 2021. Learning to Penalize Other Learning Agents. MIT Press. https://doi.org/10.1162/isal_a_00369

[18] Kyrill Schmid, Lenz Belzner, Robert Müller, Johannes Tochtermann, and Claudia Linnhoff-Popien. 2021. Stochastic Market Games. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 384–390. https://doi.org/10.24963/ijcai.2021/54

[19] Kyrill Schmid, Lenz Belzner, Thomy Phan, Thomas Gabor, and Claudia Linnhoff-Popien. 2020. *Multi-agent Reinforcement Learning for Bargaining under Risk and Asymmetric Information.* https://doi.org/10.5220/0008913901440151 Pages: 151.

[20] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nature* 550, 7676 (Oct. 2017), 354–359. https://doi.org/10.1038/nature24270 Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7676 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational science;Computer science;Reward Subject_term_id: computational-science;computer-science;reward.

[21] Meir Statman. 2004. The Diversification Puzzle. *Financial Analysts Journal* 60, 4 (July 2004), 44–53. https://doi.org/10.2469/faj.v60.n4.2636 Publisher: Routledge _eprint: https://doi.org/10.2469/faj.v60.n4.2636.

[22] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (Nov. 2019), 350–354. https://doi.org/10.1038/s41586-019-1724-z Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7782 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computer science;Statistics Subject_term_id: computer-science;statistics.

[23] Jane X. Wang, Edward Hughes, Chrisantha Fernando, Wojciech M. Czarnecki, Edgar A. Duenez-Guzman, and Joel Z. Leibo. 2019. Evolving intrinsic motivations for altruistic behavior. *arXiv:1811.05931 [cs]* (March 2019). http://arxiv.org/abs/1811.05931 arXiv: 1811.05931.

[24] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. 2020. Learning to Incentivize Other Learning Agents. *arXiv:2006.06051 [cs, stat]* (Oct. 2020). http://arxiv.org/abs/2006.06051 arXiv: 2006.06051.