Exploring the effects of punishment severity and norm update frequency in mixed-motive norm-enhanced Markov Games

Rafael Molinari Cheang[1][0000−0003−2434−1304], Marcos Menon José[2][0000−0003−4663−4386], and Jaime Simão Sichman[1,2][0000−0001−8924−9643]

[1] Laboratório de Técnicas Inteligentes (LTI), Escola Politécnica (EP)
[2] Center for Artificial Intelligence (C4AI)
Universidade de São Paulo (USP), São Paulo, Brazil
{rafael_cheang,marcos.jose}@alumni.usp.br
jaime.sichman@usp.br

Abstract. In the realm of game theory, mixed-motive games represent a subset of games where the interests of players are not entirely aligned nor entirely opposed. This duality often leads the system to a state known as the collective action problem, when individuals systematically prioritize their own rewards as opposed to greater group rewards. This problem normally occurs in mixed-motive games in the real world because people are generally good at responding to individual incentives and, with the emergence of learning techniques such as reinforcement learning, so are becoming agents in MAS. In our previous work [8], we proposed a framework composed of several learning agents, whose actions were regulated by a regulator agent to prevent the collective action problem in mixed-motive MAS when the following two conditions are not guaranteed: *a)* most agents in the system, more often than not, act in favor of the group instead prioritizing their own rewards, and *b)* agents are allowed to inflict non-negligible harm to other agents in order to punish defective behavior. In this new work, we present two experiments in order to test the effects that two variables have on the system's outcome; the frequency in which the regulator updates the system's norm and the harshness of the punishment given to agents that violate such norms. We show that higher update frequencies and harsher punishments tend to yield better outcomes.

Keywords: Reinforcement learning · Normative multiagent systems · Mixed-motive games.

1 Introduction

In the realm of game theory, mixed-motive games represent a subset of games where the interests of players are not entirely aligned nor entirely opposed. These games are distinguished by two fundamental properties [10]: *a)* each player has an incentive to pursue a strategy that may be advantageous for their individual

well-being but that can lead to suboptimal collective outcomes; and *b)*, the collective welfare of all players is maximized when they cooperate. These conflicting incentives between self and group can lead the whole system to an unfavorable state known as the collective action problem [26].

One classic example of the collective action problem in the real world is the provision of public goods. In urban life, we are all indirectly responsible for the maintenance of our roads, public spaces, and services such as the police and the fire brigade through the payment of municipal taxes. We, as individuals, have the monetary incentive to benefit from the collective efforts of others while contributing minimally or not at all (free-ride)[3]. If one person does so, it is likely that the impact on the city's public goods and services won't be substantial. However, if a significant portion of the population tax evade, it will be difficult for the city's administration to secure sufficient funding to sustain the provision.

The collective action problem is not particular to communities of people in the real-world, it may also happen in multiagent systems (MAS). This issue becomes more pronounced in MAS with the advent of new learning technologies such as reinforcement learning (RL), because as agents' learning capabilities increase, so increases their ability to optimize for their own benefit, which is reminiscent of the motto "*people respond to incentives*" [22], that is the root cause for the collective action problem in our societies.

Social norms and norm enforcement mechanisms are tools of an institutional machinery that can be used for governing mixed-motive systems in order to prevent such problem [34]. These can be implemented in a centralized way — when a central governing authority is responsible for the provision of norms and norm enforcement — or in a decentralized way — when the normative system is sustained by its agents.

Decentralized approaches share the benefit of not relying on a centralized entity to sustain the normative system nor the burden that may be norm designing and accurately predicting how the system will behave afterwards. That being said, these approaches depend on at least one of two basic assumptions, which may not hold for every mixed-motive system: *a)* most agents in the system will act pro-socially for the majority of the time instead of optimizing for their individual rewards, or *b)* it is allowed for agents to inflict non-negligible, direct or indirect punishment to other agents, in order to punish defective behavior.

We draw a parallel to a real-world scenario in order to further this point. Consider the case of burglary. In theory, this problem could be solved in case everyone acted pro-socially and no stealing ever took place, but this is not a feasible solution since we have no control over the intentions and actions of others. Another possible solution would be to punish stealing in order to discourage it, by means of physical altercation for instance, but this would not be a desirable solution since it could compromise the safety of those involved. Apart from these, what else could a victim of burglary do to prevent it from happening?

In case we cannot safely assume agents will act pro-socially, nor it is desirable for agents to retaliate against each other, we may need to resort to an overseeing

[3] Assuming we wouldn't pay a fine for doing so.

entity to regulate the system, which is a solution regularly adopted to solve problems such as burglary in the real-world.

This work further explores a general purpose framework proposed in our previous work [8] to steer mixed-motivated MAS out of socially bad outcomes when assumptions *a* and *b* cited above do not necessarily hold. We extend such work by testing how the system behaves when two of its variables vary: *a)* the frequency in which the regulator changes the norm and *b)* the fine multiplier, a variable that controls the harshness of the fine applied once the norm is violated. Another contribution of this work is an enhancement of the formal model previously introduced.

2 Previous work

Our previous work [8] proposes a norm-enhanced Markov Game (neMG) model, where a Markov Game environment is augmented with normative information. In this model, a *regulator* agent monitors the system and adjusts its norms based on system-level metrics to maximize the system's collective outcome, while *players* act according to their interests to maximize their own outcomes. The model was demonstrated through a simulation of the "tragedy of the commons" game [16], where multiple agents compete for a shared resource. The simulation showed that, with the regulator in place, agents learned to cooperate by adhering to the norm, preventing resource depletion and achieving a more sustainable and better outcome.

3 Related work

The idea of regulating systems of heterogeneous agents through a formal institution is about as old as the problem of attaining social order from local actions and interactions [7]. One significant advance in crafting a framework for social control involved the introduction of electronic institutions (EI) [25, 13, 12]. These institutions, in addition to their various other provisions, establish a set of regulations that govern the actions agents within the system should or should not take in predefined circumstances. They are inspired, and play a similar role to the one traditional norm-setting institutions play in real-world societies [3].

Though an important step, EIs had some limitations when compared to real-world institutions. For once, EIs were conceived at design time and were not capable of evolving over time [4]. This issue presented some challenges for their adoption since *a)* regulating complex systems is a hard task, especially when the rules of the game are set *a priori*, and *b)* because conceiving fully functional EIs at design time is hard, a desirable property of software may be lost, i.e., the deployed system may not be self-managed.

This latter issue gave birth to the proposal of an autonomic electronic institution (AEI) [4, 3], that, as the name suggests, is an electronic institution with autonomic capabilities (norm-evolving at run-time). The main objective of an AEI is for the institution to accomplish its goal by iterating through a two-step

process of assessing goal adherence, and adapting the system's norms in case it is not, through the use of an evolutionary algorithm.

The RL community has also seen its fair share of proposals for solving the collective action problem in mixed-motive multiagent reinforcement learning (MARL) environments. That being said, its take on the problem differs from that of the MAS community previously presented in that most of its proposals have tackled the problem from a decentralized perspective; their solutions involve tailoring agents' architectures or capabilities to the specific needs of mixed-motive games.

These solutions can work just fine in closed systems, where one has control over the agents being deployed, or even in systems where agents are allowed to punish each other, but not as much in open systems where firm retaliation[4] is not allowed. They can be generally grouped in two: strategies that leverage reciprocity mechanisms, where agents learn to punish defective behaviors, and pro-social intrinsic motivation strategies, that reward agents for pro-social behavior.

Reciprocity has been a notorious strategy for agents in mixed-motive games since the days of the Axelrod's tournaments [1, 2]. This strategy is as simple as it is effective, an agent playing a reciprocity strategy defects when it recognizes antisocial behavior and cooperates when it recognizes pro-social behavior.

These strategies have been implemented in RL agents by simply adding the capability of firmly punishing others to the agents' set of actions. By doing this, agents were capable of learning to reciprocate through self-play. Among the works that have leveraged reciprocity mechanisms to combat the collective action problem in mixed-motive MARL, we highlight those of Pérolat et al. [27], that implemented agents with the ability of tagging other agents out of the game for a period of time, Lerer and Peysakhovich [19], that implemented agents with two switchable policies, one fully cooperative and one fully defective, and Eccles et al. [11], that implemented reciprocity through imitation.

Another active avenue of research is to deviate from the rational egoist model and endow RL agents with pro-social *intrinsic motivation*. Traditional RL agents learn through the rewards given by the environment. This reward can be regarded as *extrinsic*, i.e. the reinforcement is given to the agent as a signal of how well it is solving a problem of clear practical value [30]. Conversely, *intrinsic motivation* can be modeled as a term that composes the agents' rewards together with the extrinsic; this can be understood as a reward that is not related to the specific task in hand, but is rather earned because it is inherently enjoyable [30].

Intrinsic motivation can be used as a way to model complex abstract patterns such as morality and empathy. Among the works that leverage pro-social intrinsic motivation to deal with the collective action problem in mixed-motive environments, we highlight those of Hughes et al. [17], that incorporated inequity aversion preferences in RL agents, Peysakhovich and Lerer [28], that modeled

[4] By firm retaliation we mean that the punishment inflicted by one agent to another is not negligible.

pro-sociality by including other agents' rewards as agents' intrinsic motivation, and Jaques et al. [18], that used intrinsic motivation to model social influence.

The proposed work is similar to the AEI framework in that it addresses most of the same problems (social order in MAS) by leveraging the use of norms, but different in that it uses RL for norm adaptation instead of an evolutionary algorithm. In doing so, it deviates significantly from those solutions put forward by the RL community; it does not assume anything about the agents' architectures nor that they are able to punish each other.

4 Normative MAS and the ADICO grammar of institutions

MAS hold many similarities with human societies in that, like us humans, agents may have heterogeneous preferences and may differ in how they assess their surroundings and act toward their goals. As such, MAS may also be subject to the harmful symptoms commonly found in mixed-motive human systems such as miscoordination, collusion, and negative externalities [22].

One way of preventing these issues both in the real-world and in MAS is through the use of regulation and oversight. Such apparatus involve the creation of norms that dictate the socially desired behavior of agents, as well as the establishment of oversight bodies that ensure that these norms are being followed.

A norm enhanced MAS can be regarded as a normative multiagent system (NMAS), i.e. a MAS in which norms and normative concepts may influence its overall outcome [24]. In these settings a norm is typically understood to be a standard or guideline that is widely accepted and expected to be followed within a particular group or society [33].

Within the context of NMAS, failing to adhere to the prevailing norm could lead to sanctions. These can be broadly categorized as *direct material sanctions*, that have an immediate negative effect on a resource valued by the agent, such as fines, or *indirect social sanctions*, like damaging the agent's reputation, which can shape its future standing within the system [6].

Such normative systems can be arranged either in a centralized or distributed manner [20]. They differ in whether the normative machinery is sustained and enforced by a single entity — be it an agent or an organization — (centralized), or not (distributed).

In order to formalize the conception of norms, Crawford and Ostrom [9] proposes the ADICO grammar of institutions. The grammar is defined within the five dimensions:

- ***A**ttributes:* is the set of variables that specify the individuals or entities to whom the norm is applicable.
- ***D**eontic:* is a placeholder for the three key modal operations derived from deontic logic: *may* (indicating permission), *must* (indicating obligation), and *must not* (indicating prohibition).

- ***Aim:*** describes a specific action or a collection of actions to which the deontic operator is assigned.
- ***C****onditions:* defines the contextual factors that determine when, where, how, and under what circumstances an action is deemed obliged, permissible, or forbidden.
- ***O****r else:* describes the sanctions in the event of non-compliance with the norm.

This grammar can be useful to turn the somewhat abstract concept of a norm into something tangible, and to operationalize the norm creation and norm revision processes. For instance, the norm *All citizens, who earn more than 30,000 dollars per year, must pay income tax at the beginning of the year, or else he/she will have to pay a fine of 1,000 dollars*[5] can be broken down into: ***A:*** All citizens who earn more than 30,000 dollars per year, ***D:*** must, ***I:*** pay income tax, ***C:*** at the beginning of the year, ***O:*** will have to pay a fine of 1,000 dollars.

5 Reinforcement learning and multiagent reinforcement learning

5.1 Reinforcement learning (RL)

The reinforcement learning task outlines the journey of an agent as it engages with an environment, receives positive or negative feedback for its actions in the form of rewards, and learns from them. This general description can be formalized through the Markov decision process (MDP), defined in the following.

Definition 1. A Markov Decision Process (MDP) is defined by the $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ *tuple, where*

- $\mathcal{S}$ *represents a finite set of environment states;*
- $\mathcal{A}$*, a finite set of agent actions;*
- $\mathcal{R}$*, a reward function* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow R$ *that defines the immediate — possibly stochastic — reward an agent gets for taking action* $a \in \mathcal{A}$ *in state* $s \in \mathcal{S}$*, and transition to state* $s' \in \mathcal{S}$ *thereafter;*
- $\mathcal{P}$*, a transition function* $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ *that defines the probability of transitioning to state* $s' \in \mathcal{S}$ *after taking action* $a \in \mathcal{A}$ *in state* $s \in \mathcal{S}$*; and*
- $\gamma \in [0, 1]$*, a discount factor of future rewards [31, p. 47].*

In this context, the agent's primary objective is to maximize its cumulative expected reward over the long term, denoted G_t. This cumulative reward can be computed as the discounted infinite sum of rewards: $(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... + \gamma^n R_{t+n+1})$. Solving an MDP involves finding an optimal *policy* $\pi_* : \mathcal{S} \rightarrow \mathcal{A}$, i.e., the best action to take at each state — the action a that corresponds to the highest long-term expected reward G_t subject to the discount factor γ at a given state s.

[5] This is a hypothetical scenario

5.2 Multiagent reinforcement learning (MARL)

Multiagent reinforcement learning (MARL) refers to the set of RL tasks where multiple agents — two or more – co-exist and interact with an environment and with each other. The MDP counterpart in MARL is the Stochastic Game or Markov Game [21], defined in the following.

Definition 2. A Markov Game (MG) can be formally defined by the 6-tuple $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \gamma \rangle$*, where*

- $\mathcal{N} = \{1, ..., N\}$ *denotes the set of* $N > 1$ *agents;*
- $\mathcal{S}$*, a finite set of environment states;*
- $\mathcal{A}^i$*, agent's* i *set of possible actions.*

  Let $\mathcal{A} = \mathcal{A}^1 \times ... \times \mathcal{A}^N$ *be the set of agents' possible joint actions. Then*

- $\mathcal{R}^i$ *denotes agent's* i *reward function* $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow R$ *that defines the immediate reward earned by agent* i *given a transition from state* $s \in \mathcal{S}$ *to state* $s' \in \mathcal{S}$ *after a combination of actions* $a \in \mathcal{A}$*;*
- $\mathcal{P}$*, a transition function* $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ *that defines the probability of transitioning from state* $s \in \mathcal{S}$ *to state* $s' \in \mathcal{S}$ *after a combination of actions* $a \in \mathcal{A}$*; and*
- $\gamma \in [0, 1]$*, a discount factor on agents future rewards [36].*

From an agent's point of view the goal remains the same as in the traditional RL case; to maximize its long term cumulative expected reward. Still, one key difference between RL and MARL lies on the fact that the environment transitions to a new state as a function of the combined actions of all agents on the latter, as opposed to the former, where it transitions solely as a function of one agent's action.

As a result, a game theoretic aspect which is central to multiagent systems is added to the system. Since the environment transitions as a function of the joint actions of all agents, an agent has to optimize its policy not only with respect to the state of the environment, but also, relative to the joint policy of all other agents in the system.

6 A Norm-enhanced Markov Game

We further formalize the norm-enhanced Markov Game (neMG) model proposed in our previous work [8]. A neMG comprises two types of RL agents: $N > 1$ *players* and one *regulator*. Players are simple RL agents, analogous to the ones that interact with regular versions of MG environments, with the difference that they are aware of the norm of the game, which is available to them as it is part of the environment's state. The regulator, on the other hand, is able to act exclusively on the environment's norm at a predefined frequency measured in

terms of players' steps, which we refer as a period. This agent senses the state of the environment through a social metric — i.e. a system-level diagnostic — and the efficacy of its actions is signaled back by the environment as a reward based on the system's social outcome.

Definition 3. Let $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \gamma\rangle$ *be the regular version of the Markov Game to be enhanced. Then, a norm-enhanced Markov Game (neMG) can be formally defined by a 13-tuple* $\langle \phi, \mathcal{N}_p, \mathcal{S}_p, \{\mathcal{A}_p^i\}_{i\in\mathcal{N}_p}, \{\mathcal{R}_p^i\}_{i\in\mathcal{N}_p}, \mathcal{P}_p, \gamma_p, m, \mathcal{S}_r, \mathcal{A}_r, \mathcal{R}_r, \mathcal{P}_r, \gamma_r\rangle$*, where*

- ϕ *denotes the neMG's set of possible norms;*
- $\mathcal{N}_p = \mathcal{N}$ *denotes the set of* $N > 1$ *players;*
- $\mathcal{S}_p = \mathcal{S} \times \phi$*, the players' finite set of environment states;*
- $\mathcal{A}_p^i = \mathcal{A}^i$ *player's* i *set of possible actions.*

Let $\mathcal{A}_p = \mathcal{A}_p^1 \times ... \times \mathcal{A}_p^N$ *be the set of players' possible joint actions. Then*

- $\mathcal{R}_p^i$ *denotes player's* i *reward function* $\mathcal{R}_p^i : \mathcal{S}_p \times \mathcal{A}_p \times \mathcal{S}_p \to R$ *that defines the immediate reward earned by player* i *given a transition from state* $s_p \in \mathcal{S}_p$ *to state* $s'_p \in \mathcal{S}_p$ *after a combination of actions* $a_p \in \mathcal{A}_p$*;*
- $\mathcal{P}_p$*, a transition function* $\mathcal{P}_p : \mathcal{S}_p \times \mathcal{A}_p \times \mathcal{S}_p \to [0, 1]$ *that defines the probability of the players' environment transitioning from state* $s_p \in \mathcal{S}_p$ *to state* $s'_p \in \mathcal{S}_p$ *after a combination of actions* $a_p \in \mathcal{A}_p$*;*
- $\gamma_p \in [0, 1]$*, a discount factor on players future rewards;*
- $m \in \mathbf{N}$*, the amount of players' steps per period;*
- $\mathcal{S}_r$*, the regulator's set of states;*
- $\mathcal{A}_r$*, the regulator's set of actions;*

Let r_j^i *denote the reward earned by player* i *at a relative time step* j *of a given period*[6]*, and* n *the number of players in a neMG. Then*

- $\mathcal{R}_r$ *denotes the regulator's reward function* $\mathcal{R}_r = \sum_{i=1}^{n} \sum r_j^i$[7]*, that determines the immediate reward earned by the regulator at the end of a period given by the sum of all players' rewards over that same period;*
- $\mathcal{P}_r$*, the normative transition function* $\mathcal{P}_r : \phi \times \mathcal{A}_r \to \phi$ *that defines norm update following a regulator's action; and*
- $\gamma_r \in [0, 1]$*, the regulator's discount factor.*

Following this definition, a neMG can be executed through two distinct RL loops: one relative to the regulator at the outer level, and another relative to players at the inner level. Algorithm 1 exemplifies how these could be implemented.

[6] e.g. r_3^2 refers to the third reward earned by player 2 within the period.
[7] $\sum r_j^i$ refers to the sum of rewards earned by player i in the given period.

Algorithm 1: neMG Pseudocode

```
1  algorithm parameters: number of players (n), steps per period (m);
2  initialize policy and/or value function parameters;
3  foreach episode do
4      initialize environment (set initial states s_{r0} and s_{p0});
5      foreach period do
6          regulator adjusts norm (φ) by consulting its policy π_r in state s_r;
7          for m steps do
8              set current player i;
9              current player acts based on its policy π_p^i in state s_p, state
                transitions to s'_p, player observes its reward r_p^i, and updates its
                policy π_p^i;
10         end for
11         regulator observes next state s'_r, its reward r_r and updates its policy
            π_r;
12     end foreach
13 end foreach
```

Training on an neMG happens across multiple episodes. An episode begins with the initialization of the environment's states (line 4). At every period, the regulator acts by adjusting the environment's norm based on its percept, players in the game act for m steps (combined), and the regulator receives an immediate reward, update its policy, and the environment transitions to the next state (lines 5-10). In this case, period size (m) is the variable used to control the frequency in which the regulator acts and is measured in terms of players' steps. At every step, a player acts based on its percepts, the state transitions, the player receives an immediate reward from the environment, and updates its policy (lines 8-9). The current player can be set in a round-robin, circular manner. Note that the norm does not appear anywhere in the players' loop because it is embedded within the environment state.

7 Experiments

7.1 Environment

The experiments take place in the same environment as the experiment in our previous work [8]; an environment that emulates the tragedy of the commons game [16] and that closely resembles the environment used in Ghorbani et al. [14]. In it, players consume units of a common resource that replenishes as a function of the amount of resources left in a previous step — i.e. if the resource level falls to zero, the replenishment will also be zero. Players are rewarded proportional to the amount of resources they consume, but if they all consume as much as they can in each iteration, resources soon deplete, which characterizes an instance of the collective action problem. The environment allows for the existence of

norms and a regulator agent by including all elements introduced in Section 6. The regulator can set a consumption limit for other agents, as well as the punishment for overconsumption. The environment is composed of two different but related parts: the players' environment and the regulator's environment, which are both described in the following.

Regulator's environment: The regulator's environment has the goal of exposing macro-level information about the system to the regulator, and allowing it to adapt the norms that will influence the behavior of players.

At every regulator's iteration — which we here denote period —, the regulator can observe how much resource is left (R), and a short-term and long-term sustainability measurement (S_s and S_l respectively), given by $S = \sum_{j=p-t}^{p} \frac{rp_j}{c_j}$ defined for $c_j > 0$ and $t \geq 0$, with t being the number of periods considered as short-term and long-term (respectively one and four for all simulations); rp_j, the total amount of resources replenished in period j; c_j, the total consumption in period j; and p, the current period.

The initial values at the beginning of the simulation for these variables are drawn from uniform distributions, i.e. $R_0 \sim \mathcal{U}(10000, 30000)$, $S_{s0} \sim \mathcal{U}(0.4, 0.6)$, and $S_{l0} \sim \mathcal{U}(0.4, 0.6)$.

After observing the environment's state, the regulator acts by adapting the norm regulating the system. Here, we use the ADICO grammar cited in Section 4 as the normative framework to operationalize the norm synthesis process. The A, D, and C dimensions remain fixed in this environment since *a)* the norm applies to all players, *b)* the norm always defines a forbidden action, and *c)* the norm is valid throughout the episode, no matter the conditions. Conversely, the I and O dimensions can be adapted by the regulator; i.e., at every period, the regulator may change the players' consumption limit (l) and the fine applied to those players who violate this condition ($f(c, l, \lambda)$) — by changing the fine multiplier λ. The regulator adapts the norm by increasing or decreasing the values of (l) — with changes limited to a value of 400 (Δl_{max}) and up until a maximum value of c_{max} ($l_{max} = c_{max}$) — and (λ) — with changes limited to a value of 0.5 ($\Delta\lambda_{max}$) and up until a maximum value of 3 (λ_{max}). The initial values of both the consumption limit and the fine multiplier are drawn from normal distributions in the first period of the simulation, i.e. $l_0 \sim \mathcal{N}(375, 93.75)$ and $\lambda_0 \sim \mathcal{N}(1, 0.2)$. These values become available as part of the environment state in the player's environment

At the end of the period, the environment rewards the regulator based on how well all players did during the iteration, i.e., how much of the resource all agents consumed combined. The regulator's environment relates to lines 5-12 in Algorithm 1.

Players' environment: After the regulator sets the norm for the period, players' consume, one at a time, a quantity of resources up to a maximum of 1500 units. The decision of how much to consume (c_i) is taken after the player observes the environment state available to it, which is composed of the amount of common resource left (R), and the system's norm, which includes the consumption limit

(l) and the fine multiplier (λ) set by the regulator. Upon such decision, the environment's resource level is updated following the simple rule $R := R - c_i$. This process of observing the state of the environment, and choosing how much to consume happens for a total of m steps, which controls the frequency in which the regulator acts.

At every n steps — n being the number of players, 5 for this experiment — the resource grows by a quantity given by the logistic function $\Delta R := rR(1 - \frac{R}{K})$ — akin to how some natural resources grow in the real world [14] —, with ΔR being the amount to increase; r, the growth rate, set to 0.3; R, the current resource quantity; and K, the environment's carrying capacity — an upper bound for resources —, set to 50000. The players' environment execution relates to lines 7-10 in Algorithm 1.

An episode has two stop conditions; it finishes at the end of a period in case resources are completely depleted or after a thousand steps.

Settings: We propose testing the model with changes along two axes: the harshness of the punishment applied to players that violate norms (by changing the value of the fine multiplier) and the frequency at which the regulator acts (by changing the period size). The test cases are distributed in two experiments, each one serving the purpose of testing how this implementation of the framework behaves given variations on each axis. Table 1 presents how the 8 proposed test cases vary along said axes.

Experiment	Name	Value of fine multiplier (λ)	Period size (m)
Experiment 1	*default50*	var	50
	default100	var	100
	default200	var	200
	default500	var	500
Experiment 2	*default100*	var	100
	fixedMultiplier0.5	0.5	100
	fixedMultiplier1	1	100
	fixedMultiplier2	2	100
	fixedMultiplier3	3	100

Table 1. Summary of implementation test cases. The *default100* case is used as a base case in both experiments and thus.

The environment was built using both the OpenAI gym [5] and pettingzoo [32] frameworks. Agents in this simulation were built with traditional RL architectures — SAC [15] for the regulator and A2C [23] for the players — using the Stable Baselines 3 framework [29], and players were trained on a shared policy. The learning rates for all agents were set to 0.00039. Each test case was run 10 times.

7.2 Experiment 1: testing the period size effect

This experiment provides us with a way of testing the effect that different period sizes – the frequency at which the regulator acts – have on the overall performance of the system. To this end, we use the *default100* case as a benchmark and test it against versions of the game with different period sizes (m). These were set to 50 (*default50* case), 200 (*default200* case), and 500 (*default500* case).

Results Figure 1 presents the average net and total consumption per episode for each case in the experiment. The results show that the *default50* case seems to reach — on average — a higher consumption (around 600,000) than the three other cases, before the four-thousandth episode, when it drops about 33%. We conjecture this drop occurs due to some training instability common to RL such as off-policy divergence [31, p. 260].

For the test cases in which the regulator's actions are more infrequent, total consumption did not stabilize at — in the *default200* case — or even reach — in the case of *default500* — the same levels as the test cases in which the regulator act more frequently (*default50* and *default100*). This behavior is expected since this metric is highly dependent on the regulator's ability to set the right consumption limit, and its learning is dependent on the frequency in which it acts. Also, player's learning could have been harmed in these cases, since players spend more time acting on states with depleted resources, where their actions have no effect on their rewards. A final reason that could explain the lower performance of cases with larger period sizes is that in these systems the regulator would have less time to react and thus prevent it from collapsing.

7.3 Experiment 2: testing the fine multiplier effect

In this experiment we test the effect harsher punishment has on the system's performance. This is accomplished by fixing the fine multiplier at different levels across four different test cases ($\lambda = 0.5$, $\lambda = 1$, $\lambda = 2$, $\lambda = 3$) and leaving only the task of setting the consumption limit to the regulator. Since fines are just a proxy metric for negative rewards in our environment, this experiment has the intent of testing how these mixed-motive systems behave for different scales of punishment and how these changes may affect the agents' learning path. We also compare these cases against the *default100* case, to check if there are any noticeable advantages in allowing this extra flexibility to the regulator.

Results: Figure 3 presents the average total and net consumption per episode for each of the five test cases in experiment 3 (*default100*, *fixedMultiplier0.5*, *fixedMultiplier1*, *fixedMultiplier2*, and *fixedMultiplier3*). We notice a tendency for convergence at a higher consumption level for the two cases with greatest fine multipliers (*fixedMultiplier2* and *fixedMultiplier3*) when compared to the two cases with the smallest fine multipliers (*fixedMultiplier0.5* and *fixedMultiplier1*). This effect could be due to the strength of the signal being sent to the agents in the form of fines. The smaller the fine multiplier, the lesser is the punishment

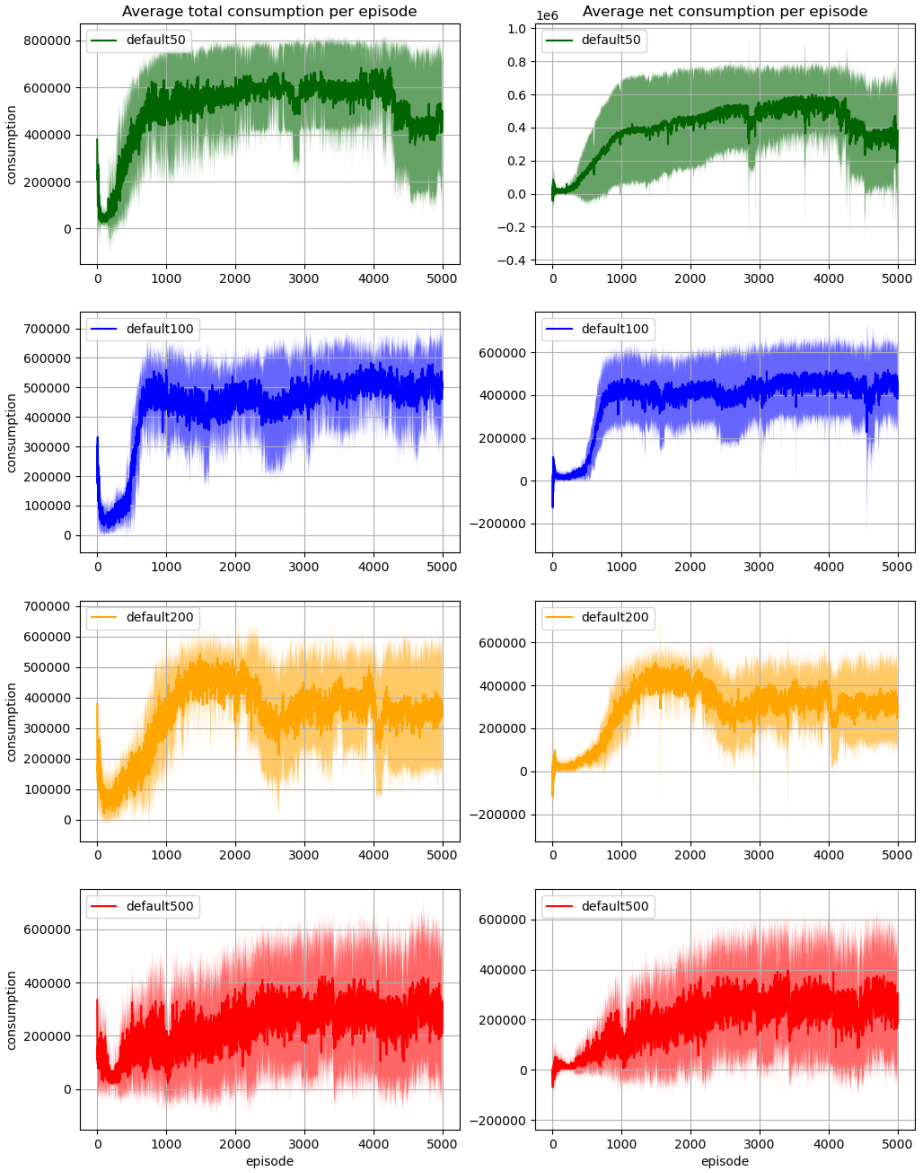

Fig. 1. The average total and net consumption per episode for all cases in experiment 2 (*default50*, *default100*, *default200*, and *default500*). The shaded area in each graph covers the area of one standard deviation above and one standard deviation below the mean for each episode.

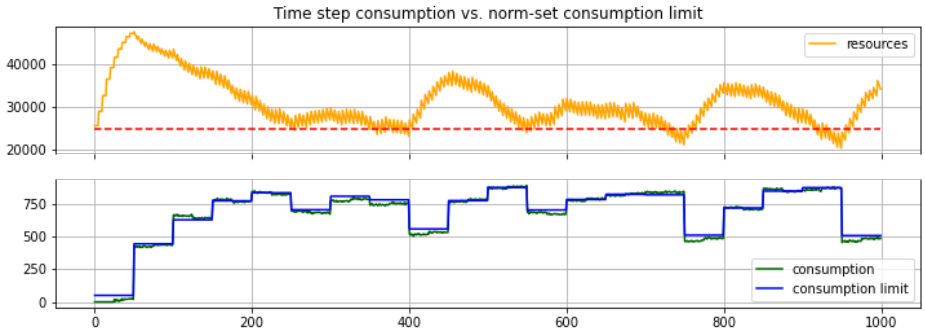

Fig. 2. Resources level at a later episode when $m = 50$. The regulator manages to keep resources near the optimal level (25000), represented by the dotted red line.

received for violating the norm and weaker is the players' learning signal. The stronger signal could be doing a better job in encouraging players to consume below the limit, which is good for them in the long run. Another finding from this experiment, is that there does not seem to exist a noticeable gain by allowing the regulator set the fine multiplier.

Discussion: The first experiment shows that the frequency in which the regulator acts in the system proved to be a sensible variable. Increasing such frequency grants the regulator greater control by allowing it a bigger margin for it to correct the system's path once the system starts to behave undesirably. This could be especially useful in dynamic systems, where negative outcomes might scale exponentially.

The second experiment gives us a hint to how the punishment variable — the **Or else** variable from the *ADICO* framework such as λ — impact learning in and the overall performance of a mixed-motive neMG. Greater punishment seems to grant more stability during training and also positively impact system's performance. That being said, we do not know the extent to which this pattern is valid, more experiments should be conducted to test if it holds for even greater values of λ.

8 Conclusions

Multiagent systems are part of a trend towards greater and widespread computational power [35] that harnesses the potential of autonomous, goal-oriented agents to solve ever so complex problems. This is reminiscent of how humans solve problems in societies. We coordinate, cooperate, and negotiate with one another in order to settle disputes, reach agreements, and move forward as collective.

Still we have come to agree that letting everyone freely pursue their goals through any means deemed necessary may take us quickly down a dangerous

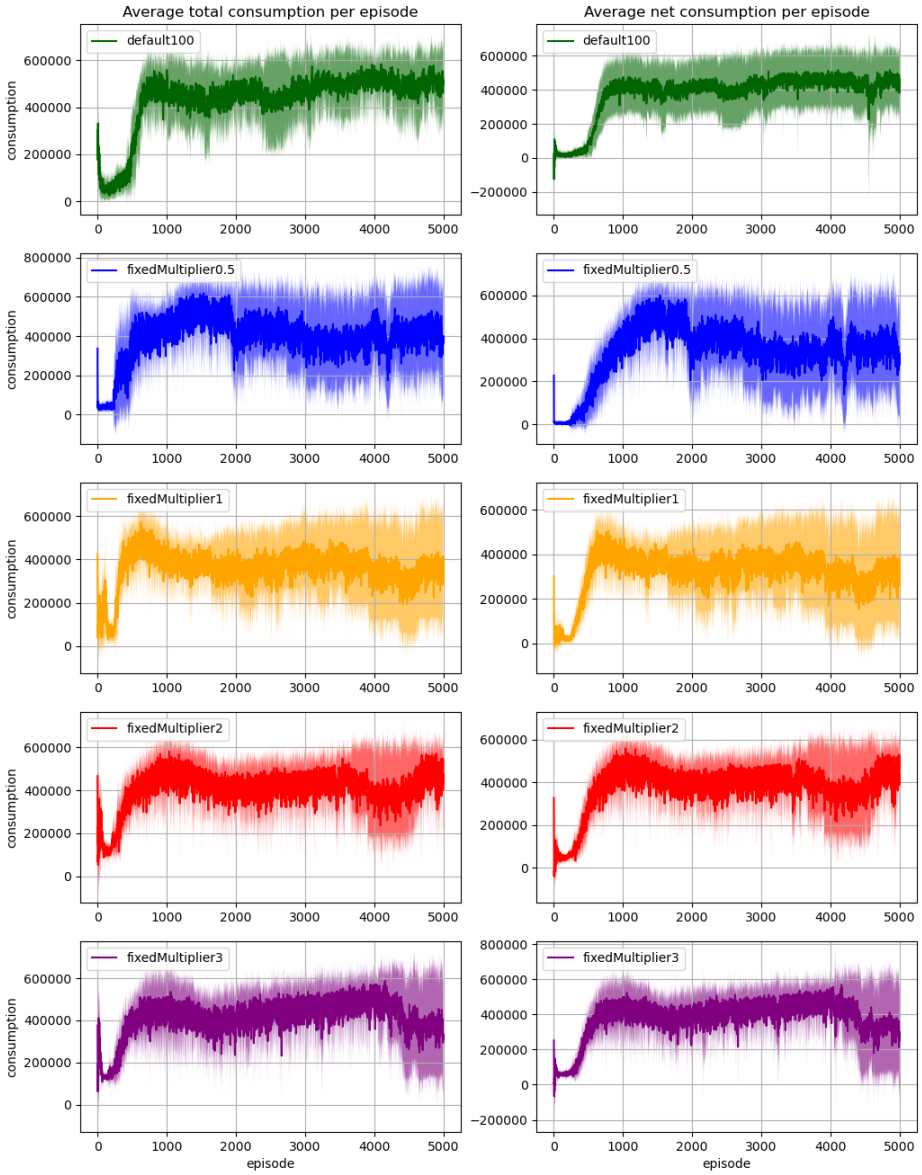

Fig. 3. The average total and net consumption per episode for all cases in experiment 3 (*default100*, *fixedMultiplier0.5*, *fixedMultiplier1*, *fixedMultiplier2*, and *fixedMultiplier3*). The shaded area in each graph covers the area of one standard deviation above and one standard deviation below the mean for each episode.

road. In a system where incentives can point to many different directions, all sorts of emergent exploits may lead to negative externalities. For instance, two people may agree on a deal beneficial to them both but that goes against the interests of one or more third parties.

In many of these cases we resort to central regulation of some shape or form. If many parallels can be drawn between multiagent systems and real-world communities, why shouldn't we exploit this apparatus that has been employed for centuries in the real-world, and is very present in our everyday lives, to solve problems in communities of artificial agents? This work is part of an effort to try and explore such solutions in MARL environments.

Acknowledgments. This research was carried out with the support of *Itaú Unibanco S.A.*, through the scholarship program of *Programa de Bolsas Itaú (PBI)*, and it is also financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Finance Code 001, Brazil. Any opinions, findings, and conclusions expressed in this manuscript are those of the authors and do not necessarily reflect the views, official policy or position of Itaú-Unibanco and CAPES. Jaime Sichman is a member of the UNBIAS team, which is a component of the THUS pillar of the USP-CNRS International Research Center.

References

1. Axelrod, R.: Effective choice in the prisoner's dilemma. Journal of Conflict Resolution **24**(1), 3–25 (1980). https://doi.org/10.1177/002200278002400101
2. Axelrod, R.: More effective choice in the prisoner's dilemma. Journal of Conflict Resolution **24**(3), 379–403 (1980)
3. Bou, E., López-Sánchez, M., Rodríguez-Aguilar, J.A., Sichman, J.S.: Adapting autonomic electronic institutions to heterogeneous agent societies. In: Organized Adaption in Multi-Agent Systems. pp. 18–35. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
4. Bou, E., López-Sánchez, M., Rodríguez-Aguilar, J.A.: Towards self-configuration in autonomic electronic institutions. In: Coordination, Organizations, Institutions, and Norms in Agent Systems II. pp. 229–244. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
5. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. arXiv preprint arXiv:1606.01540 (2016)
6. Cardoso, H.L., Oliveira, E.: Adaptive deterrence sanctions in a normative framework. In: International Joint Conference on Web Intelligence and Intelligent Agent Technology. pp. 36–43. IEEE Computer Society (2009)
7. Castelfranchi, C.: Engineering social order. In: Engineering Societies in the Agents World. pp. 1–18. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
8. Cheang, R.M., Brandão, A.A.F., Sichman, J.S.: Centralized norm enforcement in mixed-motive multiagent reinforcement learning. In: Ajmeri, N., Morris Martin, A., Savarimuthu, B.T.R. (eds.) Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XV. pp. 121–133. Springer International Publishing, Cham (2022)

9. Crawford, S.E.S., Ostrom, E.: A grammar of institutions. American Political Science Review **89**(3), 582–600 (1995). https://doi.org/10.2307/2082975
10. Dawes, R.M.: Social Dilemmas. Annual Review of Psychology **31**(1), 169–193 (1980). https://doi.org/10.1146/annurev.ps.31.020180.001125
11. Eccles, T., Hughes, E., Kramár, J., Wheelwright, S., Leibo, J.Z.: Learning reciprocity in complex sequential social dilemmas (2019)
12. Esteva, M., de la Cruz, D., Rosell, B., Arcos, J.L., Rodríguez-Aguilar, J., Cuní, G.: Engineering open multi-agent systems as electronic institutions. In: National Conference on Artifical Intelligence. p. 1010–1011. AAAI'04, AAAI Press (01 2004)
13. Esteva, M., Rodríguez-Aguilar, J.A., Sierra, C., Garcia, P., Arcos, J.L.: On the formal specifications of electronic institutions. In: Agent Mediated Electronic Commerce, The European AgentLink Perspective. p. 126–147. Springer-Verlag, Berlin, Heidelberg (2001)
14. Ghorbani, A., Ho, P., Bravo, G.: Institutional form versus function in a common property context: The credibility thesis tested through an agent-based model. Land Use Policy **102**, 105237 (2021). https://doi.org/https://doi.org/10.1016/j.landusepol.2020.105237
15. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Dy, J., Krause, A. (eds.) International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1861–1870. PMLR (10–15 Jul 2018)
16. Hardin, G.: The tragedy of the commons. Science **162**(3859), 1243–1248 (1968). https://doi.org/10.1126/science.162.3859.1243
17. Hughes, E., Leibo, J.Z., Phillips, M., Tuyls, K., Dueñez-Guzman, E., García Castañeda, A., Dunning, I., Zhu, T., McKee, K., Koster, R., Roff, H., Graepel, T.: Inequity aversion improves cooperation in intertemporal social dilemmas. In: Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
18. Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P.A., Strouse, D.J., Leibo, J.Z., de Freitas, N.: Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In: International Conference on Machine Learning. vol. 97. PMLR (2019)
19. Lerer, A., Peysakhovich, A.: Maintaining cooperation in complex social dilemmas using deep reinforcement learning (2018)
20. de Lima, I.C.A., Nardin, L.G., Sichman, J.S.: Gavel: A sanctioning enforcement framework. In: Weyns, D., Mascardi, V., Ricci, A. (eds.) Engineering Multi-Agent Systems - 6th International Workshop, EMAS 2018, Stockholm, Sweden, July 14-15, 2018, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11375, pp. 225–241. Springer (2018). https://doi.org/10.1007/978-3-030-25693-7_12, https://doi.org/10.1007/978-3-030-25693-7_12
21. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: International Conference on Machine Learning. p. 157–163. ICML'94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1994)
22. Mankiw, N.G.: Principles of Economics, 8th edition. Cengage Learning, Cambridge, Massachusetts (2018)
23. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1928–1937. PMLR, New York, New York, USA (20–22 Jun 2016)

24. Nardin, L.G., Balke-Visser, T., Ajmeri, N., Kalia, A.K., Sichman, J.S., Singh, M.P.: Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. Knowledge Engineering Review **31**(2), 142–166 (2016). https://doi.org/10.1017/S0269888916000023, https://doi.org/10.1017/S0269888916000023
25. Noriega, P.: Agent-mediated auctions: the fishmarket metaphor. Ph.D. thesis, Universitat Autònoma de Barcelona (1997)
26. Olson, M.: The Logic of Collective Action: Public Goods and the Theory of Groups. Harvard University Press, Cambridge, Massachusetts (1965)
27. Pérolat, J., Leibo, J.Z., Zambaldi, V., Beattie, C., Tuyls, K., Graepel, T.: A multi-agent reinforcement learning model of common-pool resource appropriation. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
28. Peysakhovich, A., Lerer, A.: Prosocial learning agents solve generalized stag hunts better than selfish ones. In: International Conference on Autonomous Agents and MultiAgent Systems. p. 2043–2044. AAMAS '18, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2018)
29. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research **22**(268), 1–8 (2021)
30. Singh, S., Barto, A.G., Chentanez, N.: Intrinsically motivated reinforcement learning. In: International Conference on Neural Information Processing Systems. p. 1281–1288. NIPS'04, MIT Press, Cambridge, MA, USA (2004)
31. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge, MA, USA, second edition edn. (2018)
32. Terry, J.K., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sullivan, R., Santos, L., Perez, R., Horsch, C., Dieffendahl, C., Williams, N.L., Lokesh, Y.: Pettingzoo: A standard api for multi-agent reinforcement learning. In: Advances in Neural Information Processing Systems (2021)
33. Ullmann-Margalit, E.: The Emergence of Norms. Oxford University Press (1977)
34. Verhagen, H.: Norm Autonomous Agents. Ph.D. thesis, Stockholm University (07 2000)
35. Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley Publishing, 2nd edn. (2009)
36. Zhang, K., Yang, Z., Başar, T.: Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms, pp. 321–384. Springer International Publishing (2021)