Emergence of Cooperation in N-Person Dilemmas through Actor-Critic Reinforcement Learning

João Vitor Barbosa
Escola Politécnica
Universidade de São Paulo
São Paulo, Brasil
joao.vitor.barbosa@usp.br

Anna H. Reali Costa
Escola Politécnica
Universidade de São Paulo
São Paulo, Brasil
anna.reali@usp.br

Francisco S. Melo
INESC-ID and Instituto Superior
Técnico, Universidade de Lisboa
Porto Salvo, Portugal
fmelo@inesc-id.pt

Jaime S. Sichman
Escola Politécnica
Universidade de São Paulo
São Paulo, Brasil
jaime.sichman@usp.br

Francisco C. Santos
INESC-ID and Instituto Superior
Técnico, Universidade de Lisboa
Porto Salvo, Portugal
franciscocsantos@tecnico.ulisboa.pt

ABSTRACT

The study of the emergence of cooperation remains an open challenge for many areas of knowledge. This problem may be conveniently formalised using Game Theory and an Iterated N-person dilemma games. In this work we investigate the emergent learning dynamics of this kind of problem using Reinforcement Learning (*RL*). We simulate decision-making in N-person dilemma games with players with different levels of sophistication concerning their learning policies and observation levels. We show that the combination of a simple Actor-Critic *RL* architecture with a state space that includes the number of agents who cooperated in the previous round can offer sufficient conditions for cooperation to thrive. This result is shown to depend on the size of the group and the strength of the dilemma. Moreover, cooperation is shown to increase with low exploration and learning rates while decreasing with significant discounting of future rewards. Overall, our results suggest that for each dilemma, an appropriate selection of state space and learning policy ensures coordinated efforts within a multi-agent system composed of adaptive self-interested agents.

KEYWORDS

Reinforcement Learning; Game Theory; Multi-Agent Systems; Public Goods Games.

1 INTRODUCTION

Benefits of cooperation are abundant in nature. One of the reasons why the early *Homo Sapiens* individuals replaced the physically stronger *Neanderthals* is the superior social capacities of the former over the latter [12]. Argentinian Ants can work together even from different colonies, their high level of cooperation [28] allows them to beat many other species in competition for resources [11].

However, cooperation is not easily achieved. There are obstacles to cooperation that only few species are able to overcome. One model that illustrates well this dichotomy is the Prisoner's Dilemma game (*PD*). In this game there are two players; if both cooperate they split the rewards equally, if only one cooperates it wastes its efforts and loses its rewards to the other player, if no one cooperates they have no gains. Therefore, the obstacle to achieving cooperation in this model is the conflict between what is best for the group and what is best for the individual. The game where agents play *PD* repeatedly multiple rounds is called the Iterated (or repeated) Prisoner's Dilemma (*IPD*); the iterated *NPD* is the generalisation of the *IPD* for games with more than two players.

In order to answer which factors stimulate cooperation among players, this paper proposes a set of experiments with agents that behave similarly to animals in the iterated *NPD*.

One way of approximating animal behaviour is assuming that they make decisions based on what they learn. They try out different approaches and nature punishes or rewards their behaviour; they thus use this information to improve future decisions [23, 25]. One class of algorithms inspired by this idea is called Temporal Difference Reinforcement Learning (*TD-RL*). Reinforcement Learning (*RL*) means the agents learn through repetition, punishment and reward (i.e., through trial and error). Temporal Difference (*TD*) means that learning takes place by comparing the agents current information with that observed in the immediate future. *TD-RL* algorithms achieve this by measuring not only the quality of the current action but also if that action leads to a state where it is possible to get more rewards in the next iterations. Another key aspect of learning through trial and error is to balance exploration and exploitation. The first is responsible for seeking better alternatives, and the latter is responsible for taking advantage of acquired knowledge to get high rewards.

In this paper we seek to answer the following four questions:

(1) Can *RL* agents achieve widespread cooperation when playing *NPD*? What makes it difficult for them to achieve?
(2) How do the learning parameters impact cooperation?
(3) What is the role of cognition in the emergence of cooperation among *RL* players playing *NPD*?
(4) What *RL* players learn when playing *NPD*? What the most cooperative *RL* player learn?

The first question focuses on the game parameters, the size of the group and how much wealth the group generates, to find if there is a combination of them in which the players converge to widespread cooperation. Then, the second question addresses the learning parameters, how fast they learn and how valuing future gains affect the cooperation rates. With those parameters set, in addressing the third question we define a set of *RL* players with different cognition levels to investigate how much each one of them cooperates in homogeneous groups. Finally, in the last question, we analyse the strategies the players learn in particular, the learned policies that lead to higher cooperation rates.

This approach (see also, e.g.,[2, 5, 21, 26, 27]) does not aim to analyse the convergence or optimality as most studies of reinforcement learning in game theory settings (see, e.g., [15, 17]). In [4], different heuristics are used to improve the convergence to optimal Nash-equilibrium states, where players do not have incentives to change the way they are playing. On the other hand, in [31], the authors consider *RL* players that play many different games, and in some cases, the players converged to Nash-equilibrium, in others oscillated around the equilibrium, and in the *PD* achieved better-than-Nash results. Moreover, [14] proposed *RL* players that play *PD* optimally against many fixed strategies and cooperate when playing against each other. These works focus on 2-person games and study if those players converge to a previously known Nash-equilibrium state. In this work, we study how *RL* players behave when interacting with more than one player at the same time, evaluating the emerging prevalence of cooperation and which strategies individuals learn when facing a collective action problem.

This work covers the fundamental concepts to understanding these results (section 2), then defines the different players (section 3) that appear in the experiments (section 4). Finally, we compare our approach with other works (section 5) and give our conclusions (section 6).

2 FUNDAMENTALS

In this section we define the *NPD* game and provide the basics of reinforcement learning.

2.1 Defining the Game

The N-person prisoner's dilemma (*NPD*) constitutes the most used metaphor to study public goods games (PGGs): cooperators (C) contribute an amount p to the public good; defectors (D) do not contribute. The total contribution is multiplied by an multiplication factor f and the result is equally distributed between all N members of the group, irrespectively of who contributed. Hence, defectors get the same benefit of the cooperators at no cost. In the iterated *NPD*, this entire process repeats itself for multiple rounds. The outcome of the game may differ from round to round, as individuals can base their decision to contribute on multiple criteria.

The reward function for *NPD* can be formalised by

$$R(D) = \frac{fkp}{N}, \quad R(C) = R(D) - p, \tag{1}$$

where k is the number of cooperators, f is the public good multiplier, p is the donation to the public good, and N is the number of players.

This game is the generalisation of *PD* for many players because it has the same three possible situations: the greatest overall reward is achieved when all players cooperate, mutual defection is worse than mutual cooperation, and in a mixed pool of actions defective players take advantage of cooperative players' efforts.

2.2 Learning by Experience

In reinforcement learning (*RL*), the interaction between the agent and the environment is described as a Markov decision process (*MDP*), defined as a tuple (S, A, P, R). In an *MDP*, at each time step t the agent observes the state of the environment, denoted as s_t and takes values in the set S, and selects an action $a_t \in A$. The state should include all relevant information for the agent selecting its action. Depending on s_t and a_t, the agent then receives a reward R_{s_t,a_t} and the environment transitions to a new state $s_{t+1} \in S$ according to the probabilities in P (usually unknown). The goal of the agent is to select the actions to maximise the total expected discounted reward,

$$V = \mathbb{E}\left[\sum_t \gamma^t R_{s_t,a_t}\right],$$

where γ is a scalar discount factor in $[0, 1)$. The goal of the agent is to determine a *policy*, π, that maps states to actions and maximises the value V above. During learning each player follows an exploration policy, that chooses actions based on its current knowledge. Given enough time, the exploration policy converges to a fixed learned policy, also referred to as learned strategy.

The quality of a state-action pair (s, a) in terms of the aforementioned long-term goal is represented through a number, $Q^*(s, a)$, that can be computed using a number of algorithms. In [14] and in this work the *RL* algorithm used is the *SARSA* algorithm, whose updating rule for the value of each pair action/state is:

$$Q_{s_t,a_t} \leftarrow Q_{s_t,a_t} + \alpha(R_{s_t,a_t} + \gamma Q_{s_{t+1},a_{t+1}} - Q_{s_t,a_t}), \tag{2}$$

where Q_{s_t,a_t} is the current quality of action a in state s, α and γ are the learning rate and the discounting factor, respectively; α configures how fast the agent learns, while γ discounts the value of future rewards, the higher γ the more important are future rewards for the agent. It is possible to arrange the Q_{s_t,a_t} in a table, with states as rows and actions as columns, this table is called Q-value table.

3 METHODOLOGY

In previous section, we introduced the game and the learning algorithm. By varying N and f we analyse the conditions in which cooperation prevails (see first question). Regarding the second question, to find out how the learning process impact the cooperation of the group, we investigate the impact of the parameters α and γ.

While previous section sets the foundation for answering the two first questions, this section introduces the theoretical framework for developing the last two. In the following subsections we define each player's level of perception and action selection method, from the simplest to the most complex, besides that, we show how to extract from the Q-value table what these players learn.

3.1 Perceiving more

The state-space limits the policies the player can learn. Hence, the state space can be associated with the "cognitive capabilities" of the players: the larger the state space, the more complex policies it can learn; so, the higher is its cognition. Here we define 4 different types of players, also called agents, each of them with a larger state space size than the previous one: *MemoryLess*, *MajorTD4*, *SelflessLearner* and *LevelLearner*.

The two first players are inspired, respectively, in *TD1* and *TD4* from [14]. *MemoryLess* (and *TD1*) has only one state and is expected to defect always, since it does not know anything from previous rounds. The *MemoryLess* player is going to be used as a baseline for the other players.

MajorTD4 has the exactly same state space as *TD4*: it knows its action in the last round a_{t-1} and the action of the opponent(s) in the last round $\bar{a}_{t-1}$, which sets up four states, $S = \{a_{t-1}\bar{a}_{t-1} : CC, CD, DC, DD\}$. However, for *TD4*, $\bar{a}_{t-1}$ is the single opponent last action and, for *MajorTD4*, $\bar{a}_{t-1}$ is the most frequent action executed by the opponents in the last round; both choose C over D if tied. *MajorTD4* receives this name thanks to its similarities with *TD4* and its dependence on the majority of opponents' last actions.

The other two players are based on the idea that it is better to know precisely how many players cooperated in the last round. The name *LevelLearner* comes from this idea of knowing every 'level' of cooperation. Besides how many individuals contributed in the previous round, this player also remembers its last action as *MajorTD4.* As the number of states of *LevelLearner* increases quickly with the number of players, we designed *SelflessLearner* that, differently from *MajorTD4* and *LevelLearner*, does not know its own last action, and has a space state size between the other two.

The state spaces sizes of *MemoryLess* and *MajorTD4* are independent of other parameters, and they are, respectively 1 and 4. However, for the other two players this size varies with the number of players. Since the number of cooperators may vary from 0 to N, the number of possible states for the *SelflessLearner* is $N + 1$. Since *LevelLearner* knows its own action, which has two possible values (C or D) and there are two unreachable states (DN and C0), its state space size is $2(N + 1) - 2 = 2N$. Resuming, for $N = 5$ the state spaces sizes of each of these two players are $|\{0, 1, 2, 3, 4, 5\}| = 6$ and $|\{D0, D1, D2, D3, D4, C1, C2, C3, C4, C5\}| = 10$, respectively.

3.2 Choosing Smartly

SARSA is a on-policy algorithm because it uses the exploration policy to approximate the rewards of the next state. This means that the exploration policy has great impact on the algorithm performance and on what it learns on the Q-value table.

One commonly used exploration policy is ϵ-greedy, that is the one used in [14]:

$$\pi_\epsilon(s) = \begin{cases} argmax_a Q(s,a), & \text{with probability } (1-\epsilon); \\ argmin_a Q(s,a), & \text{with probability } \epsilon. \end{cases} \tag{3}$$

It is greedy because it chooses the action with greater value for the current state with a high probability $1-\epsilon$ and chooses randomly any other action with probability ϵ, that is the exploration factor. This policy has this explicit factor to regulate agent's exploration. Formally, the ϵ-greedy policy for *IPD* and *NPD* is shown in Equation (3). Since in these games there are only two possible actions, choosing randomly any other action is just selecting the other one. In the case that the two actions have the same value in the table for a state, the agent chooses one randomly, including at the start when the whole Q-value table is initialised with zeros.

Exploration allows the player to transit through many states, which enhances learning by enabling greater exploration of the state space and thus finding the best policy.

Hence, it is urgent to increase cooperation without lowering ϵ so much. One solution for that is to have a high value of ϵ in the beginning of the game and decrease the value of ϵ through time. It is possible to define two other policies with dynamic decreasing ϵ: one decays by a linear function, the other by a logarithmic one. The linear function is given by:

$$\epsilon_{lin} = \frac{\epsilon_0}{NR+1}, \tag{4}$$

where ϵ_0 is the initial value of the exploration factor and *NR* is the number of rounds already played.

Similarly, a logarithmic decreasing ϵ-greedy policy is:

$$\epsilon_{log} = \frac{\epsilon_0}{ln(NR+2)}. \tag{5}$$

Another option is to use exploration policies that do not have an exploration factor (although they allow the agent to explore). One way of doing that is with probability distribution functions like Boltzmann that uses the Q-value table to calculate the probabilities of choosing each action in the current state, as stated in:

$$p_a(s) = \frac{e^{\beta Q(s,a)}}{\sum_{a' \in A} e^{\beta Q(s,a')}}. \tag{6}$$

In this equation, β is a constant that changes the shape of the function. An agent following this exploration policy sorts its action based on these probabilities at each time step. Note that $p_D + p_C = 1$ at any time. Since the Q-value table starts with all entries equal to zero, before simulation starts $p_D = p_C = 0.5$, this means that at the beginning the agent will choose actions randomly like the ϵ-greedy policies.

Finally, the last exploration policy tried out in this work is an actor-critic policy. Actor-critic agents learn two different things while playing. The first is the critic that is how good an action is for each state, the other is the actor that learns how to choose actions given the critic. One simple way of doing this is to use a Bernoulli distribution for each state, represented in:

$$p_{a,s} = \begin{cases} p_s, & \text{if } a = C, \\ 1 - p_s, & \text{if } a = D. \end{cases} \tag{7}$$

where p_s is the probability to cooperate in state s. The agent will learn a vector of probabilities $\mathbf{p} = [p_{s_1}, p_{s_2}, \ldots, p_{s_n}]$, where $n = |S|$ and the probabilities are initialised with 0.5.

It is then possible to rewrite the Equation (2) as:

$$\begin{gathered} Q_{s_t,a_t} \leftarrow Q_{s_t,a_t} + \alpha\delta, \\ \delta = r_{s_t,a_t} + \gamma Q s_{t+1}, a_{t+1} - Q_{s_t,a_t}. \end{gathered} \tag{8}$$

The factor δ is then used to update the value of each element in the vector of probabilities according to:

$$\Delta p_s = \alpha_p \delta(y^t - p_s^t), \tag{9}$$

where α_p is the learning rate of this exploration policy, y^t is the value of action selected in round t (it is 1 if $a_t = C$ and 0 otherwise) and p_s^t is the current value of the probability of cooperating in the current state s_t. This is a linear actor-critic policy, simplified for $|A| = 2$, as specified in [30].

3.3 Strategy Identification and Dynamics

At the beginning, agents play randomly, independently of the exploration policy they are following. However, when they start to learn they start trying out strategies, until they find the best strategy for their environment. Nevertheless, the other players are part of that environment, so when a player starts playing a different strategy, it changes the environment for the others, which may cause them to change strategy in response. This happens because *RL* agents always try to learn the optimal strategy against the other players. The search for the optimal response for the environment is what creates these dynamics.

First we need to define what is a strategy, that is only a sequence of actions, that usually can be translated into a rule, like always cooperate (*ALLC*), always defect (*ALLD*), alternate defection and cooperation (*ALT*), start cooperating then copy opponent's action (*TFT*), defect if the opponent defected twice in a row and cooperate otherwise (*TF2T*), repeat last action if in mutual cooperation or defected against cooperation and flips last action otherwise (*WSLS*).

A strategy h_1 is optimal against strategy h_2 if there is no other strategy that has greater expected reward playing against h_2. Examples of optimal strategies are abundant in the literature: *ALLC* is optimal against *TFT*, *ALLD* is optimal against *ALLC*, *TFT* is optimal against *ALLD* and *ALT* is optimal against *TF2T*. By analysing what strategies the players learn at the end of simulation it is possible to explain why some player cooperates more than others and what is the reasoning about the player's decisions. On the other hand, by checking how many times a player changes strategy during learning it is possible to measure how much the player is exploring alternative strategies.

To determine what strategy an *RL* agent is playing at a given point it is necessary to look at its Q-value table. For the greedy policies and Boltzmann policies, the Q-value gives all the information to determine how the player is playing. For actor-critic policy it is necessary to look at the probabilities learned by the actor. Another thing to notice is that greedy policies only play deterministic strategies, while Boltzmann and actor-critic may play stochastic strategies that define probabilities of playing each action in each state. For simplicity only the strategies learned by *MajorTD4* are analysed.

For this player, the set of states is $S = \{CC, CD, DC, DD\}$. Hence, strategies can be defined using four bits, b_3, b_2, b_1, b_0, where each bit corresponds to the action the player chooses in a given state[1]. By generating every possible value, there are 16 possible strategies and many of them were already mentioned, for example: $ALLD$ = 0000 = $S00$, ALT = 0011 = $S03$, TFT = 1010 = $S10$, $ALLC$ = 1111 = $S15$. Then, to extract which strategy a player is currently playing, we check at that moment the action with the highest quality for each state and set the corresponding bit accordingly. Finally, whenever an agent reevaluates the best action for a given state, it changes its strategy and we can measure exploration by the frequency that players change strategy on average.

4 RESULTS

In order to assess improvement in cooperation, it is necessary to establish an initial configuration from which variations are created by changing one trait at a time. The basic configuration is an *NPD* game with $f = 2$ and five *MajorTD4* players, all with $\alpha = 0.05$, $\gamma = 0.9$, and the ϵ-greedy learning policy with $\epsilon = 0.001$. The values for α and γ are based in [14], and, as in *NPD* is expected even higher sensibility to the exploration factor, this baseline has a smaller value of ϵ than the one used in [14], $\epsilon = 0.01$.

Besides that, there are two fixed parameters for *NPD*, the starting resources and the cooperation cost; the first is fixed in 20 and the second fixed in 1. Those parameters open a whole new set of possible experiments, regarding wealth distribution and its impact on cooperation, for example. However, this work does not evaluate the influence of these parameters.

The experiments are arranged to investigate different effects, each of them is driven to answer a question. Each study case has two phases: the learning phase and the execution phase. In the first the players learn and in the second the players only execute their learned policies and we measure the cooperation rates. In the learning phase, we create N identical and independent players that learn by playing with each other through 20000 rounds, then we sample one of them to be in the execution phase. This process is repeated N times, so we get N players in the next phase. In the execution phase these players play through 1000 rounds without learning, at the end the cooperation is measured over the last 100[2] rounds for each player. At this point we have the result of one game. This whole process, learning phase and execution phase, is then repeated 1000 times. So at the end of 1000 games we have the average cooperation rate and its standard error. This two values are used to create a single point of each figure of this work.

We perform three studies and one analysis: the Environment Study, the Learning Study, the Cognition Study, and the Strategy Analysis. The Environment Study checks if there is a scenario in which players cooperate and proposes a challenge to test which player cooperates the most. The Learning Study investigate how the learning process may affect the cooperation of the group. The Cognition Study tests different players to identify the role cognition plays in their cooperation. Finally, Strategy Analysis assesses the learned policy in order to discuss the reasons behind the improvements in cooperation.

4.1 Environment Effect

There are two parameters of the game expected to impact the cooperation: the number of players N and the public goods multiplier f. These two parameters are tuned for setting an environment hard to cooperate in order to highlight the impact of agents' cognition in cooperation.

We can see in figure 1 that as the number of players increases, cooperation decreases regardless of the type of player. For cooperation, players must coordinate efforts and it is more difficult to coordinate a larger group. However, *MajorTD4* shows an interesting phenomenon: groups with an even number of players cooperate more than groups with odd numbers; since this is specific to that player, it is probably a consequence of the approximation of opponents' behaviour to that of the majority (e.g. for both N = 5 and N = 6, the majority is 3).

Figure 2 shows that as the multiplier of public goods increases, cooperation increases sharply. The public goods multiplier reflects the amount of resources in the environment, and the smaller the

[1] Bit is 1 for C, and 0 for D.

[2] Even though in the execution phase there is no more learning, players can take a few moves to converge to a stable state, hence we take the average of the cooperation only in the last 100 turns.

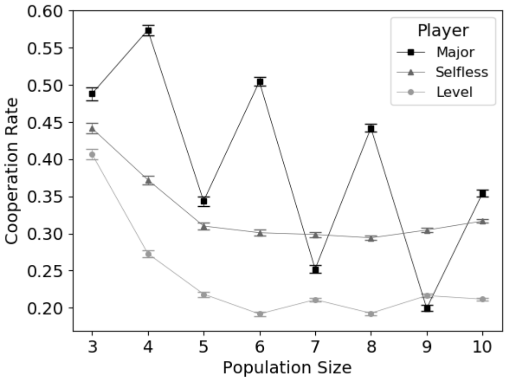

Figure 1: Cooperation rates of different players playing *NPD* ($f = 2$) following ϵ-greedy policy with $\epsilon = 0.001$ for different population sizes (*N*).

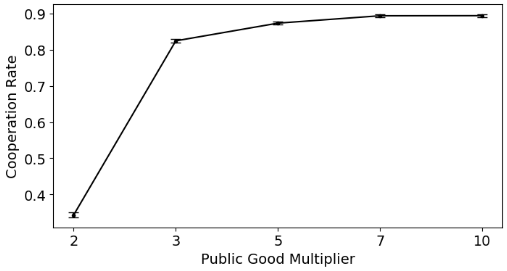

Figure 2: Cooperation rates for five *MajorTD4* playing *NPD* following ϵ-greedy policy with $\epsilon = 0.001$ for different values of public good multiplier *f*.

harder it is to cooperate. In a resource-rich scenario, with high f values, even poor cooperation yields rewards that outweigh the cost of cooperating, and this lessens the fear of being exploited by other non-cooperating players, resulting in increased cooperation.

The scenario with $f = 2$ and $N = 5$ will be used as a baseline in the following experiments because this is a hard enough setup to cooperate, with a relatively small number of players, which makes simulations less computationally costly.

4.2 Learning rates and discounting factors

There are two parameters that shape players behaviour: the learning rate (α) and the discounting factor (γ).

The learning rate sets the pace of learning. The smaller the α the more time the agent needs to learn and the more it accumulates knowledge through time. The higher the α the faster it learns and more frequently old knowledge is discarded to make room for new one. As expected a small α boosts cooperation, as shown in figure 3.

The discounting factor γ penalises rewards that are in the future. A player with γ close to one prioritises future rewards more than an agent with low values of γ. Figure 4 shows that players cooperate more when they value long-term gains over immediate gains.

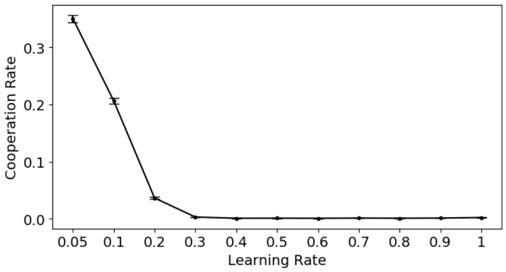

Figure 3: Cooperation rates for five *MajorTD4* playing *NPD* ($f = 2$) following ϵ-greedy policy with $\epsilon = 0.001$ for different learning rates α.

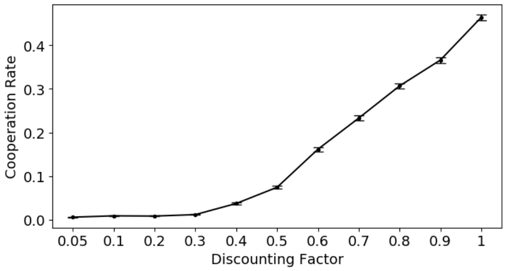

Figure 4: Cooperation rates for five *MajorTD4* playing *NPD* ($f = 2$) following ϵ-greedy policy with $\epsilon = 0.001$ for different discount factor values γ.

Although ϵ only appears in ϵ-greedy exploration policies, it has a huge impact on cooperation. As shown in Figure 5, regardless of state space, ϵ greatly affects cooperation, ranging from less than 10% to almost 80% cooperation with *MajorTD4*.

4.3 The role of cognition

Regarding cognition, it is noted that the dimension of the state space is due to the player's perception – the larger and more detailed the state space, the greater the agent's perception of the environment. Based on state space, the exploration policy makes decisions about player actuation.

Figure 5 shows how the rate of cooperation varies with respect to the value of ϵ and to the different agents with different perceptions of the environment. The results of *Memoryless* and *MajorTD4* are expected: the increase in state space allowed a huge improvement in cooperation rates, although at the cost of decreasing the exploration degree. However, the reduction in *SelflessLearner* and *LevelLearner* cooperation is not expected. It was expected that the increase in state space size would enhance cooperation. Since the agent with the best results is the *MajorTD4*, the next experiments tries out different exploration policies with this agent in order to enhance cooperation without decreasing exploration significantly.

Decreased exploration harms the learning process: it makes players less adaptive and more likely to stay in sub-optimal states. One way to measure that is to check in average how many times the player changes the strategy during learning. The more strategy changes, the more the player explored alternatives. The average

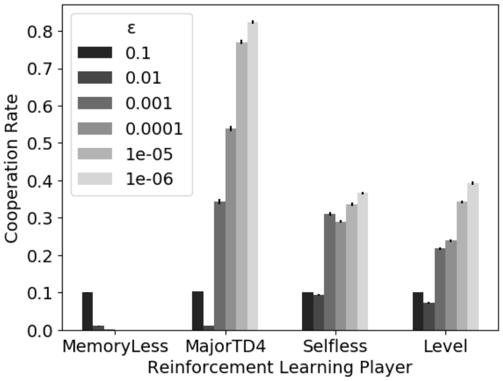

Figure 5: Cooperation rates for five agents playing *NPD* (f = 2) following ϵ-greedy policy for different values of ϵ and different information levels.

Policy		Strategy Changes	
Algorithm	Parameter	Average	Standard Deviation
ϵ-Greedy	$\epsilon = 0.01$	104.91	142.96
ϵ-Greedy	$\epsilon = 0.001$	3.812	14.92
ϵ-Greedy	$\epsilon = 0.0001$	1.84	1.90
Linear-Dynamic-ϵ	$\epsilon_0 = 0.1$	1.56	1.32
Log-Dynamic-ϵ	$\epsilon_0 = 0.001$	1.83	1.44
Boltzmann	$\beta = 0.01$	9.26	4.62
Actor-Critic	$\alpha_P = 1$	2.82	4.32

Table 1: Average number of changes on strategy for 1000 *NPD* games (f = 2), *MajorTD4* and 5 players during learning for different policies.

strategy changes for *MajorTD4* following different exploration policies are in table 1. Notice how the strategy changes decrease when ϵ is decreased in ϵ-greedy with static ϵ.

Then, the goal is to find the exploration policy that leads to higher cooperation rates and allows exploration of at least ϵ-greedy policy with $\epsilon = 0.0001$ level. After testing each policy for many variations of its parameters, the best configurations was selected. Their strategy-changes are in table 1 and their cooperation rates in figure 6.

The only policy that strictly increases cooperation and strategy exploration is the actor-critic policy, that stands out as the best result. The ϵ-greedy with linear decreasing ϵ also has cooperation improvement but at slightly less exploration. While ϵ-greedy with logarithmic decreasing ϵ and Boltzmann increase exploration at the expense of small decrease in cooperation rates.

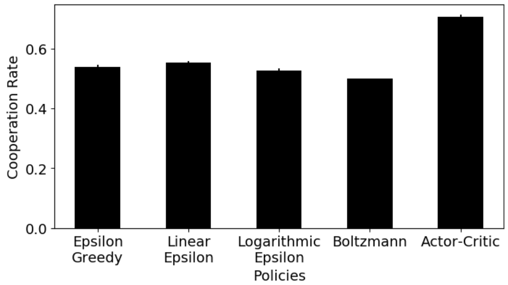

Figure 6: Cooperation rates for five *MajorTD4* playing *NPD* (f = 2) for different policies.

Actor-critic exploration policy obtained the best result with *MajorTD4*. This was expected because it allows the agent to learn slower which in turn reduces the impact of randomness during learning. Then, we checked how this exploration policy would react with players of different state space sizes, the results are in figure 7. The best result with this exploration policy was with *LevelLearner* ($\alpha_P = 0.05$) and achieved cooperation over 80% and high exploration: the value for strategy changes is 25.34 ± 8.28.

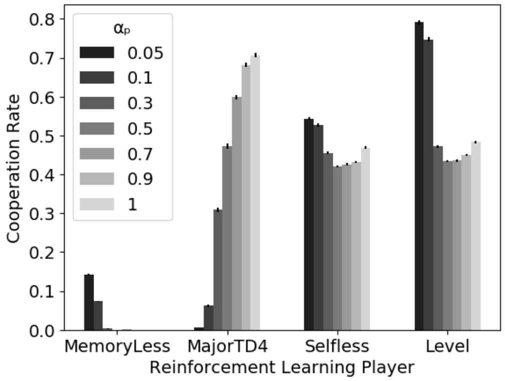

Figure 7: Cooperation rates for five agents playing *NPD* (f = 2) following actor-critic policy for different values of α_P and different information levels.

These results show how cognition is central to cooperation: it allowed the improvement from *MemoryLess* to *MajorTD4*, regarding state space, from ϵ-greedy to actor-critic, regarding exploration policy, and from *MajorTD4* to *LevelLearner* regarding state space again. Although increasing cognition in these two cases improved cooperation, fixing one state space and varying exploration policies or fixing an exploration policy and varying the state space does not reveal a steady improvement in cooperation. The improvement in cooperation seems attached to the careful combination of both dimensions.

4.4 Strategy Analysis

The state space of *MajorTD4* has the advantage of being easily translated into one of the 16 memory-one strategies of *IPD*. So this section focus on the strategies learned by *MajorTD4* and also explores the probabilities of cooperation learned by players following actor-critic.

There is significant difference between the strategies learned by *MajorTD4* when following ϵ-greedy and when following actor-critic. The first is the number of players that learn *TFT*. The second big difference is the number of *S05* strategies. *S05* is interesting because it is the *TFT* upside down: instead of copying the opponent's last action, it plays the opposite of opponent's last action. This means that when few players are cooperating, the player cooperates, and when many are cooperating, it defects. The emergence of this strategy may be responsible for the boost in $S15 = ALLC$ frequency with actor-critic policy. Besides those differences, both configurations have a low number of *ALLD* and a high number of *ALLC*, although with actor-critic the frequency of *ALLC* is considerably higher.

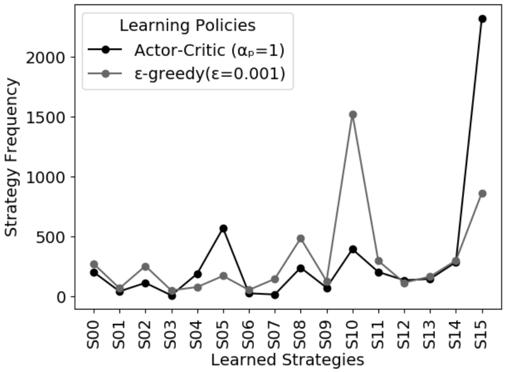

Figure 8: Frequency of strategies learned through 1000 games by five *MajorTD4* playing *NPD* ($f = 2$) following either actor-critic ($\alpha_P = 1$) or ϵ-greedy ($\epsilon = 0.001$).

Nevertheless, actor-critic policy also learns the probabilities of cooperating in each state. The average results over 1000 games are shown on table 2a for *MajorTD4*, on table 2b for *SelflessLearner* and on table 2c for *LevelLearner*. While *MajorTD4* agents learn to cooperate more in high cooperation states when compared to low cooperation states, *SelflessLearner* and *LevelLearner* players learn to cooperate around 60% frequency when no one is cooperating and to not cooperate when only one or two players are cooperating. This shows that the agent learned a recover mechanism, a way of going from a state of no cooperation to a state of high cooperation. This explains the cooperation rates of figure 7 with both *SelflessLearner* and *LevelLearner*. This mechanism resembles *WSLS*, however during learning players following actor-critic do not learn this strategy. It seems they end up differentiating in the case of *MajorTD4* and *SelflessLearner*, what explains the high standard deviations, or converging to a mixed strategy in the case of *LevelLearner* to create this mechanism.

The recover mechanism of *LevelLearner* following actor-critic stabilises with four cooperators (C4) and one defector (D4). But if by chance one of the cooperators decides to stop cooperating, it goes to state D3 and the other cooperators go to C3. Following this path the group stops cooperating very quickly, when this happens they try to start cooperating together, until they reach stability again.

Overall, the configuration that have the higher cooperation rates are the one whose cognition level allowed the agents to develop mechanism to recover from a state of widespread defection.

Table 2: Average probability to cooperate and average deviation of actor-critic with (a) *MajorTD4* ($\alpha_P = 1$), (b) *SelflessLearner* ($\alpha_P = 0.05$) and (c) *LevelLearner* ($\alpha_P = 0.05$) for each state of *S*.

State	DD	DC	CD	CC
Average	0.3758	0.3949	0.4280	0.8538
St. Dev.	0.2128	0.2069	0.1923	0.2232

(a) *MajorTD4*

State	0	1	2	3	4	5
Average	0.6317	0.0099	0.0474	0.1700	0.7801	0.6792
St. Dev.	0.0353	0.0151	0.0209	0.1368	0.4037	0.1529

(b) *SelflessLearner*

State	D0	D1	D2	D3	D4	C1	C2	C3	C4	C5
Average	0.5808	0.0588	0.0159	0.0158	0.0165	0.1833	0.2439	0.6138	0.9995	0.6033
St. Dev.	0.0244	0.0275	0.0118	0.0209	0.0568	0.0625	0.0365	0.0320	0.0041	0.0700

(c) *LevelLearner*

5 DISCUSSION

It was expected some level of cooperation for *RL* players in *NPD*, since [14] shows cooperation among *RL* players for a specific set of parameters in *IPD*. Besides that, *RL* algorithms can improve cooperation in evolutionary settings [24] and in [9], broad cooperation is achieved for different levels of language expressiveness.

In this last work, it is explored how the expressiveness of a language impact cooperation in *NPD*. Two different languages to represent agents' strategies are studied: finite automata and adaptive automata. The latter is the more expressive. When simulation results of both languages were compared, the authors concluded that in both there was a convergence to broad cooperation between the agents, without significant statistical difference on society's long-run welfare; however, when simulation started with widespread defection a higher welfare was obtained with players using adaptive automata language. In other words, a more expressive language allows the agents to recover better from defection, similarly, a richer cognition level in this work allows agents to learn strategies that recover quickly from defection. Experiments with people also show that increasing the amount of information of the group improve cooperation [29].

A possible strategy to improve cooperation is, precisely, to recover from defection, as the strategy *WSLS* does in *IPD* and *LevelLearner* following actor-critic does in *NPD*. However, there are other ways to improve cooperation. One alternative is to improve cooperation by having players in the group whose objective is to improve cooperation, as proposed by [10]. The strategy *S05* seems to have this role in *NPD* and *MajorTD4* following actor-critic learns it. This strategy increases the level of cooperation of the group when few are cooperating, what may incentives other to adhere to cooperation and takes advantage when there are a lot of cooperators. Another way to incentive players to cooperate is to reciprocate, like *TFT*. However, in this work, the higher the frequency of agents playing *TFT*, the lower the cooperation rates.

The bad performance of reciprocity in social dilemmas is expected [8]. Differently from *IPD* that it is possible to punish the opponent for defecting by defecting as well, in public goods games as *NPD*, there is no way for punishing a single defector without punishing the rest of the group. This explains why cooperation is harder in larger groups, as N increases the impact of a single defector in the group overall cooperation decreases, what favours defection.

The results of figures 5 and 7 show how a proper definition of state space can influence the cooperation of the group. This result is corroborated by [3], that achieved improvements in group coordination by improving state space carefully, since this process can hinder learning. This can explain in part the cooperation rates in figure 5, with *MajorTD4* as the most cooperative player. In [7, 22] limited cognition levels appear as the configuration that cooperate the most, when studying past reputation and evolutionary settings respectively. What may indicate that there are specific broad cooperative stable configurations of different cognition levels, as *MajorTD4* following ϵ-greedy and *LevelLearner* following actor-critic.

Finally, when [8] studied the dynamics of social dilemmas, they found that there are two stable states: one of widespread cooperation and one of widespread defection. Those states are not static, the cooperation rates of the groups stay floating around one of them, these small fluctuations are due to uncertainty of the players or when some players estimate wrongly the level of cooperation. Nevertheless, during long runs, stronger fluctuations may occur and bring the group from one stable state to the other. The consequence is that it is common for a group's behaviour in a social dilemma to remain the same for long periods, but when it changes, it changes quickly. This corroborates the results of *SelflessLearner* and *LevelLearner* following actor-critic. The recover mechanism they learn also has this property of moving quickly from widespread defection to widespread cooperation and vice versa. The difference is that the recover mechanism does not allow the group to stabilise in widespread defection.

6 CONCLUSIONS

In our experiments, we observed that increasing the public goods multiplier it is possible to achieve widespread cooperation. As the public goods multiplier decreases and the number of players increase, players stop cooperating. Hence the importance of developing more complex strategies to improve group cooperation, even in non-abundant environments.

We show that the parameters α, γ and ϵ have a significant impact on cooperation: the low values of learning rate α and exploration ϵ mean that changes are taken slowly or infrequently, so that knowledge is accumulated over time, giving time for the environment to adjust; a high discounting factor γ suggests that cooperation thrives when individuals value long-term gains over short-term ones.

Interestingly, our results also suggest that cognition plays a key role in the learning process and the emerging levels of cooperation. Indeed, cooperation emerges as a result of the combined effect of increasing the state space size and adopting more complex policies. By only enhancing cognition, we could move from 35% of cooperation, achieved by our baseline, *MajorTD4* following ϵ-greedy ($\epsilon = 0.001$), to 80% of cooperation with *LevelLearner* following actor-critic ($\alpha_P = 0.05$). Besides the improvement in cooperation, there is also a considerable improvement in exploration, from 3.81 ± 14.92 to 25.34 ± 8.28 strategy changes. This means that the player with higher cognition tries out more options and finds a better strategy consistently, as the low standard deviations on table 2c suggest.

Moreover, different learning processes lead to distinct behavioural patterns. For instance, two popular approaches to improving cooperation rely on incentives for others to cooperate, as *TFT* [1], and recover from mutual defection, as *WSLS* [16]. When analysing what the *MajorTD4* variations learned, the most important strategies are *S05*, $S15 = ALLC$ and $S10 = TFT$. Cooperation is higher for high frequencies of *S05* and *ALLC*, and low frequencies of *TFT*. Nevertheless, the dynamic of these groups is based on *S05* giving incentives to other players to play *ALLC*, by cooperating when only a few do. On the other hand, *SelflessLeaner* and *LevelLearner* following actor-critic, focus more on developing a recover mechanism that allows the group to move quickly from widespread defection to widespread cooperation. These differences may also explain why the behaviour of those agents is so different when varying α_P, as shown in figure 7.

Our work also highlights the learning of different strategies when moving from pairwise interactions to group interactions. While in 2-player is dominated by reciprocity (e.g., TFT strategies), in N-person interactions strategies focus on the recovery from mutual defection (e.g., WSLS). TFT-like strategies are solely adopted by *RL* players in *N-player* interactions when a few alternatives are explored, or when converging to low cooperation rates, which may indicate that this is indeed a sub-optimum strategy for the *NPD*. When agents have sufficient information and reasoning capacity, they explore more and converge to a recover mechanism strategy. These strategies overcome one of the reasons to defect in defection dominance dilemmas: the fear of being exploited [13]. However, it does not solve the other, the temptation to explore others. This result in groups that achieve a safe state to cooperate by allowing a small number of free riders.

The dilemma addressed here is one among many of relevance to human cooperation. Future work may address different forms of non-linear returns [18, 19], structured populations [5, 20, 27], and more complex *RL* algorithms [6] and exploration policies, closing the gap between this simple, yet illuminating models, and real-life scenarios in which conflicts of interests and free-riding prevail.

7 ACKNOWLEDGEMENTS

FSM and FCS acknowledge the support from FCT-Portugal through grants PTDC/EEI-SII/5081/2014 (FCS), PTDC/MAT/STA/3358/2014 (FCS), and UID/CEC/50021/2020 (FSM and FCS). AHRC also gratefully acknowledge financial support from CNPq-Brazil through grants 425860/2016-7 and 307027/2017-1.

REFERENCES

[1] Robert Axelrod. 1984. *The Evolution of Cooperation.* Basic, New York.

[2] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. 2015. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research* 53 (2015), 659–697.

[3] Jen Jen Chung, Damjan Miklić, Lorenzo Sabattini, Kagan Tumer, and Roland Siegwart. 2020. The impact of agent definitions and interactions on multiagent learning for coordination in traffic management domains. *Autonomous Agents and Multi-Agent Systems* 34, 1 (21 Jan 2020), 21. https://doi.org/10.1007/s10458-020-09442-1

[4] Caroline Claus and Craig Boutilier. 1998. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, July 26-30, 1998, Madison, Wisconsin, USA.* AAAI Press / The MIT Press, 746–752.

[5] Takahiro Ezaki and Naoki Masuda. 2017. Reinforcement learning account of network reciprocity. *PloS ONE* 12, 12 (2017).

[6] Jakob Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with Opponent-Learning Awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18).* International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 122–130.

[7] Julian Garcia and Matthijs van Veelen. 2018. No strategy can win in the repeated prisoner's dilemma: linking game theory and computer simulations. *Frontiers in Robotics and AI* 5 (29 8 2018). https://doi.org/10.3389/frobt.2018.00102

[8] Natalie S Glance and Bernardo A Huberman. 1994. The dynamics of social dilemmas. *Scientific American* 270, 3 (1994), 76–81.

[9] I. Guerberoff, Diego Queiroz, and Jaime Simão Sichman. 2011. Studies on the effect of the expressiveness of two strategy representation languages for the iterated n-player prisoner s dilemma. *Revue d'Intelligence Artificielle* 25, 1 (2011), 69–82. https://doi.org/10.3166/ria.25.69-82

[10] Linghui Guo, Zhongxin Liu, and Zengqiang Chen. 2017. A leader-based cooperation-prompt protocol for the prisoner's dilemma game in multi-agent systems. In *36th Chinese Control Conference (CCC).* IEEE, 11233–11237. https://doi.org/10.23919/ChiCC.2017.8029149

[11] David A. Holway, Lori Lach, Andrew V. Suarez, Neil D. Tsutsui, and Ted J. Case. 2002. The Causes and Consequences of Ant Invasions. *Annual Review of Ecology and Systematics* 33, 1 (2002), 181–233. https://doi.org/10.1146/annurev.ecolsys.33.010802.150444 arXiv:https://doi.org/10.1146/annurev.ecolsys.33.010802.150444

[12] Takanori Kochiyama, Naomichi Ogihara, Hiroki C. Tanabe, Osamu Kondo, Hideki Amano, Kunihiro Hasegawa, Hiromasa Suzuki, Marcia S. Ponce de León, Christoph P. E. Zollikofer, Markus Bastir, Chris Stringer, Norihiro Sadato, and Takeru Akazawa. 2018. Reconstructing the Neanderthal brain using computational anatomy. *Scientific Reports* 8, 1 (2018), 6296. https://doi.org/10.1038/s41598-018-24331-0

[13] Michael W Macy and Andreas Flache. 2002. Learning dynamics in social dilemmas. *Proceedings of the National Academy of Sciences* 99, suppl 3 (2002), 7229–7236.

[14] Naoki Masuda and Hisashi Ohtsuki. 2009. A Theoretical Analysis of Temporal Difference Learning in the Iterated Prisoner's Dilemma Game. *Bulletin of Mathematical Biology* 71, 8 (01 Nov 2009), 1818–1850. https://doi.org/10.1007/s11538-009-9424-8

[15] Francisco S Melo and M Isabel Ribeiro. 2008. Emerging coordination in infinite team Markov games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1.* 355–362.

[16] Martin Nowak and Karl Sigmund. 1993. A Strategy of Win-Stay, Lose-Shift That Outperforms Tit-for-Tat in the Prisoner's Dilemma Game. *Nature* 364 (08 1993), 56–8. https://doi.org/10.1038/364056a0

[17] Ann Nowe, Peter Vrancx, and Yann-Michaël De Hauwere. 2012. *Game Theory and Multi-agent Reinforcement Learning.* Springer, 441–470. https://doi.org/10.1007/978-3-642-27645-3_14

[18] Jorge Pacheco, Francisco Santos, and Max Souza. 2008. Evolutionary Dynamics of Collective Action in N-person Stag Hunt Dilemmas. *Proc. Royal Soc. B: Biological Sciences* 276 (10 2008), 315–21. https://doi.org/10.1098/rspb.2008.1126

[19] Francisco C Santos and Jorge M Pacheco. 2011. Risk of collective failure provides an escape from the tragedy of the commons. *Proc Natl Acad Sci USA* 108, 26 (2011), 10421–10425.

[20] Francisco C Santos, Marta D Santos, and Jorge M Pacheco. 2008. Social diversity promotes the emergence of cooperation in public goods games. *Nature* 454, 7201 (2008), 213–216.

[21] Fernando P Santos, Francisco C Santos, Francisco S Melo, Ana Paiva, and Jorge M Pacheco. 2016. Dynamics of fairness in groups of autonomous learning agents. In *N. Osman and C. Sierra (Eds): AAMAS 2016 Ws Best Papers, LNAI 10002.* Springer, 107–126.

[22] Fernando P. Santos, Francisco C. Santos, and Jorge M. Pacheco. 2018. Social norm complexity and past reputations in the evolution of cooperation. *Nature* 555, 7695 (01 Mar 2018), 242–245. https://doi.org/10.1038/nature25763

[23] Wolfram Schultz, Peter Dayan, and P Read Montague. 1997. A neural substrate of prediction and reward. *Science* 275, 5306 (1997), 1593–1599.

[24] Shoma Tanabe and Naoki Masuda. 2012. Evolution of cooperation facilitated by reinforcement learning with adaptive aspiration levels. *Journal of Theoretical Biology* 293 (2012), 151 – 160. https://doi.org/10.1016/j.jtbi.2011.10.020

[25] Edward Thorndike. 2017. *Animal intelligence: Experimental studies.* Routledge.

[26] Karl Tuyls, Ann Nowe, Tom Lenaerts, and Bernard Manderick. 2004. An evolutionary game theoretic perspective on learning in multi-agent systems. *Synthese* 139, 2 (2004), 297–330.

[27] Sven Van Segbroeck, Steven De Jong, Ann Nowé, Francisco C Santos, and Tom Lenaerts. 2010. Learning to coordinate in complex networks. *Adaptive Behavior* 18, 5 (2010), 416–427.

[28] Ellen Van Wilgenburg, Candice W. Torres, and Neil D. Tsutsui. 2010. The global expansion of a single ant supercolony. *Evolutionary applications* 3, 2 (Mar 2010), 136–143. https://doi.org/10.1111/j.1752-4571.2009.00114.x 25567914[pmid].

[29] Jana Vyrastekova and Yukihiko Funaki. 2010. *Market Interaction and Efficient Cooperation.* Technical Report. Njmegen Center for Economics (NiCE), Institute for Management Research, Radboud University Nijmegen, Nijmegen.

[30] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8, 3 (01 May 1992), 229–256. https://doi.org/10.1007/BF00992696

[31] Michael Wunder, Michael Littman, and Monica Babes-Vroman. 2010. Classes of Multiagent Q-learning Dynamics with ϵ-greedy Exploration. In *Proceedings of the 27th International Conference on Machine Learning, (ICML-10), June 21-24, 2010, Haifa, Israel.* Omnipress, 1167–1174.