Learning Dynamics in Populations of Actor-Critic Agents

João Vitor de Oliveira Barbosa*
joao.vitor.barbosa@usp.br
Instituto Superior Técnico - Universidade de Lisboa
Lisboa

ABSTRACT

The study of the emergence of cooperation remains an open challenge for many areas of knowledge. This problem can be conveniently formalized through the eyes of game theory and iterated N-person dilemmas. Here we investigate the learning dynamics emerging from this type of problems. We simulate decision-making in non-linear N-person dilemmas with agents portraying different levels of sophistication concerning their learning method, adopting a temporal difference learning algorithm as a baseline scenario. The results show that the combination of a simple Actor-Critic policy with a state space that allows players to distinguish how many agents cooperated in the previous round can offer a significant increase in the overall levels cooperation. These results are shown to be depend on the the nature of the dilemma, namely on the size of the group and the minimum contributions needed to produce a collective return. Cooperation is also shown to increase with low exploration and learning rates, and to decrease with the discounting of future rewards. Overall, our results suggest that, for each dilemma, a proper selection of state space and policy selection method ensures coordinated efforts within a multi-agent system made of adaptive self-regarding agents.

KEYWORDS

reinforcement learning, public goods games, game theory, multi-agent systems

1 INTRODUCTION

Benefits of cooperation are not scarce in nature. One of the reasons for the early *Homo Sapiens* have replaced the physically stronger *Neanderthals* is the superior social capacities of the first over the second [4]. Argentinian Ants can work together even from different colonies, their high level of cooperation [9] allows them to beat many other species in competition for resources [3]. But cooperation is not something easily achieved, there is an obstacle for cooperation, which only some species overcome. One model that illustrates well this dichotomy is the Prisoner's Dilemma (*PD*). In this model there are two people, 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. Hence, the obstacle to achieve cooperation in this model is the conflict of what is better for the group and what is best for the individual. The game where agents play *PD* repeatedly is called *IPD* and *NPD* is the generalization of *IPD* for more than two players. In order to answer what are the factors that enable and increase cooperation this work proposes a set of experiments with agents that behave similar to animals in *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 and they use this information to improve future decisions [6, 8]. There is a class of algorithms inspired by that, that is the Temporal Difference Reinforcement Learning algorithms. Reinforcement Learning (*RL*) means the agents learn through repetition, punishment and rewards. Temporal Difference (*TD*) means that decisions made in the present may impact on the future, those algorithms balance 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. A key aspect for learning through trial and error is to balance exploration and exploitation. The first is responsible for seeking better alternatives and the other is responsible for taking advantage of acquired knowledge to get high rewards.

This work merges two different fields of knowledge: Game Theory and Machine Learning. The first designs models and tries to find equilibrium, optimal strategies and real world applications. Since those models are usually called games, the individuals who play them are called players. The second studies how machines learn, since the machines that learn by *RL* are independent beings that interact with the environment, they are usually called agents.

Throughout this work, the terms *player* and *agent* mean the same. With this framework we answer three main questions:

(1) **Can *RL* agents achieve widespread cooperation when playing *NPD*? What does it make difficult to achieve?** The environment is composed by the game and the other players. This question focus on the game parameters in order to find if there is a combination of them in which the agents converge to widespread cooperation. Since [5] achieved cooperation with *RL* agents in *IPD* and [7] improved cooperation in evolutionary environments with *RL*, it is expected at least one configuration with widespread cooperation. After that, this parameters are tuned to achieve a challenging environment that highlights the different cooperation rates of the agents. The two parameters of the game are the number of players (N) and the public goods multiplier (f). The more players, the harder is to coordinate efforts towards cooperation, hence it is expected lower cooperation for larger groups. The public goods multiplier gives how much resources the game generates with the contribution of the cooperators, hence a high f creates an environment abundant in resources, that is expected to be easier to cooperate, a harsher and more competitive environment otherwise.

(2) **What is the role of cognition in the emergence of cooperation among *RL* playing *NPD*?** The third question focus on the agent. So, with a selected harsh scenario, vary how much information the player has at its disposal and the methodology it uses to make decisions using these information, one at a time. It is expected that increasing complexity in both aspects increases cooperation, since a higher level of cognition could improve coordination among players. An experiment made with college students, for instance, shows that increasing the information provided to players during the game improved cooperation [10]. The answer to this question includes: if the amount of information has impact in cooperation or if is the kind of information that matters, if the method to choose actions impact cooperation, if it is possible to have high cooperation without giving up to much on exploration, if there is a limit to how much cognition can improve cooperation.

(3) **What *RL* agents learn when playing *NPD*? What is the knowledge acquired by the agents that cooperate the most?** This item has two goals: explain why the results of previous question improve cooperation and give an interpretation of the results that can be translated to real scenarios. Other work already tried to improve cooperation in NPD so it is expected that the results of these work corroborate some of them. It is possible to improve cooperation by having a number of players in the population whose objective is to improve cooperation [2]. Since NPD has many players a subset of them can learn to incentive others to cooperate. Another approach is to improve cooperation by recognizing other players intention [1]. Regarding classic game theory strategies, Tit for Tat (*TFT*) and Win-Stay-Lose-Shift (*WSLS*) give insights on how to improve cooperation, the first incentive others to cooperate and the other has a mechanism to recover from mutual defection. This work answer this by analysing the most frequently learned strategies.

The results show that there is widespread cooperation for high values of f and low values of N. Reducing f already creates a challenging scenario where cooperation is improved, neither by only increasing the amount of information nor by only improving the policy for choosing actions, but by carefully selecting the right combination of the two. The most cooperative agent has over 80% of cooperation and it achieves that by developing a strategy with a recover mechanism that allows the group to move quickly from widespread defection to widespread cooperation.

2 MODEL

2.1 Defining the Game

The model to approximate the emergence and evolution of cooperation is *NPD*. In this game each player may choose to cooperate C or defect D. When a player chooses to cooperate, it donates and amount p of its resources to the public good. Then the resources of all cooperators are summed and multiplied by the public good multiplier f. Finally, the total amount is divided equally among the players independently of their actions. Hence the reward function of cooperating and defecting are very similar, the only difference is the cost p to cooperate, as appear in

$$\begin{aligned} R(D) &= \frac{fkp}{N}, \\ R(C) &= R(D) - p, \end{aligned} \tag{1}$$

where k is the number of cooperators. The reward is then added to the resources the player currently have, if a player runs out of resources it can not cooperate anymore.

This game is the generalization of *PD* for many players because it have the same three possible situations: if all cooperate is the highest global reward, mutual defection is worst than mutual cooperation and, in a mixed pool of actions, the defective players take advantage of the cooperative players efforts.

2.2 Learning Dilemmas by Experience

The definition of a TD player is a quintuple: a state space S, the possible actions A, the reward function R, the state transition and the policy. The reward function and the state transition is given by the game that the player is playing. The possible actions are C or D. The policy is the rule by which the learner chooses its actions. Finally, the state space is the possible states of the environment that the agent perceive itself into. For example, an agent may operate in a forest environment or a desert environment, thus it has a state space $S_2 = \{desert, forest\}$, and it can specialize different strategies for each state. However it cannot differentiate if it is in a boreal forest or a tropical forest, hence it can not have specialized strategies for working in each of them, even though being different, if differentiate this two situations is important we can design a larger state space $S_3 = \{desert, boreal, tropical\}$ to include this. Because of this, S can be associated with the information the agent has at its disposal, when enlarging the state space from S_2 to S_3 the agent gains the information that there is two different kinds of forest. In our case, the environment is the game and the opponents, so the state space can be designed to convey more or less information about them.

In [5] and in this work the RL algorithm used is SARSA, as in:

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

This algorithm learns the value of each action in each state, those values are displaced in a table called q-value table. This table has a row for each state in S and a column for each action in A. The table is initialized with zeros. In each time step the agent updates the value of its current action a_t for the current state s_t that is Q_{s_t,a_t}. First thing is to calculate which action (a_t) to take in the current state s_t, which is decided by the policy the agent is following. With a_t is possible to get the reward from the function R_{s_t,a_t}. Then, it is necessary to calculate the quality of the next action in the next state $(Q_{s_{t+1},a_{t+1}})$, in order to do that, we use the transition function of the TD learner and each action of each player on the last round. Then we apply the same policy again on s_{t+1} to get a_{t+1}. Finally, to get $Q_{s_{t+1},a_{t+1}}$ value is just to look in the q-value table.

Regarding the parameters: α and γ, the learning rate and the discounting factor, respectively. Both range from 0 to 1. The first configures how fast the agent learns, if the agent learns slowly it takes more iterations to converge and it accumulates the knowledge for more time, on the other hand, if the agent learns fast it converge quickly but overwrites old knowledge for newer one. The other parameter discounts the value of future benefits, the higher γ the more important is future rewards for the agent.

The starting point for defining the agents for this work is the agents defined in [5], that are RL agents for IPD. There are three learning agents with different state spaces: a learner with one state (TD1); a learner with two states (TD2), that remembers its last action, and a learner with four states (TD4), that remembers its last action (a_{t-1}) and the opponents last action $(\bar{a}_{t-1})$.

2.3 To Perceive More

The state space is not just information, it is also a factor that limits the strategies the agent can learn. Hence the state space can be associated with the capabilities of the agents, the larger the state space the more complex strategies can the agent learn, the higher is agent's cognition. In this chapter we define 4 different agents, each of them with a larger state space size than the previous: *MemoryLess*, *MajorTD4*, *SelflessLearner* and *LevelLearner*.

The two first agents are inspired, respectively, in *TD1* and *TD4*. *MemoryLess* and *TD1* have $|S| = 1$ and it is expected to defect always, since they do not know anything from previous rounds, this agent is going to be used as a baseline for the other agents. *MajorTD4* has the exact same state space as *TD4* $\{a_{t-1}\bar{a}_{t-1} : CC, CD, DC, DD\}$, the difference is that, for *TD4*, $\bar{a}_{t-1}$ is the opponent last action and, for *MajorTD4*, $\bar{a}_{t-1}$ is the most frequent action executed by the opponents in the last round, it chooses C over D if tied. *MajorTD4* receives this name thanks to its similarities with *TD4* and is dependence on the majority of opponents' last actions.

The other two are based on the idea that instead of knowing only the the majority of opponents' actions, it is better to know exactly how many players cooperated last round. The name *LevelLearner* comes from this idea of knowing every 'level' of cooperation, besides how many cooperated, this agent also remembers its last action as *MajorTD4*. As the number of states of *LevelLearner* increase 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, they are, respectively 1 and 4. However for the other two agents it 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 have two possible values (C or D), its state space size is $2 * (N + 1)$. Resuming, for $N = 5$ the state spaces sizes of each of these agents are $|\{0, 1, 2, 3, 4, 5\}| = 6$ and $|\{D0, D1, D2, D3, D4, D5, C0, C1, C2, C3, C4, C5\}| = 12$, respectively.

2.4 To Choose Smarter

SARSA is classified as a on policy algorithm, because it uses the policy to approximate the rewards of the next state. This means that the policy has great impact on the algorithm performance and on what it learns on the q-value table. One commonly used policy is the ϵ-greedy, that is the policy used in [5]. This policy is greedy because it chooses the action with greater value for the current state with 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. TD4 playing IPD against another TD4 achieves 60% of cooperation for low values of ϵ, high values of γ and low values of α.

Formally, the epsilon-greedy policy for *IPD* and *NPD* is

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

Since in these games there are only two possible actions, choosing randomly any other action is just selecting the other one, as show in equation 3. 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 initialized with zeros.

The exploration factor in epsilon greedy policies is very important to the very process of learning, without this factor, TD learners following ϵ-greedy just stick to what it learned in the first iteration. Hence, exploration allows the agent to actually use information of multiple iterations and learn solid knowledge about the game. So 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 epsilon 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 only modification needed in equation 3 is to instead of using the fixed probability ϵ, use

$$\epsilon_{lin} = \frac{\epsilon_0}{NR + 1} \qquad (4)$$

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

Similarly, to implement a logarithmic decreasing epsilon greedy policy is just use

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

instead of the static ϵ.

Another option is to use 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. These probabilities are calculated by

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

where β is a constant that changes the shape of the function. An agent following this 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 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, what is being learned by SARSA, the other is the actor that it learns how to choose actions given the critic. One simple way of doing this is to use a bernoulli distribution for each state,

$$p_{a,s} = \begin{cases} \mathrm{p}_s, if a = C \\ 1 \text{ - } \mathrm{p}_s, if a = D \end{cases} \tag{7}$$

where p_s is the probability to cooperate in state s, hence the agent will learn a vector of probabilities $P = (p_{s_1}, p_{s_2}, ..., p_{s_n})$, where $n = |S|$. To update these values we use the same value used to update the q-value table. It is possible to rewrite the equation 2 like

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

This δ is then used to update the vector of probabilities with

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

where α_p is the learning rate of the 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. This is a linear actor-critic policy, simplified for $|A| = 2$, specified in [11].

2.5 Strategy identification and dynamics

At the beginning, agents play randomly, independently of the 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, what 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 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 [1]. Examples of optimal strategies are abundant: *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 agents learn at the end of simulation it is possible to explain why some agent cooperates more than another one and what is the reasoning about the agent's decisions. On the other hand, by checking how many times an agent changes strategy during learning it is possible to measure how much is the agent 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 only the q-value gives all the information to determine how the agent is playing. For actor-critic is necessary to look at the probabilities learned by the actor. Another thing to notice is that greedy policies only play pure strategies, while Boltzmann and actor-critic may play mixed strategies. Pure strategies have an action associated with each state, while mixed strategies have probabilities of playing each action for each state. Nevertheless, it is useful to look at the q-value table and extract the strategy a greedy policy would have with those values. For simplicity only the strategies learned by *MajorTD4* are analyzed, its state space allows it to learn any pure memory-one strategy. Those strategies can be defined by four bits ($S = b_3b_2b_1b_0$), where each bit corresponds to the action the player chooses in a given state, the possible states are $\{a_{t-1}\bar{a}_{t-1} : CC, CD, DC, DD\}$ and cooperate is represented by **1** while defect is represented by **0**. Then, $b_3 = 1$ corresponds to cooperate in CC, $b_2 = 0$ to defect in CD, $b_1 = 1$ to cooperate in DC and so on. By generating every possible value, there are 16 possible strategies and many of them were already mentioned, for example: $ALLD = 0000 = S_0$, $ALT = 0011 = S_3$, $TFT = 1010 = S_{10}$, $ALLC = 1111 = S_{15}$. Hence, to extract a strategy from the q-value table, it is necessary an empty stream of bits, then for each state in $\{CC, CD, DC, DD\}$ concatenate a **0** on the right if the most valuable action for that state is *D*, concatenate **1** otherwise.

A measure of exploration is important for assure that the enhancements in cooperation do not sacrifice too much exploration. The metric for measuring exploration is the average strategy changes. In order to take this measure is just to check the strategy the agent is playing at each iteration and count how much it changes. Those changes occur whenever an agent reevaluates what is the best action for a state.

3 RESULTS

To evaluate the improvement in cooperation, it is necessary to establish a starting configuration from which variations are created varying one trait at a time. The base configuration is a *NPD* game ($f = 2$) with five *MajorTD4*, all with the learning step $\alpha = 0.05$, the weight on future knowledge $\gamma = 0.9$ and the epsilon greedy with exploration factor $\epsilon = 0.001$ as the policy for selecting actions. The values for α and γ are based in [5], 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 [5], $\epsilon = 0.01$.

[1] A strategy h_i is defined as optimal against a strategy h_j if $R(h_i, h_j) \geq R(h_k, h_j), \forall h_k \in H$. Since a strategy h defines a sequence of actions $(a_0, a_1, a_2, \ldots)$, $R(h_i, h_j)$ is defined as the expected reward of following strategy h_i against an opponent following strategy h_j, formally: $R(h_i, h_j) = \lim_{N\to\infty} \sum_{t=0}^{N} \frac{R(a_t, \bar{a}_t)}{N}$.

Besides that, there are two fixed parameters for NPD, the starting resources and the cost of cooperating, 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 measure the influence of these parameters.

The experiments are arranged in study cases, each of them can be made of any number of experiments and is driven to answer a question. All the experiments share some traits: they are made of 1000 games with the exact same parameters, each game lasts for 1000 rounds which the N players do not learn, they already learned previously, each one of the N players learned through 20000 rounds with other N-1 identical and independent players. The main measure extracted from these experiments is the average cooperation rate, that is the average cooperation in the last 100 rounds in each game.

There are two studies and an analysis: the Environment Study, the Cognition Study and the Strategy Analysis. The Environment Study, checks if the is a scenario where agents cooperate and proposes a challenging one for testing which agent cooperates more. The Cognition Study tests many agents variations to identify the role that cognition plays in cooperation among *RL* agents. Finally, the Strategy Analysis checks what the agents learned to discuss the reasoning behind the enhancements in cooperation.

3.1 Environment Study

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.

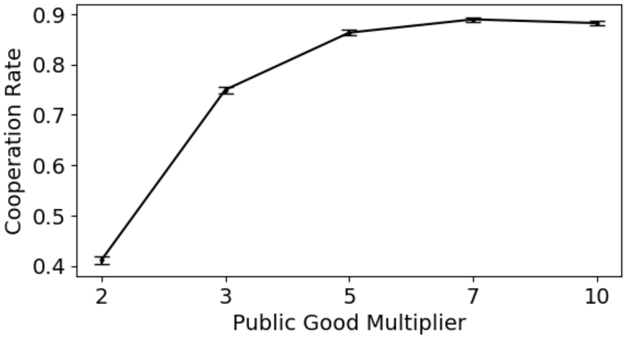

Figure 1: The percentage of cooperation in the last 100 rounds in *NPD* with five *MajorTD4* following epsilon greedy policy with $\epsilon = 0.001$ for different values of public good multiplier *f*.

The results in figures 1 and 2 are as expected. There is cooperation among RL agents in NPD and cooperation rates change with those parameters. As the number of players increase, cooperation decreases and as the public goods multiplier increases the cooperation sharply increases. In order to have cooperation, players must coordinate efforts and it is harder to coordinate a larger group. On the other hand, the public goods multiplier reflects on amount of resources in the environment and the lesser the harder is to cooperate. In a scenario with abundance of resources, high values of *f*, even little cooperation generates rewards that surpass the cost of cooperating, thus the fear of being exploited by other agents in low cooperation states disappear, what boost cooperation.

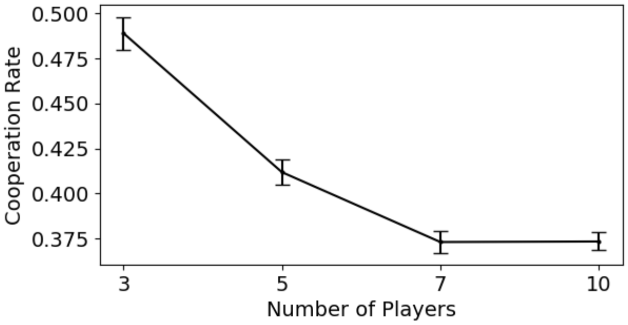

Figure 2: The percentage of cooperation in the last 100 rounds in *NPD* with five *MajorTD4* following epsilon greedy policy with $\epsilon = 0.001$ for different number of players (*N*).

The scenario to be used as baseline in other experiments was $f = 2$ and $N = 5$. Because in this configuration is difficult enough to cooperate and a relatively small number of players make simulations less demanding.

3.2 Cognition Study

Before diving into agents' cognition there are two parameters that shape agents behaviour: the learning rate (α) and the discounting factor (γ).

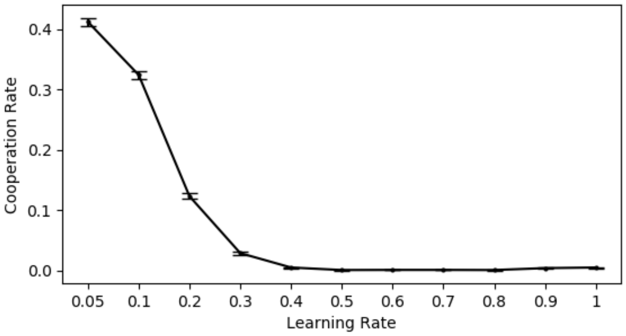

Figure 3: The cooperation rate in the last 100 rounds in NPD with five *MajorTD4* following epsilon greedy policy with $\epsilon = 0.001$ for different values of α.

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

The discounting factor penalizes rewards that are in the future. An agent with γ close to one prioritize future rewards more than

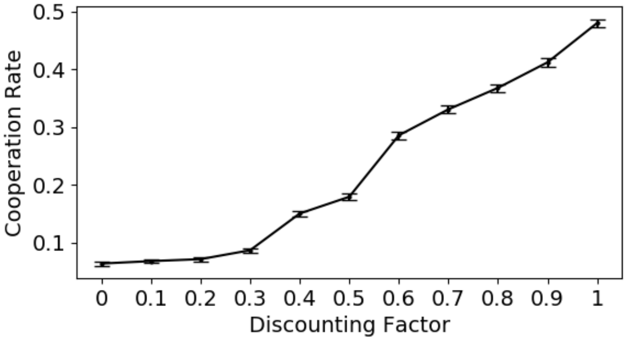

Figure 4: The cooperation rate in the last 100 rounds in *NPD* with five *MajorTD4* following epsilon greedy policy with $\epsilon = 0.001$ for different values of γ.

an agent with low values of γ. In other words, the agents cooperate more when they value long term gains over immediate gains.

Although ϵ appear only in epsilon greedy policies, it has huge impact in cooperation. As figure 5 shows, independently of the state space, ϵ impact a lot in cooperation, ranging from less than 10% to almost 80% of cooperation with *MajorTD4*.

Regarding cognition, TD learners have two traits of interest. The first is the state space is responsible for the perception of the agent, the bigger and more detailed state space, the more it perceives from the environment. While the policy is the methodology to make decisions based on the state space, what is a form of reasoning.

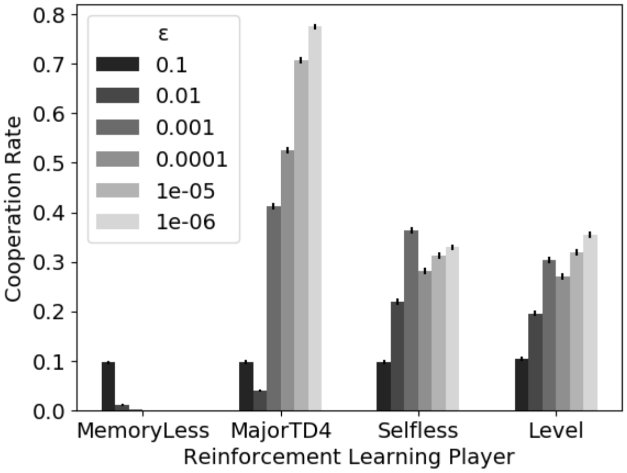

Figure 5: The cooperation rate in the last 100 rounds in *NPD* with five players following epsilon greedy policy for different values of ϵ and different agents.

In figure 5 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 factor. However the reduction in cooperation from *MajorTD4* from *SelflessLearner* 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 policies with this agent in order to enhance cooperation without decreasing exploration significantly.

Policy		Strategy Changes	
Algorithm	Parameter	Average	Standard Deviation
Epsilon Greedy	$\epsilon = 0.01$	104.91	142.96
Epsilon Greedy	$\epsilon = 0.001$	3.812	14.92
Epsilon Greedy	$\epsilon = 0.0001$	1.84	1.90
Linear Dynamic Epsilon	$\epsilon_0 = 0.1$	1.56	1.32
Log Dynamic Epsilon	$\epsilon_0 = 0.01$	4.92	22.01
Boltzmann	$\beta = 1$	7.80	3.50
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.

Decreasing exploration harms the learning process: it makes the agents more susceptible to stick in sub-optimal states and less adaptive. One way to measure that is to check in average how many times the agent changes the strategy it is playing during learning, the more changes the more it explored different ways of playing. The average strategy changes for *MajorTD4* and different policies are in table 1. Notice how the strategy changes decrease when ϵ is decreased in epsilon greedy policy with static ϵ.

Then, the goal is to find a policy that is better than epsilon greedy policy with $\epsilon = 0.0001$, in other words, that have higher cooperation rates without losing exploration. 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.

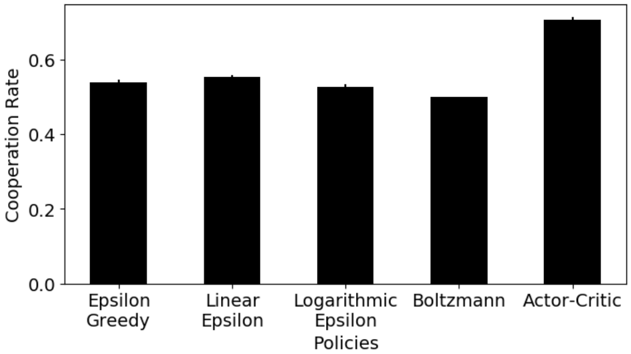

Figure 6: The cooperation rate in the last 100 rounds in *NPD* with five *MajorTD4* for different policies.

The only policy that strictly increases cooperation and strategy exploration was the Actor-Critic policy, that stands out as the best result. Epsilon greedy with linear decreasing epsilon also has cooperation improvement but at slightly less exploration. The other two policies, the epsilon greedy with logarithmic decreasing epsilon and Boltzmann, increase exploration at the expense of small decrease in cooperation rates.

Actor-Critic is the best result with *MajorTD4* and this was expected, because it decreases variation during learning. Since actor-critic has such good results, we experimented it with the other agents, the results are in figure 7. The best result is with *LevelLearner*, with $\alpha_P = 0.05$, achieved cooperation of 80% and changes strategy during learning 25.34 ± 8.28 times on average.

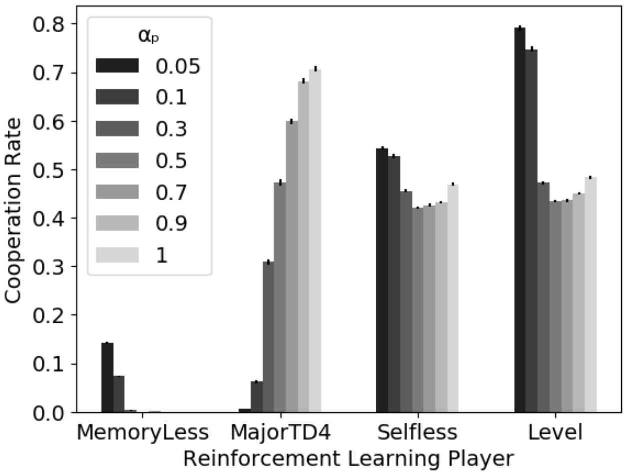

Figure 7: The cooperation rate in the last 100 rounds in *NPD* with five players following actor-critic policy for different values of ϵ and different agents.

Those results show how central cognition is to cooperation. It allowed the improvement from *MemoryLess* to *MajorTD4*, regarding state space, and again from epsilon greedy to actor-critic, regarding policy. Although increasing cognition in these two cases improved cooperation, fixing one state space and varying policies or fixing a 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 state space and policy.

3.3 Strategy Study

The state space of *MajorTD4* has the advantage of being easily translated into one of the 16 memory-one strategies of IPD. So this part focus on the strategies learned by *MajorTD4* and explores also the probabilities of cooperating learned by agents playing actor critic policies.

There are significant difference between the strategies learned by *MajorTD4* when following epsilon-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, it cooperates, when many are cooperating, it defects. The appearance 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 actor-critic the frequency of *ALLC* is higher.

S0 = 0000 = ALLD	S8 = 1000
S1 = 0001	S9 = 1001 = WSLS
S2 = 0010	S10 = 1010 = TFT
S3 = 0011 = ALT	S11 = 1011
S4 = 0100	S12 = 1100
S5 = 0101	S13 = 1101
S6 = 0110	S14 = 1110
S7 = 0111	S15 = 1111 = ALLC

Table 2: *MajorTD4* strategy mapping to binary with names of important strategies.

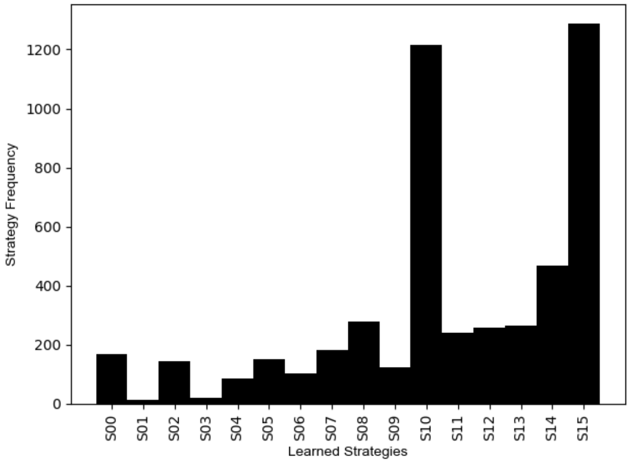

Figure 8: Strategies learned by five *MajorTD4* playing *NPD* ($f = 2$) following epsilon greedy policy ($\epsilon = 0.0001$) through 1000 games.

Nevertheless, actor-critic policy also learns the probabilities of cooperating in each state. The average results over 1000 games are shown on table 3 for *MajorTD4*, on table 4 for *SelflessLearner* and on table 5 for *LevelLearner*. The standard deviations in table **??** and the strategy distribution in figure 9 indicate a possible specialization in the group dynamics, that means the reason for those high standard deviations are that values agents converge for different strategies that stabilize together. Things become clearer with the data of *SelflessLearner* in table **??**. The more specific state space let the agent learn to cooperate with almost 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, *LevelLearner* also leanrs this mechanism, however with smaller standard deviations. This mechanism reassembles *WSLS*, however during learning agents following actor-critic do not learn this strategy, it seems they end up differentiating in the case of *MajorTD4* or seeking mixed strategies in the case of *SelflessLearner* to create these mechanism.

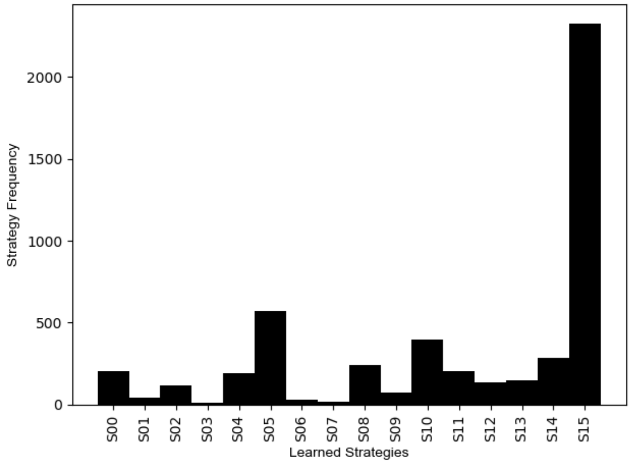

Figure 9: Strategies learned by five *MajorTD4* playing *NPD* ($f = 2$) following Actor-Critic policy ($\alpha_P = 1$) through 1000 games.

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

Table 3: Average probability to cooperate and average deviation of *MajorTD4* following Linear Actor-Critic policy for each state of *S*.

State	0	1	2	3	4	5
Average	0.5949	0.0101	0.0539	0.2350	0.7878	0.6364
St. Dev.	0.0410	0.0224	0.0332	0.2167	0.4016	0.1354

Table 4: Average probability to cooperate and average deviation of *SelflessLearner* following Linear Actor-Critic ($\alpha_P = 0.1$) policy for each state of *S*.

State	DEF0	DEF1	DEF2	DEF3	DEF4
Average	0.5808	0.0588	0.0159	0.0158	0.0165
St. Dev.	0.0244	0.0275	0.0118	0.0209	0.0568

State	COOP1	COOP2	COOP3	COOP4	COOP5
Average	0.1833	0.2439	0.6138	0.9995	0.6033
St. Dev.	0.0625	0.0365	0.0320	0.0041	0.0700

Table 5: Average probability to cooperate and average deviation of *LevelLearner* following Linear Actor-Critic policy for each state of *S*.

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.

4 CONCLUSIONS

Widespread cooperation is possible with *RL* agents playing *NPD*, for high values of *f*, more than 80% of cooperation is achieved. However, resource abundant environments are not the rule, usually individuals have to compete for resources, so we fixed $f = 2$ and $N = 5$ as the harsh scenario. In this environment the problem of managing exploration appears and sticks throughout this work.

Then we found out that the parameters α, γ and ϵ impact a lot in cooperation as well. Cooperation increases for high values of γ and low values of α and ϵ. Those relations can be understood as principles that favours cooperation: the low values of learning rate and exploration means that changes must be taken slowly and not very frequently, to accumulate the knowledge through time and give time for the environment to adjust; the high value of the discount factor means that individuals must value long term gains over short term ones.

Further on, cognition plays a key piece in the search for high cooperation rates. However the increase in cooperation is not explained solely by the increase in state space size neither by substituting the policy for another more complex. The improvement is due to a combination of the both. Analysing the results, it seems that for a given *S* it is possible to vary policies in order to increase cooperation, however some of them may perform worse than they are supposed to because of constraints of the state space, if we upgrade the state space the same policy may perform much better, this happens with actor critic. We can see that the recover mechanism learned in tables 4 and 5 is only possible because the agent is capable of differentiate when none player is cooperating from when only one or two are cooperating, *MajorTD4* does not make this clear distinction. The process is not linear but iterative, fix the best *S* and test different policies, then fix the best policy and improve *S* and so on.

The last consideration we can take from the Strategy Analysis is that, beside the recover mechanism, the higher cooperation is not with all the players. In *PD*, as in its variations, there are two reasons to defect: fear of being exploited in case of low cooperation and exploit the other players to maximize its own gains (free ride). *LevelLearner* following actor critic solved the first reason to defect with the recover mechanism, but did not solve the second, there is a decrease in cooperation from the state of COOP4 to COOP5, showing some lenience with a limited number of free riders.

For future work there is lot to be done, ranging from testing other public goods game, not only *NPD*, to testing more complex *RL* algorithms and policies and implementing this framework in real life scenarios where *NPD* naturally appear.

REFERENCES

[1] Han The Anh, Luís Moniz Pereira, and Francisco C Santos. 2011. Intention recognition promotes the emergence of cooperation. *Adaptive Behavior* 19, 4 (2011), 264–279. https://doi.org/10.1177/1059712311410896 arXiv:https://doi.org/10.1177/1059712311410896

[2] Linghui Guo, Zhongxin Liu, and Zengqiang Chen. 2017. A leader-based cooperation-prompt protocol for the prisoner's dilemma game in multi-agent systems. 11233–11237. https://doi.org/10.23919/ChiCC.2017.8029149

[3] 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

[4] 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

[5] 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

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

[7] 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

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

[9] 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].

[10] Jana Vyrastekova and Yukihiko Funaki. 2010. Cooperation in a sequential N-person prisoner's dilemma : the role of information and reciprocity. *Human Movement Science - HUM MOVEMENT SCI* (01 2010).

[11] 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