TÉCNICO LISBOA

Learning Dynamics in Populations of Actor-Critic Agents

João Vitor de Oliveira Barbosa

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Francisco António Chaves Saraiva de Melo
Prof. Francisco João Duarte Cordeiro Correia dos Santos

Examination Committee

Chairperson: Prof. Luís Manuel Antunes Veiga
Supervisor: Prof. Francisco António Chaves Saraiva de Melo
Member of the Committee: Prof. Anna Helena Reali Costa

December 2019

Acknowledgments

First I want to thank my parents. My father with his solid beliefs in the transforming power of education did not spare efforts when investing on mine, for what I will be eternally grateful. My mother for constantly teaching me that solely work does not fulfill a human life, what helped me to balance things and enjoy the path rather than the end, my deepest gratitude.

Then I want to thank all my professors that helped me to get here. Without them I would not have gone this far this fast. For helping me and countless other students my highest respect.

Finally, I want to thank all my friends. I feel in eternal debt with you for all the time we passed studying in libraries, all the conversations about my thesis that you showed interest and encouraged me to do more, the trips that you planed for us while I was studying and so on. I get constantly amazed by you.

I am overwhelmed by gratitude, this work is my first attempt to return back to society part of what I received.

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 and its previous action in the previous round can offer a significant increase in the overall level of 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; Multi-Agent Systems; Game Theory; Public Goods Games.

Resumo

O estudo sobre o aparecimento da cooperação ainda é um problema em aberto para muitas áreas do conhecimento. Esse problema pode ser formalizado através de Teoria dos Jogos e dilemas iterativos para N-jogadores. Aqui investigamos as dinâmicas de aprendizagem que aparecem nesse tipo de problema. Nós simulamos a tomada de decisão em dilemas de N-jogadores com agentes de diferentes níveis de sofisticação quanto ao método de aprendizagem, adotando um algoritmo de aprendizagem de diferença temporal como ponto de partida. Os resultados mostram que a combinação de uma simples política Actor-Critic com um estado de espaços que permite ao jogador distinguir quantos agentes cooperaram e qual foi sua última ação pode proporcionar um aumento significativo nos níveis de cooperação. Os resultados são dependentes das características do dilema, mais precisamente do tamanho do grupo e da contribuição mínima para se produzir um retorno coletivo. Cooperação também aumenta com baixo fator de exploração e taxa de aprendizagem, e diminuir com os descontos nas recompensas futuras. Em fim, estes resultados sugerem que, para cada dilema, a combinação adequada de estado de espaços e método de seleção de políticas garante coordenação de agentes adaptativos e individualistas em sistemas de multi-agentes.

Palavras Chave

Aprendizado por Reforço; Sistemas Multi-Agents; Teoria dos Jogos; Jogos de Bem Público.

Contents

List of Figures

List of Tables

Acronyms

NPD	N-person Prisoner's Dilemma
PD	Prisoner's Dilemma
IPD	Iterated Prisoner's Dilemma
RL	Reinforcement Learning
TD	Temporal Difference
ALLD	Always Defect
ALLC	Always Cooperate
TFT	Tit for Tat
TF2T	Tit for 2 Tat
ALT	Alternate Cooperation and Defection
SARSA	State-Action-Reward-State-Action
WSLS	Win Stay Lose Shift
NSH	N-Person Stag Hunt Game
N	Number of players
f	Public Goods Multiplier
SFBP	Santa Fe Bar Problem
amTFT	Approximate Markov Tit for Tat

1

Introduction

Contents

Benefits of cooperation are not scarce in nature: the early *Homo Sapiens* have replaced the physically stronger *Neanderthals* is the superior social capacities of the first over the second [1], Argentinian Ants can work together even from different colonies, their high level of cooperation [2] allows them to beat many other species in competition for resources [3]. But, how do these species achieve such widespread cooperation? [4] models the decision dilemma of cooperation with Prisoner's Dilemma (PD) and shows the importance of reciprocity to emergence of cooperation in interactions between two individuals. In PD, two individuals decide to cooperate (C) of defect (D) without communicating: 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 best for the group and what is best for the individual. Even though, ants and humans have different cognition levels, both achieve widespread cooperation states. After all, does cognition help groups of individuals to overcome selfish behaviour and start cooperating more?

A common approach to this problem is to run simulation of intelligent machines in evolutionary environments in order to measure the impact of them in the cooperation of a population [5,6]. In evolutionary environments, individuals with different strategies play PD games pairwise and accumulate rewards, after a number of iterations the least successful individuals copy the strategy of the most successful ones, this process is then repeated. In [5] the machines tries to discover the opponents' intention with Bayesian Networks, while in [6] the machines learn by trial and error. In both cases, there are improvements in cooperation, when comparing to populations of non learning individuals.

Instead of studying the impact of cognition in large populations with intelligent individuals, we focus on testing how different cognitive capabilities boosts cooperation in smaller groups that interact collectively, instead of pairwise. N-person Prisoner's Dilemma (NPD), the generalization of PD for more than two individuals, is a good model for this environment. In NPD every player has an starting amount of resources, if it chooses to cooperate, it gives a part of its resources to a public good, then this public good is enlarged and divided among all the players, independently of cooperating or not. Because this model involves giving money to a public good that is accessible to everyone, this is classified as a public good game. On the other hand, NPD is the generalization of PD because it have the three same possible states: mutual cooperation is the best solution for the group, if some are cooperating and the others are defecting, the defective players exploit the cooperative ones and if nobody cooperates there is no gains at all.

With the environment defined, it is necessary to define how the machines simulate animals' cognition. Since animals learn through trial and error [7, 8], we approximate its cognition by how they learn. 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. We examine RL agents with different cognitive levels playing NPD in order to measure the impact of cognition in cooperation.

Hence, 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 makes cooperation 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 [9] achieved cooperation with RL agents in PD and [6] improved cooperation in evolutionary environments with RL, it is expected there is 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 agents playing NPD?**

   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. Besides that, there are three learning parameters that tune how the agents learn: the learning step, the discounting factor and the exploration factor. The first adjust how fast the agents learn, the second how important is long term rewards over immediate rewards and the third adjust how much exploration some agents do during learning. The values of these parameters that boosts cooperation can be seen as standard guidelines for enhancing 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 this 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 [11]. 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 [5]. 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.

1.1 Document Structure

This work is structured in the following way: section 2 there are optional discussions related to this work, then chapter 3 explains the theory that support the experiments, chapter 4 defines the experiments and show the results that answers the three question proposed in this section and then there is the final conclusions in 5.

2

Related Work

Contents

2.1 More over Game Theory

Game theory is a discipline of Mathematics that study how to make decisions on conflicting or complex situations. PD is one of the the most studied games in the area. The story that illustrates it is with two prisoners that are going to be convicted, however the police does not have enough evidence to increase their prison time, so it offered a deal for confession. If both stay silent (mutual cooperation) they keep their sentences, if both confess (mutual defection) they have their sentences increased. However if only one confess, he gets free and the one who kept silence gets maximum sentence. Since the prisoners can not talk to each other they have no previous information, neither way to coordinate efforts, the best play is to confess, since not to do it (cooperate) opens the possibility for getting maximum sentence.

However, if they play repeatedly the risk is not as high, because the players can recover what it lost in future iterations. Besides that, past iterations give information that can be used to enhance coordination. The game where two players play PD repeatedly is called Iterated Prisoner's Dilemma (IPD). Since there are many iterations, each player has a history of actions, that is a list of the actions chosen by a player. This historic defines a strategy, a method to choose actions. Those strategies can be as simple as Always Cooperate (ALLC) and Always Defect (ALLD), dynamic changing as TFT or resilient as WSLS. The first to strategies only plays C or D and stick to it. TFT starts cooperating, then repeats the opponent's last action, hence playing against this strategy the best option is to cooperate, since defection will make TFT defect as well, changing the best play in the game from defect to cooperate. However, TFT does not recover from a state where both players are defecting, that means if two TFTs are playing against each other, they cooperate, until one of them commits an error and defect once, if this happens both will stick to defection indefinitely. WSLS on the other hand has the property of trying to restore cooperation if by any reason the opponent defects, it plays by repeating its last action if it's winning (mutual cooperation or defecting against cooperation) and flips its last action in case of losing (mutual defection or cooperating against defection).

These strategies may occur in NPD but NPD is different from IPD. TFT for example is very good strategy in IPD and not so good in NPD, because in NPD there is no way of punishing only the players who defected, either you punish everyone by playing D or do not punish anyone by playing C [12]. After all NPD is a multi-player game. The multi-player games are divided into two broad categories: Public Goods Games and Commons Dilemma [13]. In Public Goods Games, players choose to give resources to a public goods whose benefits everyone, independently of contributions. Within this class of games, the game defining trait is how the contributions are turned into the public goods, that is the production function. While the Commons Dilemma is about sharing a resource with other players, the more each player extract this resource the more benefits it gets, however, if they extract much, the resource is depleted. The key aspect for these class of games is the replenishment function, the rate in which the shared resource grows. NPD is a public goods game because in it each player starts with an amount of

resources and if it cooperates, it gives a part of this resources to the public goods. The sum of resources payed by the players who cooperated is then multiplied by f, and divided equally among all the players. This game is the generalization of PD because it maintains the same characteristics: the better result for everyone (the sum of rewards of every player) is when all cooperate, when there is much cooperation it is tempting to not cooperate and to receive the benefits without paying for them, while if there is not much cooperation it is safer to defect since the public goods divided over all players may be smaller than the cost of cooperating.

2.2 Social Dilemmas Dynamics

Understand the dynamics of social dilemmas help to understand the results of this work. In [12], there is many results regarding social dilemmas compiled.

Analytical and experimental data supports that cooperation is easier in small groups than in large groups, overall cooperation in large groups is almost impossible, repetition of the same problem and communication over the players enhances cooperation. However the, possibly, most interesting result is that states of general cooperation or defection can appear suddenly.

What usually happen in social dilemmas is 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 this states, 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 is common for the behavior of a group in a social dilemma stay the same for long periods, but when it changes, it changes fast.

A factor that may help in those transitions is when the population is heterogeneous. Heterogeneous populations have individuals who weights things differently, for example, if a small group of the population values long term gains more than the average, they may be more willing to cooperate at the start and their cooperation may incentive other groups of the population to cooperate as well, until the most conservative groups are convinced to cooperate.

2.3 RL in other Game Theory Dilemmas

One of the few works with RL and a social dilemma is [14]. The game studied is Santa Fe Bar Problem (SFBP) and the RL algorithm used was Q-learning.

SFBP is a congested resource problem. Suppose there is a bar in which it is pleasant to go only when the capacity is on 60% or less. Each person decides if go to the based based on the number of

people that went to the bar in previous weeks and decides to go only if he or she thinks that is going to be pleasant.

Q-learning is an off policy RL algorithm. This means that differently from SARSA that uses the policy to estimate future rewards, Q-learning estimates future rewards independently of the policy, by a function approximation for instance.

The goal of [14] is to find not only a efficient solution to the dilemma but also a fair solution. An efficient solution means that the total reward of the agents must be as high as possible , while fair solution means that agents with the same utility, in other words, reward function, must have the same probability of attending the bar. The way this is achieved is with Q-learning agents and a modification on the SFBP, this modification adds a taxation mechanism based on incremental changes.

2.4 Language expressiveness in NPD in evolutionary environments

Expressiveness is deeply related to information and how to use this information, although the RL learners in this work do not have a language they have a State Space and a Policy that together allow them to recognize more characteristics of the environment and build strategies based on what they recognize. In [15], the agents know the number of opponents who cooperated last turn and learn strategies based in on of two possible languages: Finite Automata and Adaptive Automata. The strategies build with the first have an initial state in which the agent cooperates of defects, then the agent transitions from one state to another depending on the number of cooperators in the last round. Adaptive Automata extends the Finite Automata framework adding rules that may change its structure by adding or removing states and transitions. Thus the Adaptive Automata may express strategies that Finite Automata can not.

Besides that, environment in [15] is quite different from this work. In [15], the agents are placed in a grid with Von Neumann's neighborhood of degree 1, that means that each agents has as neighbors only the agents strictly up, down, left or right from itself. The upper cell are connected to the bottom ones, and those of the first column with the rightmost column, thus forming a ring torus. Since each NPD is among an agent and its neighbors, those games have only 5 players. Them in order to simulate this environment, first initialize the grid with many different strategies, then simulate many NPD games across the grid, then the least successful strategies copy the most successful strategy around them. During this copy, mutations can occur and different strategies may emerge. Then repeat this procedure. This is a evolutionary environment that resembles natural selection, because the best strategies are passed forward and the worst are discarded.

The results of this evolutionary environment is that in both configurations, with Finite and Adaptive Automata, there are broad cooperation. Although in general there is no difference in global rewards between the models, if the models start in a state of widespread defection, Adaptive Automata has

higher global rewards than Finite Automata. Hence, a more complex language, that allows a more complex strategy selection, can recover better from widespread defection states.

2.5 Applications of NPD

Probably one of the most immediate applications of NPD is with Welfare states. The players would be the whole society, to cooperate is to pay correctly all the taxes and to defect is to evade taxes. No matter if a particular citizen evaded taxes, he or she can still benefit from the welfare state, like education, security and health care. Modifications over the models of this work can help to simulate which factors favors tax evasion, for example.

However simulate people's behaviour is not an easy task usually, what makes those results difficult to obtain. Luckily those social dilemmas are not limited to human and animal societies. Computational societies also suffer from these problems [16]. The internet is a public good. Imagine that in a segment of internet a group of individuals want to send files to other people. It is desirable to send those files as fast as possible, those files segmented in packets, but if everyone send its packets at a high frequency the packet loss sharply increases. In these sense, cooperation is to send packets at a lower frequency to allow everyone to benefit from the network and defection is to send packets at a higher frequency. Solving NPD with RL agents is a distributed away of solving this problem.

Another approach to apply social dilemmas solutions to concrete problems was done by [17]. This work modifies the algorithm of Deep Reinforcement Learning to learn two strategies at the same time: one cooperative and the other defective. Then the agent play a modified version of TFT, called Approximate Markov Tit for Tat (amTFT), that start playing the cooperative strategy, then when the agent starts getting exploited it starts playing the defective strategy and turn back to the cooperative one if the opponent decide to cooperate again.

3

Theoretical Framework

Contents

3.1 Defining the Game

More formally, PD is a two-player bimatrix game, what means that the game is defined by a 2x2 matrix that in each cell there is a pair of numbers, the first the reward of the first player and the second the reward of the opponent. The rows are the possible actions of the first player (a) and in the columns the actions of the opponent ($\bar{a}$). In PD each player has two possible actions Cooperate (*C*) or Defect (*D*), what gives the Action Space $A = \{C, D\}$, all the possibilities of play from both players give all the possible states, that result in the State Space $S = \{a\bar{a} : a, \bar{a} \in \{C, D\}\} = \{CC, CD, DC, DD\}$, the bimatrix can be defined as a function $R : S \rightarrow \Re^2$ as

$$R(a_t, \bar{a}_t) = \begin{cases} (R, R), \text{ if } a_t\bar{a}_t = \text{CC}, \\ (S, T), \text{ if } a_t\bar{a}_t = \text{CD}, \\ (T, S), \text{ if } a_t\bar{a}_t = \text{DC}, \\ (P, P), \text{ if } a_t\bar{a}_t = \text{DD}, \end{cases} \tag{3.1}$$

with $\{R, S, T, P\} \in \Re$ and $T > R > P > S$. In other words: the maximum gain occurs when the player defects and the opponent cooperates (*T*); the second best outcome occurs when both cooperate (*R*), the third when both defect (*P*) and the worst when the player cooperates and the opponent defects (*S*). The last requirement for this bimatrix define a PD is that $2R > T + S$ holds, this condition guarantees that the global optimum is mutual cooperation.

NPD is the generalization of IPD but there is still some differences, since it is a public good game each player now has a starting amount of resources, for instance money, and when each player cooperates it gives an amount of money (*p*) to the public good, if it does not have enough money it is obliged to defect, the current amount in the public good is enlarged by a multiplier f and then divided by all the players N. Differently from IPD the rewards depend on every other player, not just on a single opponent:

$$\begin{aligned} R(D) &= \frac{fkp}{N}, \\ R(C) &= R(D) - p, \end{aligned} \tag{3.2}$$

where *k* is the number of cooperators. The reward is then added to the money the player currently have.

3.2 Behaviour analysis in PD variations

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. This strategies represent the learned knowledge condensed in a method to make decisions. Since the other players are part of the environment, when a player starts playing a different strategy, it changes the environment for the others, what may cause the others to change strategy in response. This happens

because RL agents always try to learn the optimal strategy against the other players, what creates interactions among strategies. A strategy is only a sequence of actions, the most famous ones can usually 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. Those changes occur whenever an agent reevaluates what is the best action for a state. This measure is calculated by checking the strategy the agent is playing at each iteration and count how many times it

[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 $(a_0, a_1, a_2, ...)$, $R(h_i, h_j)$ is defined as the expected reward of following strategy h_i against an opponent following strategy h_j, formally: $R(h_i, h_j) = \lim_{N\to\infty} \sum_{t=0}^{N} \frac{R(a_t, \bar{a}_t)}{N}$.

changes, then take the average value from different agents.

3.3 Learning in IPD 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 [9] and in this work the RL algorithm used is State-Action-Reward-State-Action (SARSA), as in:

$$Q_{s_t,a_t} \leftarrow Q_{s_t,a_t} + \alpha(R_{s_t,a_t} + \gamma Q_{s_{t+1},a_{t+1}} - Q_{s_t,a_t}). \tag{3.3}$$

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 the *S* and a column for each action in the *A*. The table is initialized with zeros. In each time step the algorithm updates the value of the current action a_t for the current state s_t that is Q_{s_t,a_t}. First thing is to calculate which action 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 [9], 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 ($\bar{a}_{t-1}$).

TD1 does not have the capabilities to differentiate situations, because independently of the game it is on its single state, it learns in the general case what is the best action. As in the general case, the best to do is defect, this learner always learns to defect. The case that makes it clear the gain that TD2 has over TD1 is when we put both to play against TFT. While TD1 learns to defect against TFT (leading to sub-optimal rewards), the TD2 learns to cooperate with it (that is the optimal strategy). Moreover, TD2 learns to alternate against TF2T (optimal strategy) even though being capable of remembering only the last state, not the two previous rounds as TF2T. So playing against TF2T does not differentiate TD2 and TD4, as TD4 also learns to alternate.

There are two important situations where TD4 is better than TD2. The first is with WSLS. Depending on the reward function of the IPD the best strategy against WSLS changes, it can be ALLD or ALLC. While TD2 always learns ALLD against WSLS no matter the values of the rewards, TD4 can learn ALLD or ALLC properly depending on the values of the reward. The second situation is when TD2 plays against another TD2, and when TD4 plays against TD4. Where TD2 always learn to defect against itself and TD4 can learn to cooperate.

At last, TD4 learns optimal strategies against any memory-one strategy. A memory-one strategy can be defined by half a Byte ($S = b_3b_2b_1b_0$), where each bit corresponds to the action to be done given the state of the previous round, as there are only two players and they can only cooperate or defect, the possible states are $\{CC, CD, DC, DD\}$. Let the state be given by $a_{t-1}\bar{a}_{t-1}$ (the player's action followed by the opponent's action) and that Cooperating and Defecting value, respectively, to 1 and 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: $ALLD = 0000 = S_0$, $ALT = 0011 = S_3$, $TFT = 1010 = S_{10}$, $ALLC = 1111 = S_{15}$. TD4 can beat any memory-one strategy given that its state space is exactly the space of all possible memory-one strategies.

3.4 To Perceive More

The state space is not just information, it is also a factor that limits the strategies the agent can learn. In the previous section for example was introduced TD1 and TD4, while the state space of TD4 allows it to learn one of 16 different strategies, the state space of TD1 only allows it to learn 2 possible strategies, ALLD and ALLC. 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. In this chapter we define 4 different

agents, each of them with state spaces of different sizes: *MemoryLess*, *MajorTD4*, *SelflessLearner*, *LevelLearner*.

The two first agents are inspired, respectively, in TD1 and TD4. *MemoryLess* also has only one state and it is expected to defect always, 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 $\bar{a}_{t-1}$ is not the opponent last action since there is more than one opponent, for *MajorTD4*, $\bar{a}_{t-1}$ is the most frequent action executed by the opponents in the last round, choose *C* over *D* if tied. Hence the agent still has $|S| = 4$ and it is dependent on player's last action and the majority of opponent's last actions, this agent is called *MajorTD4*.

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 in last round. The name *LevelLearner* comes from this idea of knowing every 'level' of cooperation, besides how many cooperated, this agent also knows 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.

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 size 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.

3.5 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 [9]. 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 achieves 60% of cooperation with low values of ϵ, besides high values of γ and low values of α.

For IPD and NPD the ϵ-greedy policy is in the form of

$$\pi_{\epsilon-greedy}(s) = \begin{cases} \pi_\epsilon(s), \text{ with probability } (1-\epsilon) \\ \bar{\pi}_\epsilon(s), \text{ with probability } \epsilon \end{cases}, \\ \pi_\epsilon(s) = argmax_a(Q(s,a)), \\ \bar{\pi}_\epsilon(s) = \begin{cases} C, \text{ if } \pi_\epsilon(s) = D \\ D, \text{ if } \pi_\epsilon(s) = C \end{cases}. \tag{3.4}$$

In these formulas, $\pi_\epsilon(s)$ is the greedy part, that selects the action that has greater value in the q-value table. While the $\bar{\pi}_\epsilon(s)$ is the policy that play another action randomly, since there is only two possible actions, this function just flips the action $\pi_\epsilon(s)$ chooses. Another important aspect to stress out is that when two actions have the same value in the table for the current state, the agent chooses one randomly, including at the start when the whole table is initialized with zero.

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 ϵ based on by which function the exploration factor decays, a linear function or a logarithmic function.

In order to define a linear dynamic ϵ-greedy policy it if just necessary to modify equation 3.4 to

$$\pi_{lin-\epsilon-greedy}(s) = \begin{cases} \pi_\epsilon(s), \text{ with probability } (1-\epsilon_{lin}) \\ \bar{\pi}_\epsilon(s), \text{ with probability } \epsilon_{lin} \end{cases}, \\ \epsilon_{lin} = \frac{\epsilon_0}{R+1}, \tag{3.5}$$

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

Similar modification is necessary for defining the dynamic decreasing logarithmic ϵ-greedy policy:

$$\pi_{log-\epsilon-greedy}(s) = \begin{cases} \pi_\epsilon(s), \text{ with probability } (1-\epsilon_{log}) \\ \bar{\pi}_\epsilon(s), \text{ with probability } \epsilon_{log} \end{cases}, \\ \epsilon_{log} = \frac{\epsilon_0}{ln(R+2)}. \tag{3.6}$$

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

$$p_a(s) = \frac{e^{\beta Q(s,a)}}{\sum_{a' \in A} e^{\beta Q(s,a')}}, \tag{3.7}$$

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,

$$p_{a,s} = \begin{cases} \text{p}_s, if a = C \\ \text{1 - p}_s, if a = D \end{cases} \tag{3.8}$$

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 3.3 like

$$\begin{gathered} Q_{s_t,a_t} \leftarrow Q_{s_t,a_t} + \alpha\delta, \\ \delta = r_{s_t,a_t} + \gamma Q s_{t+1}, a_{t+1} - Q_{s_t,a_t}. \end{gathered} \tag{3.9}$$

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

$$\Delta p_s = \alpha_p \delta (y^t - p_s^t), \tag{3.10}$$

where α_p is the learning rate of the policy, y^t is the value of action chose 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 [18].

4

Results

Contents

To evaluate the improvement in cooperation rates varying different traits of reinforcement learning agents is necessary to establish a starting configuration from which there is going to be drawn variations. The base configuration is a NPD game ($f = 2$) with five *MajorTD4*, all with the learning step $\alpha = 0.05$, the discounting factor $\gamma = 0.9$ and the ϵ-greedy with exploration factor $\epsilon = 0.001$ as the policy for selecting actions. The values for α and γ are based in [9], as in *NPD* is expected even higher sensibility to the exploration factor this baseline uses a smaller value than the one used in [9], $\epsilon = 0.01$.

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 their influence.

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, create N identical and independent players that learn by playing with each other through 20000 rounds, then we sample one of them to be at 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 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.

There are two studies and an analysis: the Environment Study, the Cognition Study and the Strategy Analysis. The Environment Study, checks if there 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 improvement in cooperation rates.

4.1 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.

The first result is how the cooperation rates react when increasing the public good multiplier. The expected result is an increase in cooperation rates until achieving almost total cooperation when $f =$

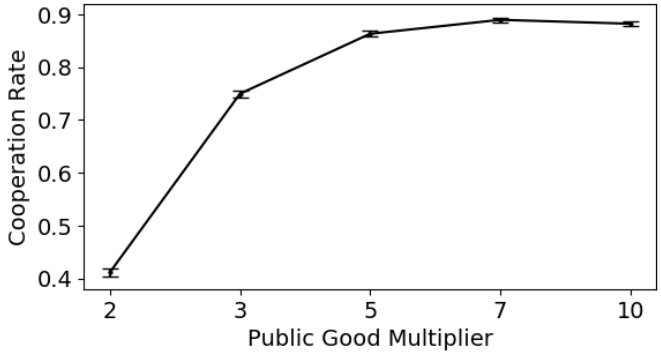

Figure 4.1: The percentage of cooperation in the last 100 rounds in NPD with five *MajorTD4* following ϵ-greedy policy with $\epsilon = 0.001$ for different values of public good multiplier *f*.

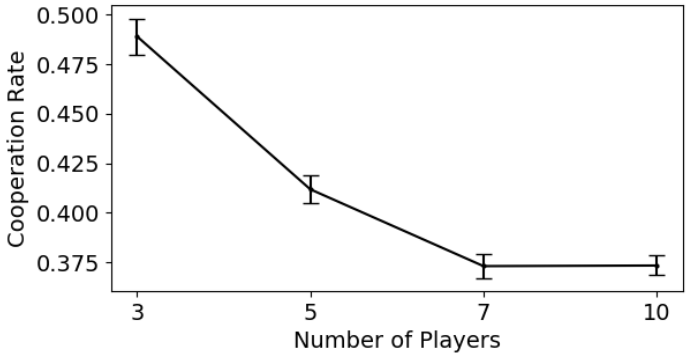

Figure 4.2: The percentage of cooperation in the last 100 rounds in NPD ($f = 2$) with five *MajorTD4* following ϵ-greedy policy with $\epsilon = 0.001$ for different number of players (*N*).

$2 * N$, where N is the number of players, the results in figure 4.1 show cooperation rates for different values of f reaching from 2 to 10. The cooperation rate sharply increases as expected. This happens because when f is almost two times N there is no reason for having fear of being exploited, the only reason to defect is to free ride in case many are cooperating. Take the example of five players with $f = 10$, if only one cooperate, the public goods will receive 1 resource and enlarge it to 10, that divided among all players results in 2 to everyone, evidently those who defected receive more than the single cooperator, but the cooperative player also increased its resources, by 1. In the case $f = 2$, cooperate alone would result in $\frac{1*2}{5} - 1 = -0.6$.

The other expected result is that if we increase the number of players cooperation decreases, since it becomes harder to coordinate more individuals towards a cooperation behavior than fewer. The values for the games with 3, 5, 7 and 10 players are disposed in figure 4.2. The cooperation rates decrease as expected.

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

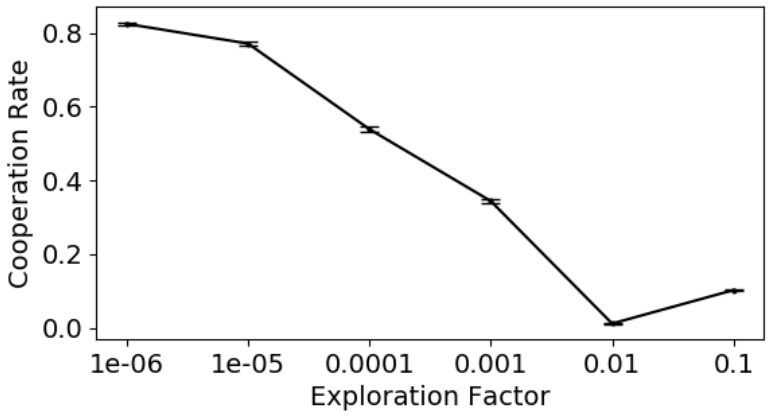

Figure 4.3: The cooperation rate in the last 100 rounds in NPD ($f = 2$) with five *MajorTD4* following ϵ-greedy policy for different ϵ.

less demanding.

4.2 Cognition Study

With the harsh environment set in previous study, this section tests the impact of the agents' capabilities in cooperation. The results of this section is divided into two parts: the first test the cooperation rate for different values of learning step (α), discounting factor (γ) and exploration factor (ϵ), with ϵ-greedy policy, the other part compare different state spaces and policies regarding the cooperation that emerge during simulations. The first part produces a set of principles that boosts cooperation for RL agents in NPD. The second part answers if more cognition leads to more cooperation, if there is a maximum limit in which cognition can improve cooperation, if it is possible to improve cooperation without reducing exploration, if there is a single policy or state space that maximizes cooperation. Moreover, this last part selects the configuration with higher cooperation achieved in this work.

4.2.1 Learning Parameters

Cooperation in IPD demands low exploration factor, because during learning if a TD learner perceives its opponents as random players (players that choose actions randomly), it always defects, because this is the rational play. Since each opponent has a probability of choosing randomly and in NPD there are more opponents than IPD, NPD has more randomness than *IPD*. Hence it is expected a reduction in ϵ in order to maintain the same level of cooperation. With that in mind, there are six experiments with $\epsilon = \{0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001\}$.

First result is that cooperation with $\epsilon = 0.01$ is, as expected, smaller in NPD then with IPD, actually cooperation does not occur in 10% of the games under these circumstances, as shown in 4.3. For $f = 2$

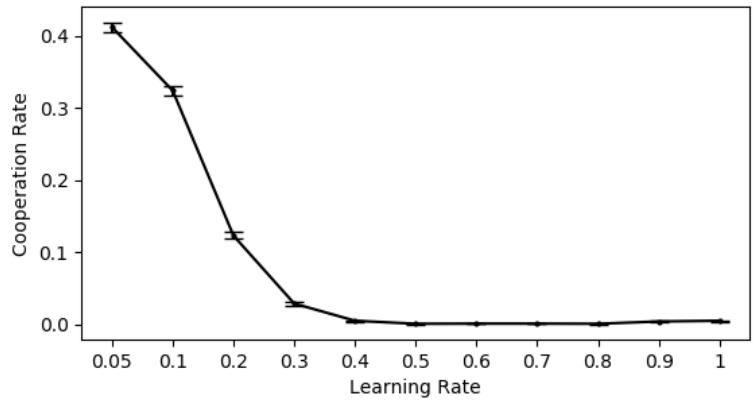

Figure 4.4: The cooperation rate in the last 100 rounds in NPD ($f = 2$) with five *MajorTD4* following ϵ-greedy policy with $\epsilon = 0.001$ for different values of α.

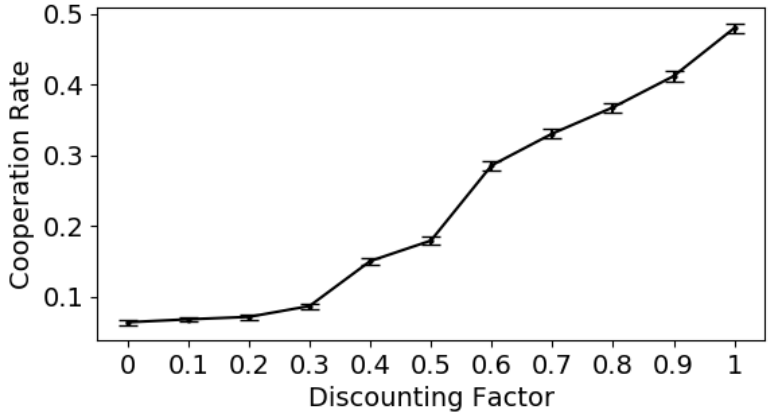

Figure 4.5: The cooperation rate in the last 100 rounds in NPD ($f = 2$) with five *MajorTD4* following epsilon greedy policy with $\epsilon = 0.001$ for different values of γ.

there will be cooperation over 60% only with $\epsilon = 0.00001$.

In IPD the cooperation is boosted by low values of α and similar results for NPD was expected, the empirical results matches the expected as shown in figure 4.4. The learning factor measures how fast the agent learns and how strongly it uses old knowledge. In other words, a high learning factor makes the agent use the knowledge it acquired recently and lessen the impact of old knowledge in decision making, while low values make the agent learn slowly but allows it to accumulate better the knowledge acquired during learning, thus using more data than high values of α would allow.

Regarding the last parameter γ empirical results match the expected results as well, as shows in figure 4.5. The discounting factor is a weight on future rewards, hence the higher the value the more the agent will prioritize long term over short term gains.

The results in this section match the results found in [9]. In other words, the generalization proposed fits well, since the relations found with RL agents in IPD and NPD are the same. Cooperation rates are

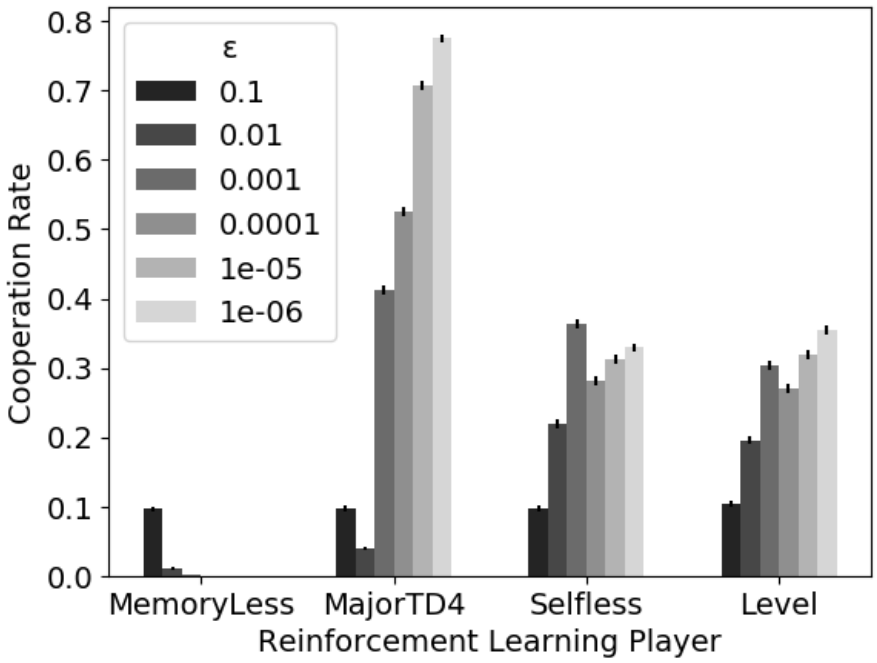

Figure 4.6: The cooperation rate in the last 100 rounds in NPD ($f = 2$) with five players following ϵ-greedy policy for different values of ϵ and different players.

boosted by high value of γ and low values of α and ϵ.

4.2.2 Cognition Levels Comparison

Regarding cognition, TD learners have two traits of interest. The first is the state space, it is responsible for the perception of the agent, the bigger and more detailed state space, the more it perceives from the environment theoretically. While the policy is the methodology to make decisions based on the state space, what is a form of reasoning. The first experiment show what happens when we fix a static ϵ-greedy policy and gradually increase the state space. Then the second experiment fixes a state space and increase the policy complexity with the caution of not decreasing exploration while doing so. Finally the last experiment selects the policy that generated the best cooperation rate in previous experiment and test the combination of other state space with it. The last experiment shows that the configuration with best cooperation rates is *LevelLearner* with actor critic policy, $\alpha = 0.05$ and $\gamma = 0.9$, actor critic policy does not have an explicit exploration factor.

The cooperation rates for the four RL players are disposed in figure 4.6. The players are disposed from the smaller state space to the largest. The first thing to notice is the expected bad result of *MemoryLess*, that as *TD1*, does not have any information of the current state of the game, hence it only learns to defect. Then we go to *MajorTD4*, that, as *TD4*, is very sensible to ϵ, but for very small ϵ can achieve high cooperation rates. The unexpected result is with *SelflessLearner* and *LevelLearner*. Since

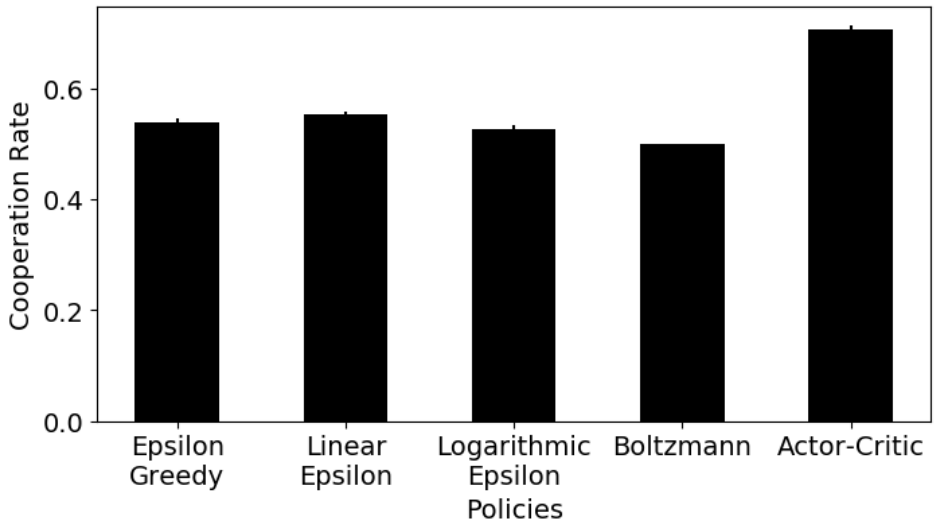

Figure 4.7: The cooperation rate in the last 100 rounds in NPD with five *MajorTD4* for different policies.

the increase in $|S|$ from *MemoryLess* to *MajorTD4* resulted in a great improvement in cooperation, it was expected that the state space enlargement from *MajorTD4* to *SelflessLearner* and from *SelflessLearner* to *LevelLearner* would have the same effect. However, from *MajorTD4* to *SelflessLearner* there is a reduction of cooperation for all ϵ besides $\epsilon = 0.01$, and from *SelflessLearner* to *LevelLearner* the results are very similar, besides the state space of *LevelLearner* being twice the size of *SelflessLearner*.

Because of these results, in the next experiment the state space used is *MajorTD4*. Then other four policies are tested. The first two of them use ϵ-greedy but instead of static values of ϵ, it starts at ϵ_0 and decays by a linear function or a logarithmic one. The other two use probability density functions, Boltzmann uses the q-value table directly to calculate the probabilities while actor critic uses the q-value table to learn a vector of probabilities, that are learned parameters. Each of these policies has an internal parameter, the cooperation rates when varying this parameters are in appendix A. The values shown in figure 4.7 are for internal parameters that better attempts to improve cooperation without giving up on exploration during learning.

The main result regarding policy improvement is shown in figure 4.7, where there are the comparison ϵ-greedy with $\alpha = 0.0001$, ϵ-greedy with Linear Decreasing Epsilon starting at $\epsilon_0 = 0.1$, ϵ-greedy with Logarithmic Decreasing Epsilon starting at $\epsilon_0 = 0.001$, Boltzmann distribution with $\beta = 0.01$ and Actor-Critic with $\alpha_P = 1$. In this way we intend to find a policy that consistently uses experience acquired during learning to enhance its cooperation rate and in this matter Actor-Critic is better than the others.

The average times an agent changes its strategy during learning and its standard deviation is shown on table 4.1 for the policies in figure 4.7. First thing to notice on the table 4.1 is how the strategy changes sharply drop when decreasing ϵ. The best static value for ϵ is $\epsilon = 0.001$, because for $\epsilon = 0.01$ the agent changes strategy on average more than 100 times during learning even though there is only 16 possible

Policy		Strategy Changes	
Algorithm	Parameter	Average	Standard Deviation
ϵ-Greedy	ϵ = 0.01	104.91	142.96
ϵ-Greedy	ϵ = 0.001	3.812	14.92
ϵ-Greedy	ϵ = 0.0001	1.84	1.90
Linear-Dynamic-ϵ	ϵ_0 = 0.1	1.56	1.32
Log-Dynamic-ϵ	ϵ_0 = 0.001	1.83	1.44
Boltzmann	β = 0.01	9.26	4.62
Actor-Critic	α_P = 1	2.82	4.32

Table 4.1: Average number of changes on strategy for 1000 NPD games ($f = 2$), *MajorTD4* and 5 players during learning for different policies.

strategies given *MajorTD4* state space, what supports the claim that with low values of ϵ the RL players are perceived as random, while for $\epsilon = 0.0001$ the players change strategy on average less than two times what is far from trying most of the possible strategies. Another issue with the ϵ-greedy policy is the high variance, which is lessened by using Linear Dynamic Epsilon but does not solve the problem with low strategy exploration. The Logarithmic Dynamic Epsilon enhance the strategy exploration but explodes the variance. Boltzmann has a good strategy exploration with a controlled variance but bad cooperation rates. Finally, Actor-Critic policy is the only policy that increases cooperation rate and exploration when compared to ϵ-greedy ($\epsilon = 0.0001$).

It was expected that actor critic policy to be the best policy since it its learning process can control better the variance during learning, the unexpected factor was the state spaces of larger size than *MajorTD4* to have worse results. *SelflessLearner* and *LevelLearner* should have higher cooperation rates because they have more detailed information on the current level of cooperation in the game. So in order to answer that other state spaces were tested with the Actor-Critic policy. The number of training rounds and execution rounds do not change from previous experiments. The results for different value of α_P are shown in figure 4.8.

In figure 4.8, it is possible to see that the best configuration is *LevelLearner* with $\alpha_P = 0.05$. Regarding the balance between exploration and exploitation, the average strategy change and standard deviation of this experiment is 25.34 ± 8.28 for this α_P. In other words, the combination of the state space of *LevelLearner* with the Actor-Critic policy resulted in the best balance between exploration and exploitation, because the agent changes more than 20 times of strategy and at the end still has a cooperation rate of almost 80%.

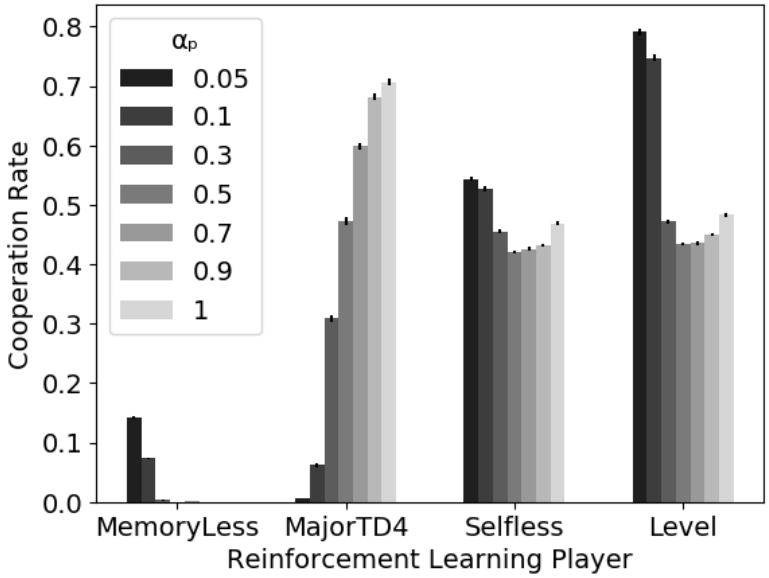

Figure 4.8: Cooperation rates of five agents playing NPD ($f = 2$) following Actor-Critic policy for different α_P through 1000 games.

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

Table 4.2: *MajorTD4* strategy mapping to binary with names of important strategies.

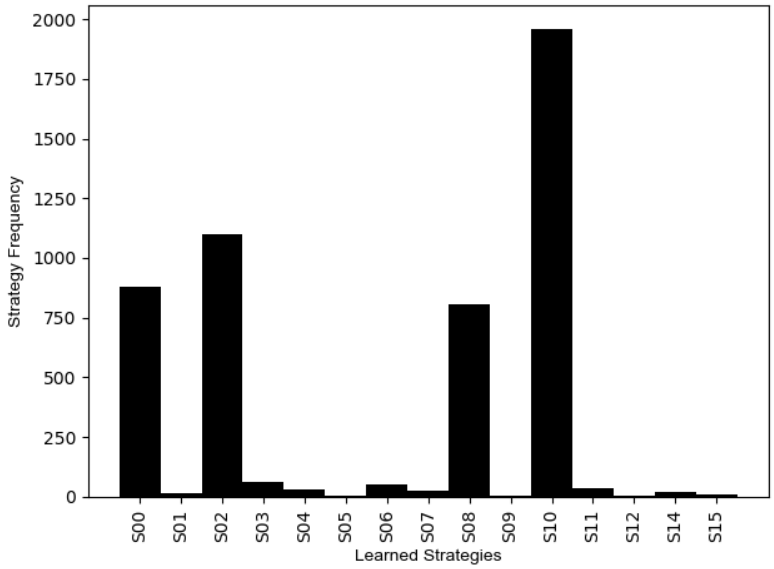

Figure 4.9: Strategies learned by five *MajorTD4* playing NPD ($f = 2$) following ϵ-greedy policy ($\epsilon = 0.01$) through 1000 games.

4.3 Strategy Analysis

This section steps out of specific parameter configurations and analyses how the agent are playing and consequently what they learned. In order to do that, we limited our scope just to *MajorTD4* due to its simplicity, as discussed in section 3.2, the q-value table of *MajorTD4* allows it to learn one of the 16 memory-one pure strategies, that means that it only learns to cooperate or to defect in each one of its four possible states. The only exception is *MajorTD4* with Actor-Critic policy, that allows the agent to learn to cooperate with a given probability for each state, allowing it to learn mixed strategies, which means that it is possible to learn to cooperate with a 70% chance in a given state for example, instead of only learning to cooperate with a 0% or a 100% chance for each state.

The experiments of this section is slightly different from the one of the previous one. The first difference is that there is no execution rounds. The information is extracted at the end of the 20000 learning rounds. The other difference is that instead of extracting the information from the actions the agents have chosen in the last turns, the information for determining which strategy the agent is playing comes from the agent's q-value table.

The first three experiments are with the standard parameters used in previous experiments and ϵ-greedy policy with $\epsilon = \{0.01, 0.001, 0.0001\}$, the data of those experiments are exposed in figures 4.9, 4.10 and 4.11. The first thing to notice is that, independently of ϵ, all three experiments showed a prominent amount of players that learned S10, TFT, even though the number of TFT slightly decreases when decreasing the exploration factor. For $\epsilon = 0.01$ the next three more learned strategies are S0, S02

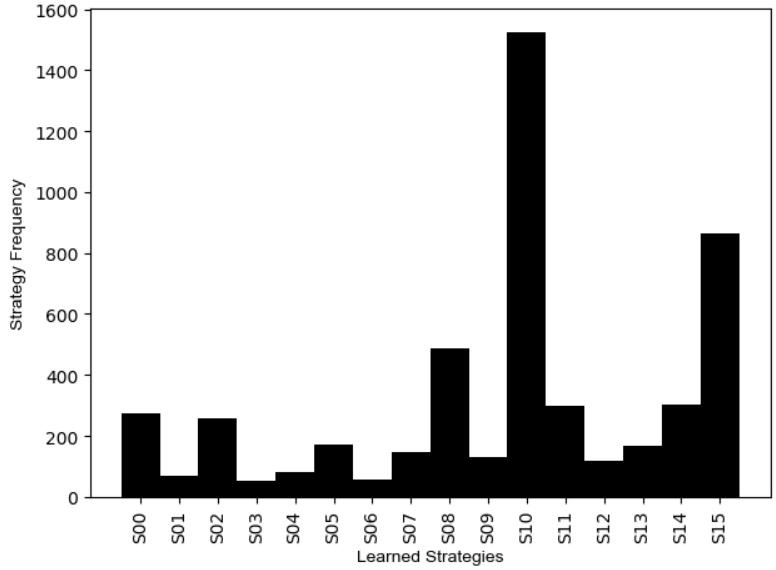

Figure 4.10: Strategies learned by five *MajorTD4* playing NPD ($f = 2$) following ϵ-greedy policy ($\epsilon = 0.001$) through 1000 games.

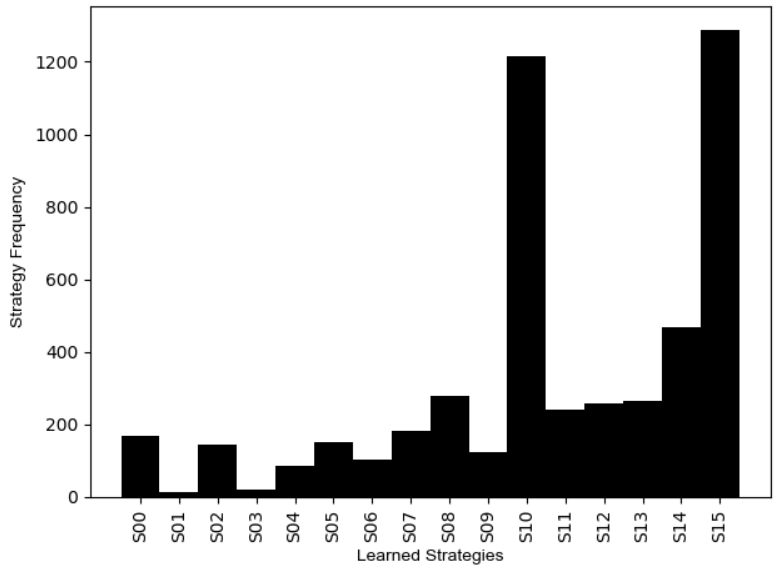

Figure 4.11: Strategies learned by five *MajorTD4* playing NPD ($f = 2$) following ϵ-greedy policy ($\epsilon = 0.0001$) through 1000 games.

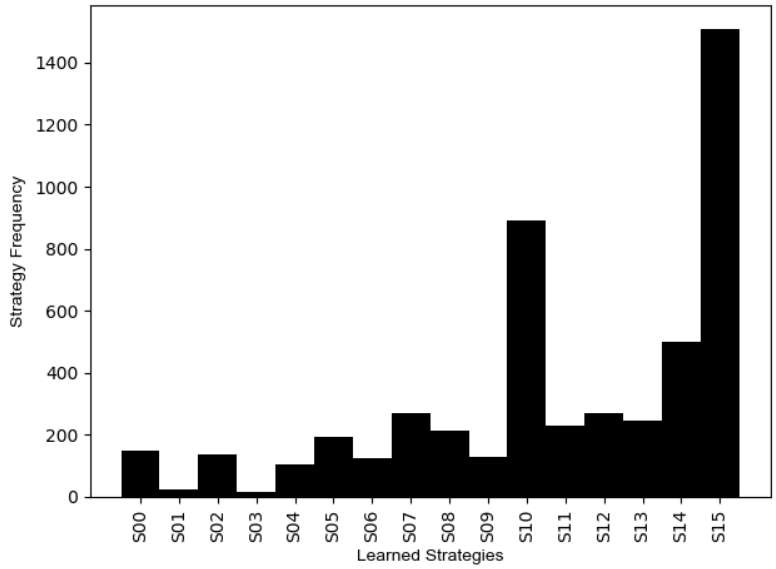

Figure 4.12: Strategies learned by five *MajorTD4* playing NPD ($f = 2$) following linear decreasing epsilon greedy policy ($\epsilon_0 = 0.1$) through 1000 games.

and S08, that are strategies bad for cooperation. S0 is ALLD, S2 only cooperates when most opponents cooperates and the agent defected last turn and S8 only cooperate when everyone, including itself, is cooperating. When the exploration factor is reduced to $\epsilon = 0.001$ there are huge changes on the strategies learned. First thing is that the occurrence of strategies S0, S2 and S8 is sharply reduced and S15 = ALLC appears as the second most frequently learned strategy. Other two strategies that worth mentioning is the S5 and S7 strategies, that are strategies that incentive cooperation but exploit cooperative players with cooperation is high, increase when ϵ is decreased. S5 is the opposite of S10, while S10 copies opponent's last action, S5 plays the opposite of the opponent's last action, hence when most players are cooperating it is defecting and vice versa. On the other hand, S7 cooperates always but when most players and itself cooperated. This characteristic of cooperating when others do not seems like an attempt to create the conditions to make others cooperate, together with defecting when cooperation is already happening create this idea of a strategy focused on free riding, in other words, focused on taking advantage of others cooperation. Besides this effect with the strategies S5 and S7, figure 4.11 shows that reducing one more time the exploration factor reduces even more defective strategies S0, S2 and S8 and increases even more S15, while maintaining high levels of TFT.

After experimenting with different values of ϵ, two experiments with decreasing exploration factor were executed, one with linear decreasing ϵ with $\epsilon_0 = 0.1$ and the other with logarithmic decreasing ϵ with $\epsilon_0 = 0.01$. One thing to notice is the similarities with static ϵ strategy distribution. The logarithmic dynamic ϵ data of figure 4.13 is similar with static $\epsilon = 0.01$ data of figure 4.9, both have high frequency of strategies S0, S2, S8 and S10. On the other hand, the data of linear dynamic ϵ in figure 4.12 is

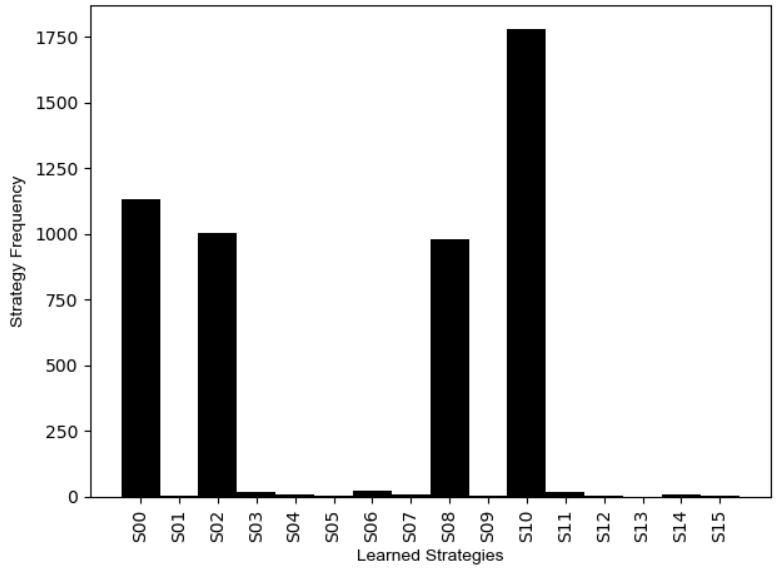

Figure 4.13: Strategies learned by five *MajorTD4* playing NPD ($f = 2$) following logarithmic decreasing ϵ-greedy policy ($\epsilon_0 = 0.01$) through 1000 games.

very similar with the data of static $\epsilon = 0.0001$ in figure 4.11, with high frequency of S10 and S15, while having smaller but still significant frequency of the defective strategies S0, S2 and S8, and the free rider strategies S5 and S7. The highlights of linear decreasing ϵ when comparing to static $\epsilon = 0.0001$ is the decrease of S10 over S15 and the slightly increase of S7 while reducing S8.

Concerning the Boltzmann policy, the first probability distribution function policy tested, the results where dramatically different. Every simulation resulted in every agent learning S0, that is ALLD. What may indicate a bad path moving from ϵ-greedy policies to probability distribution ones. However Actor-Critic uses linear probability distribution function and ended up with results completely different from Boltzmann.

Agents that follow linear Actor-Critic policy has quite different strategy distribution. ALLC is by far the most frequent strategy followed by S5 and S10. S7 almost disappear and the defective strategies S0, S2 and S8 are close to the level of static $\epsilon = 0.0001$ and linear decreasing ϵ. Besides Boltzmann (that only learns to ALLD) this is the configuration which TFT appears the least.

However an agent following Linear Actor-Critic policy does not learn only the quality of the actions for each state, it learn also how frequently it should cooperate. Hence the strategy distribution shown in figure 4.14 does not reveal exactly how these agents are playing, however, since SARSA is an on policy algorithm, the selected learning policy impacts a lot on what is learned on the q-value table. For further information, it is necessary to look at the probabilities these agents learns. The average and standard deviation for the probabilities to cooperate in each state are shown in table 4.3 for *MajorTD4*, in table 4.4 for *SelflessLearner* and in table 4.5 for *LevelLearner*.

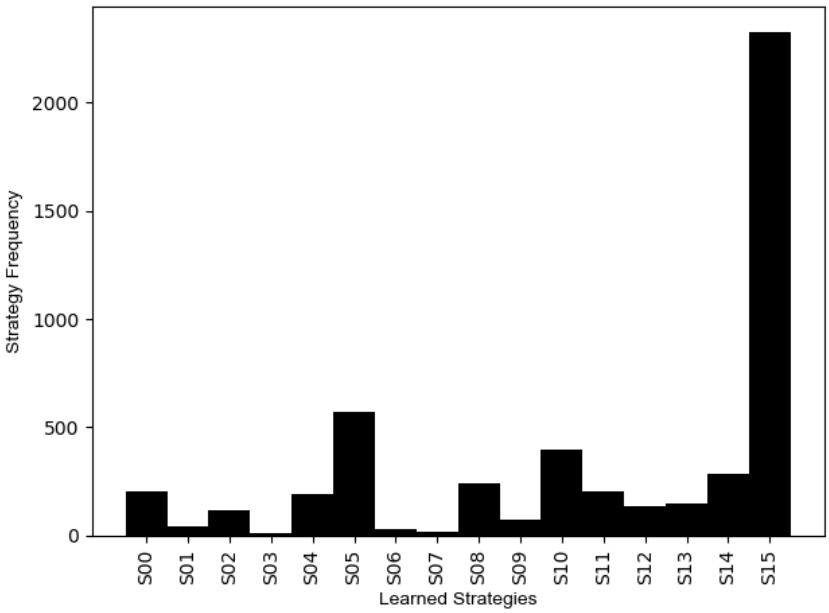

Figure 4.14: Strategies learned by five *MajorTD4* playing NPD ($f = 2$) following Actor-Critic policy ($\alpha_P = 1$) through 1000 games.

The three tables have some similarities. All of them have some degree of cooperation even on some scenarios of widespread defection and none of them achieve full cooperation in the scenarios with high cooperation. Regarding the differences, the standard deviations vary considerably when comparing the three tables. The deviations of table 4.3 are relatively high independently of the state, table 4.4 show high deviations for states with high cooperation and low deviations for states with little cooperation, while the deviations of table 4.5 are almost zero for every possible state. There is a progression, as the states of the game are specified in more detail, the variation of what is learned in each state decreases. This means that those learning agents are learning in a more similar way when increasing the state space. The high standard deviations can be explained by two things: agent specialization or ill-defined states. In the first case, the agents converge for two or more different game strategies, having different values for the probabilities of cooperating in each state. This leads to a heterogeneous group and can produce the high standard deviations of table 4.3. The second case considers the possibility of designing too generic states, that when the agent test these states it gets good and bad responses at similar probabilities, what makes difficult to extract any knowledge from, what may lead different agents converge to different probabilities in this state. In this sense, *LevelLearner* do not need more improvement in state space since the deviations for all its states are near to zero.

The explanation for a high cooperation rate with high exploration that *SelflessLearner* and *Level-Learner* have can be explained by a learned recover mechanism, that allows them to move quickly from a state of widespread defection to a high cooperation state, this can be noticed by the probability of

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 4.3: 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 4.4: Average probability to cooperate and average deviation of *SelflessLearner* following Linear Actor-Critic ($\alpha_P = 0.1$) policy for each state of *S*.

cooperating when nobody is cooperating of almost 60% for these agents. They do not cooperate when there are a few cooperating, cooperate more if more individuals are cooperating, unless everyone is cooperating, when the cooperation rate is slightly reduced.

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

5

Conclusion

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.

Less exploration means less adaptability. Adapting to environment changes is a key ability for many animals, including humans, that can achieve high cooperation among individuals. To answer the second question, we give general principles that boost cooperation and how is the dynamic between the level of cognition and the cooperation of the group.

These principles emerge from the relations of the learning step, the discounting and the exploration factor with the cooperation rates: cooperation increases for high values of γ and low values of α and ϵ. 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. So, groups of individuals that learn by trial and error and follow these principles tend to cooperate more when compared to the groups that do not.

Further on, cognition plays a key piece to increase 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. Analyzing 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. 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. This means that the cooperation does not increase by only increasing how much information the player has, neither by only improving how the player uses that information, but by selecting the method that better utilizes the information.

Another approach is to assume the player knows how to use the information. Cooperation cares about the rough total amount of information or cares about knowing specific relevant information? The amount of knowledge is measured by the state space size of the agent and only increasing the state space size does not improve cooperation. This is clear when comparing *SelflessLearner* with *LevelLearner*, both have approximately the same information, but as *LevelLearner* knows its own actions it has a much larger state space, that does not reflect in a higher cooperation rate with ϵ-greedy policy. On the other hand, when *MajorTD4* and *LevelLearner* follow actor critic, the latter show greater cooperation and much higher exploration during learning. The most significant difference between these two is that the first knows the most frequent action opponents choose, while the second knows how many cooperated each turn, thus is the quality of information not the amount that matters most.

Cognition improves cooperation among *RL* players, but is there a cognition level that any further improvement does not improve cooperation? In this work the best result is with the highest cognition level. If we consider that total cooperation is impossible and the larger the group the higher the incentives to free ride, *LevelLearner* following actor critic has the best cooperation rate possible (80%), that is with one player free riding in a group of 5 players, so further cognition improvements would not reflect on higher cooperation. However, if it is possible to overcome free riding without a central entity to punish this behavior then a higher cognition level may improve cooperation rates.

Nevertheless, the best outcome is when putting together *LevelLearner* with Actor-Critic policy, for the optimal policy parameter $\alpha_P = 0.05$ the cooperation rate is over 80% and the agent changes strategy more than 20 times on average during learning, what shows a good balance on exploitation and exploration. The probabilities the agents learn on average show that the agents learn that if defected last round they keep defecting unless everyone is defecting, in this case they cooperate with a probability of 60%, if cooperated last round the probability of cooperating increases with the number of players that cooperated in the previous turn, reaching its peak when four players (counting itself) cooperate, state in which they learn to cooperate with almost 100%. If everyone is cooperating the cooperation rates decrease to 60% what shows a certain lenience with free riding. In resume, the agents that cooperate the most have a recover mechanism to go fast from a state of no cooperation to a state of fairly high cooperation and these agents' society stabilizes when most individuals are cooperating and a few are free riding.

Actually in section 4.3 there are similar although less expressive results. The experiments with higher cooperation rates are the one where the number of TFT decays while the number of ALLC stay high and S5 grows. S5 is a strategy that cooperates when there is low cooperation and free rides when everyone is cooperating, similar mechanism encountered in the best results. However there is an important distinction between the two and this difference is due to *MajorTD4* state space. Since *MajorTD4* only analyses the action chosen by the majority of the opponents it can not distinguish when there is no one cooperating from when there is only one or two players cooperating. While for *SelflessLearner* and LevelLearner with actor critic it is important to differentiate these two situations to cooperate when no one is cooperating and defect when only a few are cooperating.

Finally this is not a good game for TFT. It is worth to highlight that it is a very frequent strategy during learning however it does not translate in good cooperation rates. This happens because TFT does not try to cooperate when everyone is defecting, it does not have a recovery mechanism. It is preferable to have a Free Riding strategy like S5 that allows the recovery mechanism but exploit others cooperation than the "righteous" TFT that does to its opponents what they have done to it. This does not mean that those players are irrational, since against TFT the best strategy is to cooperate, one argument to question these agents rationality is that when the first **TFTs!** start to appear during learning the others

should start to cooperate. The problem is that learning is happening very slowly and the other players must perceive that someone is playing TFT, what may be difficult with various players and if at some point the game just stick in Defection and TFT only defects, TFT and ALLD become undifferentiated.

From the point of view of evolution of cooperation this result resembles human societies. Where it is necessary a high percentage of cooperation, this can be translated in obeying to laws, paying taxes and keeping word. While liking or not there is a small percentage that tries and finds a way to exploit the system. Finally, when confronted with a dramatic situation like a natural disaster or war, people have this capacity of bond together and help each other, being exploited lose importance when compared to the gravity of the situation. That is basically when no one is cooperating, when the society loses the ability to produce goods, welfare and culture. While our best result resembles today human societies, a society of TFT resembles Code of Hammurabi, the law of ”an eye for an eye”, that is not acceptable anymore in most human societies.

Regarding future work, experiments with larger populations seem the first step to accomplish in order to see if these result escalate with the population. Then try to answer why Actor-Critic policy has such a nice results and try to discover if there is any seemliness with the evolution of cooperation in human societies through out history. Finally, this work focused on bringing experimental results, however there are some curves that resembles known functions like exponential, logarithmic or even polynomial, it would be a nice work to discover general formulas that explain these results.

Bibliography

[1] T. Kochiyama, N. Ogihara, H. C. Tanabe, O. Kondo, H. Amano, K. Hasegawa, H. Suzuki, M. S. Ponce de León, C. P. E. Zollikofer, M. Bastir, C. Stringer, N. Sadato, and T. Akazawa, "Reconstructing the neanderthal brain using computational anatomy," *Scientific Reports*, vol. 8, no. 1, p. 6296, 2018. [Online]. Available: https://doi.org/10.1038/s41598-018-24331-0

[2] E. Van Wilgenburg, C. W. Torres, and N. D. Tsutsui, "The global expansion of a single ant supercolony," *Evolutionary applications*, vol. 3, no. 2, pp. 136–143, Mar 2010, 25567914[pmid]. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/25567914

[3] D. A. Holway, L. Lach, A. V. Suarez, N. D. Tsutsui, and T. J. Case, "The causes and consequences of ant invasions," *Annual Review of Ecology and Systematics*, vol. 33, no. 1, pp. 181–233, 2002. [Online]. Available: https://doi.org/10.1146/annurev.ecolsys.33.010802.150444

[4] R. Axelrod, *The Evolution of Cooperation*. New York: Basic, 1984.

[5] H. T. Anh, L. M. Pereira, and F. C. Santos, "Intention recognition promotes the emergence of cooperation," *Adaptive Behavior*, vol. 19, no. 4, pp. 264–279, 2011. [Online]. Available: https://doi.org/10.1177/1059712311410896

[6] S. Tanabe and N. Masuda, "Evolution of cooperation facilitated by reinforcement learning with adaptive aspiration levels," *Journal of Theoretical Biology*, vol. 293, pp. 151 – 160, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022519311005352

[7] E. Thorndike, *Animal intelligence: Experimental studies*. Routledge, 2017.

[8] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, pp. 1593–1599, 1997.

[9] N. Masuda and H. Ohtsuki, "A theoretical analysis of temporal difference learning in the iterated prisoner's dilemma game," *Bulletin of Mathematical Biology*, vol. 71, no. 8, pp. 1818–1850, Nov 2009. [Online]. Available: https://doi.org/10.1007/s11538-009-9424-8

[10] J. Vyrastekova and Y. Funaki, “Cooperation in a sequential n-person prisoner’s dilemma : the role of information and reciprocity,” *Human Movement Science - HUM MOVEMENT SCI*, 01 2010.

[11] L. Guo, Z. Liu, and Z. Chen, “A leader-based cooperation-prompt protocol for the prisoner’s dilemma game in multi-agent systems,” 07 2017, pp. 11 233–11 237.

[12] N. S. Glance and B. A. Huberman, “The dynamics of social dilemmas,” *Scientific American*, vol. 270, no. 3, pp. 76–81, 1994.

[13] P. Kollock, “Social dilemmas: The anatomy of cooperation,” *Annual review of sociology*, vol. 24, no. 1, pp. 183–214, 1998.

[14] J. Farago, A. Greenwald, and K. Hall, “Fair and efficient solutions to the santa fe bar problem,” 02 2003.

[15] I. Guerberoff, D. Queiroz, and J. Sichman, “Studies on the effect of the expressiveness of two strategy representation languages for the iterated n-player prisoner’s dilemma,” *Revue d’Intelligence Artificielle*, vol. 25, pp. 69–82, 03 2011.

[16] N. Glance and T. Hogg, “Dilemmas in computational societies.” 01 1995, pp. 117–124.

[17] A. Lerer and A. Peysakhovich, “Maintaining cooperation in complex social dilemmas using deep reinforcement learning,” 07 2017.

[18] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

A

Policies cooperation rates when varying internal parameters

This appendix show the value of cooperation rates and strategy changes for each policy besides for static ϵ-greedy and actor critic that are show respectively in figures 4.6 and 4.8. The data show in figure 4.7 are the values of static ϵ-greedy ($\epsilon = 0.0001$) and the policies that produce higher cooperation rates with higher exploration (or very close). The average strategy changes for some policy configurations are in table A.1.

For both policies with dynamic epsilon the results are alike, as figures A.1 and A.2 show. The cooperation rates increase when comparing the results of $\epsilon_{fixed} = \epsilon_0 = 0.001$, for example, in both cases, even though in Linear Dynamic Epsilon the results are more accentuated.

Differently from previous policies, Boltzmann policy does not have ϵ factor, it has only the β parameter. As this parameter approaches zero, the policy increasingly starts ignoring the values of the q-value table, until converts itself into the Random policy (chooses actions randomly) when $\beta = 0$. Probably the cooperation rates for $\beta = 0.01$ and $\beta = 0.1$, shown in figure A.3, do not show a learned strategy, like

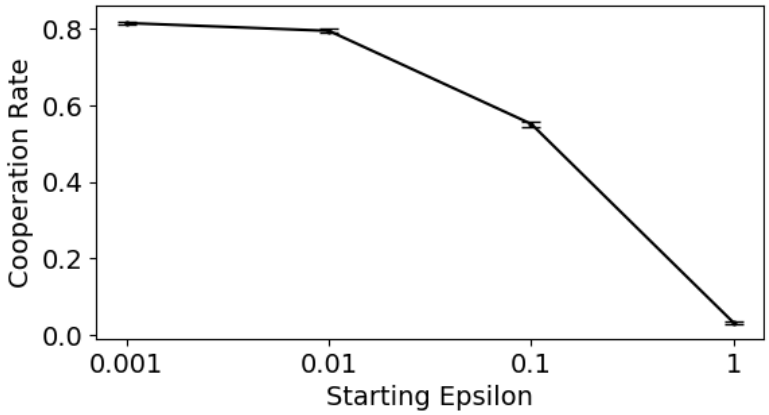

Figure A.1: Cooperation Rates in NPD ($f = 2$) with five *MajorTD4* following ϵ-greedy with Linear Dynamic Epsilon policy for different values of ϵ_0.

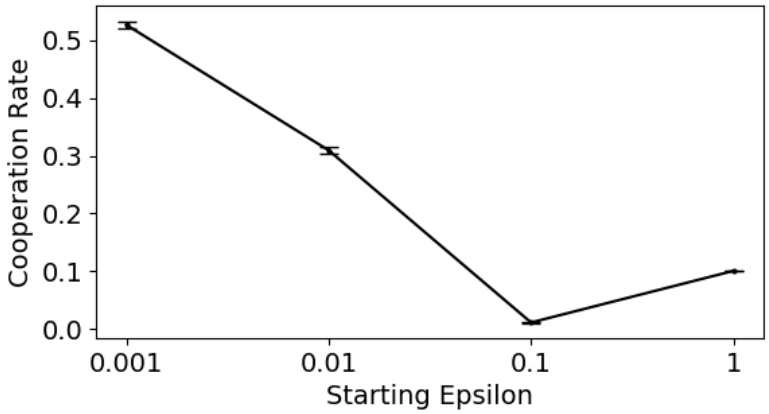

Figure A.2: Cooperation Rates in NPD ($f = 2$) with five *MajorTD4* following ϵ-greedy with Logarithmic Dynamic Epsilon policy for different values of ϵ_0.

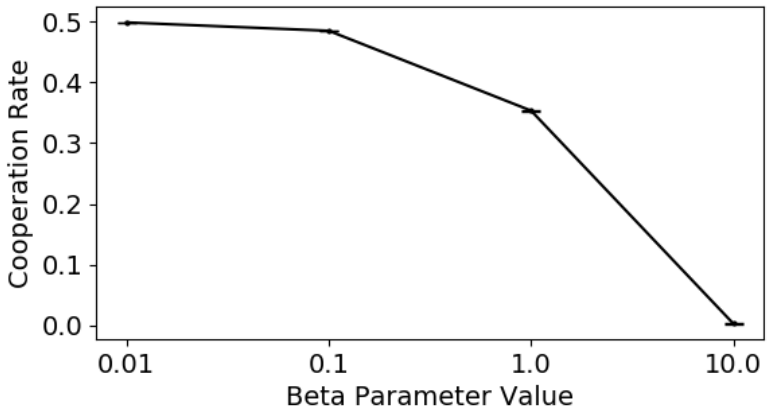

Figure A.3: Cooperation Rates in NPD ($f = 2$) with five *MajorTD4* following Boltzmann policy for different values of β.

Policy		Strategy Changes	
Algorithm	Parameter	Average	Standard Deviation
Linear-Dynamic-ϵ	ϵ_0 = 0.1	1.56	1.32
Linear-Dynamic-ϵ	ϵ_0 = 0.01	0.84	0.93
Log-Dynamic-ϵ	ϵ_0 = 0.01	4.92	22.01
Log-Dynamic-ϵ	ϵ_0 = 0.001	1.83	1.44
Boltzmann	β = 1	7.80	3.50
Boltzmann	β = 0.1	8.98	4.35
Boltzmann	β = 0.01	9.26	4.62
Actor-Critic	α_P = 1	2.82	4.32
Actor-Critic	α_P = 0.9	2.63	4.01
Actor-Critic	α_P = 0.7	3.14	2.89

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

Alternate (ALT), but indicate already that those values are too low, hence Boltzmann policy is already behaving randomly. However when we increase the β, the agent just learn to defect most of times. Hence we select the intermediate value of $\beta = 1$ as the most reasonable result for comparing with other policies.

The reason Boltzmann distribution with the q-value table values is not a good policy is probably due to either it becomes Random policy or uses the slightest differences of q-value table to choose one action over the other too soon without exploring enough, for high values of β.

B

Cooperation in N-Person Stag Hunt Game (NSH)

Beyond NPD, this appendix investigates NSH in order to compare to NPD. The only difference is that in NSH the number of cooperators must surpass a threshold to have a reward at all:

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

where the *H(x)* function is the Heaviside function, if $x < 0$ the function returns 0 and if $x \geq 0$ the function returns 1. This means that, if the number of cooperators *k* is not at least equal to the threshold *T*, in other words, if $k - T \geq 0$ does not hold, the Heaviside function returns zero, resulting in zero reward. Otherwise, the Heaviside function is 1 and the reward function is exactly equal to NPD.

In this study we put five *MajorTD4* players to play Stag Hunt Game ($f = 2$) against each other following epsilon greedy policy with $\epsilon = 0.001$ for all possible values of threshold.

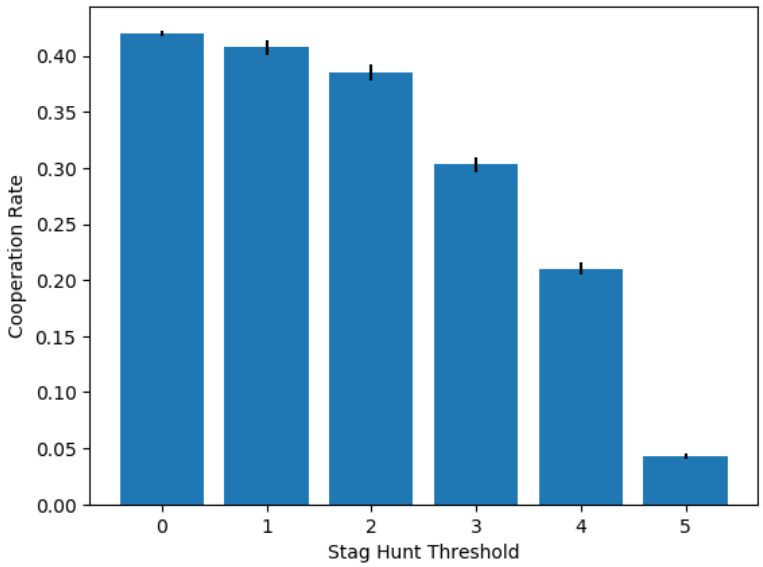

Figure B.1: Cooperation rate for five *MajorTD4* playing Stag Hunt Game ($f = 2$) for different threshold values.

The Stag Hunt game is very similar to NPD, actually when threshold is zero SH becomes NPD. Figure B.1 show that increasing threshold only results on reduction of cooperation. In other words, for *MajorTD4*, increase the threshold just turns the game harder for cooperation.