Cooperation is the rule, not the exception
Reinforcement Learning in the Battle of the Sexes

Bernat Puig Camps

Supervisor: Vicenç Gómez Cerdà
Co-Supervisor: Martí Sanchez Fibla

September 2018
Cooperation is the rule, not the exception
Reinforcement Learning in the Battle of the Sexes

Bernat Puig Camps

Supervisor: Vicenç Gómez Cerdà
Co-Supervisor: Martí Sanchez Fibla

September 2018
Abstract

Society is highly influenced by conventions, which are a form of cooperation. In many situations, individuals act together for the benefit of the group. This phenomenon is easy to understand when all individuals share the same interest. However, when there exists conflict, it is not clear if altruism is required or pure self-interest can lead to cooperation. The repeated version of the Battle of the Sexes game can summarize this situation. Although conflict is present, players need to cooperate to obtain good rewards. Here we show experimentally that two selfish reinforcement learning agents learn to cooperate in this conflictive scenario. We found that two Q-learning agents playing this game modeled as a Markov Game reach a cooperative fair solution. That is, two agents that learn based solely on their own self-interest end up cooperating. Furthermore, we found that Q-learning is able to converge in this multi-agent situation. Our results demonstrate that cooperation among individuals in this particular conflictive scenario can be explained by means of pure self-interest. Moreover, cooperation in this setting is the rule, not the exception as the convergence to it is robust to parameter asymmetry between agents. We also introduced opponent modeling into the players as a Beta binomial model. It worked well in modeling the adversary but agents fail to properly exploit that knowledge.

Keywords: Cooperation; Selfish agents; Q-Learning; Game Theory; Opponent modeling
Chapter 1

Introduction

Cooperation can be understood as the process of many individuals acting together for the benefit of the group, rather than competing solely for the selfish benefit. Nature is full of examples of this kind of behavior. Consider, for instance, mutualistic symbioses: the fungus and alga that compose a lichen or the ants and ant-acacias, where the trees house and feed the ants which, in turn, protect the trees [1].

Moreover, the human world is also highly influenced by cooperation: the way we live in society and interact with others is governed by strong societal conventions [2]. Traffic regulations pose a clear example of the situation: a driver in a hurry ignoring them might indeed arrive faster to his or her destination, but there is also a high chance that he or she does not arrive at all. Thus, people tend to follow the established regulations. Note, however, that this only makes sense as long as everyone follow the same convention. That is, as long as all drivers cooperate.

These examples raise the question if cooperation can be explained by means of pure self interest and altruism plays no role in its emergence, specially when conflict is present (i.e. the interests among individuals are not shared). This question has already been addressed, from the game theoretic point of view, concluding that the emergence of cooperation in a world of egoists is possible [3].

Starting from this point, we want to test whether or not cooperation in this setting
can be reached by remarkably simple independent (reinforcement) learning agents that are motivated exclusively by their self-interest. That is, agents that learn and choose their actions independently based only on maximizing their own well-being.

The basic idea of Reinforcement Learning (RL) is that in order to learn, an agent tries different strategies, observe how they affect it and progressively learn to adopt the most rewarding ones. Thus, RL is particularly suited to address the problem in which an individual agent wants to learn the actions that benefits it the most in a stationary environment [4]. Nevertheless, the multi-agent situation is more intricate: the actions of one agent can affect the other agents which in turn, choose their actions based on the observed consequences. Again, these actions also affect the rest. In the end, this establishes an endless recursivity that causes the environment perceived by one agent to be non-stationary. Hence, this makes it hard to determine whether or not it is possible for an agent to learn something in this situation, not to mention if learned cooperative behavior is even possible.

Different works addressed this problem with a myriad of variations and results. In some cases, tailored methods (to different extents) have been used to solve the generic multi-agent problem [5, 6, 7, 8, 9]. In others, only the case of full-cooperativity is considered (i.e. all agents have the same shared interest and there is no conflict) [10, 11, 12].

In contrast, our approach is to use directly off-the-shelf RL methods in a proper setting in which conflict is present and cooperation is also required. Such setting can be modeled by means of game theory.

Game theory is a powerful framework that has been widely used in the literature to model multiple agents and their interests and to study the situations generated by their interactions [13].

Section 1.1 introduces the basics of game theory, based in [13], along with the description of Battle of the Sexes, the particular game we consider in order to formalize and investigate our problem in a more rigorous context. Section 1.2, based in [4], deals with the necessary foundations of RL and describes Q-learning, the learn-
ing method implemented. Both game theory and RL approach similar problems in
the sense that both have rational agents with the goal of maximizing their reward.
Nevertheless, game theory is the analytical approach (i.e. theorem proving) while
the roots of RL are computational (i.e. how to design algorithmically an artificial
learning agent).

Game theory

Game theory is the mathematical study of interaction among independent, self-
interested agents. That is, agents that each of which has its own description of
which states of the world it likes.

The dominant approach to modeling an agent’s interests is utility theory. Specifi-
cally, a utility function is a mapping from states of the world to real numbers. These
numbers can be interpreted as measures of an agent’s level of happiness in the given
state.

When agents have utility functions, acting optimally in an uncertain environment
is conceptually straightforward: just choose the action that lands to the state with
the higher utility. However, things can get considerably more complicated when
the world contains two or more utility-maximizing agents whose actions can affect
each other’s utilities. To study such settings, we must turn to non-cooperative game
theory (the term “non-cooperative” only refers to the fact that the basic modeling
unit is the individual, not the group).

Normal-form games

A (finite, n-person) normal-form game is a tuple \((N, \mathcal{A}, r)\) where \(N\) is a finite set
of \(n\) players, indexed by \(i\); \(\mathcal{A} = A_1 \times \cdots \times A_n\) where \(A_i\) is a finite set of actions
available to player \(i\) (Each vector \(a = (a_1, ..., a_n) \in A\) is called an action profile);
\(r = (r_1, ..., r_n)\) where \(r_i : A \mapsto \mathbb{R}\) is a real-valued utility (also payoff or reward)
function for player \(i\).

A natural way to represent games is via an \(n\)-dimensional matrix. Figure 2 presents
such matrix for the Battle of the Sexes game. In general, each row corresponds to a possible action for player 1, each column corresponds to a possible action for player 2, and each cell corresponds to a possible outcome. Each player’s reward for an outcome is written in the cell corresponding to that outcome, with player 1’s utility listed first.

Although there are many different games, there are some restricted classes of normal-form games that deserve special mention.

The first is the class of common-payoff games. These are games in which, for every action profile, all players have the same payoff. These games represent pure coordination.

At the other end of the spectrum lie zero-sum games (meaningful primarily in the context of two-player games) which correspond to pure competition. These type of games are characterized by the fact that $r_1(a) + r_2(a) = 0$, where $a \in A_1 \times A_2$ is the strategy profile.

Finally, the games that do not fall within one of the previous categories are called general-sum games.

**Strategies in normal-form games**

The set of strategies (or policies) $\Pi_i$ of each player $i$ can be understood as its available choices. One kind of strategy is to select a single action and play it. We call such policy a pure strategy and a choice of pure strategy for each agent a pure-strategy profile.

Players could also follow another, less obvious type of strategy: randomizing over the set of available actions according to some probability distribution. Such a policy is called a mixed strategy.

By $\pi_i(a_i)$ we denote the probability that an action $a_i$ will be played under mixed strategy $\pi_i$. The subset of actions that are assigned to positive probability by the mixed strategy $\pi_i$ is called the support of $\pi_i$. 
The payoff of players given a particular strategy profile can be generalized using a basic notion of decision theory: the \textit{expected utility}. Formally, and overloading notation, the expected utility for agent $i$, $i = 1, \ldots, n$ following the joint policy $\pi$ is defined by

$$r_i(\pi) = \sum_{a \in A} r_i(a) \prod_{j=1}^{n} \pi_j(a_j)$$ \hfill (1.1)

\textbf{Analyzing Games: From Optimality to Equilibrium}

Reasoning about games is not straightforward. In single-agent decision theory the key notion to do so is that of an \textit{optimal strategy}, that is, a strategy that maximizes the agent’s expected payoff for a given environment in which the agent operates. However, the situation is remarkably complex in a multi-agent setting. In this case, the notion of an optimal strategy for a given agent is not meaningful; the best strategy depends on the choices of others.

Game theorists deal with this problem by identifying certain subsets of outcomes, called \textit{solution concepts}, that are interesting in one sense or another. In this section we describe two of the most fundamental solution concepts: Pareto optimality and Nash equilibrium.

\textbf{Pareto Optimality}

Our problem is to find a way of saying that some outcomes are better than others. While it is not usually possible to identify the best outcome, there \textit{are} situations in which you can be sure that one outcome is better than another.

A strategy profile $\pi$ \textit{Pareto dominates} strategy profile $\pi'$ if for all $i \in N$, $r_i(\pi) \geq r_i(\pi')$, and there exists some $j \in N$ for which $r_j(\pi) > r_j(\pi')$. In other words, in a Pareto-dominated strategy profile some player can be made better off without making any other player worse off.

Pareto domination gives us a partial ordering over strategy profiles. We cannot generally identify a single "best" outcome. Instead, we may have a set of non-comparable optima.
A strategy profile $\pi$ is *Pareto optimal*, or *strictly Pareto efficient*, if there does not exist another strategy profile $\pi' \in \Pi$ that Pareto dominates $\pi$.

We can easily draw several conclusions about Pareto optimal strategy profiles. First, every game must have at least one such optimum, and there must always exist at least one such optimum in which all players adopt pure strategies. Second, some games will have multiple optima.

### Defining Best Response and Nash Equilibrium

Now we will look at games from an individual agent’s point of view, rather than from the vantage point of an outside observer. This will lead us to the most influential solution concept in game theory, the *Nash equilibrium*.

Our first observation is that if an agent knew how the others were going to play, his strategic problem would become simple. Specifically, he would be left with the singe-agent problem of choosing a utility-maximizing action. Formally, define $\pi_{\setminus i} = (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n)$, a strategy profile $\pi$ without agent $i$’s strategy. Thus we can write $\pi = (\pi_i, \pi_{\setminus i})$. If the agents other than $i$ (whom we denote $\setminus i$) were to commit to play $\pi_{\setminus i}$, a utility maximizing agent $i$ would face the problem of determining his best response.

Player $i$’s *best response* to the strategy profile $\pi_{\setminus i}$ is a mixed strategy $\pi_i^* \in \Pi_i$ such that $r_i(\pi_i^*, \pi_{\setminus i}) \geq r_i(\pi_i, \pi_{\setminus i})$ for all strategies $\pi_i \in \Pi_i$.

Of course, in general an agent will not know what strategies the other players will adopt. We can leverage the idea of best response to define what is arguably the most central notion in non-cooperative game theory, the Nash equilibrium.

A strategy profile $\pi = (\pi_1, \ldots, \pi_n)$ is a *Nash equilibrium* if, for all agents $i$, $\pi_i$ is a best response to $\pi_{\setminus i}$. Intuitively, a Nash equilibrium is a *stable* strategy profile: no agent would want to change his strategy if he knew what strategies the other agents were following. That is, given a Nash equilibrium, an agent deviating from it would only worse off itself.
We can divide Nash equilibria into two categories, strict and weak, depending on whether or not every agent’s strategy constitutes a unique best response to the other agents’ strategies.

A strategy profile \( \pi = (\pi_1, \ldots, \pi_n) \) is a strict Nash equilibria if, for all agents \( i \) and for all strategies \( \pi'_i \neq \pi_i \), \( r_i(\pi_i, \pi_{\setminus i}) > r_i(\pi'_i, \pi_{\setminus i}) \). A strategy profile \( \pi = (\pi_1, \ldots, \pi_n) \) is a weak Nash equilibria if, for all agents \( i \) and for all strategies \( \pi'_i \neq \pi_i \), \( r_i(\pi_i, \pi_{\setminus i}) \geq r_i(\pi'_i, \pi_{\setminus i}) \), and \( \pi \) is not a strict Nash equilibrium.

Intuitively, weak Nash equilibria are less stable than strict Nash equilibria, because in the former case at least one player has a best response to the other players’ strategies that is not his equilibrium strategy. Mixed-strategy Nash equilibria are necessarily always weak, while pure-strategy Nash equilibria can be either strict or weak, depending on the game.

Moreover, the existence of Nash equilibria is granted by the Nash theorem:

**Theorem** [14] *Every game with a finite number of players and action profiles has at least one Nash equilibrium.*

### Repeated games

In repeated games, a given game (often thought of in normal form) is played multiple times by the same set of players. The game being repeated is called the *stage game*.

A stage game played a finite number of times is called a *finitely repeated game*. Typically, the agents do not know the other agent’s actions until all players have played a stage. Moreover, the payoff function of each agent is additive; that is, it is the sum of payoffs at each stage game. It is easy to notice that the strategy space of a repeated game is much richer than the strategy space in the stage game. Finitely repeated games can be represented in normal form (although grow exponentially with the number of repetitions) and thus the same solution concepts mentioned early apply.

In *infinitely repeated games*, the sum of the payoffs diverges. To avoid this, the
notion of *discounting* is introduced (see Section 1.2.1)

**Markov Games**

Intuitively speaking, a Markov game is a collection of normal-form games; the agents repeatedly play games from this collection, and the particular game played at any given iteration depends probabilistically on the previous game played and on the actions taken by all agents in that game. Markov games are a very broad framework, generalizing both Markov decision processes (MDPs; see 1.2) and repeated games. An MDP is simply a Markov game with only one player, while a repeated game is a Markov game in which there is only one stage game.

A *Markov game* (also known as *stochastic game*) is a tuple $(\mathcal{S}, N, \mathcal{A}, \mathcal{P}, \mathcal{R})$ where $\mathcal{S}$ is a finite set of games; $N$ is a finite set of $n$ players; $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$, where $\mathcal{A}_i$ is a finite set of actions available to player $i$; $\mathcal{R} = r_1, \ldots, r_n$ where $r_i$ is a real valued payoff function for player $i$; and $\mathcal{P}$ is the transition probability function, $p(s', r | s, a)$ is the probability of transitioning from state $s$ to state $s'$ and receiving reward $r$ after action profile $a$.

Equilibria in Markov games is a topic that is fraught with subtleties. Not in all games an equilibrium is granted and in the cases it is, finding it is a non-trivial problem.

**Reinforcement Learning**

Reinforcement learning (RL) is learning what to do - how to map situations to actions - so as to maximize a numerical reward signal (also utility or payoff). The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics - trial-and-error search and delayed reward - are the two most important distinguishing features of reinforcement learning [4].
In RL, there is the trade-off between *exploration* and *exploitation*. To obtain a lot of reward, a RL agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions, it has to try actions that it has not selected before. The agent must try a variety of actions and progressively favor those that appear to be best.

### Finite Markov Decision Processes

In this section we introduce more formally the problem of finite Markov decision processes, or finite MDPs previously mentioned in Section 1.1.4. MDPs are meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The learner and decision maker is called the *agent*. The thing it interacts with, comprising everything outside the agent, is called the *environment*. These interact continually, the agent selecting actions and the environment responding to these actions and presenting new situations to the agent. The environment also gives rise to rewards, special numerical values that the agent seeks to maximize over time through its choice of actions. See Figure 1.

![Figure 1: The agent-environment interaction in a Markov decision process](image)

In a *finite* MDP, the sets of states, actions, and rewards ($S$, $A$, and $R$) all have a finite number of elements. In this case, the random variables $R_t$ and $S_t$ have well defined discrete probability distributions. These distributions depend only on the immediately preceding state and action and this property is called the *Markov property*. That is, for particular values of these random variables, $s' \in S$ and $r \in R$, there is a probability of those values occurring at time $t$, given particular values of
Chapter 1. Introduction

the preceding state and action. This probability is defined as

\[ p(s', r|s, a) \triangleq Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\} \]  

(1.2)

\[ \sum_{s' \in S} \sum_{r \in R} p(s', r|s, a) = 1 \]  

(1.3)

for all \( s', s \in S \), \( r \in R \) and \( a \in A(s) \). Notice that from the point of view of a single agent, MDPs are very similar to the normal-form games introduced in Section 1.1. Nonetheless, in this case, there appears the notion of state which was not present before. This state \( s \) can be understood, in terms of game theory, as the game that is played at the moment of choosing the action. Thus, the agent has a different set of available actions and rewards depending on this state. Markov Games are a generalization of MDPs in the sense that they consider multiple agents, while MDPs only account for a single agent and its environment.

Goals and Rewards

In reinforcement learning, the purpose or goal of the agent is formalized in terms of a special signal, called the reward, passing from the environment to the agent. At each time step, the reward is a simple number, \( R_t \in \mathbb{R} \). Informally, the agent’s goal is to maximize the total amount of reward it receives. This means maximizing not immediate reward, but cumulative reward in the long run. This rewards are equivalent to the payoffs introduced in Section 1.1: after an agent chooses an action, it receives a reward that depends on its choice (and other factors).

Returns and Episodes

So far, we have said that the agent’s goal is to maximize the cumulative reward it receives in the long run. In general, we seek to maximize the expected return or expected future reward, where the return, denoted \( G_t \), is defined as some specific function of the reward sequence. In the simplest case the return is the sum of the rewards. When the task continues infinitely, this quantity can diverge. To avoid this, the concept that we need is that of discounting. According to this approach,
the agent tries to select actions so that the sum of the discounted rewards it receives over the future is maximized. In particular, it chooses $A_t$ to maximize the expected discounted return:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma G_{t+1}, \quad (1.4)$$

where $\gamma$ is a parameter, $0 \leq \gamma \leq 1$, called the discount rate. This formulation is the same used for the infinitely repeated games.

The discount rate determines the present value of future rewards: a reward received $k$ time steps in the future is worth only $\gamma^{k-1}$ times what it would be worth if it were received immediately. If $\gamma < 1$, the infinite sum in (1.4) has a finite value as long as the reward sequence $\{R_k\}$ is bounded. If $\gamma = 0$, the agent is "myopic" in being concerned only with maximizing immediate rewards: its objective in this case is to learn how to choose $A_t$ so as to maximize only $R_{t+1}$. In general, acting to maximize immediate reward can reduce access to future rewards so that the return is reduced. As $\gamma$ approaches 1, the return objective takes future rewards into account more strongly; the agent becomes more farsighted.

**Policies and Value Functions**

Almost all reinforcement learning algorithms involve estimating value functions, functions of states (or of state-action pairs) that estimate how good it is for the agent to be in a given state (or how good it is to perform a given action in a given state). The notion of "how good" here is defined in terms of future rewards that can be expected, or, to be precise, in terms of expected return. Of course the rewards the agent can expect to receive in the future depend on what actions it will take. Accordingly, value functions are defined with respect to particular ways of acting, called policies or strategies.

The value of a state $s$ under a policy $\pi$, denoted $v_\pi(s)$, is the expected return when
starting in $s$ and following $\pi$ thereafter. For MDPs, we can define $v_\pi$ formally by

$$
v_\pi(s) \triangleq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right], \text{ for all } s \in \mathcal{S}, \quad (1.5)
$$

where $\mathbb{E}_\pi[\cdot]$ denoted the expected value of a random variable given that the agent follows policy $\pi$, and $t$ is any time step. We call this function $v_\pi$ for the state-value function for policy $\pi$.

Similarly, we define the value of taking action $a$ in state $s$ under a policy $\pi$, denoted $q_\pi(s, a)$, as the expected return starting from $s$, taking action $a$, and thereafter following policy $\pi$:

$$
q_\pi(s, a) \triangleq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right] \quad (1.6)
$$

We call $q_\pi$ the action-value function for policy $\pi$.

A fundamental property of value functions used throughout reinforcement learning is that they satisfy recursive relationships similar to that which we have already established for the return (1.4). For any policy $\pi$ and any state $s$, the following consistency condition holds between the value of $s$ and the value of its possible successor states:

$$
v_\pi(s) \triangleq \mathbb{E}_\pi[G_t \mid S_t = s] = \sum_a \pi(a \mid s) \mathbb{E}_\pi\left[\sum_{s', r} p(s', r \mid s, a) \left[r + \gamma v_\pi(s')\right]\right] \quad (1.7)
$$

For each triple, we compute its probability, $\pi(a \mid s)p(s', r \mid s, a)$, weight the quantity in brackets by the probability, then sum over all possibilities to get an expected value.

Equation (1.7) is the Bellman equation for $v_\pi$. It expresses a relationship between the value of a state and the value of its successor states. It states that the value of the start state must equal the (discounted) value of the expected next state, plus the reward expected along the way.

The value function $v_\pi$ is the unique solution to its Bellman equation. This equation
forms the basis of a number of ways to compute, approximate and learn $v_\pi$.

**Optimal Policies and Optimal Value Functions**

Solving a reinforcement learning task means, roughly, finding a policy that achieves a lot of reward over the long run. Value functions define a partial ordering over policies. A policy $\pi$ is defined to be better than or equal to a policy $\pi'$ if its expected return is greater than or equal to that of $\pi'$ for all states. In other words, $\pi \geq \pi'$ if and only if $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in S$. There is always at least one policy that is better than or equal to all other policies. This is an **optimal policy**. Although there may be more than one, we denote all the optimal policies by $\pi_*$. Notice that in game theory notation, this policy is actually a Pareto optimal policy. They share the same state-value function, called the **optimal state-value function**, denoted $v_*$, and defined as

$$v_*(s) = \max_{\pi} v_\pi(s)$$  \hspace{1cm} (1.8)

for all $s \in S$.

Optimal policies also share the same **optimal action-value function**, denoted $q_*$, and defined as

$$q_*(s, a) = \max_{\pi} q_\pi(s, a)$$  \hspace{1cm} (1.9)

for all $s \in S$ and $a \in A$. For the state-action pair $(s, a)$, this function gives the expected return for taking action $a$ in state $s$ and thereafter following an optimal policy. Thus, if we have the optimal action-value function, acting optimally is straightforward:

$$\pi_*(s) = \max_{a} q_*(s, a)$$  \hspace{1cm} (1.10)

**Q-learning**

*Temporal-difference* (TD) learning methods are a variety of RL methods that can learn directly from raw experience without a model of the environment’s dynamics. TD methods update estimates based in part on other learned estimates, without waiting for a final outcome (they bootstrap).
Note that in these type of methods, a behavioral policy is always required. The simplest action selection rule, or policy, is to select one of the actions with the highest estimated value. We write this *greedy* action selection method as

\[ A_t(s) = \text{arg max}_a Q_t(s, a) \quad (1.11) \]

Greedy action selection always exploits current knowledge to maximize immediate reward; it spends no time at all sampling apparently inferior actions to see if they might really be better. A simple alternative is to behave greedily most of the time, but every once in a while, say with small probability \( \epsilon \), instead select randomly from among all the actions with equal probability, independently of the action-value estimates. We call methods using this near-greedy action selection rule \( \epsilon \)-*greedy* methods. An advantage of these methods is that, in the limit as the number of steps increase, every action will be sampled an infinite number of times, thus ensuring that all the \( Q_t(s, a) \) converge to \( q^*(s, a) \).

That is, all non-greedy actions are given the minimal probability of selection, \( \frac{\epsilon}{|A(s)|} \), and the remaining bulk of the probability, \( 1 - \epsilon + \frac{\epsilon}{|A(s)|} \), is given to the greedy action.

One of the early breakthroughs in reinforcement learning was the development of an off-policy TD control algorithm known as *Q-learning* [15]. Q-learning is defined by

\[ Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \quad (1.12) \]

In this case, the learned action-value function, \( Q \), directly approximates \( q^* \), the optimal action-value function, independent of the policy being followed. The policy still has an effect in that it determines which state-action pairs are visited and updated. However, all that is required for correct convergence is that all pairs continue to be updated. The following box presents the full Q-learning algorithm.
Q-learning (off-policy TD control) for estimating $\pi \approx \pi^*$:

Initialize $Q(s, a)$ for all $s \in S, a \in A(s)$, arbitrarily,
Repeat (for each episode):
   Initialize $S$
   Choose $A$ from $S$ using policy derived from $Q$ (e.g. $\epsilon$-greedy)
   Repeat (for each step of the episode):
      Take action $A$, observe $R, S'$
      $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
      $S \leftarrow S'$

1.3. Battle of the Sexes

A version of the Battle of the Sexes game is illustrated in Figure 2. In this version of the game (with the standard assumption of $h > l > 0$), two people live in a very remote area of the countryside. Each day, early in the morning, one of them needs to go from A to B and the other from B to A. The problem is that these two points are only connected by two different roads (one much longer than the other) where only one car fits in each path and there is no established circulation direction. Figure 3 schematizes the situation. If they pick different roads, the one in the shortest gets a high payoff of $h$ because it spends little time driving and the other gets a low payoff of $l$ because his or her trip is quite long. If both choose the same path, they bump into each other and no one gets to their respective destination. Thus, each gets a (normalized) payoff of 0.

There are both coordination and conflict elements in the Battle of the Sexes [16, 17]. While both players want to reach their destination, the conflict element is present because they both prefer the same road. The coordination element is also present because they may end up crashing if communication between them is limited. In fact, we can define the level of conflict by $\theta = \frac{h}{l}$. This game has several interesting properties. There are three Nash equilibria for the one-shot version of the game [13]:
two (unfair) pure strategies where one player always goes along the shortest road and the other always takes the long journey, and one (inefficient) mixed strategy where players randomly pick between the two paths with an optimal probability that only depends on the conflict level. In the (infinitely) repeated version of the game, there are additional equilibria that are both fair and efficient in the sense that both players earn the same reward and it is the maximum possible considering both of them together. These two equilibria require more sophisticated patterns of coordination across rounds that can only be reached by conventionalization [2].

One possible convention is an alternation equilibrium: the two players take turns going to the shortest road, so that they never run into each other and each gets the short trip equally often [18] [19] [20]. Notice that the periodicity of this alternation can be greater than 1, that is, drivers can alternate roads after each 3 days, for instance. Nonetheless, [18] showed that the one which maximizes the expected reward for both agents is the alternation at each round of the game. Moreover, experimental results showed that humans also tend to reach this particular equilibrium in similar
1.4. Objectives of this thesis

In this thesis, we take an empirical approach to analyze whether two Q-learning agents reach cooperation in the repeated version of the Battle of the Sexes game.
In particular, we aim to answer the following questions:

1. Does Q-Learning applied in its simplest form converge in this multi-agent setting?

2. Can we define a simple state representation so that Q-Learning converges?

3. If so, does it converge to the optimal fair solution? (i.e. TT)

4. How does the level of conflict affect the solution?

5. How does the interplay between step size and decay rate affect the convergence and solution of the algorithm?

6. Can we characterize the path to convergence?

7. How does asymmetry between agents affect the convergence?

8. How does a simple opponent modeling affect the learning procedure and the outcome?
Chapter 2

Methods

In this work, we want to study the experimental outcome of two learning agents playing the infinitely repeated version of the game Battle of the Sexes. Our approach is to model the repeated game as a Markov game where each player is a Q-learning agent that starts with no knowledge. That is, agents that start in a given state with a set of available actions but no knowledge of which is best.

Framework

To test whether or not cooperation can emerge between simple self-interested learning agents, we need to define such agents and the situation where they operate.

The agents

We are interested in modeling how two agents learn a policy in the game. Our choice of algorithm is Q-learning, the most popular RL algorithm. This means that the behavior of these agents critically depends on the two parameters $\alpha$ and $\gamma$. Despite the popularity of Q-learning, there are no general convergence guarantees in a game setting like the one considered here. Therefore we will empirically analyze its behavior.

Aside from that, we are also interested in explore how affects the agents learning if
they also model their opponent explicitly. That is, how they learn to behave if they also try to explicitly learn how the other player acts.

**Opponent modeling**

In the game we consider, coordination between the two agents is necessary to obtain a reward greater than 0 (see Section 1.3). Intuitively, being able to anticipate the action of the opponent should facilitate such task (as showed in [20]). That is, knowing the opponent’s next action, the agent is able to play its best response to it. Thus, we want to analyze the effect that opponent modeling has in our particular scenario.

Learning a model of the opponent while it is also learning and, thus, changing its policy, is no trivial task. Nonetheless, we propose to model the other player by simple probabilities that represent the belief the modeler have that the opponent will play a certain action.

Note that in the game we consider, each agent only has two possible actions. Hence, inferring the probability that one of the agents chooses a certain action given a series of observed rounds is analogous to the problem of inferring the probability that a coin shows up heads, given a series of observed coin tosses. One subtlety is that this is only true if the probability to be inferred is static, which might not the case in a learning agent. Nevertheless, one can consider that it will be static in the limit, when the learner converged to a given policy.

The **Beta-binomial model** is a tool that allows to solve this problem by modeling the probability distribution of a coin shows up head [21]. Thus, it allows to model the probability distribution of opponent taking action $o$ based on observed data $D$ (i.e. the observed history of opponent’s actions). The **probability distribution function** (pdf) defined by the Beta distribution is expressed in equation (2.1).

$$Pr\{p_o|D\} = \text{Beta}(p_o|N_o + 1, N_{\neg o} + 1) = p_o^{N_o}(1 - p_o)^{N_{\neg o}} \tag{2.1}$$

Where $p_o$ is the probability that the opponent picks action $o$ in the next turn, $p_{\neg o}$
corresponds to the complementary action, $N_o$ the number of times the opponent has been observed to pick action $o$ from the current state and $N_{-o}$ the same with action $-o$.

### Battle of the Sexes as a Markov Game

The repeated Battle of the Sexes is modeled as a Markov Game. This approach is able to sufficiently capture the dynamics of the game in such a way that there exist the possibility for agents to reach dominance or turn-taking equilibria. A Markov Game can be understood by a collection of games and at each round one of them is played. Different games mean different situations.

In order to clarify notation, we present the game as the subjective view of one of the agents and we denote the other one opponent and represent its actions by $o$. Since the game is symmetric it poses no real difference which agent is which.

**States** The set of games (or states) are $S = \{tie, low, high\}$ which correspond to the observed outcome of the previous round. The state high means the agent obtained the reward $h$ in the previous round, whilst low stands for the complementary outcome. This state representation is enough to capture the strategies we are interested in and keeps the number of states remarkably low. A single state (playing the game equally each round) would not capture the TT. Only DOM (or ties) would be a possible with a deterministic strategy. Having information about the previous round allows to capture both DOM and TT.

**Actions** Actions are taken simultaneously and as presented in Figure 2 the set of available actions to both agent and opponent is $A = \{S, T\}$.

**Rewards** The set of rewards $R = \{r_a, r_o\}$ are the same for each player. That is $r_i = \{0, l, h\}$ if there is a tie, they get the low payoff or they get the high payoff respectively.
**Transitions** Given a state $s$, an agent action $a$ and an opponent action $o$, the transition probabilities $p(s', r \mid s, a, o)$ are defined as:

\[
p(tie, (0, 0) \mid s, a = S, o = S) = 1 \tag{2.2}
\]
\[
p(tie, (0, 0) \mid s, a = T, o = T) = 1 \tag{2.3}
\]
\[
p(agent, (2, 1) \mid s, a = T, o = S) = 1 \tag{2.4}
\]
\[
p(opponent, (1, 2) \mid s, a = S, o = T) = 1 \tag{2.5}
\]

For any state $s$.

Note that the transitions and rewards depend only on the action profile of each round, which is consistent with the game and the previous definitions. All transitions not specified above have a probability of 0.

Figure 4 presents the transition graph of the Markov game defined here.

![Transition Graph](image)

**Measures**

In order to analyze the results obtained, we need to define measures that allow us to rapidly interpret the learned policy of both agents. This section introduces the
two kinds that we have used for that matter.

**Efficiency and Fairness**

We have introduced the two equilibria that can be reached with deterministic policies within a deterministic environment, that is, by choosing an action without randomization in a world where things do not change. These two equilibria are the dominance and the turn-taking. In order to identify and characterize them we need the efficiency and fairness measures introduced by Hawkins and Goldstone [2].

Efficiency is defined as the sum $\rho_1 + \rho_2$, where $\rho_i$ is player $i$’s total payoff over a number of rounds of playing the stage game. It quantifies the total amount of money the players were collectively able to earn. We divide by the total amount it was possible to earn in order to normalize all efficiency scores to the $[0, 1]$ interval and compare across conditions with different payoff structures. If a pair of players achieves the maximum efficiency of 1, they are *optimally efficient*. Note that both dominance and turn taking strategies are optimally efficient.

To distinguish among these different outcomes, we need the measure of fairness, defined as the normalized payoff ratio

$$\text{Fairness} = \frac{\min \rho'_1, \rho'_2}{\max \rho'_1, \rho'_2}$$

where $\rho'_i$ is the number of rounds that player $i$ earned the higher payoff. This normalization maps the fairness of all conditions to the same $[0, 1]$ interval. If one player gets the higher pay-off every round, this measure of fairness will be zero; if the players finish the experiment with an equal number of times of earning the high payoff, it will be one. Both measures can be defined over a number of sequentially played $N$ rounds of the stage game. Note that $N$ must be even for the fairness measure to be well defined.
Binary variables

A more simplistic approach is to consider simply binary variables corresponding to each interesting policy.

The value of these variables is 1 if the learned policy is the one corresponding to the variable and 0 otherwise. In particular, we have used (abusing notation) the \( TT \) and \( DOM \) for TT and DOM.

In order to compute them, we take a window of \( W \) rounds after the learning phase and observe the policy adopted by the agents.

Notice that when \( TT = 1 \), \textit{Fairness} = 1 and \textit{Efficiency} = 1. In the case of \( DOM = 1 \), \textit{Efficiency} = 1 but \textit{Fairness} = 0. It is straightforward to see that \( TT \) and \( DOM \) are excluding.

Fraction of turn taking

Since each instantiation of the experimental procedure does present a considerable amount of randomness, we need to repeat it many times in order to obtain significant results.

We define the fraction of turn taking (FTT) in order to rapidly interpret whether or not a parameter set leads to the TT strategy. It is simply defined as the fraction of trials the variable \( TT \) is 1 in the final window \( W \).

\[
FTT = \frac{\# TT = 1}{\# trials} \quad (2.7)
\]

Experimental approach

In order to test whether or not reinforcement learning agents playing the game can reach the equilibrium strategies, we designed one single experimental procedure.

This experiment consists on making two independent Q-learning agents with no initial knowledge play the game a number of rounds and observe what is the final
strategy that they adopted. Each time step \( t \) is a round where both players observe state transitions and rewards. There is no defined final state, the experiment is just stopped at the end time \( T_{\text{max}} \). Then, the final strategy is assessed by looking the outcomes of the final window \( W \) of 100 rounds. At the end, we compute the \( TT \) and \( DOM \) using this \( W \).

These agents are independent Q-learners. Despite they can observe the outcome of each round and infer the actions of the opponent, in the basic version, they do not model the other player in any explicit way. Thus, the \( Q \) of the agent can be expressed as \( Q(s,a) \). Both learning methods have two parameters: the step-size \( \alpha \) and the discount factor \( \gamma \). In order to test the robustness of the outcomes, we allow them to have different \( \alpha \) and \( \gamma \) parameter values.

The actions taken by each agent follow a \textit{decayed} \( \epsilon \)-greedy policy. It is an \( \epsilon \)-greedy policy on \( Q(s,a) \) which \( \epsilon \) parameter decays linearly in the first \( d \) rounds (decay time) from \( \epsilon_0 = 0.99 \) to 0. This policy makes agents be essentially exploratory at the beginning and evolve to a deterministic behavior based on their learned knowledge.

By construction, \( T_{\text{max}} \) has to fulfill that \( T_{\text{max}} \geq d + W = d + 100 \). One might consider taking \( T_{\text{max}} \gg d + W \) in order to ensure convergence of the \( Q(s,a) \) values.

The following box presents the algorithm of the experiment.
Experimental approach with independent Q-learners

$t \leftarrow 0$

Repeat (for each agent $i$):

- Initialize $Q_i(s, a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily
- Initialize $S_i$

Repeat (for each episode):

- $\epsilon \leftarrow \max \left((1 - \frac{t}{T_d})\epsilon_0, 0\right)$

Repeat (for each agent $i$):

- Choose $A_i$ from $S_i$ using policy derived from $Q_i$ ($\epsilon$-greedy)

Repeat (for each agent $i$):

- Observe $R_i, S'_i$

\[
Q_i(S_i, A_i) \leftarrow Q_i(S_i, A_i) + \alpha_i[R_i + \gamma_i \max_a Q_i(S'_i, a) - Q_i(S_i, A_i)]
\]

$t \leftarrow t + 1$

until $t > T_{max}$

Obtain $TT$ and $DOM$ from the final window $W$ outcomes

Opponent modeling

In the framework described so far, agents consider the opponent as part of the environment. We also wanted to see which effect does an explicit modeling of the opponent has in the outcome of the game.

To do so, we implement into the agents a new dimension to their action-value function $Q$ such that it becomes $Q(s, a, o)$. Note that in order to be able to follow a greedy policy in this $Q$, agent should know which action $o$ takes the rival. Since actions of both agents are chosen simultaneously, this is not possible in practice.

Despite there are many ways to approach this problem, we opted for the simplest approach: estimate the opponents action before choosing how to proceed. If the agent has a good model of the opponent, its estimations should suffice. The intuitive idea of our implementation is to observe every action taken by the opponent from each given state and use them to deduce the probability that the other player chooses
2.2. Experimental approach

a given action from a given state using the described Beta-binomial model. Then estimate the rival actions using these probabilities. Formally, this estimation is performed as follows:

1. A history of \( N_o(s) \) and \( N_{\neg o} \) is stored by each agent. It corresponds to the number of observed times the opponent has picked \( o \) or \( \neg o \) from a given state \( s \).

2. Every time an opponent’s action estimation has to be done, a probability \( p_o \) is drawn from the Beta-binomial model using \( N_o(s) \) and \( N_{\neg o}(s) \) in 2.1.

3. A random number \( z \) is obtained from a uniform distribution between \((0, 1)\). If \( z < p_o \), \( \hat{o} = o \). Otherwise, \( \hat{o} = \neg o \).

Finally, the experimental procedure implemented to study the opponent modeling case is the following:
Chapter 2. Methods

Experimental approach with opponent modeling

\( t \leftarrow 0 \)

Repeat (for each agent \( i \)):

- Initialize \( Q_i(s, a) \) for all \( s \in \mathcal{S}, a \in \mathcal{A}(s) \), arbitrarily
- Initialize \( N_o(s) = N_{-o}(s) = 0 \) for all \( s \in \mathcal{S} \)
- Initialize \( S_i \)
- Estimate \( \hat{O}_i \)

Repeat (for each episode):

\[ \epsilon \leftarrow \max\left((1 - \frac{t}{T})\epsilon_0, 0\right) \]

Repeat (for each agent \( i \)):

Choose \( A_i \) from \( S_i, \hat{O}_i \) using policy derived from \( Q_i \) (\( \epsilon \)-greedy)

Repeat (for each agent \( i \)):

- Observe \( R_i, S_i', O_i \)
- \( N_{O_i}(S_i) \leftarrow N_{O_i}(S_i) + 1 \)
- Estimate \( \hat{O}_{i+1} \)
- \( Q_i(S_i, A_i, O_i) \leftarrow Q_i(S_i, A_i, O_i) + \]
- \[ +\alpha_i[R_i + \gamma_i \max_a Q_i(S_i', a, \hat{O}_{i+1}) - Q_i(S_i, A_i, O_i)] \]

\( t \leftarrow t + 1 \)

until \( t > T_{\text{max}} \)

Obtain \( TT \) and \( DOM \) from the final window \( W \) outcomes
Chapter 3

Results

In this chapter, we present the results obtained in the previously described setting. First, we focus on equilibria, then we analyze the evolution of the policies and finally we test the effects of parameter asymmetry for the basic setting. At the end, we consider the opponent modeling agents.

Different settings have been tested in order to evaluate the emergence of cooperative behavior. Each one is a unique combination of different factors:

- Two sets of rewards: One corresponding to a high conflict case with \((l, h) = (1, 4), \theta = 4\); and another one corresponding to a low conflict case with \((l, h) = (1, 2), \theta = 2\). The results of [2] suggest that the difficulty of reaching TT depends on \(\theta\).

- Seven step-sizes \(\alpha \in \{0.1 : 0.1 : 0.7\}\).

- Nineteen discount factors \(\gamma \in \{0.9 : 0.01 : 0.99\} \cup \{0.991 : 0.001 : 0.999\}\).

- Thirteen \(\epsilon\) decay times \(d \in \{5e3 : 5e3 : 25e3\} \cup \{3e4 : 1e4 : 10e4\}\). Higher values were discarded due to computational limitations.

- The number of trials for each experiment is set to 100.
Low conflict scenario

Here we present the results of Q-learners playing the repeated Battle of the Sexes in a low conflict scenario, that is, with rewards \((l, h) = (1, 2)\). Figure 5 presents the \(F\!T\!T(\alpha, \gamma)\). This measure corresponds to the observed \(F\!T\!T\) for each combination of \(\alpha\) and \(\gamma\) with \(d = 1e5\). Figures 6 and 7 show the \(F\!T\!T(\alpha, \gamma, d)\). In this case, the measure represents \(F\!T\!T\) for each combination of \(\alpha\), \(\gamma\) and \(d\), which allows to interpret the effect each parameter has in the emergence of TT.

![Figures 5, 6, and 7 showing \(F\!T\!T(\alpha, \gamma)\) and \(F\!T\!T(\alpha, \gamma, d)\).

From these results we can observe that the tendency of the system is to converge to the TT equilibrium. That is, given the proper combination of parameters, learning the TT strategy is assured. Nevertheless, this is only possible if the decay time \(d\) is sufficiently long. For smaller values of \(d\), the progression from exploration to exploitation is too fast and the agents are not able to converge to a robust policy. The learning is truncated and the resulting policy is not necessarily TT. Interestingly, they always end up either with DOM or TT (at least for the decay times considered). That is, there are no ties in the final window in any case.

Despite the co-existence of two competing learning agents, only two robust equilibria are reached, DOM and TT. Both equilibria correspond to \(E\!f\!f\!i\!c\!i\!e\!n\!c\!y = 1\) so the two agents as a whole always maximize the reward they get from the system. The ties
3.1. Low conflict scenario

Figure 6: $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.9 : 0.01 : 0.99\}$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination.
Figure 7: $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.991 : 0.001 : 0.999\}$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination.
penalize both agents and that is why they always learn to avoid them. Nonetheless, DOM corresponds to one agent getting much more reward than the other and as long the learning procedure is smooth enough, it does not happen. Agents tend to equilibrate because both of them try to maximize their own reward. TT is the policy in which both agents maximize their reward equally.

We also observe that the combination of the step-size $\alpha$ and the discount rate $\gamma$ is crucial to reach the TT equilibria. Smaller step-sizes favour the convergence but lead to longer decay times needed whilst the opposite effect is found in $\gamma$. Bigger discount factors increase the FTT but also increase the required decay time.

The high contrast between the FTT obtained with $\alpha = 0.1$ and $\gamma = 0.999$ with respect to the value for $\alpha = 0.2$, is probably caused by the fact that a decay time $d = 1e5$ is not enough for the agents to learn the cooperative solution.

In this section we focus on the different equilibria reached by the agents. In the next section, we analyze how the learned policies (Q-values) evolve during the learning process. For that, we consider the same setting as in this section as it proves to be robust.

**Policy evolution**

Another illustrative result is how the policies of the agents evolve over time. This can be easily plotted by means of the $Q(s,a)$ values of each agent at each time step. Since the target policy is the greedy policy and there are only two actions, the biggest $Q(s,a)$ value from each state $s$ corresponds to the action that the agent will take in that given state. Thus, if $Q(s,T) - Q(s,S) > 0$ the agent will pick $T$ in $s$ and $S$ if $Q(s,T) - Q(s,S) < 0$.

Figures 8 and 9 show the characteristic evolution of such a difference in the $Q(s,a)$ values for both agents. We analyze different parameter combinations. The former corresponds to a setting in which DOM is sometimes learned and the latter corresponds to a setting that always leads to TT.
Chapter 3. Results

Figure 8: Policy evolution of both agents in a low conflict scenario. \(\alpha = 0.4\), \(\gamma = 0.999\) and \(d = 2e4\). \(Q(s, T) - Q(s, S) > 0\) means that \(\pi(s) = T\) and \(Q(s, T) - Q(s, S) < 0\) corresponds to \(\pi(s) = S\). In this figure agents learn to dominate (left) and to be dominated (right).

Figure 9: Policy evolution of both agents in a low conflict scenario. \(\alpha = 0.4\), \(\gamma = 0.999\) and \(d = 1e5\). \(Q(s, T) - Q(s, S) > 0\) means that \(\pi(s) = T\) and \(Q(s, T) - Q(s, S) < 0\) corresponds to \(\pi(s) = S\). In this figure both agents learn the TT equilibrium.

Notice that, in both situations, \(Q(s, a)\) values oscillate considerably. This is the expected behavior since agents are mainly exploring and sometimes they get the reward of the action and sometimes they get 0. At the beginning this oscillation derives into continually changing policies. Progressively some tendencies appear and from some point to the end, oscillations of the differences keep confined within the positive or negative zone. Hence, at some point, the policies converge and do not change anymore (although the \(Q(s, a)\) values keep changing). Notice that when agents do converge to the TT strategy, they also learn how to properly solve ties.
Each of them picks a different action after a tie.

Additionally, we also want to evaluate how the fairness and efficiency of the system evolves over time when agents progressively learn the TT strategy. Figure 10 presents such evolution for the same parameter set as Figure 9.

Figure 10: Fairness and efficiency evolution of both agents in a low conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 1e5$. In this figure both agents learn the TT equilibrium. Results averaged over 100 trials. Each point corresponds to the measures computed in a window of 1000 time steps.

At the beginning the fairness is high but the efficiency is relatively low. This is because agents are mainly exploratory and choose actions at random. Thus, the probability of winning one, the other or a tie are similar. As they learn, the fairness reduces and efficiency starts raising. Thus, there appear some dominance periods but ties start fading off. At some point, there is a transition and this tendency changes dramatically. Fairness starts rising until it reaches the value of 1 at the end, corresponding to a learned deterministic TT. This transition matches the same transition observed in the evolution of the policies.

**Parameter asymmetry between learners**

All previous settings consider two agents that are exactly alike in terms of learning rate and discount factor. We question whether or not this plays an important role in learning the TT strategy. In order to study this phenomenon, we performed two experiments. On one hand, we test the effect of different step sizes by fixing one agent to $\alpha_2 = 0.2$ and varying $\alpha_1$ of the other agent and $\gamma$ and $d$ for both. Figure 11 presents the obtained results.

We clearly see that this particular asymmetry in the step-size favors the learning of
Chapter 3. Results

Figure 11: $FTT(\alpha, \gamma)$ with $d = 1e5$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination. $\alpha_2 = 0.2$

the TT. That is, if $\alpha_2$ is fixed to 0.2, bigger values of $\alpha_1$ than in the symmetric case lead to the TT equilibrium.

On the other hand, we explore discount rate asymmetry by confronting agents with $\alpha = 0.2$ and combinations of $\gamma \in \{0.9, 0.99, 0.999\}$. Results in the symmetric setting show that these values always converge to TT as long as both agents share the same parameter set. With this experiment we expect to see if this convergence is lost when the agents have very different discount rates.

As we discussed earlier, the discount rate can be interpreted as how farsighted is the agent. Then, it might be the case that if the agents have different $\gamma$, ties are less penalizing for the one with the greater value because it looks more far into the future. Thus, it is willing to sacrifice more rounds in order to obtain a better outcome in the end. This may end up in a DOM strategy because it can learn to dominate after the other player surrenders to domination before. Figure 12 and 13 correspond to the results of this experiment.

In this case, the convergence to the TT is maintained for small asymmetry in the discount factor. Nevertheless, for bigger differences, the robust convergence is lost. In this case, the learning of the TT strategy is no longer assured. Moreover, when TT is not reached, the agent with the highest $\gamma$ dominates its opponent.
3.2. High conflict scenario

In this section, we present the results of Q-learners playing the repeated Battle of the Sexes in a high conflict scenario, that is, with rewards \((l, h) = (1, 4)\). Figure 14 presents the \(FTT(\alpha, \gamma)\). Figures of \(FTT(d)\) can be found in Appendix B.

Here we can observe the same tendency and behavior as the one discussed in the low conflict scenario. Nonetheless, in this case the TT solution is less robust, in the sense that it requires smaller values of the step size and greater values of the discount rate in order to be obtained. Additionally, the system requires higher decay times to converge to the TT equilibrium.
Figure 14: $FTT(\alpha, \gamma)$ with $d = 1e5$ of Q-learning independent agents in a high conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination.

Again, the high contrast in the $FTT(\alpha, \gamma)$ observed between $\alpha = 0.1$ and $\alpha = 0.2$ is because $d = 1e5$ is not sufficient for the agents to properly learn the cooperative solution (although with greater decay times they would).

**Policy evaluation**

In this setting we can also evaluate how the policies of the agents evolve over time. Repeating the same procedure, we can represent this evolution with the same parameters as Figure 9. Figure 15 presents the results. Additionally, we can also study how the fairness and efficiency evolve in the same particular setting. The results are presented in Figure 16.

The behavior observed in the temporal evolution of the learned policies by the agents in the high conflict scenario is considerably similar to the one in the low conflict case. However, we detect some differences. Notice that, in this case, the oscillations in the $Q(s,a)$ differences are higher than in the lower conflict case and the transition to the convergence of the policy is more abrupt. Aligned with that, we also observe that the behavioral change in the fairness is also more abrupt than in the case of lower conflict.
3.2. High conflict scenario

Figure 15: Policy evolution of both agents in a high conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 1e5$. $Q(s, T) - Q(s, S) > 0$ means that $\pi(s) = T$ and $Q(s, T) - Q(s, S) < 0$ corresponds to $\pi(s) = S$. In this figure both agents learn the TT equilibrium.

Figure 16: Fairness and efficiency evolution of both agents in a high conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 1e5$. In this figure both agents learn the TT equilibrium. Results averaged over 100 trials. Each point corresponds to the measures computed in a window of 1000 time steps.

Opponent modeling

As a final question, we analyze what effect would have to introduce the explicit modeling of the opponent into the system. Intuitively, one might think that this should guarantee the convergence even with lower decay times. It should be easier to coordinate with an agent that you understand and you can anticipate. In the opposite direction, however, is the infinite recursivity that appears when two agents are modeling the other. The actions of the first affect the actions of the second, which affect back the choices of the first and so on. Thus, it is not clear if introducing this feature will favor the emergence of TT or will worse it off.

Firstly, we introduced the opponent modeling only in one of the agents. That way,
the recursivity is avoided. Figure 17 corresponds to a representative policy evolution with the agent without the improvement and Figure 18 shows the policy evolution and the model evolution of the opposite agent which explicitly considers the first one.

Figure 17: Representative policy evolution of a basic agent in a low conflict scenario confronted to a modeling agent. \( \alpha = 0.2, \gamma = 0.999 \) and \( d = 1 \times 10^5 \). \( Q(s, T) - Q(s, S) > 0 \) means that \( \pi(s) = T \) and \( Q(s, T) - Q(s, S) > 0 \) corresponds to \( \pi(s) = S \). In this figure, the agent learns to dominate.

Figure 18: Policy and model evolution of a modeling agent in a low conflict scenario confronted to a basic agent. \( \alpha = 0.2, \gamma = 0.999 \) and \( d = 1 \times 10^5 \). \( Q(s, T) - Q(s, S) > 0 \) means that \( \pi(s) = T \) and \( Q(s, T) - Q(s, S) > 0 \) corresponds to \( \pi(s) = S \). Policies separated by the estimation of the opponent’s action. In this figure, the agent learns to simply avoid ties. The right figure represents the mean of the Beta distribution (the variance is almost 0 after less than 1% of the time steps, thus the mean and the drawn samples coincide).

What we can observe in these results is that when there is only one modeling agent, it rapidly learns to avoid ties. After very few iterations, the policy of the modeling agent is to choose the complementary action of the estimated next opponent’s action, independently of the state. Since the basic agent does not account for the
3.2. High conflict scenario

opponent’s actions, it always prefers the biggest reward. Therefore, it learns to dominate because the modeling agent avoids ties and offers free choice to the basic agent without any kind of punishment. Moreover, we observe that the modeling agent properly learns that the opponent will dominate. With more iterations, the probabilities of the opponent choosing $S$ would simply decrease to 0. At the end, the only probability decreasing is the one corresponding to the low state because is the only one visited. As a final remark, the learned policy at the end contains mainly domination but ties are also present. Although the model is quite correct, the modeling agent sometimes fails to properly anticipate the adversary with the method implemented, thus ending up in ties.

Finally, we confronted two modeling agents. Since they are symmetric, we only present the policy and model evolution of one of the agents in Figure 19. This is representative enough.

Figure 19: Policy and model evolution of a modeling agent in a low conflict scenario confronted with another modeling agent with $\alpha = 0.2$, $\gamma = 0.999$ and $d = 1e5$. $Q(s, T) - Q(s, S) > 0$ means that $\pi(s) = T$ and $Q(s, T) - Q(s, S) < 0$ corresponds to $\pi(s) = S$. Policies separated by the estimation of the opponent’s action. In this figure, the agent learns to simply avoid ties. The right figure represents the mean of the beta distribution (the variance is almost 0 after less than 1% of the time steps, thus the mean and the drawn samples coincide).

Interestingly, both agents rapidly learn that ties are the worse outcome. Thus, given that they could exactly anticipate the opponent’s action, they will always pick the complementary. This pattern can only lead to TT as long as one of the agents already plays this way (and the predictions are exact). If the other agent learns to dominate, these modeling agents just learn to be dominated (as we have seen in the
previous setting).

As we suspected, confronting two modeling agents lead to an infinite recursivity that is not properly properly. This is clearly exposed by the right plot of Figure 19. Each agent starts with a model of the opponent corresponding to a random behavior (which is correct), that is, the probability of the adversary choosing either action is 0.5. At each time step, the agents draw a sample of the distribution and based on this random value, they estimate the opponent’s next choice. Since they rapidly learn to avoid ties, the agents simply pick the complementary action of the estimated one. Since this estimation is totally random, their behavior is too. Then, when they try to model each other, they can only see random behavior and this randomness is perpetuated to the end.

**Summary**

In this section, we summarize the main ideas that can be interpreted from the results presented in this chapter:

- Basic independent Q-learning agents following a decaying $\epsilon$-greedy policy can learn to take turns in the repeated Battle of the Sexes game. This is observed in the case where the game is modeled as a Markov Game and the states correspond to the outcome of the previous round. Q-learning applied to the naive representation (i.e. single state) does not converge to the TT because it simply can not capture the TT. Moreover, if convergence to a stable solution is possible, TT seems to be the natural equilibrium to which the system converges.

- The convergence of the system to the fair cooperative solution depends on four factors: the step size $\alpha$, the discount rate $\gamma$, the decay time $d$ and the conflict level of the game $\theta$. There is always a minimum $d$ required which depends on the other three factors for the TT convergence to be reached. After this minimum, any $d$ leads to the same solution. Smaller values of $\alpha$ and greater values of $\gamma$ favor this convergence but increase the minimum $d$ required.
3.3. Summary

Smaller conflict levels $\theta$ also favor the appearance of this equilibrium. For the combinations of relatively small $\gamma$ and relatively high $\alpha$, the convergence cannot be reached, independently of $d$. The set of parameter combinations for which the agents do not learn the TT increases with the conflict $\theta$.

- The learning procedure of the agents when they reach the TT is characterized by a transition. The efficiency of the system monotonously increases but the fairness initially decreases and after some particular time step it augments until it reaches the maximum value at time $d$. This transition is also observed in the policy evolution: the learned policy changes during the first part but after the same particular time as the fairness, it remains constant to the end. Moreover, the higher the conflict level $\theta$, the abrupter this transition is.

- The exact symmetry between agents is not required for them to always learn to take turns. Symmetry in the $\alpha$ parameters does not impair the learning to the turn-taking as long as one of the agents have a sufficiently small step size value. Moreover, if one of the $\alpha_i$ is small enough, it can compensate a bigger step size of the opponent and also reach the TT. If the $\gamma$ asymmetry is important enough, it erases the robust convergence to the fair cooperative solution. In these cases, the agent with higher discount rate sometimes learns to dominate the adversary (which learns to be dominated).

- Opponent modeling as a Beta-binomial model seems to properly model the opponent but agents fail to take advantage of this information with our implementation. If a modeling agent is matched with a basic player, it learns to avoid ties which derives in being dominated. In the case where both players are modeling agents, they learn to avoid ties. Despite that, they enter a recursive state of random actions due to the predicting mechanism. Thus, there appear rounds of DOM, rounds of TT and ties during the testing window at the end of the episode.
Chapter 4

Discussion

In this work, we have tested whether or not selfish independent Q-learners can learn to cooperate playing the repeated version of the Battle of the Sexes game. We modeled the framework as a Markov Game where each state is only characterized by the outcome of the previous round. In this particular setting, two agent types have been introduced to be the players: basic learners that do not model the opponent explicitly and modeling agents that do take the adversary into account using a Beta-binomial distribution model. Both kinds implement the Q-learning algorithm and follow an $\epsilon$-greedy decaying policy.

In our approach, the policies learned by the agents are deterministic at the end of the learning process (when their strategy is tested). They choose one of the two possible actions given the outcome of the previous round. The only possible equilibria in this setting are the TT and the DOM (mixed strategies cannot appear). Although it could be the case that agents learn policies that lead to some ties, the observed strategies at the end of an episode are always either TT or DOM (i.e. equilibrium strategies) independently of the parameters used (provided that $d \geq 5000$). Moreover, our main result is that TT is the only solution to which the agents converge. That is, it is the only equilibrium that is reached with probability 1 given some of our considered combinations of parameters. DOM appears but it never does with probability 1.
The parameters of the learning algorithm, the policy and the game play a determining role in the emergence of the TT. The Q-learning in its simplest form only depends on two parameters: the step size $\alpha$ and the discount rate $\gamma$. The policy of both agents depend on the decay time $d$ which corresponds to the time it is required for an agent to evolve from acting randomly to choose actions deterministically. Finally, the main feature of the game is the conflict level $\theta$ which measures how better is the best reward.

For each combination of $\alpha$, $\gamma$ and $\theta$, there is always a minimum value of $d$ for the agents to converge to the TT. Values greater than this minimum always lead to the same solution. Smaller values of $\alpha$ and greater values of $\gamma$ favor the TT but increase the minimum $d$ required. For the combination of relatively small $\gamma$ and relatively high $\alpha$, the convergence cannot be reached independently of $d$. The step size determines the learning speed and, therefore, smaller values require longer learning times. For Q-learning to be able to converge to an optimal solution, $\alpha$ has to be sufficiently small [4]. Thus, greater values of it prevent the convergence to the stable solution. The discount rate $\gamma$ can be interpreted as how far-sighted is the agent. Then, smaller values of $\gamma$ make immediate rewards more important. Hence, ties are more penalizing as $\gamma$ decreases. This explains how the discount rate affects the convergence. For smaller values of $\gamma$, the agent wants to maximize rewards that are closer in time. Thus, it learns in a smaller time-scale, speeding up the convergence. Nonetheless, being too myopic may limit the possibility for the agent to properly learn the TT.

The conflict level $\theta$ of the game also affects the convergence of the TT. Higher values difficult the learning of the strategy. This can be understood by the fact that the higher the $\theta$, the more penalizing are ties with action $T$, and choosing $S$ derives into a higher sacrifice. Then, agents require higher $d$ and $\gamma$ and smaller $\alpha$ in order to robustly learn the TT.

We also analyzed which consequences have the presence of different types of asymmetries in the agents. In particular, we tested $\alpha$ and $\gamma$ asymmetry. The former do not affect negatively the convergence to the TT. In fact, given that one of the agents $i$
has a relatively small $\alpha_i$, the system can converge to TT for a wider range of parameter values. The latter affects negatively the robust learning of TT if the asymmetry is considerable. In these cases, what happens is that the agent with the greater $\gamma$ sometimes learns to dominate the opponent. This is because that agent is more far-sighted. Thus, one agent feels more penalized by ties and conforms with the low reward before the TT can be learned by both players.

The learning procedure presents interesting results. Essentially, two main phases can be observed in each episode: the exploration phase and the stable phase, separated by a transition the smoothness of which decreases with the conflict level of the game. The exploration is characterized by rather high values of $\epsilon$, resulting in a mainly random behavior. During this stage, the policy of the agents changes continually due to the oscillations of the $Q(s,a)$ values. These oscillations arise because sometimes the actions do not return a reward (ties) and sometimes they do. At some point, although the oscillations continue, there is a transition in which the policy becomes stable. After this point, the TT is already learned but $\epsilon$ is still greater than 0. Thus, the fairness and efficiency are not 1 until $t = d$. The fact that the efficiency monotonously increases is because the goal of Q-learning is precisely to maximize the receive reward. Hence, as the agents learn and use their knowledge, the received reward is greater than the one obtained following a random policy.

The behavior we observe when agents learn the TT in our experiments is similar to the turn taking with iterated randomizations (TTIR) proposed in [18] for two agents to learn and use TT. This TTIR strategy allows players to engage in intertemporal sharing of the gain from cooperation. The TTIR strategy consists in randomizing actions until an outcome that do not correspond to a tie is obtained. After that, the TT is adopted. The similarity with our results is that agents act essentially randomly during a time period and, after that, they adopt the TT strategy. An important difference between both approaches is that, while the TTIR is a predefined strategy, in our case, the TT is the equilibrium resulting from two reinforcement learning agents.

The robust convergence towards the TT strategy, while surprising, is in agreement
with previous results on the literature. In [3], Axelrod proved that cooperation can emerge from selfish individuals in scenarios where conflict is present and that cooperative strategies can overcome more egoistic ones. He worked with the Prisoner’s dilemma game. Nonetheless, the results can be extrapolated since, in his work, conflict is also present and the lack of cooperation was also highly penalizing. Additionally, Browning and Colman, using genetic algorithms, showed that generations of players evolve to a form of asymmetric reciprocity through coordination strategic alternation [22]. While they considered a slightly different version of the game, this form of cooperation is, essentially, the TT strategy. The only difference in the game is that, in their work, both agents are penalized if they both choose $S$ (if the payoff matrix is normalized).

Moreover, experiments performed with humans in similar situations also presented considerable amounts of learned TT [19]. In [19], they tested people in a version of the Prisoner’s dilemma where there is also a trade-off between cooperation and conflict and TT is the optimal fair solution. In most of the games, people learn to take turns after $\sim 150$ iterations of the game. The experiments in [2], which comprised a smaller (50) number of rounds per episode, showed less prevalence of TT. The fact that some rounds are required in order to adopt the cooperation strategy is because this type of coordination requires communication [23] and these preliminary rounds are a sufficient form of communication [24].

Nevertheless, the required iterations of our game for the agents to learn the TT ($\sim 60000$) is much greater that the one observed in humans ($\sim 150$ in [19]). In [19], they suggest that intelligence of the players has an important role in how fast the TT appears. Hence, it is likely that the simplicity of our agents explains the remarkably slow convergence observed. Probably agents considering more factors and/or with more complex decision making procedures would converge faster.

The most important missing feature in our agents that we could think of is the lack of opponent consideration. That is, our basic agents consider the opponent as part of the environment and they cannot react or anticipate to its actions as humans would. Moreover, the actions of the opponent are only accessible indirectly in the
state through the result of the previous round. For instance, punishment to greedy adversaries, which is required for the emergence of the cooperative behavior [19], cannot be actively done if the opponent and its actions are not explicitly considered.

Therefore, we introduced the modeling of the opponent into the system. To do so, agents keep and update a model of the opponent in the form of a Beta binomial distribution based on the opponent’s action history. The agents learned to avoid ties considerably fast (∼20000 iterations) but failed to converge to the TT. In the case where only one of the players is a modeling agent, it rapidly learns to avoid ties. Then, since the adversary does not account for the actions of the modeling agent, it has a higher preference for the higher reward. Progressively, the modeling agent understands that the most probable behavior of the basic agent is to choose T. Thus, it ends choosing S most of the time. That is, it ends up learning to be dominated. Parameters variations in this setting always lead to the same result; they only affect the required time scale to reach the final policy without ties.

When both agents maintain a distribution of the actions of the other, although they also learn to avoid ties, they enter an endless recursivity of randomness. They start modeling the opponent as random (0.5 probability of each action). Then, each time they have to choose an action, they make an estimation based on this probability. Since both actions have the same probability, the estimation is also random. Finally, as they rapidly learn to avoid ties, agents pick the complementary behavior of the estimation. In turn, this means that their behavior is random too and what the other agent models is a random agent. Therefore, despite they have learned to avoid ties, they act totally random which is what the adversary models, leading to the endless recursivity of randomness.

In conclusion, the Beta-binomial model succeed at modeling the opponent but the way our agents exploit this knowledge makes them essentially reactive. They rapidly learn that the worst outcome is when both players choose the same action, thus to avoid ties. Nevertheless, at the time of action picking, modeling agents do not have initiative. They act reacting to what they estimate the opponent will do. In the case where only one agent models the adversary, the basic agent learns to dominate
because the modeling agent rapidly learns to adopt the complementary behavior. This did not happen with two basic agents because both preferred the higher reward and none of them allowed the other to dominate. When both agents are modeling agents, none takes any initiative so they keep acting randomly forever.

Finally, we obtained an unexpected result with our setting: we found a way of finding an experimental solution to a Markov Game using RL (the TT). As we considered it, our Markov Game is a *separable reward state independent transition* (SR-SIT) game. They are characterized by the fact that the reward can be split into the reward obtained due to the action profile and due to the state (ours only depend on the state) and that the transitions to a new state only depends on the action profile, not the previous state (as in our case) [13]. In these games, analytical solutions can be found by means of what is called *linear programming* but the formulation is not straightforward and depends on the particular game [25].

We believe that we were able to find a solution to this game using Q-learning (which only has the convergence guaranteed in static environments [15]) because the combination of Q-learning and the decaying $\epsilon$-greedy policy slowly shifts the problem from a Markov Game to a MDP. That is, agents become more and more deterministic with time and this makes the environment more static at each time step. Moreover, this might explain the required minimum decay time $d$: the shifting to stationarity must be slow enough for the Q-learning to properly converge to a solution, in this case, the TT strategy.

In summary, we experimentally proved that independent Q-learning agents following a decaying $\epsilon$-greedy policy learn to cooperate in the repeated version of the the Battle of the Sexes game as long as the parameter set allows a sufficiently smooth convergence. That is, the fair cooperative solution emerges naturally from purely selfish learning agents. In the end, we showed that, in this setting, cooperation is the rule, not the exception.
List of Figures

1. The agent-environment interaction in a Markov decision process [4].

2. Payoff matrix of the game Battle of the Sexes. Action $T$ (tough) corresponds to the shortest road and action $S$ (soft) to the longest road. $(h, l) = (2, 1)$.

3. Illustrative representation of a situation that can explain the payoff matrix of the Battle of the Sexes game. Points A and B are connected by two different roads, one shorter than the other, with no defined circulation direction.

4. Transition graph the Battle of the Sexes repeated game modeled as a Markov Game. Circles represent states and arrows represent transitions, labeled by the action profile and the rewards associated to them.

5. $FTT(\alpha, \gamma)$ with $d = 1e5$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination.

6. $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.9 : 0.01 : 0.99\}$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination.

7. $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.991 : 0.001 : 0.999\}$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Policy evolution of both agents in a low conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 2e4$. $Q(s,T) - Q(s,S) &gt; 0$ means that $\pi(s) = T$ and $Q(s,T) - Q(s,S) &lt; 0$ corresponds to $\pi(s) = S$. In this figure agents learn to dominate (left) and to be dominated (right).</td>
</tr>
<tr>
<td>9</td>
<td>Policy evolution of both agents in a low conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 1e5$. $Q(s,T) - Q(s,S) &gt; 0$ means that $\pi(s) = T$ and $Q(s,T) - Q(s,S) &lt; 0$ corresponds to $\pi(s) = S$. In this figure both agents learn the TT equilibrium.</td>
</tr>
<tr>
<td>10</td>
<td>Fairness and efficiency evolution of both agents in a low conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 1e5$. In this figure both agents learn the TT equilibrium. Results averaged over 100 trials. Each point corresponds to the measures computed in a window of 1000 time steps.</td>
</tr>
<tr>
<td>11</td>
<td>$FTT(\alpha, \gamma)$ with $d = 1e5$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination. $\alpha_2 = 0.2$.</td>
</tr>
<tr>
<td>12</td>
<td>$FTT(\gamma)$ with $d = 1e5$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination. $\alpha = 0.2$.</td>
</tr>
<tr>
<td>13</td>
<td>$FTT(\gamma, d)$ with $d = 1e4$ of Q-learning independent agents in a low conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination. $\alpha = 0.2$.</td>
</tr>
<tr>
<td>14</td>
<td>$FTT(\alpha, \gamma)$ with $d = 1e5$ of Q-learning independent agents in a high conflict setting of the Battle of the Sexes repeated game. Obtained with 100 trials of each parameter combination.</td>
</tr>
<tr>
<td>15</td>
<td>Policy evolution of both agents in a high conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 1e5$. $Q(s,T) - Q(s,S) &gt; 0$ means that $\pi(s) = T$ and $Q(s,T) - Q(s,S) &lt; 0$ corresponds to $\pi(s) = S$. In this figure both agents learn the TT equilibrium.</td>
</tr>
</tbody>
</table>
16 Fairness and efficiency evolution of both agents in a high conflict scenario. $\alpha = 0.4$, $\gamma = 0.999$ and $d = 1e5$. In this figure both agents learn the TT equilibrium. Results averaged over 100 trials. Each point corresponds to the measures computed in a window of 1000 time steps. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

17 Representative policy evolution of a basic agent in a low conflict scenario confronted to a modeling agent. $\alpha = 0.2$, $\gamma = 0.999$ and $d = 1e5$. $Q(s,T) - Q(s,S) > 0$ means that $\pi(s) = T$ and $Q(s,T) - Q(s,S) > 0$ corresponds to $\pi(s) = S$. In this figure, the agent learns to dominate. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

18 Policy and model evolution of a modeling agent in a low conflict scenario confronted to a basic agent. $\alpha = 0.2$, $\gamma = 0.999$ and $d = 1e5$. $Q(s,T) - Q(s,S) > 0$ means that $\pi(s) = T$ and $Q(s,T) - Q(s,S) > 0$ corresponds to $\pi(s) = S$. Policies separated by the estimation of the opponent’s action. In this figure, the agent learns to simply avoid ties. The right figure represents the mean of the Beta distribution (the variance is almost 0 after less than 1% of the time steps, thus the mean and the drawn samples coincide). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

19 Policy and model evolution of a modeling agent in a low conflict scenario confronted with another modeling agent with $\alpha = 0.2$, $\gamma = 0.999$ and $d = 1e5$. $Q(s,T) - Q(s,S) > 0$ means that $\pi(s) = T$ and $Q(s,T) - Q(s,S) < 0$ corresponds to $\pi(s) = S$. Policies separated by the estimation of the opponent’s action. In this figure, the agent learns to simply avoid ties. The right figure represents the mean of the beta distribution (the variance is almost 0 after less than 1% of the time steps, thus the mean and the drawn samples coincide). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

20 $FTT(\alpha, \gamma)$ with $d = 1e5$ of SARSA independent agents in a high conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>$F_{TT}(\alpha, \gamma, d)$ for $\gamma \in {0.9 : 0.01 : 0.99}$ of SARSA independent agents in a high conflict setting of the Battle of the Exes repeated game.</td>
<td>Obtained with 100 trials of each parameter combination. 60</td>
</tr>
<tr>
<td>22</td>
<td>$F_{TT}(\alpha, \gamma, d)$ for $\gamma \in {0.991 : 0.001 : 0.999}$ of SARSA independent agents in a high conflict setting of the Battle of the Exes repeated game.</td>
<td>Obtained with 100 trials of each parameter combination. 61</td>
</tr>
<tr>
<td>23</td>
<td>$F_{TT}(\alpha, \gamma)$ with $d = 1e5$ of SARSA independent agents in a low conflict setting of the Battle of the Exes repeated game.</td>
<td>Obtained with 100 trials of each parameter combination. 62</td>
</tr>
<tr>
<td>24</td>
<td>$F_{TT}(\alpha, \gamma, d)$ for $\gamma \in {0.9 : 0.01 : 0.99}$ of SARSA independent agents in a low conflict setting of the Battle of the Exes repeated game.</td>
<td>Obtained with 100 trials of each parameter combination. 63</td>
</tr>
<tr>
<td>25</td>
<td>$F_{TT}(\alpha, \gamma, d)$ for $\gamma \in {0.991 : 0.001 : 0.999}$ of SARSA independent agents in a low conflict setting of the Battle of the Exes repeated game.</td>
<td>Obtained with 100 trials of each parameter combination. 64</td>
</tr>
<tr>
<td>26</td>
<td>$F_{TT}(\alpha, \gamma, d)$ for $\gamma \in {0.9 : 0.01 : 0.99}$ of Q-learning independent agents in a high conflict setting of the Battle of the Exes repeated game.</td>
<td>Obtained with 100 trials of each parameter combination. 66</td>
</tr>
<tr>
<td>27</td>
<td>$F_{TT}(\alpha, \gamma, d)$ for $\gamma \in {0.991 : 0.001 : 0.999}$ of Q-learning independent agents in a high conflict setting of the Battle of the Exes repeated game.</td>
<td>Obtained with 100 trials of each parameter combination. 67</td>
</tr>
</tbody>
</table>
Bibliography


Appendix A

SARSA: On-Policy TD Control

We also tried the performance of the SARSA learning method with the basic experimental approach. Nonetheless, we did not include this in the main document because the results were not significant. Despite this fact, we include them here for the sake of completeness.

The method

In this section, we present an on-policy TD control method. On-policy methods attempt to evaluate and improve the policy that is used to make decisions. The first step is to learn an action-value function. In particular, for an on-policy method we must estimate $q_\pi(s, a)$ for the current behavior policy $\pi$ and for all states $s$ and actions $a$.

We consider transitions from state-action pair to state-action pair, and learn the values of state-action pairs. The theorems assuring convergence of state values under TD(0) also apply to the corresponding algorithm for action values:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$  \hspace{1cm} (A.1)

This update is done after every transition from a nonterminal state $S_t$. This rule uses every element of the quintuple of events, $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$, that make
up a transition from one state-action pair to the next. This quintuple gives rise to the name Sarsa for the algorithm.

It is straightforward to design an on-policy control algorithm based on the Sarsa prediction method. We continually estimate $q_\pi$ for the behavior policy $\pi$, and at the same time change $\pi$ toward greediness with respect to $q_\pi$. The general form of the Sarsa control algorithm is given in the box below.

### Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize $Q(s,a)$ for all $s \in S$, $a \in A(s)$, arbitrarily, and $Q(\text{terminal-state}, \cdot) = 0$

Repeat (for each episode):
- Initialize $S$
- Choose $A$ from $S$ using policy derived from $Q$ (e.g. $\epsilon$-greedy)

Repeat (for each step of the episode):
- Take action $A$, observe $R, S'$
- Choose $A'$ from $S'$ using policy derived from $Q$ (e.g. $\epsilon$-greedy)
- $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
- $S \leftarrow S'$; $A \leftarrow A'$

until $S$ is terminal

The convergence properties of the Sarsa algorithm depend on the nature of the policy’s dependence on $Q$. For example, one could use $\epsilon$-greedy policies. Sarsa converges with probability 1 to an optimal policy and action-value function as long as all state-action pairs are visited and infinite number of times and the policy converges in the limit to the greedy policy (which can be arranged, for instance, by setting $\epsilon = 1/t$).

### Results

In this section, the results obtained by both agents using SARSA as a learning method are presented.
A.2. Results

High conflict

Here are presented the results of SARSA learners playing the repeated Battle of the Sexes in a high conflict scenario, that is, with rewards \((l, h) = (1, 4)\). Figure 20 presents the \(FTT(\alpha, \gamma)\) and Figures 21 and 22 show the \(FTT(\alpha, \gamma, d)\).

![Figure 20: FTT(α, γ) with d = 1e5 of SARSA independent agents in a high conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.](image)

From these results one can easily observe that the SARSA setting in a high conflict scenario is not able to robustly reach the cooperative solution for any of the parameter combinations and with the ones that seem promising, a decay time of \(d = 1e5\) is surely insufficient.

Low conflict

Here are presented the results of SARSA learners playing the repeated Battle of the Exes in a low conflict scenario, that is, with rewards \((l, h) = (1, 2)\). Figure 23 presents the \(FTT(\alpha, \gamma)\) and Figures 24 and 25 show the \(FTT(\alpha, \gamma, d)\).

Finally, this last setting presents, again, an improvement with respect to its high conflict counterpart. The agents are able to robustly learn the cooperative strategy for some parameter combinations and for some others they would probably learn it given a sufficiently large decay time.
Figure 21: $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.9 : 0.01 : 0.99\}$ of SARSA independent agents in a high conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.
Figure 22: $\text{FTT}(\alpha, \gamma, d)$ for $\gamma \in \{0.991 : 0.001 : 0.999\}$ of SARSA independent agents in a high conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.
Figure 23: $FTT(\alpha, \gamma)$ with $d = 1e5$ of SARSA independent agents in a low conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.
Figure 24: $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.9 : 0.01 : 0.99\}$ of SARSA independent agents in a low conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.
Figure 25: \( FTT(\alpha, \gamma, d) \) for \( \gamma \in \{0.991 : 0.001 : 0.999\} \) of SARSA independent agents in a low conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.
Appendix B

Extra convergence results

Here are included extra convergence results that were not considered in the main document. This decision was made because the information provided by them is not necessary for the work but it is illustrative. Nevertheless, the dimensions of them really did saturate the document.
Figure 26: $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.9 : 0.01 : 0.99\}$ of Q-learning independent agents in a high conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.
Figure 27: $FTT(\alpha, \gamma, d)$ for $\gamma \in \{0.991 : 0.001 : 0.999\}$ of Q-learning independent agents in a high conflict setting of the Battle of the Exes repeated game. Obtained with 100 trials of each parameter combination.