# Q-Learning to Cooperate*

Emilio Calvano[+#], Giacomo Calzolari[+#&], Vincenzo Denicolò[+&] and Sergio Pastorello[+]

[+]University of Bologna, [&]CEPR and [#] Toulouse School of Economics

*Preliminary and incomplete*

May 31, 2018

**Abstract**

This is a preliminary report on ongoing research that uses Artificial Intelligence algorithms as experimental subjects to study infinitely repeated games. We focus on the repeated prisoner dilemma and on repeated games of pricing. We uncover several "anomalies" with respect to the standard theoretical predictions.

*Keywords*: Artificial intelligence; Q-learning; iterated prisoner dilemma; cooperation; collusion

*J.E.L. numbers*:[].

*Financial support from Almaidea and the Digital Chair of the Toulouse school of Economics are gratefully acknowledged.

# 1 Introduction

It is generally acknowledged that the theory of repeated games is largely inconclusive. In a variety of different settings, folk theorems have shown that if players are sufficiently patients, many different outcomes can be sustained as a sub-game perfect equilibrium of the infinitely repeated game. In the face of this wide multiplicity of equilibria, there seems to be no consensus on what refinements should be used to make sharper predictions.

In view of this, the experimental approach can usefully supplement theory and as such has received considerable attention in the last decades: see Dal Bò and Frechette (2018) for a recent, extensive survey of the literature. This research project contributes to this approach. The main difference with the previous experimental literature is that we use Artificial Intelligence algorithms, rather than humans, as our experimental subjects.

Specifically, we focus on machine learning algorithms which in the Artificial Intelligence literature have been proposed as a model of human boundedly rational behaviour. These algorithms do not have any *a priori* knowledge of the problem at hand (e.g., the structure of the game they play) but are able to learn from experience. In order to acquire experience, they perform random experimentation. A learning mechanism then processes the results of the experimentation and instructs the algorithm to take advantage of them.

We are especially interested in modelling the behaviour of firms in the marketplace. Our working hypothesis is that, after a suitable learning phase, machine learning algorithms may approximate the behaviour of experienced and intelligent managers reasonably well.

By contrast, the experimental literature has generally taken unexperienced human subjects as an approximation for sophisticated decision makers. While both approximations may be useful, which one is better probably depends on the problem at hand. For example, suppose one aims to figure out how professional chess players would play a chess game. One possibility would be to recruit a number of undergraduate students who have never played chess before, teach them the rules of the game, have them play a dozen training matches, and then look at how they play. One may doubt, however, whether this would provide a good approximation of professional play. A perhaps better approach might be to use chess programs based on artificial intelligence. Their style might be slightly peculiar, but the quality of play would compare to professional players'. In fact, the best chess programs available today can easily beat Magnus Carlsen, the current world

champion. To get a reasonable model of professional human play, one would have to downgrade the programs, or resort to relatively outdated ones.

This is precisely what we do in this research project. We use relatively simple machine learning algorithms, i.e. Q-learning, to study the emergence of cooperation in repeated strategic interactions. Q-learning is described in greater detail in Section 2 below. It is not the state of the art in machine learning, but is well understood, easy to use and is the building block of many sophisticated AI algorithms.[1] In future research, one might study how results would change with more (or still less) sophisticated algorithms.

While the goodness of approximation may be open to debate, there are several evident practical advantages of using algorithms rather than humans as experimental subjects. First, with algorithms we can observe directly not only the players' actions but also their strategies. Recovering strategies from observed actions in experiments with humans, in contrast, requires considerable ingenuity on the part of the experiment designer. Second, a problem that has long bothered experimentalists is how to model infinitely repeated games in experiments of finite duration. Our algorithms do not suffer from this problem as they are specifically designed to play an infinitely repeated game but can be stopped at any time. Finally, algorithms are cheaper, quicker and more reliable than most undergraduate students.

This allows us to run experiments on an unprecedented scale. Even if this project is still ongoing, we have already considered several hundreds configurations of model parameters. For each configuration, we have run 1000 experiments each involving enough repetitions of the stage games to allow the algorithms to complete their learning (this may easily take some 150,000-200,000 repetitions). Given the fineness of the parameter grid and the number of experiments for each cell of the grid, the resulting estimates are extremely precise and allow a very detailed comparative statics analysis.

The analysis is divided in two parts. In the first part, we consider the classic prisoner dilemma with two symmetric players each of whom has two possible actions. In the second part, we focus on pricing games played by firms in the marketplace. Here there may be more than two firms, firms may asymmetric, and the action space may be bigger. One additional motivation for this latter part of the analysis is that pricing algorithms are increasingly used in electronic commerce and

[1] Such as with the famous deep reinforcement learning experiment in Mnih et al. (2015).

elsewhere (Chen et al. 2016). Our analysis has direct implications for the study of these markets.

Before describing our preliminary results and ongoing research, we provide a brief introduction to Q-learning. After that, the paper comprises two more sections, one for the prisoner dilemma and the other for pricing games.

## 2 Q-learning

Even if we are interested in repeated games, it may be useful to consider a single decision maker first. The decision maker controls a stationary Markov system. Let $s \in S$ denote the state of a system and $a \in A$ an action. Both $S$ and $A$ are taken to be finite. The system starts at some state $s_0$ and moves from state to state according to a Markov transition function

$$s_{t+1} = F(s_t, a_t). \tag{1}$$

The transition might be stochastic but for the sake of simplicity we abstract from uncertainty in this presentation.

The decision maker observes the current state and aims to maximize the discounted payoff

$$V = \sum_{t=1}^{\infty} \pi(s_t, a_t)\delta^t \tag{2}$$

where $\pi(s_t, a_t)$ is the period-$t$ payoff and $\delta$ is the discount factor.

To solve this problem, a Q-learning algorithm attaches to each pair $(s, a)$ a number $Q(s, a)$ that is meant to represent the estimated value of choosing action $a$ in state $s$, in the same spirit as the Bellman value function in dynamic programming. The initial matrix $Q_0$ consists of arbitrarily assigned values. The core of a Q-learning mechanism is the learning equation

$$Q_{t+1}(s,a) = \begin{cases} (1-\alpha)Q_t(s,a) + \alpha[\pi(s,a) + \delta \max_{a \in A} Q_t(F(s,a),a)] & \text{if } s_t = s \text{ and } a_t = a \\ \\ Q_t(s,a) & \text{if } s_t \neq s \text{ or } a_t \neq a. \end{cases} \tag{3}$$

The parameter $\alpha \in [0, 1]$ is called the learning rate. Equation (3) says that the value $Q_t(s, a)$ is not updated if the current state is different from $s$ and/or the current action is different from $a$. If $s_t = s$ and $a_t = a$, the value is updated, the new value being a convex combination of the old value and the current one-period payoff plus the discounted value of the new state which is reached.

Given the table of $Q$-values, in each period the algorithm either exploits, i.e. chooses the action with the highest $Q$-value for the current state, or experiments. The choice between exploitation and experimentation will be described in greater detail shortly. For now, just keep in mind that for deterministic problems it is generally assumed that the probability of experimentation declines and eventually converges to zero.

For a single decision maker, it has been shown that under certain regularity conditions the Q-learning algorithm converges to the policy that maximizes (3).

Next, consider $N$ Q-learning algorithms that play a repeated game. In the repeated game model, we assume that players have bounded recall; specifically, each player keeps track of the actions chosen by itself, and possibly also by the other players, in the last $k$ repetitions of the game. For example, with perfect observability of the opponents' actions and a $k$-period memory, a state $s$ is the list of the actions chosen by the $N$ players in the last $k$ stages.

The per-period payoff of agent $i$ may now depend on the opponents' choices, denoted by $\mathbf{a}_{-i}$: $\pi_i(s, a_i, \mathbf{a}_{-i})$. Likewise, the transition function may depend on all players' actions, $s_{t+1} = F(s_t, \mathbf{a}_t)$, where $\mathbf{a}_t$ is the $N$-dimensional vector of the players' period-$t$ choices. The algorithms however continue to learn in a non-strategic way. That is, they regard opponents as part of the environment and still update their $Q_i$-matrix according to (3).

To complete the description of the model, one needs to specify how actions are chosen and the matrix $Q$ is initialized. This is what we do in the next subsections.

### 2.1 Experimentation

Following the computer science literature, we consider two models of experimentation, the $\varepsilon$-greedy model and the Boltzman model.

In the $\varepsilon$-greedy model, the algorithm chooses the action with the highest $Q$-value with probability $1-\varepsilon$, and randomizes uniformly across every feasible action $a \in A$ with probability $\varepsilon$. Thus, $\varepsilon$ is the fraction of times the algorithm is in *experimentation mode*, while $1-\varepsilon$ is the fraction of times it is in *exploitation mode*. The probability $\varepsilon$ may vary over time. In our experiments, we have set $\varepsilon_1 = 1$ and let $\varepsilon_t$ decrease exponentially with speed $\beta$, i.e.

$$\varepsilon_t = e^{-\beta(t-1)}. \tag{4}$$

In the Boltzman experimentation model, in contrast, actions are chosen with probabilities that reflect their current $Q$-values

$$\Pr(a_t = a \,|_{s_t=s}) = \frac{e^{Q_t(s_t,a)/T}}{\sum_{a' \in A} e^{Q_t(s_t,a')/T}} \tag{5}$$

where the parameter $T$ is often referred to as the "temperature" of the system. Like $\varepsilon$, $T$ is also taken to decrease over time, converging to 0. When $T = 0$, the algorithm chooses the action with the highest $Q$-value with probability one, so it does not experiment any longer.

Summarizing, a Q-learning algorithm is characterized by two parameters: the learning rate $\alpha$, and the intensity of experimentation. The latter is captured by parameter $\beta$ in the $\varepsilon$-greedy model with exponentially decreasing probability, and by the function $T(t)$ in the Boltzman experimentation model.

## 2.2 Initialization

We have initialized the matrix $Q$ in two ways. In some experiments, $Q_{i0}(s, a)$ was set at

$$Q_{i0}(s, a_i) = \frac{\pi_i(s, a_i, a_{-i}^{\text{Nash}})}{1 - \delta} \tag{6}$$

where $a_{-i}^{\text{Nash}}$ is the list of player $i$'s opponents' Nash equilibrium actions in the stage game (which is assumed to have a unique Nash equilibrium). In other words, the initialization assumes that the other players do not cooperate. This presumably makes it more difficult for players to learn to cooperate.

In other experiments, on the other hand, the initialization was

$$Q_{i0}(s, a_i) = \frac{\sum_{a_{-i} \in A^{n-1}} \pi_i(s, a_i, a_{-i})}{(1 - \delta) \,|A|^{n-1}} \tag{7}$$

That is, player $i$ initially assumes that its opponents randomize uniformly across all possible strategies. This is consistent with the $\varepsilon$-greedy model of experimentation for $\varepsilon_1 = 1$. When players do learn to cooperate, convergence to cooperation is faster with this latter mode of initialization, but overall the observed differences between the two cases are minor.

### 2.3 Related literature

Q-learning algorithms are related to several strands of the economics literature on learning and experimentation in games. [...]

## 3 Repeated prisoner dilemma

Our first set of experiments uses the prisoner dilemma as the stage game. The game is infinitely repeated with bounded recall.

### 3.1 The stage game

In the stage game, there are two players denoted by $i$ and $j$, each with two strategies: cooperate ($C$) or defect ($D$). The game is described in the following table:

| | | player 2 | |
|---|---|---|---|
| | | $D$ | $C$ |
| player 1 | $D$ | $0, 0$ | $g, \ell$ |
| | $C$ | $\ell, g$ | $1, 1$ |

We normalize to 1 the players' payoffs under cooperation, and to 0 the payoffs if both defect. If player $i$ cooperates and $j$ defect, player $i$ gets the sucker's payoff $-\ell$, player $j$ the deviation payoff $g$. We assume that

$$g - \ell < 2 \tag{8}$$

so that $CC$ dominates alternating between $CD$ and $DC$.

### 3.2 Strategies and equilibria with bounded recall

Players have bounded recall. In the $k$-memory model with perfect observability, each player recalls his and the opponent's actions in the last $k$ stages. Since each agent has two actions, there are $|S| = 4^k$ states in a $k$-memory game. So far, in our experiments we have focused on the cases $k = 1$ and $k = 2$.

When $k = 1$ there are four possible states. These may be conveniently denoted as $DD$, $DC$, $CD$ and $CC$, where the first letter denotes the last period action of the first player and the second letter that of the second.

For each state, a player can choose between two actions, so we have $4^2 = 16$ possible strategies. Some of these strategies are noteworthy. For example, one strategy is always to defect, another always to cooperate. Since these strategies do not effectively condition actions on the current state, they do not actually require any memory. However, they are also obviously feasible with a one-period memory.

Tit-for-tat, i.e. the strategy of choosing the same action as the opponent did in the last stage, is a strategy which requires a one-period memory and is therefore feasible in this setting. In contrast, a grim trigger strategy would in principle require unbounded recall. However, even in a one-period memory framework one may define a strategy in the same spirit as grim trigger as follows: cooperate if both players cooperated in the last stage (i.e., in state $CC$), defect otherwise. The "incentive compatibility" condition for this strategy (that is, the necessary and sufficient condition for the strategy to be an equilibrium in the one-period memory game ) is exactly the same as the corresponding condition for true grim trigger strategies to be an equilibrium with unbounded recall. Intuitively, players interpret the fact that at least one of them defected in the last stage as a signal that a player defected in some earlier stage and thus keep "punishing." For this reason, we shall call that strategy "grim trigger."

Another noteworthy strategy is the "one-period punishment" one (sometimes also called the "Pavlov" strategy). This strategy prescribes to cooperate if both players made the same choice in the last stage (i.e., in states $CC$ and $DD$), defect if players made different choices. In a one-period memory model, the incentive compatibility constraint for this strategy looks like that of the genuine one-period punishment strategy with unbounded recall. Intuitively, players interpret a unilateral choice of $D$ as defection, but a coordinated choice of $D$ as the joint punishment of a past defection, which is followed by a return to cooperation.

If the discount factor $\delta$ is sufficiently large, there are three equilibria of the infinitely repeated game with 1-period memory: always defect, "grim trigger", and "one-period punishment". The equilibrium where both players always defect results in a long-run payoff of zero, the latter two in a long-run payoff of $\frac{1}{1-\delta}$.

Theory does not say which of this equilibria prevails. Furthermore, there are other pairs of strategies that may come close to being an equilibrium, in the sense that they too deliver a long-run payoff of $\frac{1}{1-\delta}$, and, while they are not deviation-proof, entail a small gain from deviation. If the algorithms get stuck on such a pair of strategies for a sufficiently long time and then stop learning, the outcome may be "stable" even if it is not an equilibrium.

When $k = 2$, there are sixteen possible states and hence $16^2 = 256$ possible strategies. The strategies just described remain available and many more become feasible. [...]

### 3.3 Preliminary results

So far we have explored only the $\varepsilon$-greedy model of experimentation. We plan to analyze the Boltzman case in the near future.

In the $\varepsilon$-greedy model, the Q-learning algorithm is characterized by two parameters, $\alpha$ and $\beta$. In the infinitely repeated prisoner dilemma, we have four additional parameters: the gain from defection $g$, the sucker's payoff $\ell$, the discount factor $\delta$, and the length of the memory $k$.

We have conducted a number of experiments varying each of these parameters. For each combination of parameter values, we have run 1,000 experiments, each of which consisted of a large number of repetitions of the stage game.

#### 3.3.1 Convergence

When two Q-learning algorithms play against each other, even if the outside environment is stationary each player faces a non-stationary problem as the other player is experimenting and learning. Since we aim to understand the behaviour of experienced players, we focus on the outcome that emerges after a number of repetitions sufficiently large that learning may be regarded as completed. We take this to mean that the optimal strategy (i.e., the value of $a$ that maximizes $Q(s, a)$ for each value of $s$) stays constant for at least 25,000 consecutive repetitions of the stage game, for both players.

In principle, such "convergence" may not be achieved in finite time. Classical convergence results for single decision makers do not apply here, and we are not aware of any convergence results for a game-theoretic framework. However, in all of our experiments the algorithms did manage to

converge in finite time. The average number of repetitions needed to achieve "convergence" is around 150,000-200,000.

**Result 1.** *Q-learning algorithms playing repeated prisoner dilemma converge.*

Figure 1 illustrates how convergence is achieved by plotting the difference between the $Q$-values associated with actions $D$ and $C$, respectively, for a given state (in this case, $CC$). In an initial learning phase, the difference oscillates between the positive and the negative region, leading to repeated switches in the optimal strategy. Eventually, however, the sign of the difference stays constant; in particular, the difference becomes constantly negative, pointing to the optimality of action $C$.

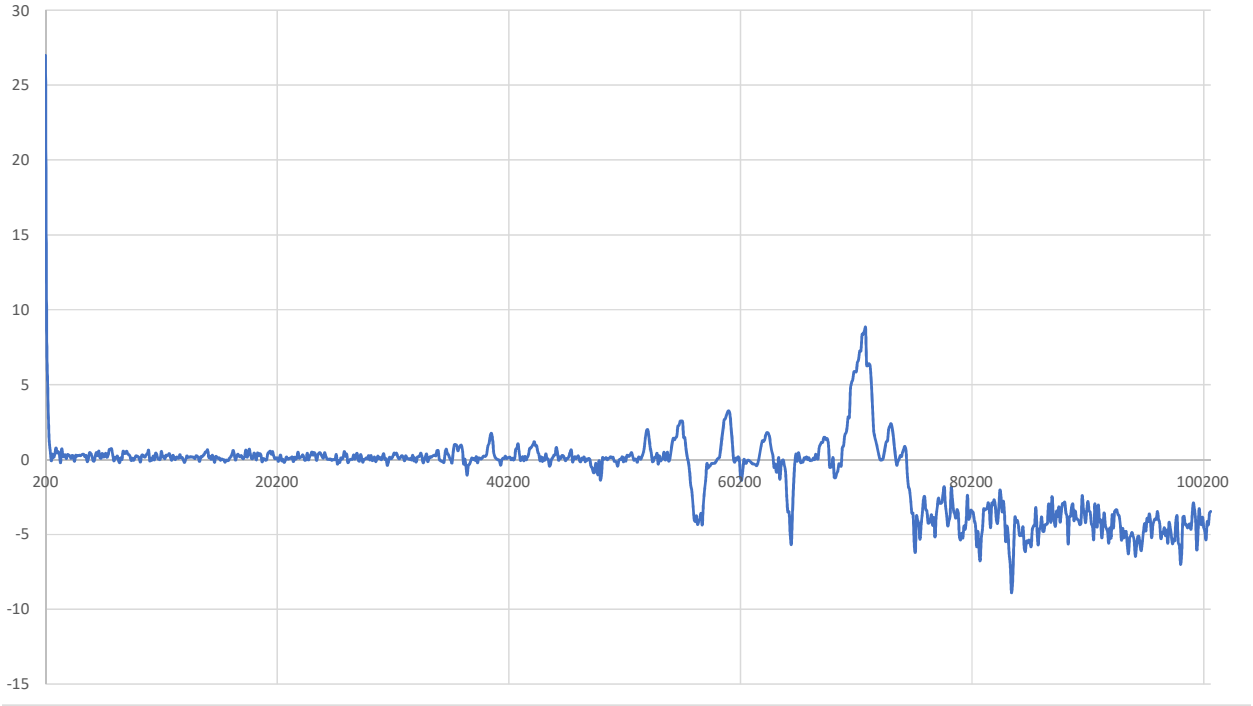


Figure 1: The value of defection relative to cooperation following "cooperative" play of the stage game: $Q((CC), D) - Q((CC), C)$.

The system often converges to a pair of equilibrium strategies, but not always. The fraction of cases where we have convergence to an equilibrium depends on the parameters of the model. A regularity that emerges, however, is that convergence to an equilibrium pair of strategies is significantly faster than convergence to a non-equilibrium.

In view of Result 1, we have let the algorithms play until convergence is achieved. The results reported below refer to the last 25,000 repetitions of the stage game.

### 3.3.2 Learning to cooperate

Q-learning algorithm do learn to cooperate.

**Result 2.** *With appropriate levels of learning and experimentation, a substantial amount of cooperation may be obtained.*

We measure cooperation in two ways, which lead to similar result. The first one is the fraction of the last 25,000 repetitions in which players end up playing $CC$. The second measure is the "average gain," i.e. the ratio between the players' average payoff and the cooperative payoff of 1. According to both measures, for many configurations of the parameters we observe levels of cooperation greater than 90% and also greater than 95%.

These values are significantly larger than those reported in the computer science literature, where the level of cooperation hardly exceeds 30% (Sandholm and Crites, 1996; Waltman and Kaymak, 2008). We conjecture that the reason why we find more cooperation is that we have allowed for more repetitions and more experimentation than in the previous literature.

As it turns out, cooperation is typically most likely when the learning rate $\alpha$ is in the range of 10%-20% and the experimentation parameter $\beta$ is in the range 60%-80%, as illustrated in Figure 2.

It may be interesting to consider how cooperation is achieved: *via* grim-trigger strategies, one-period punishment strategies, or else with strategies which are not a sub-game perfect equilibrium of the game? The next result shows that the role played by grim trigger strategies in sustaining cooperation is limited.

**Result 3.** *Q-learning players almost never converge to grim trigger strategies.*

The intuitive reason for this is that while grim trigger strategies are good for intelligent players who correctly forecast the deterministic equilibrium response of the opponent, they are quite ineffective when opponents experiment because in this case a single episode of experimentation may trap the players in state $DD$ for a very long time, until enough further experimentation may get the

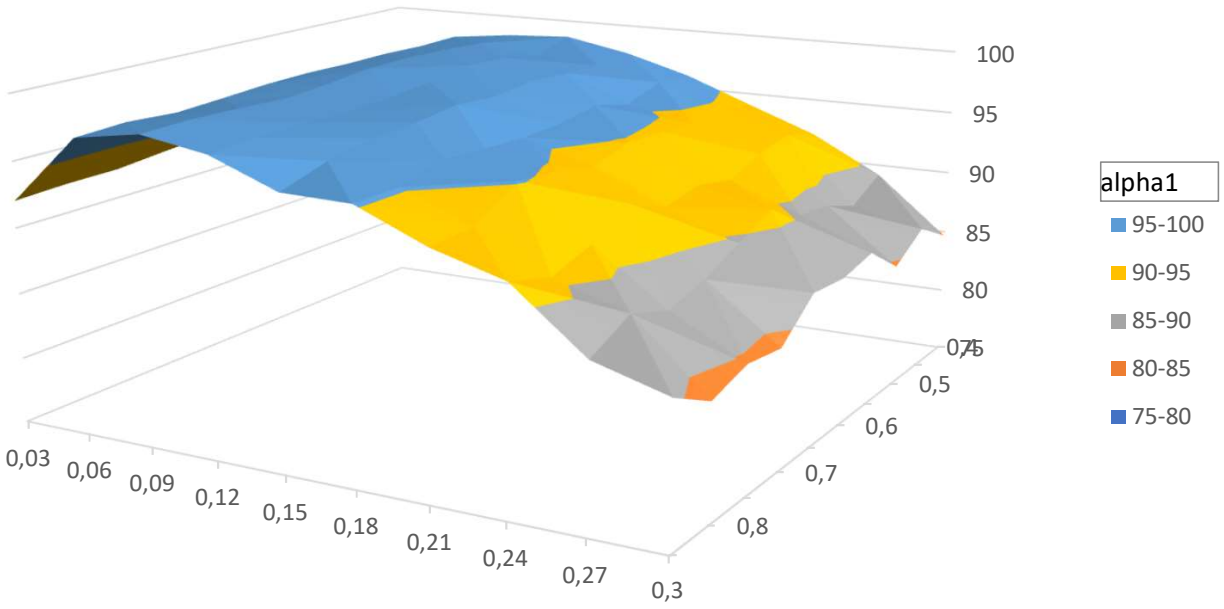


Figure 2: The effect of the learning rate $\alpha$ and the experimentation parameter $\beta$ on cooperation (measured as the frequency of state $(CC)$).

players out of the trap. This is very costly, and thus the Q-learning programs learn to avoid these strategies.

### 3.3.3 Patience

The literature on infinitely repeated game has largely focused on the discount factor as a critical determinant of cooperation. Higher discount factors, which correspond to more patient players, or more frequent interaction, are almost always regarded as conducive to cooperation. A special attention has been paid to the critical discount factor that makes grim trigger strategies an equilibrium.

Our analysis casts doubts on this traditional approach.

**Result 4.** *Patience initially facilitates cooperation, but when $\delta$ is sufficiently high a further increase in $\delta$ decreases cooperation.*

This is illustrated in Figure 3. Several remarks are in order. First, for values of $\delta$ rather low (less than $\frac{1}{3}$ for the given parameter values), the only equilibrium of the game with one-period memory is always to defect. Accordingly, the extent of cooperation is limited, practically non existent.

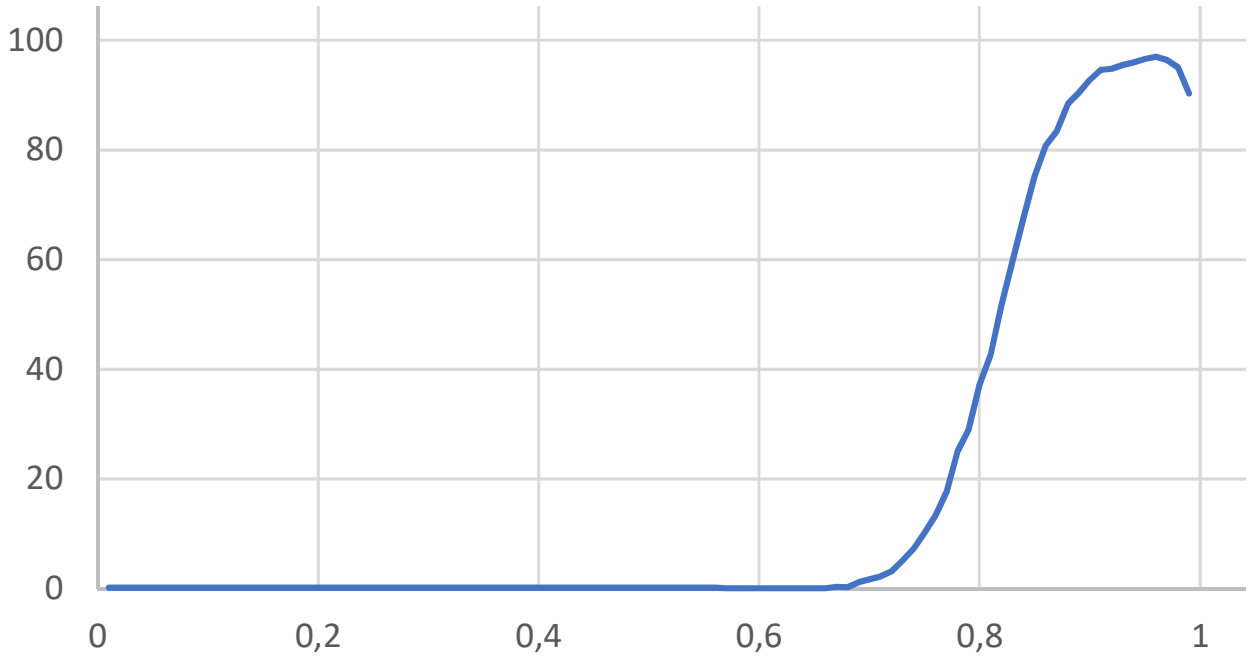


Figure 3: The effect of the discount factor $\delta$ on cooperation (frequency of state $(CC)$).

Surprisingly, however, we observe no cooperation even for values of $\delta$ substantially higher than $\frac{1}{3}$. On reflection, this is partly explained by Result 3. The critical level of the discount factor of $\frac{1}{3}$ is the level for which grim trigger strategies become an equilibrium of the repeated game. But we have seen that algorithms almost never converge to grim trigger strategies. A more relevant critical threshold is the value of $\delta$ for which one-period punishment becomes an equilibrium. For the parameter values of Figure 3, this is $\frac{1}{2}$. Yet, Figure 3 shows that cooperation starts emerging only for significantly higher values of $\delta$, around $\frac{2}{3}$. These results suggest that the focus on critical thresholds of the discount factor, which is ubiquitous in the economics literature on cooperation and collusion, may actually be misplaced.

Second, increases in the discount factor $\delta$ increase the overall rate of cooperation up to $\delta \approx 0.95$. However, for higher values further increases in $\delta$ may actually impede cooperation. This runs counter to conventional wisdom according to which increasing $\delta$ makes cooperation easier to sustain. Sannikov and Skrzypacz (2007) find that this is no longer true with imperfect monitoring, but in our experiment we have perfect monitoring of the rival's last action. We conjecture that the non-monotonicity may be due to the fact that with a constant learning rate $\alpha$, changes in $\delta$ may affect

the speed with which the algorithms actually learns.

Third, it may be interesting to observe that for intermediate values of $\delta$ cooperation is achieved mostly by means of one-period punishment strategies (which are then equilibrium strategies). As $\delta$ further increases, however, our artificial players no longer converge to equilibrium strategies but achieve cooperation by means of strategies that systematically lead to the "good" state $CC$ in the absence of experimentation but are not a best response to each other (or they are so only approximately).

### 3.3.4 Stage-game payoffs

We next turn to the impact on cooperation of the payoffs in the stage game, $g$ and $\ell$.

**Result 5.** *Changes in the sucker's payoff $\ell$ have a substantial impact on the level of cooperation.*

This is illustrated in Figure 4. The result is surprising as the sucker's payoff does not enter the

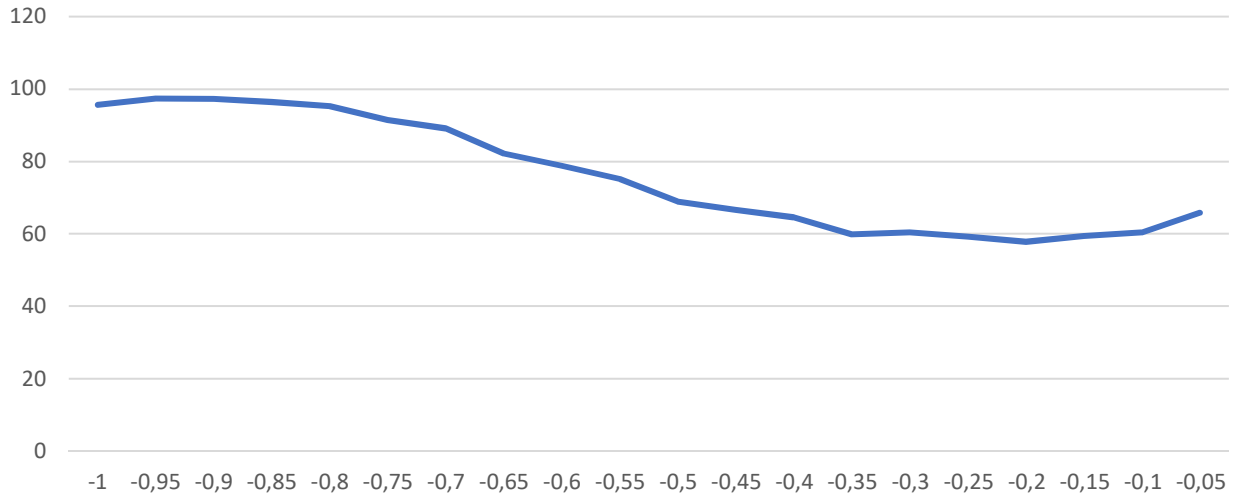


Figure 4: Level of cooperation (frequency of state ($CC$)) for different values of the sucker's payoff $\ell$.

incentive compatibility conditions that determine when grim trigger or one-period punishment are Nash equilibria of the repeated game. A similar result has been noted also in the experimental literature with human subjects (see Dal Bò and Frechette, 2011, Blonski and Spagnolo, 2011). In that literature, the explanation that has been proposed for this finding is that subjects are risk

averse, as the sucker's payoff affect the riskiness of strategies. However, algorithms are not risk averse. This suggests that the role of the sucker's payoff may be due to learning rather than, or in addition to, risk aversion.

**Result 6.** *Changes in the gain from defection $g$ and the sucker's payoff $\ell$ have a non-monotonic impact on the level of cooperation.*

The non-monotonicity with respect to $\ell$ is already apparent in Figure 4; Figure 5 illustrates that for $g$.

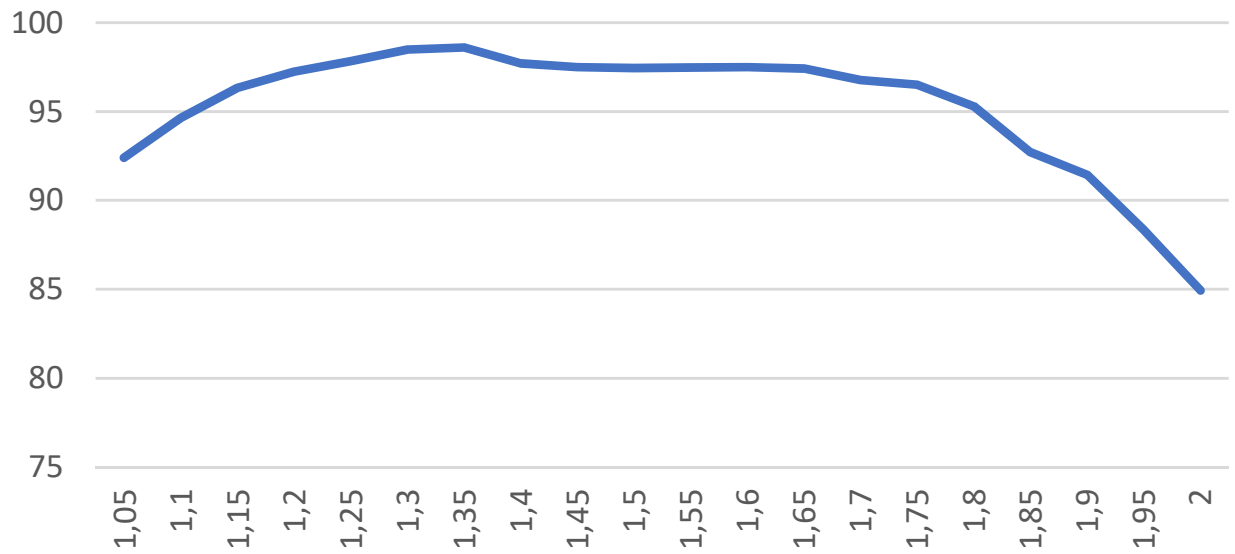


Figure 5: Level of cooperation (frequency of state $(CC)$) for different values of the gain from defection $g$.

Both non-monotonicities are surprising, and for both we still lack a convincing explanation. One would expect that cooperation is more difficult when $g$ increases and that, if anything, cooperation should become more difficult as $\ell$ increases. But our results indicate that both parameters have a non-monotonic impact on the level of cooperation. If $g$ is small, the level of cooperation does not depend on $\ell$ significantly. If $g$ is high, the level of cooperation first decreases and then increases with $\ell$. If $\ell$ is large, the level of cooperation does not depend on $g$ significantly. If $\ell$ is small, the level of cooperation first increases and then decreases with $g$. Overall, cooperation seems to be most difficult when $g$ is high but $\ell$ is low.

### 3.3.5 Memory

We have just started to explore the effects of longer memory, considering the case $k = 2$. For now, the results we have obtained are as expected.

**Result 7.** *Longer memory facilitates cooperation.*

Anything else constant, we have observed higher levels of cooperation when players have a 2-period memory than when they have 1-period memory. This result is illustrated in Figure 6, which plots the level of cooperation as a function of the learning rate $\alpha$ for the cases $k = 1$ and $k = 2$. Extension to $k > 2$ is still to be done.

## 4 Games of pricing

In this section we describe experiments on the interaction between Q-learning algorithms in pricing games. This may be viewed as a stylized model of experienced and intelligent human decision, but also as a model of the interaction of two or more algorithmic pricing programs. Indeed, in electronic commerce and elsewhere, firms' pricing decisions are increasingly delegated to software programs that incorporate the latest developments of artificial intelligence. This has lead lawyers and economists to voice concerns that algorithmic pricing may facilitate collusion, posing new challenges to antitrust authorities (see, for instance, Ezrachi and Stucke (2015), Mehra (2016) and Harrington (2017)). Our analysis may shed light on this important policy issue: see Calvano et al. (2018) for a more thorough discussion.

So far, we have conducted experiments for pricing games in which demand is derived from Singh and Vives (1984) preferences. In particular, with two firms and two products the buyer's utility function, in monetary terms, is taken to be

$$u\left(q_1, q_2\right) = q_1 + q_2 - \frac{1}{2}(q_1^2 + q_2^2) - \gamma q_1 q_2. \tag{9}$$

The associated demand functions are:

$$p_i = 1 - q_i - \gamma q_j. \tag{10}$$

Marginal costs $c_i$ are constant.

## 4.1 Preliminary results

So far we have explored only two of the many issues that arise in this richer framework: whether cooperation is easier or more difficult to achieve when the number of possible strategies increases, which makes coordination less straightforward, and the impact on cooperation of the degree of product differentiation. In both cases, the effects we have found are non monotonic.

**Result 8.** *Complexity has a non monotonic impact on cooperation.*

As the number of possible price firms can choose from increases, the level of cooperation first decreases and then increases.

**Result 9.** *Product differentiation has a non monotonic impact on cooperation.*

As the degree of product differentiation increases, the level of cooperation first decreases and then increases.

## 4.2 Extensions

In future work we plan to explore the following:

### 4.2.1 Asymmetries

Conventional wisdom maintains that collusion is more difficult when firms are asymmetric. This follows mechanically from the fact that with symmetric firms there is just one "incentive compatibility" constraint for the sustainability of collusion, whereas with asymmetric firms there are more. But we suspect that things may be more complicated. We plan to introduce asymmetries in terms of cost, demand and the level of patience.

### 4.2.2 Number of competitors

Conventional wisdom is that collusion becomes more difficult as the number of firms increases. This is an important issue which is highly ranked in our research agenda.

#### 4.2.3 Stochastic demand

We plan to consider the case of stochastic demand, using a logit model with a Poisson process of arrival of buyers.

#### 4.2.4 Imperfect observability

In this extension, we shall consider the possibility that prices are observable but also that each firm can observe its current profit but not the rivals' prices. With stochastic demand, imperfect observability may make it more difficult to detect defection and hence to sustain collusion. Whether this still holds true for Q-learning algorithms remains to be seen.

## 5 Conclusion

[to be added]

## References


[1] Blonski, Matthias, Peter Ockenfels, and Giancarlo Spagnolo, 2011. Equilibrium selection in the repeated prisoner's dilemma: Axiomatic approach and experimental evidence. *American Economic Journal: Microeconomics*, 3(3): 164–192.

[2] Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò and Sergio Pastorello (2018), Algorithmic pricing: What implications for competition policy?, *Review of Industrial Organization*, forthcoming.

[3] Chen, L., A. Mislove, and C. Wilson, 2016, An empirical analysis of algorithmic pricing on Amazon marketplace. *Proceedings of the 25th International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee, 2016.

[4] Dal Bó, Pedro and Fréchette, Guillaume R. (2011). The evolution of cooperation in infinitely repeated games: Experimental evidence. *American Economic Review*, 101(1):411–429.

[5] Dal Bó, Pedro and Fréchette, Guillaume R. (2018). On the determinants of cooperation in infinitely repeated games: A survey. *Journal of Economic Literature*, 56(1):60–114.

[6] Ezrachi, A. and M. E. Stucke, 2015, Artificial Intelligence and Collusion: When Computers Inhibit Competition, Oxford Legal Studies Research Paper No. 18/2015, University of Tennessee Legal Studies Research Paper No. 267.

[7] Harrington J. E. 2017, Developing Competition Law for Collusion by Autonomous Agents, working paper, The Wharton School, University of Pennsylvania.

[8] Mehra, S. 2016, Antitrust and the Robo-Seller: Competition in the Time of Algorithms, *Minnesota Law Review* 1323.

[9] Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529–533.

[10] Sandholm, Tuomas, and Robert H. Crites (1996) Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* 37.1-2 : 147-166.

[11] Sannikov, Y., and A. Skrzypacz (2007), Impossibility of Collusion Under Imperfect Monitoring With Flexible Production, *American Economic Review*.

[12] Singh, Nirvikar, and Xavier Vives (1984) Price and quantity competition in a differentiated duopoly. *The RAND Journal of Economics*: 546-554.

[13] Waltman, L. and U. Kaymak. (2008) Q-learning agents in a Cournot oligopoly model, *Journal of Economic Dynamics and Control* 32, 10: 3275-3293.