Strategies and equilibria on selected markets: a multi-agent simulation and stochastic modeling approach

Master's thesis Defense

Aldric Labarthe

Ecole Normale Supérieure Paris-Saclay

23/05/2024

Introduction

Abstract

In this paper, we introduce the first agent-based model of competition in quantities featuring a *Deep Deterministic Policy Gradient* (DDPG) algorithm. This algorithm has been selected as a replacement for the traditional Q-Learning algorithm to examine two current unsolved questions in the economic literature: the tendency of algorithmic markets to converge toward a collusive equilibrium, and the chaotic behavior of the dynamic Cournot oligopoly. We show that the DDPG algorithm is a relevant tool to model oligopolies with independent learning agents. We find that our model consistently converges toward the Nash-equilibrium in every market structure we have tested, except for the Cournot oligopoly with well-tuned parameters. We estimate the effect of these parameters on the decision process and explain why collusion may occur in this situation. Overall, we show that algorithmic collusion remains an exception when algorithmic complexity increases. We also place our model in chaotic settings and find that the chaotic behavior of the dynamic Cournot model was only theoretical and never observed in simulations.

Theorem

The Cournot oligopoly when the number of competitors is at least 3, does not necessarily converge toward a stable equilibrium.

Theorem

The Cournot oligopoly when the number of competitors is at least 3, does not necessarily converge toward a stable equilibrium.

- Theocharis 1960 is the first to have mathematically demonstrated that Cournot's "*adjustment mechanism*" was no guarantee for the equilibrium to exist.

Theorem

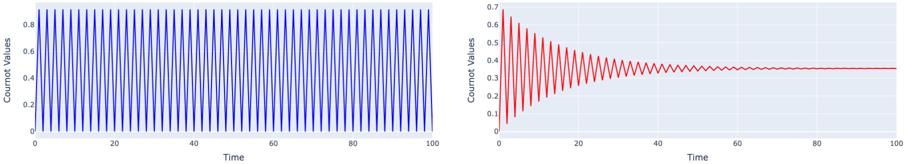
The Cournot oligopoly when the number of competitors is at least 3, does not necessarily converge toward a stable equilibrium.

- Theocharis 1960 is the first to have mathematically demonstrated that Cournot's "*adjustment mechanism*" was no guarantee for the equilibrium to exist.
- Several papers have since explorated different demand structures and costs functions, as Puu 2008; Agiza and Elsadany 2003; Agiza and Elsadany 2004.

Theorem

The Cournot oligopoly when the number of competitors is at least 3, does not necessarily converge toward a stable equilibrium.

- Theocharis 1960 is the first to have mathematically demonstrated that Cournot's "*adjustment mechanism*" was no guarantee for the equilibrium to exist.
- Several papers have since explorated different demand structures and costs functions, as Puu 2008; Agiza and Elsadany 2003; Agiza and Elsadany 2004.
- While some further developments have been made with exotic evolutionary approaches (Hommes, Ochea, and Tuinstra 2011), the issue of the instability of the Cournot oligopoly remains an unsolved question and no economical explanation has been suggested, let alone proved.



(a) $c = 0.2$ (b) $c = 0.55$

Figure 1: **Numerical simulations illustrating the behavior of the Cournot system of quantities** (simulations conducted with $D = 2.2$ and $N = 4$, *quantities are bounded between 0 and 1*).

Sketch of the proof

$$\text{argmax}_{q_t^i} \Pi_t^i(q_t^i, q_{t-1}^{-i}) = (D - q_t^i - q_{t-1}^{-i}) q_t^i - c(q_t^i)^2 \Rightarrow q_t^i(q_{t-1}^{-i}) = \frac{D - q_{t-1}^{-i}}{2(1+c)}$$

$$\begin{cases} q_t^1 = \frac{D - \sum_{i \neq 1} q_{t-1}^i}{2(1+c)} \\ \vdots \\ q_t^N = \frac{D - \sum_{i \neq N} q_{t-1}^i}{2(1+c)} \end{cases} \Leftrightarrow \underbrace{\begin{bmatrix} q_t^1 \\ \vdots \\ q_t^N \end{bmatrix}}_{Q_t} = \frac{-1}{2(1+c)} \left(\underbrace{\begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \ddots & \vdots \\ 1 & 1 & \ddots & \ddots & 1 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} q_{t-1}^1 \\ \vdots \\ q_{t-1}^N \end{bmatrix}}_{Q_{t-1}} - \begin{bmatrix} D \\ \vdots \\ D \end{bmatrix} \right)$$

$$Q_t = \left(\frac{-1}{2(1+c)} A\right)^t \left[Q_0 + \left(\frac{-1}{2(1+c)}\right)^2 A \left(\frac{-1}{2(1+c)} A - I_N\right)^{-1} B \right] + \frac{1}{2(1+c)} \left[\left(\frac{-1}{2(1+c)} A - I_N\right)^{-1} \right] B$$

Algorithmic collusion

The literature about **algorithmic collusion** studies the **possibility that independent agents**, modeled as independent learners, **could learn not to play the Nash-equilibrium**, i.e. what it is rational to play in a non-communication static game, and **to play an action that is closer to what they should play if they were communicating and colluding**.

Algorithmic collusion

The literature about **algorithmic collusion** studies the **possibility that independent agents**, modeled as independent learners, **could learn not to play the Nash-equilibrium**, i.e. what it is rational to play in a non-communication static game, and **to play an action that is closer to what they should play if they were communicating and colluding**.

- Waltman and Kaymak 2008 were, one of the firsts, or the firsts, to prove that Q-Learning independent agents can converge collectively toward an equilibrium that is deviating from the Nash equilibrium in the direction of the collusive one.
  This work has been replicated many times: Calvano, Calzolari, and Denicolò 2019; Asker, Fershtman, and Pakes 2022; Banchio et al. 2022; Kerzreho 2024...

- Byrne and Roos 2019 empirically demonstrate that firms can collude without necessarily forming a cartel with communication

- Byrne and Roos 2019 empirically demonstrate that firms can collude without necessarily forming a cartel with communication
- Assad et al. 2024 is the first empirical work which exhibited a link between algorithmic pricing and collusive outcome by studying the case of Germany's retail gasoline market.

- Byrne and Roos 2019 empirically demonstrate that firms can collude without necessarily forming a cartel with communication
- Assad et al. 2024 is the first empirical work which exhibited a link between algorithmic pricing and collusive outcome by studying the case of Germany's retail gasoline market.

Algorithmic collusion is also contested

Some very recent works have come to challenge these widely accepted results. Indeed, the former almost all rely on the same technology: Q-Learning (or even simpler algorithms), which is widely accepted as an outdated algorithm. Abada, Lambin, and Tchakarov 2022 state that over-simplified algorithms like Q-Learning, or not well-tuned exploration processes, could be the source of these strange results.

Definition

We consider a Markov-game (Littman 1994) composed of:

- $N \subset \mathbb{N}$ the set of agents (for simplicity, $\text{card}(N) = N \in \mathbb{N}$)
- $\mathcal{A}_{j \in N} \subset \mathbb{R}$ (finite) the set of actions available for each agent
- S ($\subset \mathbb{R}$ for simplicity) the set of all possible states
- The markovian transition function $p : S \times \left(\bigtimes_{j \in N} \mathcal{A}_j \right) \longrightarrow S$ such that $s_{t+1} = p(s_t, a_1, ..., a_N)$
- Each agent has a policy $\pi_{i \in N} : S \longrightarrow \mathcal{P}(\mathcal{A})$ where $\mathcal{P}(\mathcal{A})$ is the set of probability measures on the action space.
- At each round t, each agent observe a state s_t, select an action $(a_t^i)_{i \in N}$ and receive a reward according to the reward function $r : S \times \mathcal{A}_j \longrightarrow \mathbb{R}$.

Definition

In a Markov-game, we say that an agent is self-learning if it is associated with an algorithm whose task is to find an optimal policy, without prior knowledge on its shape.

Definition

In a Markov-game, we say that an agent is self-learning if it is associated with an algorithm whose task is to find an optimal policy, without prior knowledge on its shape.

Agents maximize their expected total reward: $R = \sum_t^{\infty} \gamma^t r(s_t, a_t)$ (with $\gamma \in [0, 1]$ a discount factor). To select their action with respect to the state they observe, they use the Q-function, which can be deduced from the Bellman equation of the problem:

$$\begin{aligned} V(s) &= \max_{a \in \mathcal{A}} \{\mathbb{E}[r(s,a)|\ s,a] + \gamma \mathbb{E}[V(s')|\ s,a]\} \\ &= \max_{a \in \mathcal{A}} \{Q(s,a)\} = \max_{a \in \mathcal{A}} \{\mathbb{E}[r(s,a)|\ s,a] + \gamma \mathbb{E}[\max_{a' \in \mathcal{A}} Q(s',a')|\ s,a]\} \end{aligned}$$

Q-Learning algorithm (Watkins 1989)

We want to estimate:

$$Q(s,a) = \underbrace{\mathbb{E}[r(s,a)|\ s,a]}_{\text{Observable}} + \gamma \underbrace{\mathbb{E}[\max_{a' \in \mathcal{A}} Q(s',a')|\ s,a]}_{\text{Unobservable}}$$

To do so, Q-Learning discretizes the state and action space, and uses a Q-Table, that is filled by an exploration process, that is by trying several combinations of states and actions to estimate $Q(s,a)$.
As $\mathbb{E}[\max_{a' \in \mathcal{A}} Q(s',a')|\ s,a]$ is unobservable, Q-Learning, to estimate Q at point (s,a), plays a at state s, observe $\mathbb{E}[r(s,a)|\ s,a]$, and estimate $\mathbb{E}[\max_{a' \in \mathcal{A}} Q(s',a')|\ s,a]$ by $\mathbb{E}[\max_{a' \in \mathcal{A}} Q(s',a')|\ s',a]$ with s' the new state after playing a.

The dangers of the Q-Learning algorithm

However, for $N > 1$,

$$\mathbb{E}[\max_{a'\in\mathcal{A}} Q(s', a')|\ s, a] \neq \mathbb{E}[\max_{a'\in\mathcal{A}} Q(s', a')|\ s, s', a]$$
$$\Leftrightarrow \mathbb{E}_{s'}[\max_{a'\in\mathcal{A}} Q(s', a')|\ s, a] \neq \max_{a'\in\mathcal{A}} Q(s', a')$$

Moreover, Q-Learning needs to discretize both $\mathcal{A}$ and S which are in our settings continuous. Not only has this discretization not been proven without consequences on the outcome, but also every increase in the quality of the discretization jeopardize the computational performance of the algorithm, limiting researchers to only use very rough approximations.

The dangers of the Q-Learning algorithm

However, for $N > 1$,

$$\mathbb{E}[\max_{a' \in \mathcal{A}} Q(s', a')|\ s, a] \neq \mathbb{E}[\max_{a' \in \mathcal{A}} Q(s', a')|\ s, s', a]$$
$$\Leftrightarrow \mathbb{E}_{s'}[\max_{a' \in \mathcal{A}} Q(s', a')|\ s, a] \neq \max_{a' \in \mathcal{A}} Q(s', a')$$

Moreover, Q-Learning needs to discretize both $\mathcal{A}$ and S which are in our settings continuous. Not only has this discretization not been proven without consequences on the outcome, but also every increase in the quality of the discretization jeopardize the computational performance of the algorithm, limiting researchers to only use very rough approximations.

Hence, new algorithms need to be used to strengthen research methodologies and results. In our case, we will use the **Deep Deterministic Policy Gradient algorithm (DDPG)**.

Objectives and Methodology

Research question 1

Is the DDPG Algorithm a relevant tool in a multi-agent setup, in particular in the case of oligopolies with competition in quantities?

A non-myopic agent is an agent whose value function has $\gamma > 0$ *in* $V(s) = \max_a \{\mathbb{E}[r(s,a) + \gamma V(s') | \; s,a]\}$

Research question 1

Is the DDPG Algorithm a relevant tool in a multi-agent setup, in particular in the case of oligopolies with competition in quantities?

A non-myopic agent is an agent whose value function has $\gamma > 0$ *in* $V(s) = \max_a \{\mathbb{E}[r(s,a) + \gamma V(s') | \ s,a]\}$

Research question 2

How does the equilibrium evolve when non-myopic agents are introduced?

Research question 1

Is the DDPG Algorithm a relevant tool in a multi-agent setup, in particular in the case of oligopolies with competition in quantities?

A non-myopic agent is an agent whose value function has $\gamma > 0$ *in* $V(s) = \max_a \{\mathbb{E}[r(s,a) + \gamma V(s') | \ s,a]\}$

Research question 2

How does the equilibrium evolve when non-myopic agents are introduced?

Research question 3

How do learning agents behave in a setting without any stable analytical solution?

Definition: Deterministic policy

In our setting, we say that the policy selected by agents is deterministic and is:

$$\mu : S \longrightarrow \mathcal{A}, \qquad s \longmapsto \text{argmax}_a Q(s, a)$$

Reminder: $Q(s,a) = \mathbb{E}[r(s,a)|\ s,a] + \gamma\mathbb{E}[\max_{a' \in \mathcal{A}} Q(s',a')|\ s,a]$

Definition: Deterministic policy

In our setting, we say that the policy selected by agents is deterministic and is:

$$\mu : S \longrightarrow \mathcal{A}, \qquad s \longmapsto \text{argmax}_a Q(s, a)$$

Reminder: $Q(s,a) = \mathbb{E}[r(s,a) |\ s,a] + \gamma \mathbb{E}[\max_{a' \in \mathcal{A}} Q(s',a') |\ s,a]$

Definition: Critic estimator

We define $Q^{\pi}_{\theta_Q}$ the critic estimator, a neural network approximation of the function Q (unobservable), parametrized by the weights vector θ_Q. The Critic neural network is fitted *via* a gradient descent on the loss function:

$$\mathcal{L}(\theta_Q) = \mathbb{E}[(Q^{\pi}(s_t, a^i_t | \theta_Q) - (r(s_t, a^i_t) + \gamma Q^{\pi}(s_{t+1}, \mu(s_{t+1}))))^2 | r, a_t, s_t, s_{t+1}]$$

Definition: Actor estimator

We define μ_{θ_π} the actor estimator, a neural network approximation of the function μ (the policy), parametrized by the weights vector θ_μ. The purpose of this estimator is to find the argmax of the critic estimator, i. e. the optimal action that maximize the critic given the observed state.

Definition: Actor estimator

We define μ_{θ_π} the actor estimator, a neural network approximation of the function μ (the policy), parametrized by the weights vector θ_μ. The purpose of this estimator is to find the argmax of the critic estimator, i. e. the optimal action that maximize the critic given the observed state.

Deterministic policy gradient theorem

To fit the actor estimator, we perform a gradient ascent, but as the objective function is itself an estimator, we use the *Deterministic policy gradient theorem* introduced by Silver et al. 2014:

$$\nabla_{\theta_\pi} J^\pi = \int_{\mathcal{S}} \rho(s) \nabla_{\theta_\pi} \mu(s|\theta_\pi) \nabla_a Q^\pi(s, a|\theta_Q)|_{a=\mu(s|\theta_\pi)} ds$$

with $\rho(s)$ a discounted state distribution factor made accordingly to our markovian transition function p.

Definition: Target networks

Following Mnih et al. 2013, we define target networks, or target estimators, as lagged versions of the actor and critic estimators, that are used in every training steps. The objective of this adjustment is to avoid circular references: without them, we would fit the actor using the critic which is itself fitted using the actor network.

$$\begin{cases} \theta_\pi^{\text{target}} = \tau\theta_\pi + (1-\tau)\theta_\pi^{\text{target}} \\ \theta_Q^{\text{target}} = \tau\theta_Q + (1-\tau)\theta_Q^{\text{target}} \end{cases} \quad \text{with } \tau \in [0,1]$$

Definition: Target networks

Following Mnih et al. 2013, we define target networks, or target estimators, as lagged versions of the actor and critic estimators, that are used in every training steps. The objective of this adjustment is to avoid circular references: without them, we would fit the actor using the critic which is itself fitted using the actor network.

$$\begin{cases} \theta_\pi^{\text{target}} = \tau\theta_\pi + (1-\tau)\theta_\pi^{\text{target}} \\ \theta_Q^{\text{target}} = \tau\theta_Q + (1-\tau)\theta_Q^{\text{target}} \end{cases} \quad \text{with } \tau \in [0,1]$$

Definition: Replay buffer

Following Lillicrap et al. 2015, a replay buffer ($\mathcal{B}$) is the computational set of all previous experiences $(s_t, r_t^i, a_t^1, ..., a_t^N, s_{t+1})$. This set is used to provide training data for neural networks estimators.

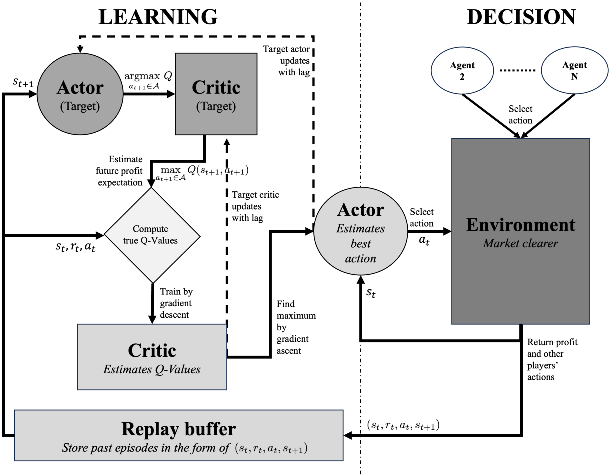


Figure 2: **A summary of our algorithm design**

Definition

All agents are firms with the same quadratic cost function $C_i(q_i) = cq_i^2$ ($c \in \mathbb{R}_+$) on the same market with a linear demand $D(Q) = D - Q$ ($D \in \mathbb{R}_+$). Competition is in quantities. Agents are in incomplete information: they only observe the total quantity produced (Q) and the price and have no information on their competitors (neither their number nor their cost structures).

Definition

All agents are firms with the same quadratic cost function $C_i(q_i) = cq_i^2$ ($c \in \mathbb{R}_+$) on the same market with a linear demand $D(Q) = D - Q$ ($D \in \mathbb{R}_+$). Competition is in quantities. Agents are in incomplete information: they only observe the total quantity produced (Q) and the price and have no information on their competitors (neither their number nor their cost structures).

Proposition

In this setting, in the Cournot N-oligopoly, there exists a stable and attractive equilibrium iff $\frac{N-3}{2} < c$ with N the number of firms. The equilibrium point is the same as the one of the static Cournot game.

Simulation program:

- First, to challenge the relevance of the DDPG algorithm (**Q1**) we conduct two types of simulations: a Cournot duopoly, and a Stackelberg duopoly with fully myopic agents.

Simulation program:

- First, to challenge the relevance of the DDPG algorithm (**Q1**) we conduct two types of simulations: a Cournot duopoly, and a Stackelberg duopoly with fully myopic agents.
- Second, we introduce non-myopic agents in the Cournot duopoly to evaluate the effect on the equilibrium (**Q2**).

Simulation program:

- First, to challenge the relevance of the DDPG algorithm (**Q1**) we conduct two types of simulations: a Cournot duopoly, and a Stackelberg duopoly with fully myopic agents.
- Second, we introduce non-myopic agents in the Cournot duopoly to evaluate the effect on the equilibrium (**Q2**).
- Last, we will focus on the Cournot 4-oligopoly: we will try to simulate chaotic cases (**Q3**), and non-myopic agents (**Q2**).

Results

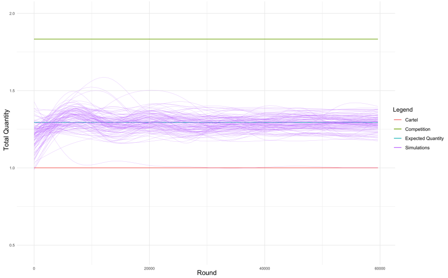
We begin by the Cournot duopoly, as it features a stable analytical solution and a unique equilibrium. We keep $\gamma = 0$ (fully myopic agents) and use as parameters $D = 2.2$, $c = .2$ and compute $q_C^* = .647, Q_C^* = 1.294$.

We begin by the Cournot duopoly, as it features a stable analytical solution and a unique equilibrium. We keep $\gamma = 0$ (fully myopic agents) and use as parameters $D = 2.2$, $c = .2$ and compute $q_C^* = .647, Q_C^* = 1.294$.

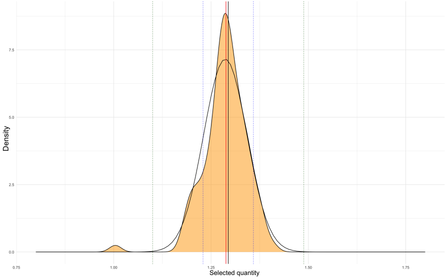
	Global dispersion			Shapiro test		Distribution	
Sample	$Q_C^* \pm 5\%$	$Q_C^* \pm 10\%$	$Q_C^* \pm 15\%$	W	p-value	$\bar{x}$	$\hat{s}$
Uncorrected	0.794	0.991	0.991	0.932	3.9e-05	1.29	0.056
Without outlier	0.802	1	1	0.990	0.61	1.29	0.049

Table 1: **Descriptive statistics of the simulation sample of the Cournot duopoly** (107 simulations conducted with $m = 2000$, $\varsigma = .15$, $\gamma = 0$).
79.4% of simulations have converged toward an equilibrium that is in the interval $Q_C^* \pm 5\%$ with Q_C^* the Cournot analytical total solution.

Performance of our DDPG algorithm in standard games with myopic agents



(a) **Evolution of the total quantity ($q_t^1 + q_t^2$).** The Cournot predicted quantity is in blue, the cartel equilibrium in red and the perfect competition equilibrium in green.



(b) **Distribution of the total quantity ($q_t^1 + q_t^2$) at convergence.** In black a normal law with the distribution's parameters (μ, σ) and the Cournot expected quantity; in red the median of simulations; in blue the 5% confidence interval; in green the 15%; in orange the density function of $(q_t^1 + q_t^2)$.

Figure 3: **Quantity chosen by the learning agents in a Cournot duopoly** (107 simulations conducted with $m = 2000$, $\varsigma = .15$, $\gamma = 0$)

Performance of our DDPG algorithm in standard games with myopic agents

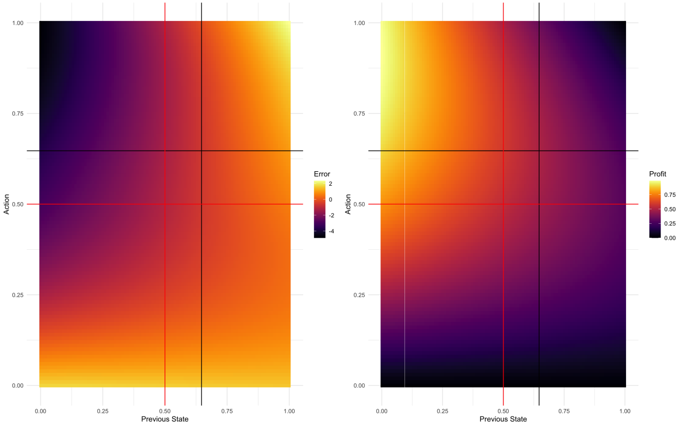


Figure 4: **Estimated critic of agents in a Cournot duopoly** (subsample of 15 critics over 107 simulations conducted with $m = 2000$, $\varsigma = .15$, $\gamma = 0$).
Black lines are the Cournot equilibrium, red lines are the Cartel equilibrium. On the left, the heat-map represents the average estimation error (standardized), whereas the right heat-map reminds profit values (the previous state being the previous quantity selected by the opponent).

We now try to use our DDPG Algorithm in the Stackelberg framework. This model is more challenging than the Cournot one as one agent has to play first. The results of our simulations are to some extent disappointing.

We now try to use our DDPG Algorithm in the Stackelberg framework. This model is more challenging than the Cournot one as one agent has to play first. The results of our simulations are to some extent disappointing.

		Dispersion			Shapiro Test		Distribution	
Sample	q_S^*	$q_S^* \pm 5\%$	$q_S^* \pm 10\%$	$q_S^* \pm 15\%$	W	p-value	$\bar{x}$	$\hat{s}$
Follower	0.568	0.214	0.536	0.726	0.9891	0.708	0.541	0.077
Leader	0.837	0.202	0.512	0.619	0.961	0.013	0.795	0.107
Total quantity	1.405	0.536	0.821	0.940	0.978	0.159	1.337	0.083

Table 2: **Implementation performance of the DDPG Algorithm in the Stackelberg duopoly** (84 simulations, $m = 500$, $\gamma = 0$, $\varsigma = .3$).
72.6% of simulations have converged toward an equilibrium where the follower has chosen its quantity in the interval $q_S^* \pm 5\%$ with q_S^* the Stackelberg analytical solution.

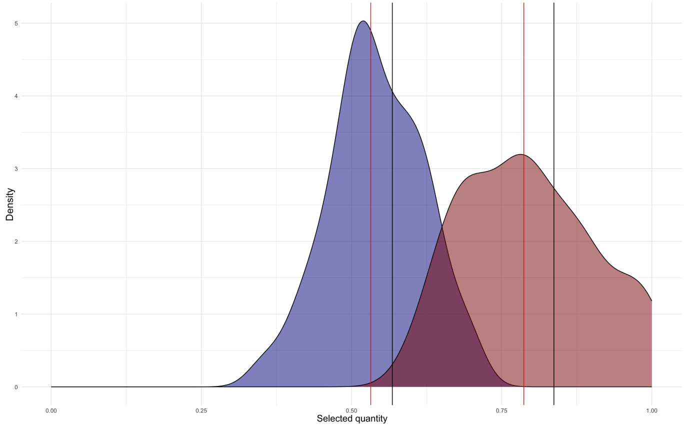


Figure 5: **Distribution of selected quantities in the Stackelberg duopoly** (with the follower in blue and the leader in red, 107 simulations conducted with $m = 500$, $\varsigma = .3$, $\gamma = 0$)
Black lines are the analytical optimal quantities, red lines are the median of observed distributions.

Alteration of the Q-function

We slightly edit the Q-function (derived from the Bellman equation) to make it a convex combination with $\gamma \in [0, 1]$.

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim E}[(1-\gamma) r(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, \mu(s_{t+1})) | s_t, a_t]$$

What are the effects of an increase of the γ parameter on the outcome ?

Alteration of the Q-function

We slightly edit the Q-function (derived from the Bellman equation) to make it a convex combination with $\gamma \in [0, 1]$.

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{s_{t+1}\sim E}[(1-\gamma)r(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, \mu(s_{t+1}))|s_t, a_t]$$

What are the effects of an increase of the γ parameter on the outcome ?

Definition: Δ-score

Following the work of Calvano, Calzolari, and Denicolò 2019, we assess the collusiveness of the equilibrium by evaluating the deviation from the Nash-equilibrium:

$$\Delta = \frac{\bar{\Pi} - \Pi^*_{\text{Cournot}}}{\Pi^*_{\text{Cartel}} - \Pi^*_{\text{Cournot}}}$$

Sample			Inter-dispersion $Q^*_{\text{Car.}}$	$Q^*_{\text{Car.}}$	$Q^*_{\text{Car.}}$		Collusion		t-test p-value		
γ	m	T	$\pm 5\%$	$\pm 10\%$	$\pm 15\%$	$\bar{\Pi}$	$\bar{\delta}$	$\bar{\Delta}$	$\bar{\Delta} > \bar{\Delta}_{\gamma=0}$	$\bar{\Delta} > 0$	$\#\mathcal{S}$
0	1000	9×10^4	0.01	0.05	0.12	1.005	0.939	0.214	-	0.001	74
0.1	1000	9×10^4	0.03	0.03	0.03	1.008	0.946	0.252	0.356	0.003	39
0.2	1000	9×10^4	0	0.04	0.09	0.992	0.96	0.074	0.903	0.399	54
0.3	1000	9×10^4	0	0	0	1.012	0.947	0.292	0.207	0	36
0.4	1000	9×10^4	0	0	0.02	1.004	0.965	0.212	0.511	0.005	48
0.5	1000	9×10^4	0.03	0.07	0.09	1.009	0.973	0.261	0.322	0.001	58
0.6	1000	9×10^4	0	0	0.13	1.006	0.979	0.225	0.471	0.104	30
0.7	1000	9×10^4	0.02	0.02	0.05	0.981	0.985	-0.042	0.978	0.696	55
0.8	1000	9×10^4	0	0.03	0.09	0.997	0.995	0.13	0.758	0.211	34
0.5	2000	9×10^4	0	0	0.11	0.992	0.967	0.077	0.904	0.357	72
0.5	3000	9×10^4	0	0	0	0.996	0.978	0.122	0.794	0.195	29
0	500	6×10^4	0	0	0	0.982	0.952	-0.028	0.984	0.763	72
0	1000	6×10^4	0.01	0.02	0.03	0.994	0.953	0.099	0.928	0.034	150

Table 3: **Counterfactual analysis of the effects of non-myopic agents with our DDPG Algorithm** ($\varsigma = .3$, 751 simulations).
The two last lines are implemented as a reminder, to assess the effect of the increase of the simulation length.

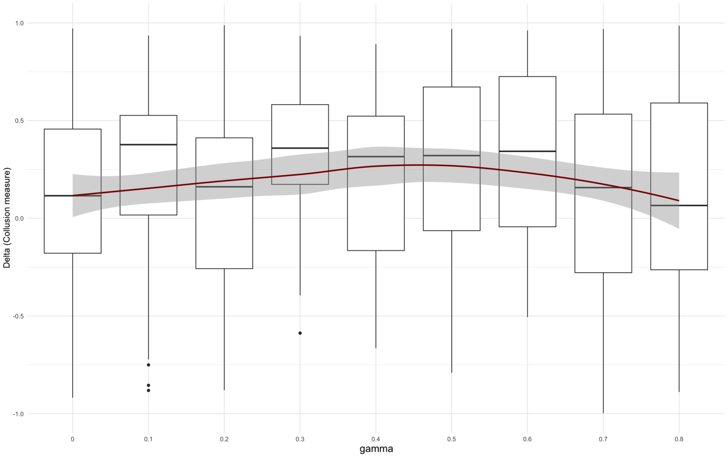


Figure 6: **The relation between the γ parameter and the Δ-score** (518 simulations conducted with $m = 1000$, $\varsigma = .3$)

If we cannot prove a link between γ (the preference for future) and the collusive outcome, how can we explain collusion ?

If we cannot prove a link between γ (the preference for future) and the collusive outcome, how can we explain collusion ?

We find that the increase of the simulation duration and of the memory buffer size has caused the statistically significant increase of the Δ-score. Interestingly, if we keep increasing m above $m = 1000$, collusion decreases. **This suggests that collusion is only possible with rare tuples of values for (m, T, ω) and is in fact quite rare.**

Can we explain how algorithms are shifting towards the collusive behavior?

Definition: δ-score

For this purpose, we have created a new measure of the relative valuation of collusion for an agent:

$$\delta = \frac{\displaystyle\iint_{B(q^*_{Ca.}, 0.15 q^*_{Ca.})^2} Q^\pi(s_t, a_t) ds_t da_t}{\displaystyle\iint_{B(q^*_{Co.}, 0.15 q^*_{Co.})^2} Q^\pi(s_t, a_t) ds_t da_t}$$

We introduce two linear models:

$$\min(\delta_i, \delta_{-i}) = \beta_0^f + \beta_1^f \max(\delta_i, \delta_{-i}) + \beta_2^f \gamma + \epsilon^f \quad (1)$$

$$a_i = \beta_0^S + \beta_1^S \delta_i + \beta_2^S \delta_{-i} + \epsilon^S \quad (2)$$

We introduce two linear models:

$$\min(\delta_i, \delta_{-i}) = \beta_0^f + \beta_1^f \max(\delta_i, \delta_{-i}) + \beta_2^f \gamma + \epsilon^f \tag{1}$$

$$a_i = \beta_0^S + \beta_1^S \delta_i + \beta_2^S \delta_{-i} + \epsilon^S \tag{2}$$

These two models allow us to evaluate the global effect of these valuations on the total equilibrium. If we assume without any loss of generality that $\delta_i \leqslant \delta_{-i}$:

$$\begin{cases} a_i = \beta_0^S + \beta_1^S \left(\beta_0^f + \beta_1^f \delta_{-i} + \beta_2^f \gamma + \epsilon^f\right) + \beta_2^S \delta_{-i} + \epsilon^S \\ a_{-i} = \beta_0^S + \beta_1^S \delta_{-i} + \beta_2^S \left(\beta_0^f + \beta_1^f \delta_{-i} + \beta_2^f \gamma + \epsilon^f\right) + \epsilon^S \end{cases} \tag{3}$$

$$\Rightarrow a_i + a_{-i} = 2\beta_0^S + (\beta_1^S + \beta_2^S)\beta_0^f + (\beta_1^S + \beta_2^S)(1 + \beta_1^f)\delta_{-i} + \beta_2^f(\beta_1^S + \beta_2^S)\gamma + (\beta_1^S + \beta_2^S)\epsilon^f + 2\epsilon^S \tag{4}$$

For the sake of simplicity and clarity, first stage estimations are not presented here.

	Estimate	t. value	p-value
(cste)	0.425 (0.175)	2.43	0.016 *
δ_i	-1.139 (0.142)	-8.04	$1.22e^{-13}$ ***
δ_{-i}	1.398 (0.142)	9.86	$< 2e^{-16}$ ***

(a) Zero-collusion group (182 observations, $R_a^2 = .43$)

	Estimate	t. value	p-value
(cste)	0.478 (0.089)	5.35	$1.17e^{-07}$ ***
δ_i	-1.107 (0.07)	-15.84	$< 2e^{-16}$ ***
δ_{-i}	1.275 (0.07)	18.25	$< 2e^{-16}$ ***

(b) Collusion group (788 observations, $R_a^2 = .4$)

Table 4: **Second stage regression to link a_i with (δ_i, δ_{-i}).**

For the sake of simplicity and clarity, first stage estimations are not presented here.

	Estimate	t. value	p-value
(cste)	0.425 (0.175)	2.43	0.016 *
δ_i	-1.139 (0.142)	-8.04	$1.22e^{-13}$ ***
δ_{-i}	1.398 (0.142)	9.86	$< 2e^{-16}$ ***

(a) Zero-collusion group (182 observations, $R_a^2 = .43$)

	Estimate	t. value	p-value
(cste)	0.478 (0.089)	5.35	$1.17e^{-07}$ ***
δ_i	-1.107 (0.07)	-15.84	$< 2e^{-16}$ ***
δ_{-i}	1.275 (0.07)	18.25	$< 2e^{-16}$ ***

(b) Collusion group (788 observations, $R_a^2 = .4$)

Table 4: **Second stage regression to link a_i with (δ_i, δ_{-i}).**

We recover a famous result in game theory: cooperative equilibria can have two opposite effects on the decision of the player. First, they allow it to increase its profit, what incentivizes it to lower its quantity. Second, they are an opportunity to increase its profit by taking advantage of the other player.

In this section, our goal is to study the possible shifts in behaviors that can be observed when our markets are populated by more than two firms. We consider three settings:

- We use $D = 5$, $\eta_\pi = .25$, $H_\pi = .15$, with $c = .6$ for the "stable" case, with $\gamma = 0$ for the fully-myopic case, and $\gamma = .5$ for the collusive case. The equilibrium tuple is (0.962, 0.289, 0.806, 0.410, 0.543, 0.489).
- We use $D = 5$, $\eta_\pi = .25$, $H_\pi = .15$, with $c = .5$ for the "unstable" case. The equilibrium tuple is (1, 0.275, 0.833, 0.410, 0.556, 0.497).

In this section, our goal is to study the possible shifts in behaviors that can be observed when our markets are populated by more than two firms. We consider three settings:

- We use $D = 5$, $\eta_\pi = .25$, $H_\pi = .15$, with $c = .6$ for the "stable" case, with $\gamma = 0$ for the fully-myopic case, and $\gamma = .5$ for the collusive case. The equilibrium tuple is (0.962, 0.289, 0.806, 0.410, 0.543, 0.489).
- We use $D = 5$, $\eta_\pi = .25$, $H_\pi = .15$, with $c = .5$ for the "unstable" case. The equilibrium tuple is (1, 0.275, 0.833, 0.410, 0.556, 0.497).

Definition: Intra-dispersion score

We create an intra-dispersion score that assesses the deviation between quantities selected by firms:

$$\check{S} = \frac{1}{\text{card}(\mathcal{S})} \sum_{s \in \mathcal{S}} \max_{i,j \in \mathcal{A}} \left(\frac{1}{T + 1 - .98 \cdot T} \sum_{t \in [\![.98 \cdot T, T]\!]} ||q_t^i - q_t^j|| \right)$$

Sample			Inter-dispersion			Distribution		Intra-dispersion			
c	γ	m	$Q^*_{Co.}$ $\pm 5\%$	$Q^*_{Co.}$ $\pm 10\%$	Shapiro p-value	$\bar{Q}$	sd(Q)	mean$(\check{S})$	sd$(\check{S})$	$\bar{\Delta}$	$\#\mathcal{S}$
0.6	0	500	0.94	1	0.001	3.20	0.089	0.232	0.093	0.028	113
0.6	0	1000	0.90	1	0.001	3.15	0.079	0.193	0.086	0.120	97
0.6	0	2000	0.93	1	0.296	3.16	0.071	0.185	0.070	0.117	43
0.6	0.5	500	0.68	0.92	0.008	3.33	0.152	0.280	0.101	-0.258	84
0.6	0.5	1000	0.86	0.99	0.057	3.25	0.107	0.255	0.091	-0.081	83
0.5	0	500	0.94	1	0.122	3.34	0.096	0.240	0.083	-0.025	105
0.5	0	1000	1	1	0.230	3.31	0.074	0.203	0.070	0.034	30

Table 5: **Implementation performance of the DDPG Algorithm in the Cournot 4-oligopoly.**

On the matter of unstable equilibria, our model seems not to be affected by the chaotic nature of the analytical equilibrium: we find that our model has converged toward the same mean (after correcting for the effect of the change in c) with a p-value of .022. These results are corroborated by the Kolmogorov-Smirnov test, that gives us a p-value of .8352, suggesting that the underlying distribution is the same for both samples.

Discussion and conclusion

- Our main contribution has been to introduce the first agent-based model of competition in quantities featuring a *Deep Deterministic Policy Gradient* (DDPG) algorithm. This algorithm has been selected as a replacement for the traditional Q-Learning algorithm that impose dramatic simplifications.

- Our main contribution has been to introduce the first agent-based model of competition in quantities featuring a *Deep Deterministic Policy Gradient* (DDPG) algorithm. This algorithm has been selected as a replacement for the traditional Q-Learning algorithm that impose dramatic simplifications.
- Overall, our results tend to support the thesis developed by Abada, Lambin, and Tchakarov 2022: collusive outcomes have been very rare in our settings, and appear to be an exception caused by well-chosen parameters.

- Our main contribution has been to introduce the first agent-based model of competition in quantities featuring a *Deep Deterministic Policy Gradient* (DDPG) algorithm. This algorithm has been selected as a replacement for the traditional Q-Learning algorithm that impose dramatic simplifications.
- Overall, our results tend to support the thesis developed by Abada, Lambin, and Tchakarov 2022: collusive outcomes have been very rare in our settings, and appear to be an exception caused by well-chosen parameters.
- We do not find any reliable link between the training length and the γ parameter and collusive behaviors, nor that we can find any punishment behaviors as Calvano, Calzolari, and Denicolò 2019.

- Our main contribution has been to introduce the first agent-based model of competition in quantities featuring a *Deep Deterministic Policy Gradient* (DDPG) algorithm. This algorithm has been selected as a replacement for the traditional Q-Learning algorithm that impose dramatic simplifications.
- Overall, our results tend to support the thesis developed by Abada, Lambin, and Tchakarov 2022: collusive outcomes have been very rare in our settings, and appear to be an exception caused by well-chosen parameters.
- We do not find any reliable link between the training length and the γ parameter and collusive behaviors, nor that we can find any punishment behaviors as Calvano, Calzolari, and Denicolò 2019.
- Following the work of Abada, Lambin, and Tchakarov 2022, we do not find any reason to adjust anti-trust policies, as more sophisticated algorithms seem more likely to serve rational competition than collusion

Avenue for future works

Despite our efforts to implement fully decentralized learning algorithms, we have only tested fully identical algorithms, without pre-training and without temporal differences (all firms join the market at the same time): implementing heterogeneity, and question its implications on the obtained equilibrium, could make a very interesting work to allow these theoretical results to be more useful for policy-makers.

- Our artificial markets with self-learning agents are not significantly affected by market situations that are analytically unstable.

- Our artificial markets with self-learning agents are not significantly affected by market situations that are analytically unstable.
- Worse, we find no element that could corroborate the best response adjustment process described by Cournot 1838 and studied analytically by Theocharis 1960 and his followers. We observe, as expected with a DDPG algorithm, a slow convergence toward a stable value, and not a converging (or diverging) oscillation around an equilibrium.

- Our artificial markets with self-learning agents are not significantly affected by market situations that are analytically unstable.
- Worse, we find no element that could corroborate the best response adjustment process described by Cournot 1838 and studied analytically by Theocharis 1960 and his followers. We observe, as expected with a DDPG algorithm, a slow convergence toward a stable value, and not a converging (or diverging) oscillation around an equilibrium.

These results are confusing as they raise questions about the validity of the Cournot adjustment behavior.

References I

[1] Ibrahim Abada, Xavier Lambin, and Nikolay Tchakarov. "Collusion by Mistake: Does Algorithmic Sophistication Drive Supra-Competitive Profits?" In: *Social Science Research Network* (2022). DOI: 10.2139/ssrn.4099361.

[2] Hamdy N. Agiza and Abdelalim A. Elsadany. "Nonlinear dynamics in the Cournot duopoly game with heterogeneous players". In: *Physica A: Statistical Mechanics and its Applications* 320 (2003), pp. 512–524. ISSN: 0378-4371. DOI: 10.1016/S0378-4371(02)01648-5.

[3] Hamdy N. Agiza and Abdelalim A. Elsadany. "Chaotic dynamics in nonlinear duopoly game with heterogeneous players". In: *Applied Mathematics and Computation* (2004). DOI: 10.1016/s0096-3003(03)00190-5.

References II

[4] John Asker, Chaim Fershtman, and Ariél Pakes. "Artificial Intelligence, Algorithm Design, and Pricing". In: *AEA papers and proceedings* (2022). DOI: 10.1257/pandp.20221059.

[5] Stephanie Assad et al. "Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market". In: *Journal of Political Economy* 132.3 (2024), pp. 723–771. DOI: 10.1086/726906.

[6] Martino Banchio et al. *Adaptive Algorithms, Tacit Collusion, and Design for Competition*. 2022. DOI: 10.48550/arXiv.2202.05946.

[7] David P. Byrne and Nicolas de Roos. "Learning to Coordinate: A Study in Retail Gasoline". In: *American Economic Review* 109.2 (Feb. 2019), pp. 591–619. DOI: 10.1257/aer.20170116.

References III

[8] Emilio Calvano, Giacomo Calzolari, and Vincenzo Denicolò. "Artificial Intelligence, Algorithmic Pricing, and Collusion". In: *American Economic Review* (2019). DOI: 10.2139/ssrn.3304991.

[9] Antoine Augustin Cournot. *Recherches sur les Principes Mathématiques de la Théorie des Richesses*. Économistes. Paris: Hachette, 1838.

[10] Cars Hommes, Marius I. Ochea, and Jan Tuinstra. *On the Stability of the Cournot Equilibrium: An Evolutionary Approach*. 2011. DOI: 11245/1.354907.

[11] Yann Kerzreho. "Spontaneous Collusion with Synchronous Learning and States". MA thesis. Ecole Normale Supérieure Paris-Saclay, 2024.

References IV

[12] Timothy P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2015. DOI: 10.48550/arXiv.1509.02971.

[13] Michael L. Littman. "Markov games as a framework for multi-agent reinforcement learning". In: *Proceedings of the International Conference on Machine Learning*. 1994. DOI: 10.1016/b978-1-55860-335-6.50027-1.

[14] Volodymyr Mnih et al. *Playing Atari with Deep Reinforcement Learning*. 2013. DOI: 10.48550/arXiv.1312.5602.

[15] Tönu Puu. "On the stability of Cournot equilibrium when the number of competitors increases". In: *Journal of Economic Behavior and Organization* (2008). DOI: 10.1016/j.jebo.2006.06.010.

References V

[16] David Silver et al. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the International Conference on Machine Learning*. June 2014.

[17] Reghinos D. Theocharis. "On the Stability of the Cournot Solution on the Oligopoly Problem". In: *The Review of Economic Studies* (1960). DOI: 10.2307/2296135.

[18] Ludo Waltman and Uzay Kaymak. "Q-learning agents in a Cournot oligopoly model". In: *Journal of Economic Dynamics and Control* (2008). DOI: 10.1016/j.jedc.2008.01.003.

[19] Chris Watkins. "Learning from delayed rewards". PhD thesis. 1989. URL: https://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf.