Will Algorithms Learn to Collude?

Insights from the Natural Policy Gradient Method

by

Yanwen Wang

Submitted to the Distinguished Majors Program
Department of Economics
University of Virginia
April 15, 2022
Advisor: Federico Ciliberto

Will Algorithms Learn to Collude?

Insights from the Natural Policy Gradient Method

Yanwen Wang *

Abstract

Strategic processes are increasingly delegated to algorithms. The autonomous nature of decision-making algorithms gives rise to concerns that algorithms might learn to collude in the absence of any external guidance and communication. To measure the risk of algorithmic collusion, I use the Natural Policy Gradient method with a Gaussian policy to simulate the behaviors of algorithms in simultaneous Bertrand and Cournot duopoly and oligopoly environments. I show that the Natural Policy Gradient method converges to Nash equilibrium in all markets simulated and is less prone to algorithmic collusion. To the best of my knowledge, this is the first study of algorithmic collusion in continuous space while providing evidence that algorithms can learn to reach competitive results in certain contexts. The convergence to Nash equilibrium is robust to asymmetries in marginal costs and changes in demand functions.

*I am grateful for invaluable guidance, advice, and support from my advisor, Professor Federico Ciliberto. I would also like to thank Professor Gaurub Aryal for inspiring me to explore this area as well as Professor Amalia Miller for insightful suggestions and comments. Finally, I wish to thank my fellow DMP cohort for helpful discussions in the early stage of this study. All errors remain my own.

1 Introduction

With the advancement of technology and the wide application of artificial intelligence, an increasing number of firms have started using artificial intelligence algorithms to optimize their selling strategies. Indeed, algorithms help firms find optimal strategies, make timely adjustments to market changes, and enhance decision-making efficiency. However, some real-world cases and studies have suggested that algorithmic decision-making would lead to supra-competitive results in a coordinated fashion (U.S. Department of Justice, 2016; Assad et al., 2020; Brown and MacKay, 2021). While algorithms intentionally designed to achieve price-fixing or other collusive behaviors are less of a concern, the tendency for self-learning algorithms, which are programmed to maximize profits, to arrive at supra-competitive results and sustain collusive outcomes is the main problem.[1] The convergence to collusive results inhibits competition, harms consumer welfare, and gives rise to a challenging antitrust issue as tacit algorithmic collusion escapes scrutiny from antitrust enforcers who mainly target explicit agreement among competitors and require evidence that tends to rule out the possibility of independent actions (Harrington, 2018; Ezrachi and Stucke, 2020). There will be a gap between current antitrust policy and tacit algorithmic collusion because of the independence between algorithms, the detachment between programmers and the behaviors of algorithms, the unfamiliarity of underlying mechanisms, and the difficulty of detecting collusive evidence. Thus, will algorithmic collusion actually happen and how real the risk of algorithmic collusion are questions of economists' and policymakers' investigations.

[1] See Ezrachi and Stucke (2017) for detailed categorization of algorithmic collusion.

Related Literature

This paper investigates the impact of decision-making algorithms on equilibrium outcomes in Bertrand and Cournot competitions and contributes to the growing literature on decision-making algorithms and algorithmic collusion. In the economics literature, researchers have mainly relied on empirical, theoretical, and experimental approaches to assess the risk of algorithmic collusion. Some focus on one particular market that has adopted algorithmic pricing years ago and try to find empirical evidence of the impact of adopting algorithmic pricing. Some develop new frameworks that capture features of algorithmic sellers to explain the underlying mechanisms behind the results. Some construct artificial intelligence algorithms to simulate possible outcomes learned by algorithms. The following two examples present the results of recent studies on algorithmic collusion that use empirical and theoretical approaches.

Assad et al. (2020) focus on the German retail gasoline market, where algorithmic pricing has been widely adopted since 2017. Using price data for every German gas station from 2014 to 2019, they compare the retail margins of adopting and non-adopting gas stations through regressions and show a 9% increase in the mean station-level margins after adoption. By examining the timing of adoption effects, Assad et al. suggest margins gradually increased after a year of adoption and conclude that algorithms learned to collude tacitly.

In a recent research paper, Brown and MacKay (2021) also find evidence of supra-competitive prices in e-commerce markets. They first collect hourly data from online retailers and discover that online retailers update prices at regular intervals, and retailers with faster pricing technology react to the price changes of retailers with slower

pricing technology. Based on the patterns they detect, Brown and MacKay construct a new theoretical model that incorporates pricing technology to explain the behaviors of algorithmic sellers. They demonstrate that the asymmetry in pricing technology enables firms to charge higher equilibrium prices. In cases where firms have different pricing frequencies, they prove that the equilibrium prices fall on the faster firm's Bertrand best-response function, which is between the Bertrand and Stackelberg equilibrium.

It is not common to have sufficient data of markets that have adopted decision-making algorithms. Theoretical studies are often built on simplified assumptions of the dynamic market. Thus, another strand of the literature takes an experimental approach to simulate stochastic markets and observes the interactions between algorithms. Using Q-learning, a model-free value-based reinforcement learning algorithm, to simulate Bertrand and Cournot environments, researchers have shown that through self-learning, decision-making algorithms could independently arrive at collusive results without explicit human design and sustain supra-competitive behaviors using reward and punishment strategies. For instance, Calvano et al. (2020) construct Q-learning algorithms that consist of multiple pricing agents and allow them to interact in repeated Bertrand duopoly and oligopoly environments. Abada and Lambin (2020) simulate the environment of an electricity market and implement Q-learning algorithms in repeated Cournot games, and Klein (2021) focuses on sequential pricing competitions. However, several drawbacks of the Q-learning algorithm, such as slow learning process, no theoretical convergence guarantees, and applicability to only discrete space, make the algorithm hard to be applied in real-world scenarios. In the study of Calvano et al. (2020), Q-learning takes around 850,000 iterations, which means more than three

years of learning in real-time, to find the optimal price (Hettich, 2021). In all their simulations, convergence guarantees that exist for single-agent Q-learning do not hold for multi-agent Q-learning since the multi-agent environment is no longer stationary. Also, Q-learning can only study discrete space while price and quantity are continuous in actual situations. As Q-learning only represents a part of the artificial intelligence algorithms, there is still great research potential for further improvement in the choice of algorithms that better accord with real-world scenarios. Hettich (2021) has taken the first step in using a more realistic algorithm. Hettich extends the experimental design of Calvano et al. (2020) using Deep Q-Network that lies in the same value-based category as Q-learning. Though Deep Q-Network still focuses on discrete space, the algorithm has a much faster convergence rate to collusive prices.

In addition to assessing the risk of algorithmic collusion, several papers also propose possible solutions to avoid collusive outcomes and increase competition in marketplaces where decision-making algorithms are commonly adopted. From an artificial intelligence perspective, Abada and Lambin (2020) suggest that policymakers may regulate the market by requiring training to be performed at the individual level instead of at the aggregator level. They also propose that regulators could intervene in the learning process of algorithms by introducing agents that aim to maximize social welfare or consumer welfare and guide other agents towards socially desirable outcomes. Using their Q-learning algorithms, they prove that agents could learn to avoid collusive prices when welfare-oriented agents are in the market. From a platform design perspective, Johnson et al. (2020) propose two platform design rules, price-directed prominence and dynamic price-directed prominence, aiming to steer demand towards sellers with lower prices. These suggestions all provide unique insights.

Contribution

The literature on decision-making algorithms is expanding. This paper contributes to the experimental branch, extending the results of prior literature using reinforcement learning algorithms that better accord with real scenarios. Instead of using the value-based Q-learning algorithm as Calvano et al. (2020), Abada and Lambin (2020), and Klein (2021), I build multi-agent algorithms that rely on the Natural Policy Gradient method, a policy-based reinforcement learning algorithm, to simulate simultaneous Bertrand and Cournot games with continuous action and state space in different market structures. This paper discusses the implementations of the Natural Policy Gradient method and shows that algorithms can learn to converge to Bertrand-Nash or Cournot-Nash equilibria in simulated markets. This result resonates with the theoretical analysis of Hambly et al. (2021), which proves the convergence to Nash equilibrium in general-sum multi-agent policy gradient dynamics with a certain level of noise, and echos the study of Shi and Zhang (2020) that shows policy gradient dynamics converge to Nash equilibrium in concave Cournot games with either two players or a linear price function.

To the best of my knowledge, this paper would be the first study of algorithmic pricing and decision-making algorithms in continuous space while providing evidence that algorithms can learn to reach competitive outcomes in certain contexts. The exceptions in algorithmic collusion would provide useful insights into designing regulatory policies for decision-making algorithms.

The remaining part of the paper proceeds as follows. Section two introduces the background of reinforcement learning and the Natural Policy Gradient method. Sec-

tion three goes on to the experimental design of simulating simultaneous Bertrand and Cournot competitions. Section four presents the impact of the Natural Policy Gradient method on equilibrium outcomes and discusses robustness checks. Section five concludes with a discussion of the study.

2 Background

This paper adopts the policy gradient method, which is a type of reinforcement learning algorithm, to simulate Bertrand and Cournot competitions for the following reasons. First, policy gradient methods can model the environment in continuous state and action spaces. As price and quantity are continuous variables, policy gradient methods are more appropriate than value-based reinforcement learning algorithms, such as Q-learning and Deep Q-Network that prior researchers used in their studies, which need to discretize the continuous action space to enumerate possible actions. Second, policy gradient methods are more efficient in large state and action spaces as they directly optimize policy, while the classic Q-learning indirectly extracts policy after estimating the expected value of taking each action. Third, policy gradient methods have proven to be useful and successful in a range of real-world applications (Peters and Schaal, 2006; Amari, 1998) and also served as the foundation of one of the most popular and the state of the art reinforcement learning frameworks, the Actor-Critic methods.

There are several reasons for choosing to use the Natural Policy Gradient method. First, it achieves better performance and a faster convergence rate compared with the Vanilla Policy Gradient (Kakade, 2001). Second, it is among the most widely-used policy optimization algorithms. Other mainstreams algorithms, such as Trust Region

Policy Optimization and Proximal Policy Optimization, are generations and variants of the Natural Policy Gradient method (Cen et al., 2021).

This paper discusses the implementations of policy-based Natural Policy Gradient in simultaneous Bertrand and Cournot competitions, showing the impact of the Natural Policy Gradient method on equilibrium outcomes if all firms in the market adopt the same algorithm to set price or quantity and do not intervene in the algorithmic decision-making process. The rest of this section contains a brief introduction to reinforcement learning in general and the main categories of reinforcement learning, including value-based and policy-based methods. The difference between these two categories will become clear shortly.

2.1 Reinforcement Learning

Following the definition of Russell and Norvig (2020), a Markov Decision Process (MDP) is a sequential decision problem for a fully observable stochastic environment with a Markovian transition model and additive rewards. An MDP consists of a single agent, a set of states $s_t \in S$ that represent configurations of the environment, a set of actions $a_t \in A(s_t)$ that the agent can select, a transition function $T(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, a_t)$ that maps s_t, a_t to s_{t+1}, and a reward function $R(s_t, a_t)$ that defines the value of picking an action at a state. The agent's goal is to choose the action that leads to the highest total reward.

In an unknown MDP environment, people use reinforcement learning that enables agents to learn about optimal policy through trial-and-error. Similar to MDP, the core of reinforcement learning includes agents, states, actions, and rewards. In each iteration $t = 0, 1, 2, ...$, the agent observes the state s_t, takes an action a_t, progresses

to the next state s_{t+1}, and receives a reward. The interaction between agent and the environment is then repeated until convergence is reached. The agent follows a predefined rule to find the optimal policy. Value-based and policy-based reinforcement learning are two main approaches to optimize policy.

2.1.1 Value-based

In value-based reinforcement learning, agents estimate the value of state $V(s)$ and state-action pair $Q(s, a)$ and pick the action that maximizes the value function. Bellman equations are used to characterize the optimal values:

$$V^{\pi*}(s_t) = \max_{a_t} Q^{\pi*}(s_t, a_t) = \max_{a_t} \sum_{s_{t+1}} T(s_t, a_t, s_{t+1})[R(s_t, a_t, s_{t+1}) + \delta V^{\pi*}(s_{t+1})]$$

$V^{\pi*}(s)$ represents the expected utility starting in s and acting optimally under policy π. $Q^{\pi*}(s, a)$ is the expected utility starting from taking action a at state s and acting optimally. δ is the discount factor. The optimal policy is defined as: $\pi^*(s) = \arg\max_a Q^{\pi*}(s, a)$.

An example of a value-based approach is the Q-learning algorithm that learns optimal strategy through updating Q-values. Using finite S and A, $Q(s, a)$ can be represented as a $|S| \times |A|$ matrix where each entry is the value of taking action $a \in A$ at state $s \in S$ (Calvano et al., 2020). In each iteration, agents take actions and update one cell in the matrix: $Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha[r_t + \delta \max_{a_t+1} Q(s_{t+1}, a_{t+1})]$ where α is the learning rate. Q-learning guarantees optimality so long as each entry has been visited sufficiently many times. As Q-learning only updates one entry each time, Q-learning has a slow learning process when S and A are large.

2.1.2 Policy-based

Instead of estimating value functions and retrieving optimal policy from Q-values, agents learn parametrized policies π_θ by directly optimizing the expected returns with respect to θ (Levine and Koltun, 2013). At each time step, the agent's decision is characterized by $\pi_\theta(a_t|s_t)$ which represents probability distributions of actions over states (Sutton et al., 1999). In each iteration, agents pick actions a_t according to π_θ, enter the next state s_{t+1}, and receive rewards $r(a_t, s_t)$. Agents update their policy parameters according to the gradient of long-term expected rewards (Sutton et al., 1999). So better actions will be picked with a higher probability. Using gradient ascent, agents find the optimal policy that maximizes their long-term total rewards: $\eta(\theta) = E_{\tau\sim\pi_\theta(\tau)}[r(\tau)] = E_{\tau\sim\pi_\theta(\tau)}[\sum_t r(a_t, s_t)]$ where τ is the trajectory under policy π_θ. The gradient $\nabla_\theta\eta(\theta)$ is defined as:

$$\begin{aligned}
\nabla_\theta\eta(\theta) &= \int \nabla_\theta\pi_\theta(\tau)r(\tau)\,\mathrm{d}\tau \\
&= \int \pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau)r(\tau)\,\mathrm{d}\tau \\
&= E_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau)r(\tau)] \\
&\approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau^{(i)})r(\tau^{(i)})
\end{aligned} \tag{1}$$

A standard update of policy parameter can be written as: $\theta \leftarrow \theta + \alpha\nabla_\theta\eta(\theta)$ where α is the learning rate (Williams, 1992).

The Natural Policy Gradient is another popular policy-based method that achieves better performance and a faster convergence rate. Instead of using the normal gradient, the Natural Policy Gradient method follows the steepest direction with respect to the Fisher information matrix that measures the curvature (Kakade, 2001). Therefore, the

improvement of policy parameter in each step is written as:

$$\begin{aligned} \theta &\leftarrow \theta + \alpha \nabla_\theta \tilde{\eta}(\theta) \\ &= \theta + \alpha F(\theta)^{-1} \nabla_\theta \eta(\theta) \end{aligned} \tag{2}$$

where $F(\theta)$ is the Fisher information matrix defined as $E_{\tau \sim \pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^T]$. More details regarding the setup of the Natural Policy Gradient method are discussed in the next section.

3 Experimental Design

3.1 Economics Environment

3.1.1 Bertrand

Consider a differentiated product market with N firms. Each firm i competes in price p_i and maximizes profits $R_i(p_i, p_{-i}) = (p_i - c_i)q(p_i, p_{-i})$, where $q(p_i, p_{-i})$ is the demand function and c_i is the marginal cost. A strategy profile $p^* = (p_1^*, p_2^*, ..., p_n^*)$ is a Bertrand-Nash equilibrium if for every firm i and any other $\tilde{p_i}$ firm i can choose, $R_i(p_i^*, p_{-i}^*) \geq R_i(\tilde{p_i}, p_{-i}^*)$. In the baseline experiment, assume for linear demand and constant and identical marginal costs.

3.1.2 Cournot

Consider a homogeneous product market with N firms. Each firm i chooses its production level $q_i \geq 0$ simultaneously for infinitely repeated periods. The profit for each firm is denoted as $R_i(q_i, q_{-i}) = p(q_i, q_{-i})q_i - C_i(q_i)$ where $p(q_i, q_{-i})$ is the inverse demand

function and $C_i(q_i)$ is the cost of producing q_i.

Following the standard assumption in the literature (Frank and Quandt, 1963, Szidarovszky and Yakowitz, 1977), I focus on Cournot competitions where the following conditions hold:

1. The price function is strictly decreasing, twice-differentiable, and concave

2. The cost function is strictly increasing, twice-differentiable, and convex

Under the assumptions where $p'(q_i, q_{-i}), -C_i'(q_i) < 0$ and $p''(q_i, q_{-i}), -C_i''(q_i) \leq 0$, there is an unique Nash equilibrium (Szidarovszky and Yakowitz, 1977).

For the baseline experiment, consider a symmetric Cournot competition where the market price is denoted as $p(q_i, q_{-i}) = a - b(\sum_i q_i)$ and $C_i(q_i) = cq_i$ $(a, b, c \in \mathbb{R})$.

3.2 Natural Policy Gradient Setup

Following the standard setup of a stochastic policy in continuous space, I use the Natural Policy Gradient with a Gaussian policy to characterize the action $a_i \sim \pi_{\theta i}$ firm i chooses. The policy is defined as: $\pi_\theta(a|s) = \frac{1}{\sqrt{2\pi}\sigma_\theta(s)} \exp(-\frac{(a-\mu_\theta(s))^2}{2\sigma_\theta(s)^2})$, where μ_θ denotes the mean of firm's action. The action is reparametrized by the mean of action as $a = \mu_\theta(s) + \sigma_\theta(s)\xi$, where $\xi \sim N(0, 1)$ (Heess et al., 2015), and the action is rectified to be non-negative (Shi and Zhang, 2020). In each iteration, firm i chooses a_i, receives reward η_i, and updates θ_i according to the natural gradient. Following the derivation of gradient in 1, the gradient with respect to the policy parameter θ can be written as: $\nabla_{\theta i}\eta_i(\theta_i) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\theta_i} \log \pi_{\theta_i}(a_i) R_i(a_i)$, where $R_i(a_i)$ is the profit, $\pi_{\theta_i}(a_i)$ is the parametrized policy for firm i.

Therefore, in each step, the policy parameter is updated as:

$$\begin{aligned}\theta_i &\leftarrow \theta_i + \alpha F(\theta)^{-1}(\frac{1}{N}\sum_{i=1}^{N}\nabla_{\theta_i}\log\pi_{\theta_i}(a_i)R_i(a_i)) \\ &= \theta_i + \alpha F(\theta)^{-1}\frac{1}{N}\sum_{i=1}^{N}\frac{(a_i-\theta_i)^2}{\sigma_i^2}R_i(a_i)\end{aligned} \tag{3}$$

Algorithm 1 below shows the pseudocode of the algorithm used for simulations.

Algorithm 1 Natural Policy Gradient for Cournot competition with linear demand

Initialize θ_i, σ_i, learning rate α, a, b, marginal cost c_i, stop time T
for t = 0, 1, 2, ...T **do**
  Pick $q_i \sim N(\theta_i, \sigma_i)$
  Calculate $p = a - b(\sum_i q_i)$ and $R_i(q_i) = (p - c_i)q_i$
  Evaluate gradient $\nabla_\theta \eta_i(\theta) = \frac{1}{N}\sum_{i=1}^{N}\nabla_{\theta_i}\log\pi_{\theta_i}(q_i)R_i(q_i)$
  Update policy parameter $\theta_i \leftarrow \theta_i + \alpha F(\hat{\theta_i})^{-1}\nabla_{\theta i}\eta_i(\theta_i)$
end for

4 Outcomes and Result Analysis

4.1 Baseline Experiments

Consider symmetric Bertrand duopoly and oligopoly with three firms and linear demand functions. Assume $q(p_i, p_{-i}) = 0.5 - p_i + 0.5\sum p_{-i}$ and $c = 0.5$. Figure 1 tracks the learning process of the Natural Policy Gradient method with learning rate $\alpha = 0.08$ and standard deviation $\sigma = 0.08$ for each firm in the stated environments. The dotted red lines represent the Bertrand-Nash equilibria. In both duopoly and oligopoly environments, the algorithm learns to converge to the Bertrand-Nash equilibrium.

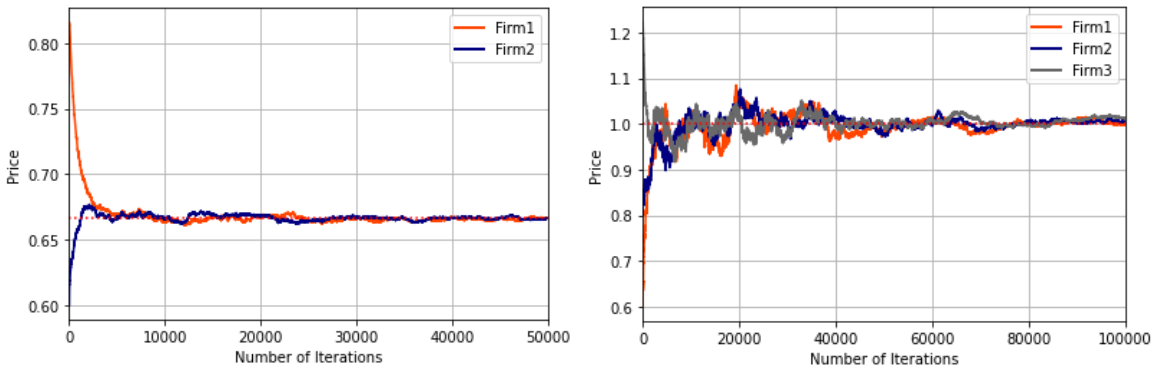

Figure 1: The learning process of the Natural Policy Gradient method in symmetric Bertrand duopoly (left) and oligopoly (right) environments

For symmetric Cournot competitions, consider the two-player and three-player cases with linear price functions $p(q_i, q_{-i}) = 2 - \sum_i q_i$ and constant marginal cost $c = 0.5$. Figure 2 displays the quantities chosen by the Natural Policy Gradient method with learning rate $\alpha = 0.08$ and standard deviation $\sigma = 0.08$ for each firm in each iteration. As expected, the convergence to Cournot-Nash equilibrium is verified in both Cournot duopoly and oligopoly environments.

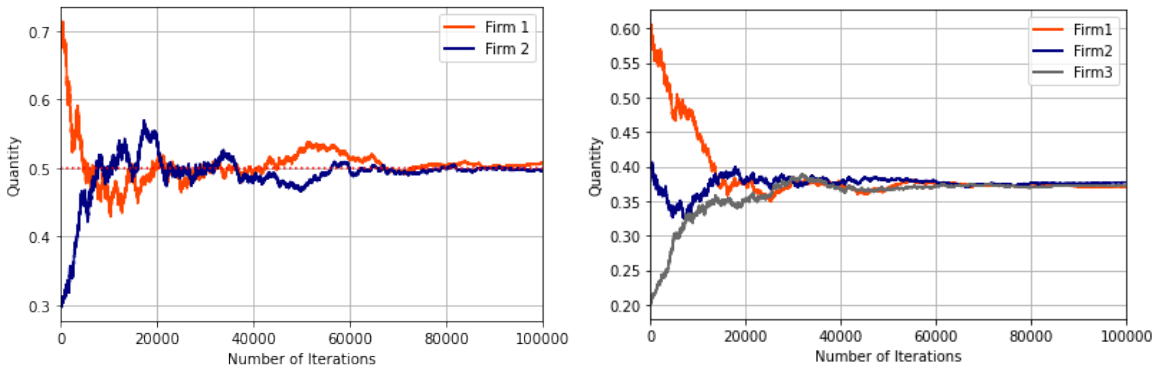

Figure 2: The learning process of the Natural Policy Gradient method in symmetric Cournot duopoly (left) and oligopoly (right) environments

These simulated markets are simplified versions of real-world scenarios containing both algorithmic and non-algorithmic sellers. As developing and training high-

performance algorithms to automate decision-making processes is time-consuming and costly for individual sellers, it is very likely that most algorithmic sellers on Amazon, eBay, and other e-commerce platforms choose to employ existing optimization software built and designed by professional experts using reinforcement learning. Sellers have incentives to pay for these repricing tools or price optimization products to make timely adjustments to market changes and gain competitive advantages, such as winning the Amazon Buy Box (Chen et al., 2016). Based on google search data, each of the most recommended Amazon price optimization products has hundreds or even thousands of users.[2] While algorithmic sellers can customize and configure their own features on the repricing platform, their algorithms share exactly the same architecture as those who also choose the same platform. Therefore, the simulated markets here can model the markets with algorithmic sellers who produce similar but differentiated products and use identical pricing software.

The convergence to pure strategy Nash equilibrium of the Natural Policy Gradient method suggests that algorithmic sellers who only use policy gradient based decision optimization tools to set price or quantity would adopt Nash strategies after a period of time, and algorithmic collusion does not exist under this setup.

4.2 Why Different from Q-learning?

The Natural Policy Gradient method leads to a completely different result from what prior researchers get using Q-learning. Instead of approaching supra-competitive equilibrium and sustaining collusive outcomes through reward and punishment strategies,

[2]For instance, over 2000 Amazon sellers choose RepricerExpress, and more than 500 firms use Feedvisor. Intelligence Node, which mainly targets retailers and category leaders, has hundreds of users worldwide.

the simulations in this paper identify the exceptions that would not lead to algorithmic collusion. Here are some possible reasons for getting the opposite result. First, policy gradient methods are theoretically more likely to reach competitive outcomes than Q-learning. While there is no general consensus among computer scientists and mathematicians that multi-agent policy gradient algorithms are guaranteed to converge to Nash equilibrium in continuous action and state space (Mazumdar et al., 2019), some researchers prove the convergence of policy gradient methods in general-sum multi-agent games under certain conditions. The findings in this paper resonate with recent studies of Hambly et al. (2021) and Shi and Zhang (2020). Hambly et al. (2021) demonstrate the global linear convergence for a class of linear-quadratic games with a certain level of noise in the dynamic system. Shi and Zhang (2020) derive that policy gradient dynamics converge to Nash equilibrium in concave Cournot games with either two players or a linear price function. Both of these studies provide theoretical support for this paper. The case for multi-agent Q-learning with ϵ-greedy exploration in general-sum games is different as no theoretical convergence guarantee has been verified.

Second, as Abada and Lambin (2020) suggest, seemingly collusive outcomes learned by Q-learning originate in imperfect exploration that lies in the nature of Q-learning. As Abada and Lambin manually force the algorithm to explore hardly explored states and the Nash equilibrium, firms deviate from the supra-competitive outcome and reach more competitive strategies instead. Therefore, this study and their result demonstrate that the choice of reinforcement learning algorithm and the corresponding exploration strategy significantly impact the equilibrium outcomes.

4.3 Robustness Checks

This section reports robustness checks. Figure 3 and 4 consider asymmetries in marginal costs. All parameters remain the same as the baseline experiment except that marginal cost varies across firms. Specially, I use $c_1 = 0.4$ and $c_2 = 0.8$ for the duopoly case and $c_1 = 0.2$, $c_2 = 0.4$, and $c_3 = 0.8$ for the three-player game. Figure 5 shows the simulations of Bertrand competitions with logit demand and constant marginal cost $c = 1$. The logit demand for product i is defined as $q_i = \frac{\exp \frac{2-p_i}{0.25}}{\sum_j \exp \frac{2-p_j}{0.25}+1}$.[3] The dotted red lines in the figures below represent the Nash equilibria. The result of this paper holds for markets with asymmetric costs or logit demand.

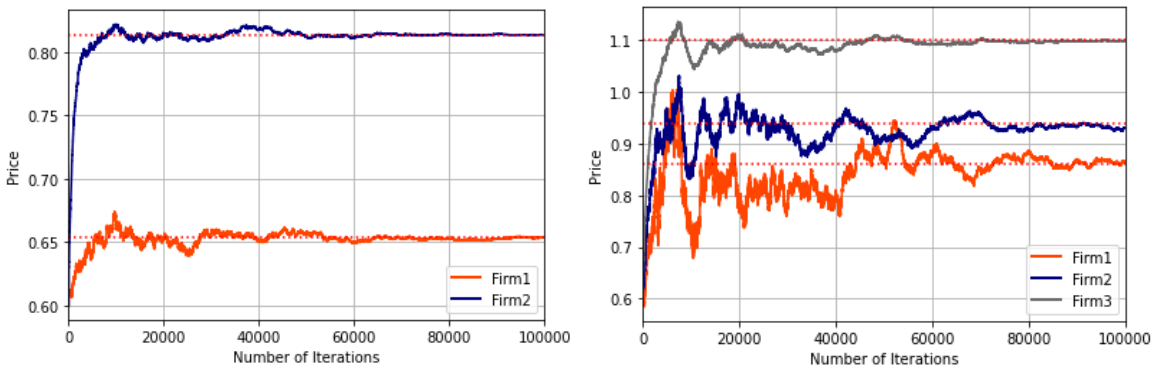

Figure 3: The learning process of the Natural Policy Gradient method in Bertrand duopoly (left) and oligopoly (right) environments with asymmetric marginal costs

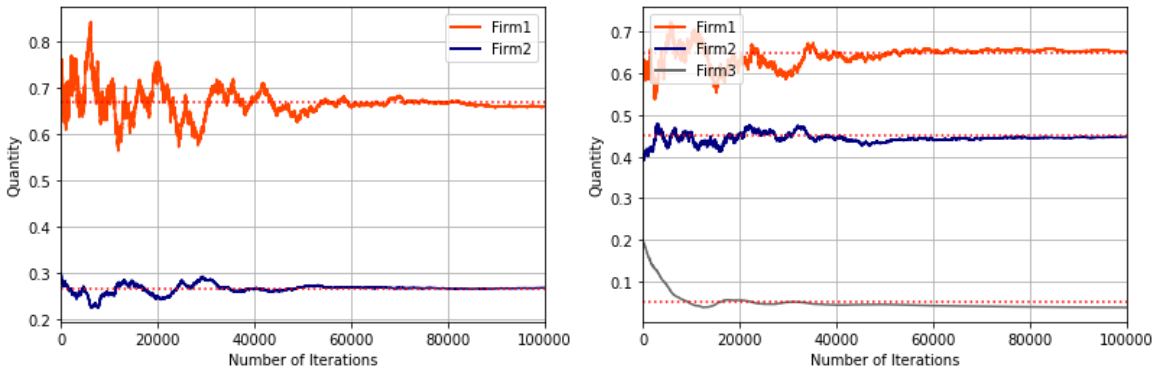

Figure 4: The learning process of the Natural Policy Gradient method in Cournot duopoly (left) and oligopoly (right) environments with asymmetric marginal costs

[3]This demand function is the one that Calvano et al. (2020) use for their baseline experiment.

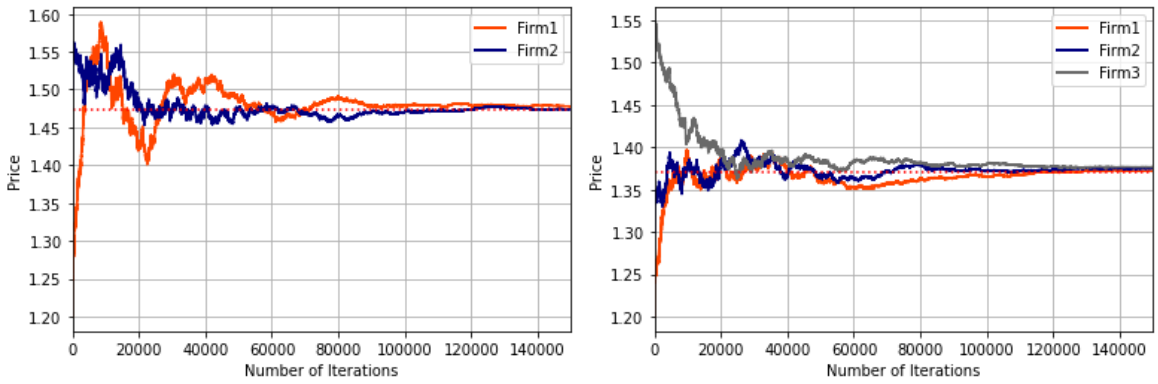

Figure 5: The learning process of the Natural Policy Gradient method in Bertrand duopoly (left) and oligopoly (right) environments with logit demand

5 Conclusion

In this paper, I take an experimental approach to explore the impact of decision-making algorithms on equilibrium outcomes in simultaneous Bertrand and Cournot competitions. Using the Natural Policy Gradient method, which is a type of policy-based reinforcement learning algorithm, I simulate repeated Bertrand and Cournot competitions in different market structures and observe the interactions between autonomous decision-making agents. The computer-simulated markets in this study can be considered as simplification of real-world online marketplaces involving algorithmic sellers who use policy gradient based decision-making software to optimize selling strategies. Through simulations, I show that decision-making algorithms based on the Natural Policy Gradient method consistently converge to Nash equilibria.

As far as I am aware, this is the first study in algorithmic collusion that employs algorithms other than value-based methods and identifies situations where algorithms can learn to reach competitive outcomes. Compared with value-based reinforcement learning algorithms, including Q-learning and Deep Q-Network that discretize action

space, the Natural Policy Gradient method studies continuous action and state space that better accords with real-world situations. The exceptions in algorithmic collusion would also provide helpful insights into developing regulatory policies for decision-making algorithms that carry the risk of reaching supra-competitive outcomes and sustaining tacit collusion.

The convergence to Nash equilibria of policy gradient dynamics combined with the collusive behaviors of Q-learning algorithms illustrates that the risk of algorithmic collusion depends on various factors, including the choice of machine learning algorithms. Compared with decision-making algorithms that are mainly based on Q-learning, optimization algorithms with policy gradient methods are less prone to algorithmic collusion. A theoretical generalization of this study and the analysis of the convergence properties of multi-agent Q-learning are left for future research.

One limitation regarding the Natural Policy Gradient method is that the gradient estimator suffers from large variance, slowing down the learning process. Variants of Actor-Critic methods could possibly mitigate this problem and achieve similar results with a faster convergence rate.

References

Abada, I. and X. Lambin (2020). Artificial intelligence: Can seemingly collusive outcomes be avoided? Available at SSRN 3559308.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation 10*(2), 251–276.

Assad, S., R. Clark, D. Ershov, and L. Xu (2020). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. No. 8521. CESifo Working Papers.

Brown, Z. Y. and A. MacKay (2021). Competition in pricing algorithms. *American Economic Journal: Microeconomics*.

Calvano, E., G. Calzolari, V. Denicolo, and S. Pastorello (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review 110*(10), 3267–97.

Cen, S., C. Cheng, Y. Chen, Y. Wei, and Y. Chi (2021). Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*.

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*, 1339–1349.

Ezrachi, A. and M. E. Stucke (2017). Artificial intelligence collusion: When computers inhibit competition. *University of Illinois Law Review* (5).

Ezrachi, A. and M. E. Stucke (2020). Sustainable and unchallenged algorithmic tacit collusion. *Northwestern Journal of Technology and Intellectual Property 17*.

Frank, C. R. and R. E. Quandt (1963). On the existence of cournot equilibrium. *International Economic Review*, 92–96.

Hambly, B., R. Xu, and H. Yang (2021). Policy gradient methods find the nash equilibrium in n-player general-sum linear-quadratic games. *arXiv preprint arXiv:2107.13090*.

Harrington, J. E. (2018). Developing competition law for collusion by autonomous artificial agents. *Journal of Competition Law and Economics 14*(3), 331–363.

Heess, N., G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa (2015). Learning continuous control policies by stochastic value gradients. *Advances in neural information processing systems 28*.

Hettich, M. (2021). Algorithmic collusion: Insights from deep learning. Available at SSRN 3785966.

Johnson, J., A. Rhodes, and M. Wildenbeest (2020). Platform design when sellers use pricing algorithms. No. 15504. CEPR Discussion Papers.

Kakade, S. M. (2001). A natural policy gradient. *Advances in neural information processing systems 14*, 1531–1538.

Klein, T. (2021). Autonomous algorithmic collusion: Q-learning under sequential pricing. *The RAND Journal of Economics 52*(3), 538–558.

Levine, S. and V. Koltun (2013). Guided policy search. *International conference on machine learning*, 1–9.

Mazumdar, E., L. J. Ratliff, M. I. Jordan, and S. S. Sastry (2019). Policy-gradient algorithms have no guarantees of convergence in linear quadratic games. *arXiv preprint arXiv:1907.03712*.

Peters, J. and S. Schaal (2006). Policy gradient methods for robotics. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2219–2225.

Russell, S. and P. Norvig (2020). *Artificial intelligence: a modern approach fourth edition*. Pearson.

Shi, Y. and B. Zhang (2020). Multi-agent reinforcement learning in cournot games. *IEEE Conference on Decision and Control (CDC)*, 3561–3566.

Sutton, R. S., D. A. McAllester, S. P. Singh, , and Y. Mansour (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*.

Szidarovszky, F. and S. Yakowitz (1977). A new proof of the existence and uniqueness of the cournot equilibrium. *International Economic Review*, 787–789.

U.S. Department of Justice (2016). Former e-commerce executive charged with price fixing in the antitrust division's first online marketplace prosecution.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning 8*(3), 229–256.