Algorithmic Collusion on Online Marketplaces[*]

Matteo Courthoud[†]

October 27, 2021

Abstract

The use of algorithms to set prices is particularly popular in online marketplaces, where sellers need to take quick decisions in complex dynamic environments. In this article, I investigate the role of online marketplaces in facilitating or preventing collusion among sellers that use pricing algorithms. In particular, I investigate a platform that has the ability to give prominence to certain products and automates this decision through a reinforcement learning algorithm, that maximizes the platform's profits. Depending on whether the business model of the platform is more aligned with consumer welfare or with sellers' profits (e.g., if it collects quantity or profit fees), the platform either prevents or facilitates collusion among algorithmic sellers. If the platform is also active as a seller, the so-called dual role, it is able to both induce sellers to set high prices and appropriate most of the profits. Importantly, self-preferencing only happens during the learning phase but is not observed in equilibrium. I investigate a potential solution: separating the sales and marketplace divisions. The policy is effective but does not fully restore the competitive outcome when the fee is distortive, as in the case of a revenue fee.

Keywords: Artificial Intelligence, Collusion, Platforms, Antitrust

JEL Classification: D21, D43, D83, L12, L13

[*] Preliminary, latest version here: `https://matteocourthoud.github.io/project/alg_platform/`.

[†] University of Zürich, email: matteo.courthoud@econ.uzh.ch.

1 Introduction

Algorithms are gradually replacing humans in many decision making processes. Modern artificial intelligence agents are able to take quick decisions in complex dynamic environments, often outperforming human decision making. In this paper, I am going to focus on pricing algorithms. These algorithms are designed to automate pricing strategies in online markets. The advantages of employing an algorithm over a human are a quicker reaction time, and the ability to deal with a high degree of complexity (OECD, 2017). Indeed, pricing algorithms are often deployed in markets that present a very high frequency of interaction. One of the most common venues where pricing algorithms are deployed is online marketplaces. Online marketplaces satisfy the pre-requisites cited above: a complex environment in which hundreds or thousands of sellers compete in multiple markets with prices that are updated more and more frequently[1]. Therefore, it is not surprising that a considerable amount of the pricing decisions in online marketplaces is automated and delegated to algorithms[2].

While, at the beginning, algorithms on online marketplaces where not very sophisticated[3], the most recent reinforcement learning algorithms come from state-of-the-art artificial intelligence research and are able to generate complex dynamic strategies (UK Competition and Markets Authority, 2021). Indeed, the seminal paper of Calvano et al. (2020) has shown how pricing algorithms are able to learn grim-trigger strategies in order to keep supra-competitive prices. Contrarily to previous evidence[4], they show that these strategies are not just coordinated high prices, but reinforcement learning algorithms are able to learn reward-punishment schemes that allow them to sustain higher prices and profits in the long run. They are able to learn grim-trigger strategies even if they interact sequentially (Klein, 2019) or if there is noise in the environment (Calvano et al., 2021). The use of these algorithms is not only allowed by online marketplaces but also supported through the availability of APIs to check competitors prices[5], guides and tutorials[6], and sometimes the platform itself

[1] Cavallo (2018) shows the dramatic increase in the frequency of interaction on Amazon Marketplace.

[2] Chen et al. (2016) estimate that around 1/3 of the prices set on Amazon Marketplace in the 250 most sold categories, are automated and set by algorithms.

[3] A famous example is the competition between two price over-cutting algorithms that lead to a book about flies having a posted price of over 2 million dollars (see detailed post by Michael Eisen: *Amazon's 23,698,655.93 book about flies*.

[4] For example, Waltman and Kaymak (2008) show that algorithms are able to learn to coordinate on supra-competitive prices, however, they do not show evidence of the fact that these strategies are robust to unilateral deviations.

[5] See for example the wide range of APIs that Amazon Marketplace offers in its *Amazon Marketplace Web Services*.

[6] See for example Amazon Marketplace *Guide on Low Price Matching*.

provides the algorithms to firms[7].

On the other hand, online marketplaces have recently drawn policy-makers attention for their role of market makers or "gatekeepers". Many competition authorities have released reports in the last years, documenting the growing market power and conflicts of interest of the largest digital platforms[8]. These platforms enjoy significant market power, which in turn translates into the economically unchallenged ability to set the rules of the game. Of particular interest for this paper, are online marketplaces such as Amazon Marketplace, the Apple Store or Google's Play Store. On these marketplaces, sellers offer their products to users and the platform acts as an aggregator[9] and a filter by providing users the best products for their queries. As documented in the literature[10], platforms are able to exert a significant control over consumers' attention through the selection and ranking of products[11]. While the ability of a platform to filter results and provide consumers with the best search results can be crucial for its competitive success, it also gives it significant power over which products enter consumers' consideration sets. The ability of online marketplaces to steer consumers' attention, in turn can have an impact on sellers' conduct and on market outcomes.

In this article, I study the role that online marketplaces can have in facilitating or preventing algorithmic collusion. In particular, I study a setting in which multiple sellers compete among each other on an online marketplace and employ reinforcement learning algorithms in order to set prices. I assume that the platform enjoys significant market power, which implies that sellers do not face competition from outside the platform, and that the platform is able to freely "set the rules of the game" in the marketplace. In particular, I assume that the platform is able to redirect consumers' attention to any subset of sellers, at no cost. This assumption can be seen either as the ability of platforms to control consumers' query results, or to redirect their attention to specific products with the use of UI elements such as badges or banners[12]. Lastly, I assume that the marketplace delegates the choice-set

[7] Amazon itself offers an *Automate Pricing* algorithm.

[8] The list of reports includes the Australian Competition and Consumer Commission (2019) "Digital Platforms Inquiry" report, the Furman et al. (2019) "Unlocking Digital Competition" report for the UK Competition and Market Authority, the Crémer et al. (2019) "Competition in the Digital Era" report for the European Commission, the Duch-Brown (2017) "The Competitive Landscape of Online Platforms" report for the Joint Research Center on Digital Economy, the Stigler Platforms (2019) Final Report on Digital Platforms, and the Schallbruch et al. (2019) "Competition Law 4.0" report for the German Commission.

[9] Ben Thompson defines an Aggregator platform in his *Aggregation Theory* article.

[10] See an excellent literature review of how platforms are able to control consumers' attention in Derakhshan et al. (2020).

[11] For example, only 10% of consumers browse to Amazon Marketplace second page of results and even less go to the third page (see AdWeek article *Why It's So Hard for Brands to Get Noticed on Amazon*).

[12] See for example this article by EyeSee on how consumers' attention changes across online marketplaces: *Decoding the Product Page: Why Walmart's product pages drive 14% higher sales than its competition's.*

selection decision to a reinforcement learning algorithm with the objective of maximizing platform's profits. Importantly, I will analyze different platform business models, and how the equilibrium outcome will change depending on how the platform monetizes its business.

In my setting, both the sellers and the platform automate their decision-making through reinforcement learning algorithms. The sellers employ pricing algorithms to dynamically set their prices and the platform reacts to the prices deciding which products will receive consumers' attention. This paper is closely related to Johnson et al. (2020) that explore a similar setting where the platform uses different heuristics to select the consideration sets, such as showing only the lowest priced item. The difference with their paper is twofold. First, in my setting, the platform is maximizing profits and therefore the facilitation or prevention of algorithmic collusion is not an objective of the platform but a byproduct of its optimization problem. Second, I allow the platform to use a reinforcement learning algorithm to construct the consideration sets. This allows me to endogenize the decision making of the platform, and, at the same time, to contribute to the computer science literature exploring interactions between artificial intelligence agents with separate (possibly conflicting) objectives.

I show that depending on the business model of the marketplace, the platform is able to either prevent or facilitate tacit collusion among algorithmic pricing sellers. In particular, if the platform collects a profit fee, and hence its objective function is aligned with firms' incentives, it facilitates collusion between sellers. If, on the other hand, the platform collects a quantity or per-sale fee, it is able (and has the incentives) to hinder collusion and increase competition among sellers. The platforms uses its ability to punish firms - by directing consumers' attention away from them - in order to steer pricing algorithms away from or towards collusive policies. This intervention takes place both during the pricing algorithms' learning phase and in equilibrium. In fact, we observe that when the platform has the incentive to facilitate collusion, it intervenes in the punishment phase of the grim-trigger strategies, by strengthening it. When, on the other hand, it has the incentive to prevent collusion, it intervenes in the punishment phase of the grim-trigger strategies, by relaxing it. Moreover, the platform also intervenes during the learning process increasing or decreasing the speed of pricing algorithms' learning, depending on its incentives.

In the second part of the paper, I investigate the case in which the platform is also active as a seller, the so called "dual role". Independently from the business model, in case the platform jointly maximizes the profits of the marketplace and the retail divisions, it induces sellers to set high prices. Importantly, it also induces a very specific set of strategies so that most profits are appropriated by its own sale division. In order to obtain this outcome, it learns a "bullying" strategy that always excludes the independent sellers if it does not perform the platform's preferred strategy. Importantly, exclusion is not observed in equilibrium, but only during the learning phase. In equilibrium an external observer would only observe that the platform's products are cheaper that the independent seller's product. I propose a potential solution to this issue: separate the marketplace and retail divisions of the platform and have the marketplace division act as if it was collecting a fee also on its own products. This solution restores the non-dual outcomes, in case the platform is collecting a profit fee. Otherwise, since quantity and revenue fees are distortive, the separation between the divisions introduces a bias in favor of the platform's product.

The paper is structured as follows. Section 2 contains a review of artificial intelligence algorithms in decision making. In particular, I cover recent advancements in computer science, with a focus on Q-learning algorithms. In Section 3, I present the specific formulation used in the simulations. In Section 4, I explore the baseline setting in which two Q-learning pricing algorithms compete on an online marketplace that collects fees. Depending on the fee structure, the platform has the incentives and is able to facilitate or prevent tacit collusion among sellers. Section 5 focuses on the dual role of the platform: when it is both active as a marketplace and a seller. I first show how the platform induces sellers to set high prices and coerces the independent seller to a strategy that allows the platform's seller to reap most of the profits. Then, I analyze a potential policy solution: splitting the marketplace and sales divisions. This solution enables to solve most, but not all the self-preferencing problems. Section 6 concludes.

2 Q-Learning

Reinforcement learning algorithms are a class of learning algorithms that try to solve optimization problems where rewards are delayed over multiple time periods. Moreover, these rewards depend on the sequence of actions that the algorithm has to take over different time periods. These two characteristics make reinforcement learning fundamentally different from standard supervised learning problems. The algorithm objective is not simply to learn a reward function but also to take an optimal sequence of actions. The most important feature of reinforcement learning is indeed the dual role of the algorithm: prediction and optimization.

Q-learning is a popular reinforcement learning algorithm and constitutes the baseline model for the most successful advancements in Artificial Intelligence in the last decade (Igami, 2020). The most popular algorithms such as Bonanza or AlphaGo are based on the same principle, while using deep neural networks to provide a more flexible functional approximation of the policy function (Sutton and Barto, 2018).

One advantage of Q-learning algorithms is their interpretability, especially for what concerns the mapping from the algorithm to the policy and value functions. The policy function has a matrix representation that can be directly observed and interpreted. This makes it possible to understand not only the logic behind any decision of the algorithm at any point in time, but also to know what the algorithm would have done in any counterfactual scenario.

In this section, I first explore the general formulation of Q-learning algorithms. Since these algorithms have been developed to work in single-agent environments, I comment on their use in repeated games. Lastly, I analyze the baseline algorithm used in the simulations throughout the paper.

2.1 Single Agent Learning

Reinforcement learning algorithms try to solve complex dynamic optimization problems by adopting a model-free approach. This means that the algorithm is not provided any structure regarding the relationship between the state it observes, its own actions, and the payoffs it receives. The algorithm only learns through experience, associating states and actions with the payoffs they generate. Actions that bring higher payoffs in a state are preferred to actions that bring lower payoffs. Since the state can include past states or actions, reinforcement learning algorithms can learn complex dynamic strategies. The more complex the state and action space of the learning algorithm, the more complex the strategies it can learn.

The objective function of the learning algorithm is the expected discounted value of future payoffs

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t \pi_t\right]. \tag{1}$$

In many domains faced by computer scientists, payoffs have to be hard-coded together with the algorithm. For example, with self-driving cars, one has to establish what is the payoff of an accident, or the payoff of arriving late. Clearly, the decision of these payoffs directly affects the behavior of the algorithm. However, in some cases, the payoffs are directly provided by the environment. For example, in the setting analyzed in this paper, the algorithm sets prices and observes the profits coming from the sales of an item.

The main trade-off faced by reinforcement learning algorithms is the so-called *exploration-exploitation* trade-off. Since the algorithm is not provided with any model of the world, it only learns through experience. In order to learn different policies, the algorithm explores the action space by taking sub-optimal actions. However, since the objective of the algorithm is to maximize the expected discounted sum of future payoffs, at a certain point, the algorithm needs to shift from exploration to exploitation, i.e., it needs to start taking optimal actions, given the experience accumulated so far.

In each period, the algorithm observes the current state of the world, $\boldsymbol{s}$ and takes an action a with the intent to maximize its objective function. I refer to the total discounted stream of future payoffs under optimal actions as the value function. I can express the value function of algorithm i in state s recursively as

$$V_i(\boldsymbol{s}) = \max_{a_i \in \mathcal{A}} \left\{ \pi_i(\boldsymbol{s}, \boldsymbol{a}) + \delta \mathbb{E}_{\boldsymbol{s}'}[V_i(\boldsymbol{s}')|\boldsymbol{s}, \boldsymbol{a}] \right\}. \tag{2}$$

Q-learning algorithms are based on a different representation of the value function, the action-specific value function, which is called the Q function. I can write the action-specific value function of algorithm i in state $\boldsymbol{s}$ when it takes action a_i as

$$Q_i(\boldsymbol{s}, a_i) = \pi(\boldsymbol{s}, \boldsymbol{a}) + \delta \mathbb{E}_{\boldsymbol{s}'} \left[\max_{a_i' \in \mathcal{A}} Q_i(\boldsymbol{s}', a_i') \middle| \boldsymbol{s}, \boldsymbol{a} \right]. \tag{3}$$

When the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite, we can express the Q function as a $|\mathcal{S}| \times |\mathcal{A}|$ matrix. Many of the advancements in reinforcement learning involve a functional representation of the Q function that can encompass more complex state or action spaces. The most successful function approximations involve deep neural networks.

The objective of the algorithm is to learn the Q function. Once the algorithm has learned the Q function, in each period it will take the optimal action $a_i^* = \arg\max_{av \in \mathcal{A}} Q(\boldsymbol{s}, a_i)$. Learning and taking the optimal actions are the two main tasks of the algorithm, and the choice between the two behaviors constitutes the main trade-off of reinforcement learning: *exploration* versus *exploitation*.

Exploitation. Since the final objective of the algorithm is to maximize the expected discounted sum of payoffs, in the exploitation phase, the algorithm picks the best available action, given the experience accumulated so far. After a sufficient amount of exploration, in a stationary environment, the algorithm expected discounted sum of future payoffs is guaranteed to converge to a local optimum of the value function (Beggs, 2005). Exploration is needed in order to discover other better locally optimal policies.

Exploration. During the exploration phase, the objective of the algorithm is to explore the state-action space in order to discover new policies. Since exploration involves testing suboptimal actions, there is a trade-off between exploration and exploitation. More exploration implies lower short-term profits but can lead to the discovery of policies than bring higher long-term profits. Moreover, when reinforcement learning algorithms are deployed in a dynamic environment, exploration gives the algorithm the flexibility to adapt to changes in the environment.

There exists may different ways in which one algorithm can explore the state-action space and the simplest one is the ε-greedy model. In each period, the algorithm decides to explore with probability ε and to exploit with probability (ε). In the exploration phase, the algorithm chooses one action uniformly at random.

Optimality Results. In a Markov single agent environment, under mild conditions, it has been proven that a reinforcement learning algorithm learns locally optimal strategies, given that the exploration rate ε converges to zero as time goes to infinity (Sutton and Barto, 2018).

Learning Speed. Since learning is a noisy process, the Q matrix is only partially updated. In particular, given an action a_i^*, irrespectively of whether the action comes from exploration or exploitation, the update policy is

$$Q_i(\boldsymbol{s}, a_i^*) = \alpha Q_i^{OLD}(\boldsymbol{s}, a_i^*) + (1 - \alpha) Q_i^{NEW}(\boldsymbol{s}, a_i^*) \tag{4}$$

where we refer to α as the *learning rate.* A higher α implies a faster but noisier learning. The policy function takes less time to converge but it is less likely to adopt better strategies.

2.2 Repeated Games

In my setting, multiple Q-learning algorithms play a repeated game in which they set per-period prices with the objective to maximize the expected discounted sum of profits. Most of the reinforcement learning literature in computer science focuses stationary environments. One common example is video games, where algorithms receive the video feed as an input and have to pick the optimal actions in order to perform best in the game. Another common area of research in reinforcement learning is robotics, from logistics to self-driving cars. All these examples involve mostly stationary environments.

The behavior of reinforcement learning algorithms competing with each other in a repeated game is still under research and there exist no general result concerning their behavior. In particular, there is no result on whether algorithmic behavior will converge on collaborative behavior, depending on the context.

3 Setting

As shown in the seminal paper by Calvano et al. (2020), pricing algorithms, when deployed in controlled environments, are able to learn complex grim-trigger strategies with the intent of keeping supra competitive prices. In Calvano et al. (2020) setting, two algorithms compete in pricing facing logit demand and no source of external noise. This finding has been supported by a growing strand of literature that has confirmed the robustness of this finding, for example when the stage pricing game is sequential, à la Stackelberg (Klein, 2019).

There is evidence that many of these algorithms are deployed in high-frequency markets (Assad et al., 2020) and online marketplaces (Chen et al., 2016). Indeed, an essential requirement for reinforcement learning algorithms to work well is to be provided with a sufficient amount of data, which means either very frequent interactions, or many parallel ones. In particular, Chen et al. (2016) estimate that approximately 30% of pricing on Amazon Marketplace is automated. However, they do not provide evidence of the fact that this automation is in the form of reinforcement learning algorithms.

Online marketplaces support the use of algorithms to perform pricing, by providing access to APIs to track competitors' prices in real time. Moreover, many private companies offer out-of-the-box algorithms to automate pricing[13]. This has lead to a substantial increase in the diffusion of pricing algorithms[14] and a parallel increase in their sophistication.

Moreover, online marketplaces usually have control over consumer attention. Indeed, their role is to aggregate a wide variety of sellers on a unique platform and provide consumers an efficient way to scan them. The most common way in which platforms perform this selection of attention is through the ranking of alternatives. When a user performs a search, products are ranked from top to bottom, according to some metric, often unknown to the consumer. The amount of attention is non-linear in the ranking with considerable drops at the end of the screen and at each subsequent page[15]. Online marketplaces also perform other types of attention filtering by providing further UI elements such as banners or flags or dedicating special UI space to certain products. In the model, I will model attention as a binary choice of each item, therefore reducing the need to parametrization.

Because of these reasons, I explore a setting in which sellers producing a differentiated good compete on an online marketplace. Sellers employ pricing algorithms to select their

[13] Amazon itself offers an *Automate Pricing* algorithm.

[14] See for example the recent reports by OECD (2017), the UK Competition and Markets Authority (2018) on algorithmic pricing.

[15] See an excellent literature review of how platforms are able to control consumers' attention in Derakhshan et al. (2020)

price, with the objective of maximizing their discounted stream of profits. The platform instead collects a fee from the seller's transactions and is able to direct consumer attention to either seller or both. The platform employs a reinforcement learning algorithm with the objective to maximize the discounted stream of profits of the platform. I abstract from the participation decision of the sellers, which however constitutes an essential element of the market design. However, I will comment on sellers' participation whenever it seems reasonable to be an unreasonable assumption. The details of the model are provided in the remaining part of the section.

3.1 Model

I adopt the baseline model of Calvano et al. (2020). First of all, this choice allows a direct comparison with their simulation results. Moreover, their setting is particularly simple and easy to interpret. Lastly, since many competition policy and law papers are based on their results, I can directly speak to that literature using the same exact model and parametrization.

Time is discrete and the horizon is infinite. At each point in time, n firms are active and compete in prices with differentiated products. I discretize the possible actions on a grid of dimension m: $\{p_1, ..., p_m\}$. Differently from Calvano et al. (2020), there is also an extra agent, the online marketplace. The online marketplace, after observing the pricing decision of the firms, decides which products to display to consumers and which not to. Firms instead, base their decision on the prices in the previous period and the platform's decision in the previous period[16]. Therefore, the state of firm i is represented by a vector $\boldsymbol{s}_{i,t} = \{p_{1,t-1}, ..., p_{i,t-1}, ..., p_{n,t-1}, r_{t-1}\} \in S = \{p_1, ..., p_m\}^k \times \{0,1\}^n$, where $s_{i,t}$ represents the state of firm i at time t and r_{t-1} the decision of the platform at time $t-1$ (where r stands for *ranking*). The decision of the platform is a vector of dimension n, in which each element i indicates whether seller i's product is shown to consumers ($r_i = 1$) or not ($r_i = 0$). In the case in which only two sellers are active, the set of possible actions of the platform is the collection of tuples $R = \{(1,0), (0,1), (1,1)\}$. I exclude the case $(0,0)$ where no product is shown to consumers. The discount factor is $\delta \in [0,1)$.

Demand. There is a continuum of consumers of unit mass. Consumer j's utility from

[16] In principle, agents' state space could include more than one period. However, as noted by Calvano et al. (2020), at least in this setting, a higher number of periods is not necessary in order to allow for higher strategy sophistication. In fact, by just increasing the number of grid points in the state space, one could obtain the same result.

buying one unit of product i is given by

$$u_{j,i} = v_j - \mu p_i + \varepsilon_{j,i} \tag{5}$$

where v_j is the value of the product i for consumer j, p_i is the price of product i, μ is the price elasticity and $\varepsilon_{j,i}$ is the idiosyncratic shock preference of consumer j for product i. The random shocks $\varepsilon_{j,i}$ are assumed to be independent and type 1 extreme value distributed so that the resulting demand function has the logit form. For example, the demand of product i is:

$$q_i(\boldsymbol{p}) = \frac{e^{-\mu p_i}}{e^{-\mu p_i} + \sum_{-i} e^{-\mu p_{-i}}}. \tag{6}$$

Depending on the business strategy of the platform, the profit functions can take one of the forms reported in Table 1.

	Firm $i : \pi_i(\boldsymbol{p}, r)$	Platform $p : \pi_p(\boldsymbol{p}, r)$
Without fee	$q_i(\boldsymbol{p}, r) \cdot (p_i - c)$	0
Profit fee	$q_i(\boldsymbol{p}, r) \cdot (p_i - c) \cdot (1 - f)$	$f \cdot (\pi_i + \pi_j)$
Revenue fee	$q_i(\boldsymbol{p}, r) \cdot (p_i \cdot (1 - f) - c)$	$f \cdot (p_i q_i(\boldsymbol{p}, r) + p_j q_j(\boldsymbol{p}, r))$
Quantity fee	$q_i(\boldsymbol{p}, r) \cdot (p_i - c - f)$	$f \cdot (q_i + q_j)$

Table 1: Profit functions and business models

When all firms are visible to customers, $R = \{1\}^n$, the static game has a unique Nash Equilibrium which we refer to as the competitive outcome. To ensure that the platform never has the incentive to exclude any firm in equilibrium, I impose a malus to the platform profits equal to the maximum state-wise difference between the platform profits with and without exclusion. This preserves the incentives but ensures, by construction, that the platform never has the incentive to exclude firms in the static game. However, it might decide to do so in the dynamic game. In fact, this decision can impact the learning process of the pricing algorithms and their equilibrium strategies. One might interpret this assumption as a reduced form assumption regarding the participation decision of firms. The platform is allowed to exclude firms occasionally but it cannot do so in equilibrium. The reasons might be reputational concerns, direct economies of scale or indirect economies of scale (consumers have a preference for searching for products on a single platform).

Exploration/Exploitation. I use a ε-greedy exploration method as in Calvano et al. (2020). In each period, each algorithm has a probability ε_t of exploring and a probability

$1 - \varepsilon_t$ of exploiting. I refer to ε as the exploration rate. The value of ε_t in period t is given by

$$\varepsilon_t = 1 - e^{-\beta t} \tag{7}$$

where β is the convergence parameter that governs how quickly the algorithms shifts from exploration to exploitation. The exploration/exploitation probability distribution is the same for all algorithms, however the realizations are not. As shown in Calvano et al. (2020), the probability of collusion is generally increasing in β. The more the algorithms are allowed to explore, the more likely it is that they "stumble upon" a reward-punishment scheme. Once they discover these schemes, they are likely to adopt them since they are more profitable than playing Nash Equilibrium in the long run. On the other hand, the platform might or might not facilitate the learning of these collusion strategies by removing some sellers from the consumers' consideration set.

Q Matrix. Each algorithm's policy function is defined by a Q-matrix. The Q-matrices of the pricing algorithms have dimension $m^n \times r \times m$ where the first $n+1$ dimensions correspond to the state space, i.e., the previous actions of all firms plus the action of the platform, and the last dimension corresponds to the current action. I initialize the Q matrix to the sum of discounted value of future profits, given action a_i, and averaging over the possible actions of the opponents. Therefore, the initial values do not depend on the state $\boldsymbol{s}$, but only on the action a:

$$Q_i^0(\boldsymbol{s}, a_i) = \frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{a}_{-i} \in \mathcal{A}^{n-1}} \frac{\pi_i(a_i, \boldsymbol{a}_{-i})}{1 - \delta} \tag{8}$$

The Q-matrix of the platform instead has dimension $m^n \times n$, where the first m^n dimensions correspond with the current actions of the firms and the last n dimensions correspond with the action of the platform: the ranking of firms. Indeed, the platform conditions its decision on the current sellers' prices.

Policy Update. Irrespectively of whether an algorithm is exploring or exploiting, the Q matrix is updated by averaging it's new value with the previous one, according to a parameter α, the learning rate. In particular, the updating formula is the following:

$$Q_i(\boldsymbol{s}, a_i^*) = \alpha Q_i(\boldsymbol{s}, a_i^*) + (1 - \alpha) \Big[\pi(\boldsymbol{s}, a_i^*) + \delta \max_{a_i'} Q_i(\boldsymbol{s}', a_i') \Big]. \tag{9}$$

The new value of Q is state $\boldsymbol{s}$ for action a_i^* is an average of the old value, $Q_i(\boldsymbol{s}, a_i^*)$, and the new one. The new value is given by the static payoff of action a_i^* in state $\boldsymbol{s}$, plus the discounted value of the best action the next state, $\boldsymbol{s}'$. There are different ways to update the continuation value of the Q-function (Sutton and Barto, 2018). In this setting, I select the optimistic updating given by the *max* operator to be consistent with the choice of Calvano et al. (2020). Other possible solutions are weighted averages, where the softmax is a popular weighting function. I update the Q-functions of the firms and the Q-function of the platform in the same way.

Algorithm. I summarize the full algorithm in Figure 1

```
Algorithm 1: Q-learning
initialize $Q_i^0(\boldsymbol{s}, a_i)\ \forall i = 1...n, \boldsymbol{s} \in \mathcal{S}, a_i \in \mathcal{A}$ ;
initialize $\boldsymbol{s}^0$ ;
while convergence condition not met do
    for $i = 1...n$ and (last) $p$ do
        exploration$_i = \mathbb{I}\left(r_i < e^{-\beta t}\right)$ where $r_i \sim U(0,1)$ ;
        if exploration$_i$ then
            $a_i^* = a \in A$ chosen uniformly at random ;
        else
            $a_i^* = \arg\max_{a_i} Q_i(\boldsymbol{s}, a_i)$ ;
        end
    end
    observe $\boldsymbol{\pi}$ given $(\boldsymbol{s}, a^*)$ ;
    observe $\boldsymbol{s}'$ given $(\boldsymbol{s}, a^*)$ ;
    $Q_i(\boldsymbol{s}, a_i^*) =$
      $\alpha Q_i(\boldsymbol{s}, a_i^*) + (1-\alpha)\left[\pi(\boldsymbol{s}, a^*) + \delta \max_{a_i'} Q_i(\boldsymbol{s}', a_i')\right]$
      $\forall i$ ;
    $\boldsymbol{s} = \boldsymbol{s}'$ ;
end
```

Parametrization. I summarize the parameters of the baseline model in Table 2. The parametrization closely follows Calvano et al. (2020) so that the simulation results are directly comparable with theirs. The main difference concerns The exploration rate β In the original paper, the exploration rate was $\beta = 4 \cdot 10^{-6}$, but since now we have another extra state of dimension 3, I reduce β to $2 \cdot 10^{-6}$ to keep approximately the same number of explorations per state as in the original paper (~ 20).

Parameter Description	Parameter	Value
Learning rate	α	0.15
Exploration rate	β	$2 \cdot 10^{-6}$
Discount factor	δ	0.95
Marginal cost	c	1
Number of past observed states	k	1
Dimension of the action grid	m	15
Number of firms	n	2
Convergence parameter	T	$U[0,1]$
Price elasticity	μ	0.25

Table 2: Model parametrization

3.2 Convergence

There is no guarantee of convergence for the learning algorithm. Algorithms react to each others' policies and therefore it is possible that they get stuck in a loop. Moreover, as long as there is a non-zero probability of exploration, there is always a chance that one algorithm suddenly adopts a totally different policy.

I use the same convergence criterion of Calvano et al. (2020): convergence in actions. The algorithm stops if for T periods the index of the highest value of the Q matrix in each state has not changed i.e. $\arg\max_{a_i} Q_{i,t}(\boldsymbol{s}_t, a_i) = \arg\max_{a_i} Q_{i,t+\tau}(\boldsymbol{s}_{t+\tau}, a_i)\ \forall i, \boldsymbol{s}, \tau = 1...T$. In practice, I choose a value of $T = 10^5$.

The advantage of this convergence criterion is that it does not require the algorithms to adopt a full exploitative strategy in order to converge. Algorithms' actions can stabilize even if the algorithms regularly take random actions, but with low probability. In fact, as long as the learning-rate α is lower than 1, firms only partially update their Q function. In practice, we observe convergence around $3M$ to $4M$ iterations, i.e. when the exploration probability ε is around e^{-10} to e^{-15}.

It is important to remark that one can achieve convergence in a shorter amount of periods. What is crucial to achieve convergence is a sufficiently little exploration rate. However, with a lower exploration rate, the algorithms are less likely to discover reward-punishment collusive strategies.

4 Baseline

In this section, I present the simulation results from the baseline setting presented in Section 3. Two pricing algorithms compete in prices, on an online marketplace. The marketplace collects a fee, according to its business model. I explore three different business models in this section:

1. The online marketplace collects a profit fee
2. The online marketplace collects a quantity (or per-item) fee
3. The online marketplace collects no fee

I abstract from the case in which the online marketplace collects a revenue fee since it falls in between the profit and the revenue fee. Whether the outcomes from an environment in which the platform collects a revenue fee are more similar to a scenario with a profit or a quantity fee, it depends on the industrial organization of the market. Moreover, the objective function of the platform in these two scenarios is unambiguously aligned with either firm profits or consumer surplus, respectively. This makes the analysis more transparent, for what concerns the platform's incentives.

In order to make the different scenarios comparable, in each setting I compute the platform fees as follows. First, I select a profit fee. In particular, I will use a profit fee equal to 0.5 in the baseline model, which means that the platform and the sellers split the profits in equal measure. Then, I compute the revenue that the platform would gather if it was charging that profit fee when sellers set Nash equilibrium prices. Lastly, I set the quantity and revenue fees so that, given those same prices, the platform would collect the same fee. The resulting set of fees in the baseline model is the following:

- $f^{\text{profit}} = 0.5$
- $f^{\text{quantity}} \simeq 0.2$
- $f^{\text{revenue}} \simeq 0.15$

These fees imply a platform profit of 0.34 (the marginal cost is $c = 1$), when sellers set Nash equilibrium prices. Importantly, this fees were chosen so that the equivalent revenue fee is approximately equal to 0.15, which is the fee that many platform markets, such as Amazon Marketplace, the Play Store, and the Apple Store, charge.

All the results presented below are averages over 100 simulations, unless otherwise stated.

4.1 Results

I first present what happens when the online marketplace does not charge any fee. I set the ranking algorithm so that in case of indifference, it decides not to exclude any seller. Therefore, no seller will ever be excluded and this makes the setting equivalent to that of Calvano et al. (2020). I report the price impulse response function to an unilateral deviation of Algorithm 1 in equilibrium, in Figure 1.

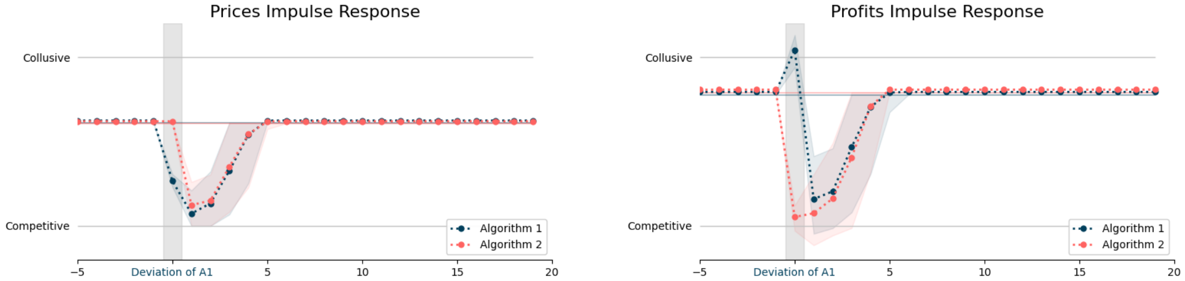


Figure 1: Timelines of prices and profits of both algorithms, when the platform does not charge a fee. In period 0, Algorithm 1 is manually forced to set the static best response price to Algorithm 2's price. Afterwards, the algorithms play according to their Q-functions. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2. The actions and payoffs have been normalized to each simulation average. The dots represent the median response and the areas around them represent the 25th to 75th percentiles.

As we can see, the equilibrium average prices and profits are close to the collusive ones. Moreover, in order to confirm that this is not due to noise, but rather it is the outcome of a collusive strategy, we observe the reaction to an unilateral deviation of Algorithm 1, manually imposed. In the deviation period, in grey in Figure 1, Algorithm 1 sets a lower price and earns higher profits. In the following periods, I allow both algorithms to respond according to their equilibrium Q-function. We observe that the deviating algorithm is punished for a couple of periods, but afterwards collusive play is restored. The behavior depicted in Figure 1 is going to be the baseline of the rest of the analysis.

After having analyzed the benchmark collusive behavior, we introduce a third player: the online marketplace. We start from the case in which the online marketplace is charging a profit fee, i.e. is taking a cut on sellers' profits, as described in Table 1. As reported in Section 3, the platform fee is equal to $f^{\text{profit}} = 0.5$. I report the impulse response function of prices and payoffs of a unilateral deviation of Algorithm 1 in Figure 2.

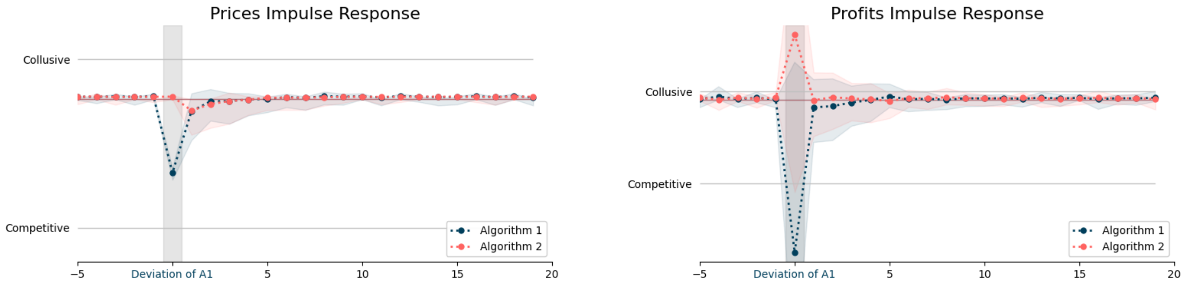


Figure 2: Timelines of prices and profits of both algorithms, when the platform is charging a profit fee. In period 0, Algorithm 1 is manually forced to set the static best response price to Algorithm 2's price. Afterwards, the algorithms play according to their Q-functions. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2. The actions and payoffs have been normalized to each simulation average. The dots represent the median response and the areas around them represent the 25th to 75th percentiles.

As we can see in Figure 2, when the platform is charging a profit fee, sellers obtain higher equilibrium profits. Despite the fact that in the baseline model sellers were already obtaining substantial supracompetitive profits, the active role of the platform allows them to get even closer to the collusive outcome. From the inspection of the deviation and punishment phases of the collusive strategy we can better understand the mechanism. First, in this setting, the deviating seller does not obtain higher profits by deviating but, on the contrary, it gets punished. This happens because the platform immediately intervenes and punishes the defector by excluding it. This action is not only efficient because it provides an external commitment, but also because it allows prices to return *faster* to the collusive level. In fact, the intervention of the platform obviates the need of a punishment phase, since there is no deviation surplus that needs to be compensated with a loss. The last difference with the baseline model consists in a lower dispersion of equilibrium actions and payoffs. From Figure 3, we can see that the platform's active role enables firms to collude on higher profits more consistently, decreasing the variance in the equilibrium prices.

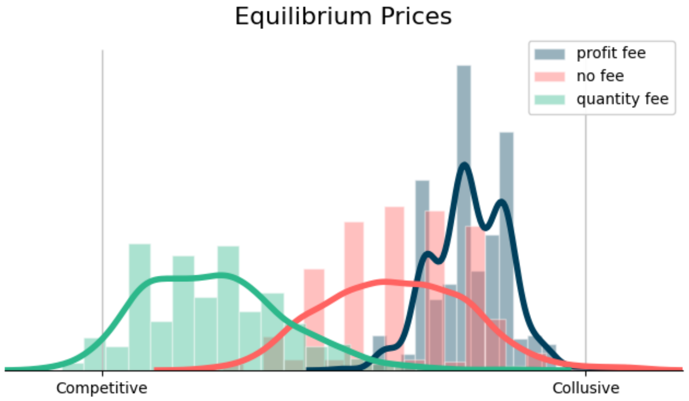


Figure 3: Equilibrium prices of Algorithm 1, over different platform business models. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

When instead the online marketplace is charging a quantity, or per-item, fee, the platform's incentives are aligned with consumer surplus since the platform prefers higher quantities and therefore lower prices. As reported in Section 3, the platform fee is equal to $f^{\text{quantity}} \simeq 0.2$. I report the impulse response function of prices and payoffs from the equilibrium grim-trigger strategy of Algorithm 1 in Figure 4.

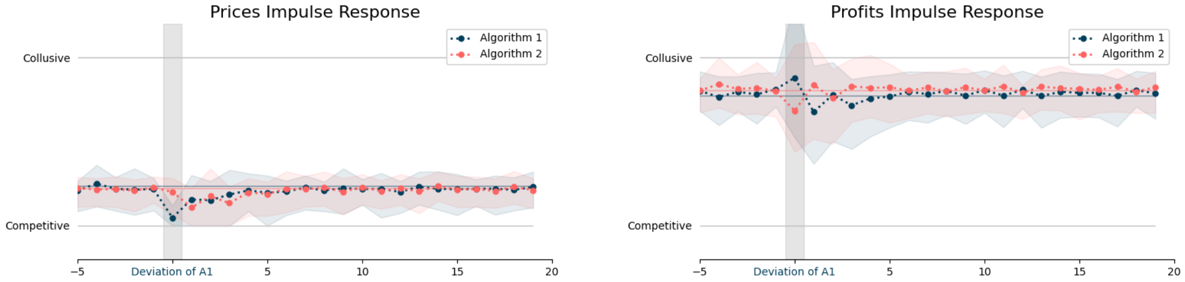


Figure 4: Timelines of prices and profits of both algorithms, when the platform is charging a quantity fee. In period 0, Algorithm 1 is manually forced to set the static best response price to Algorithm 2's price. Afterwards, the algorithms play according to their Q-functions. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2. The actions and payoffs have been normalized to each simulation average. The dots represent the median response and the areas around them represent the 25th to 75th percentiles.

From Figure 4, we can see that, when the platform is charging a quantity fee, it not only has the incentive to keep prices low, but it also manages to do it. Prices are much closer to competitive ones than in the baseline model, and so are sellers' profits. The strategy employed by the platform seems similar: punishing defectors, even though it does it to a lower extent. Lastly, we also observe a higher dispersion of equilibrium prices in Figure 3, as if the platform was able to disrupt the learning process, the opposite of what was happening when the platform was charging a profit fee.

It is also important to note that in no equilibrium outcome the platform excludes a seller. Exclusion only happens off the equilibrium path. Indeed, the platform has no incentives to consistently exclude any seller, since it would gain lower profits.

4.2 Mechanism

In this Section, I am going to further explore the mechanism through which the platform is able to influence sellers' equilibrium actions. I am going to explore two different dimensions in which the platform could be influencing algorithmic collusion: the learning path and equilibrium policies.

First, I explore the equilibrium strategy of the platform. As we already mentioned, the platform does no intervene on-path, rather, as we have seen in Figure 2 and in Figure 4, it intervenes off-path to punish sellers that deviate from equilibrium actions. Now we are going to have a broader look, starting from the setting in which the platform charges a profit fee. In Figure 5, I plot the frequency with which the platform decides to exclude either seller, depending on their actions, across 100 simulations.

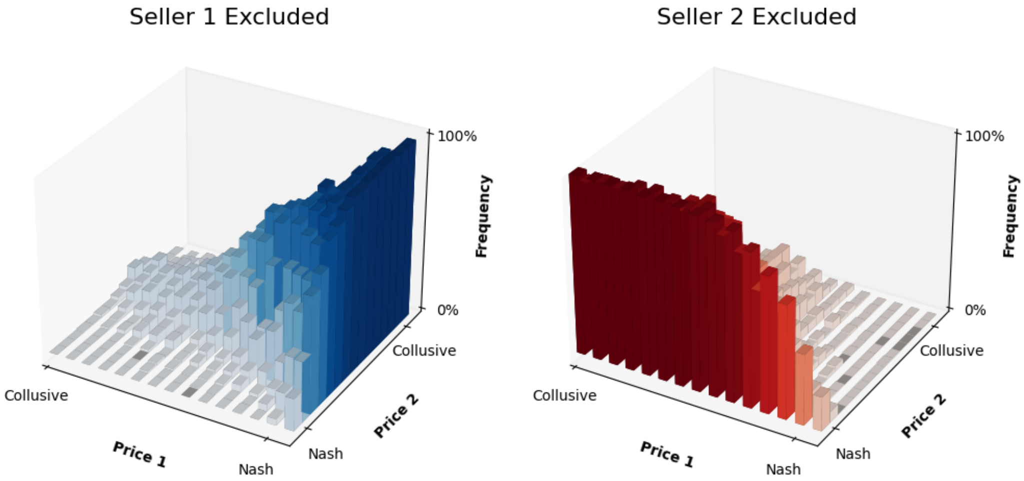


Figure 5: Equilibrium platform's strategy when the platform charges a profit fee. On the left, the exclusion frequency of Algorithm 1, and, on the right, the exclusion frequency of Algorithm 2. Color intensity is proportional to the height of the bars. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

From Figure 5, we can see that the platform employs a symmetrical exclusion strategy for both sellers. In particular, it excludes sellers that play competitive (static Nash equilibrium) prices when their rival is playing a collusive strategy. We observe in general a high frequency of exclusion for low prices and almost no exclusion for high prices. This strategy directly both algorithms towards collusive prices.

We now turn to the setting in which the online marketplace is charging a quantity fee. In this case its incentives are aligned with consumer surplus and we have seen that it is indeed able to steer equilibrium prices towards lower levels. In Figure 6, I plot the frequency with which the platform decides to exclude either seller, depending on their actions, across 100 simulations.

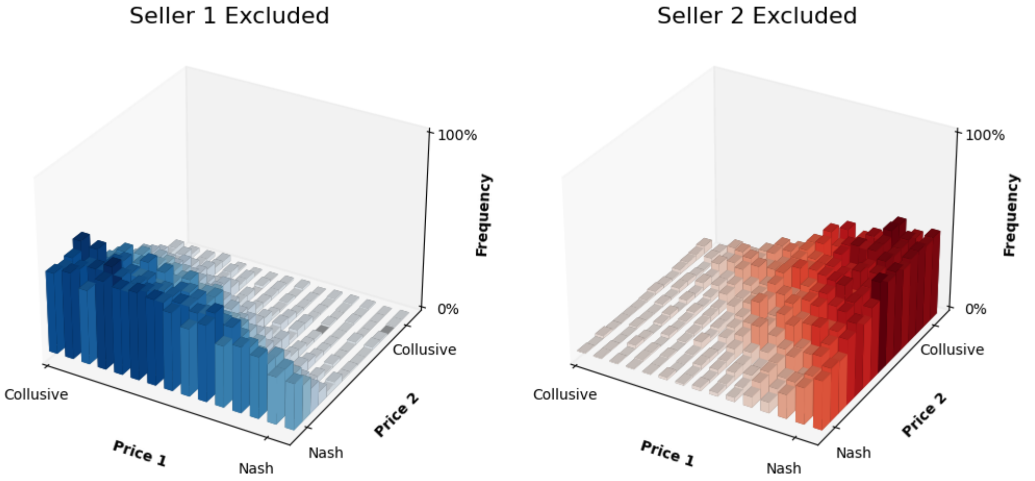


Figure 6: Equilibrium platform's strategy when the platform charges a quantity fee. On the left, the exclusion frequency of Algorithm 1, and, on the right, the exclusion frequency of Algorithm 2. Color intensity is proportional to the height of the bars. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

First, we observe that Figures 5 and 6 are essentially symmetrical. Where before the platform was punishing firms that were undercutting collusive prices, now, when it is charging a quantity fee, the platform punishes sellers that increase the price from the competitive level. This pattern is very stark and, in particular, we see that the platform never punishes undercutting of collusive prices.

These two figures together underline how, depending on the business model of the online marketplace, the platform behaves in completely opposite ways. Its strategy is exactly symmetrical, depending on whether it charges a profit or quantity fee. And, as we have seen in the previous section in Figure 3, these strategies are effective and they translate in symmetrical outcomes.

Lastly, I am going to inspect how the platform can affect the learning processes of the pricing algorithms. In Figure 7, I plot the prices of the two algorithms through the learning process, across different platform business models.

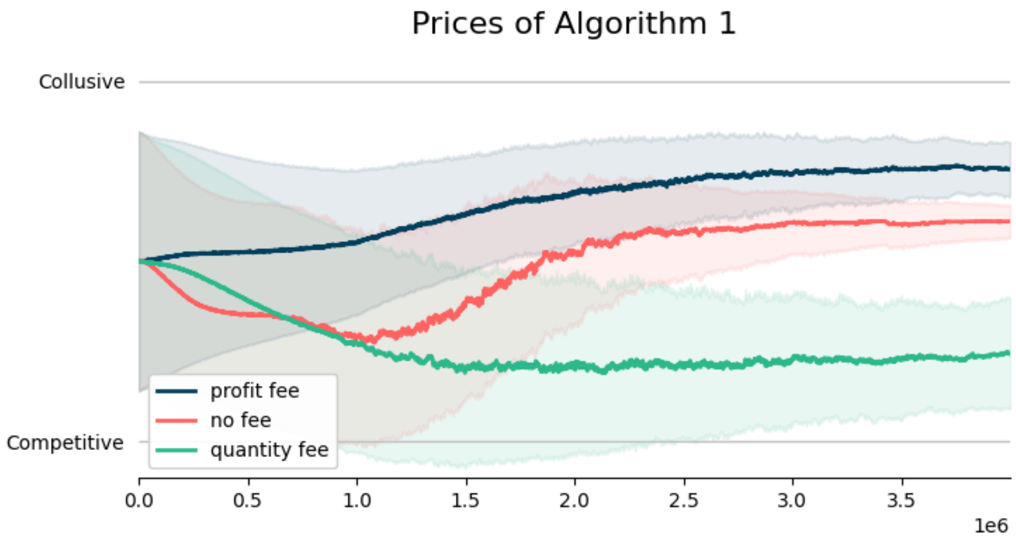


Figure 7: Median prices over time across different platform business models. The confidence bands represent the areas between the 25th and 75th percentile. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

From Figure 7, we observe that the learning process changes significantly depending on the incentives of the platform. In the baseline model, without the platform, the algorithms initially explore lower prices but, at a certain point, start gradually increasing their prices until they converge to supra-competitive actions. The same pattern is present when the platform is charging a profit fee, but it is extremely attenuated. On the other hand, with a quantity fee, it seems that the platform manages to avoid the upward rebound of prices and is able to keep sellers to their lowest explored prices.

4.3 Robustness

In this Section, I briefly show what happens when key parameters of the model are modified. In particular, I inspect changes in three main parameters:

- The value of the outside option a_0, which we can interpret as the market size.

- The value of the platform fee f.
- The elasticity of substitution μ, which we can interpret as the degree of competitiveness of the market.

First, I plot the equivalent of Figure 3 for different values of the outside option. The baseline value is $a_0 = 0$.

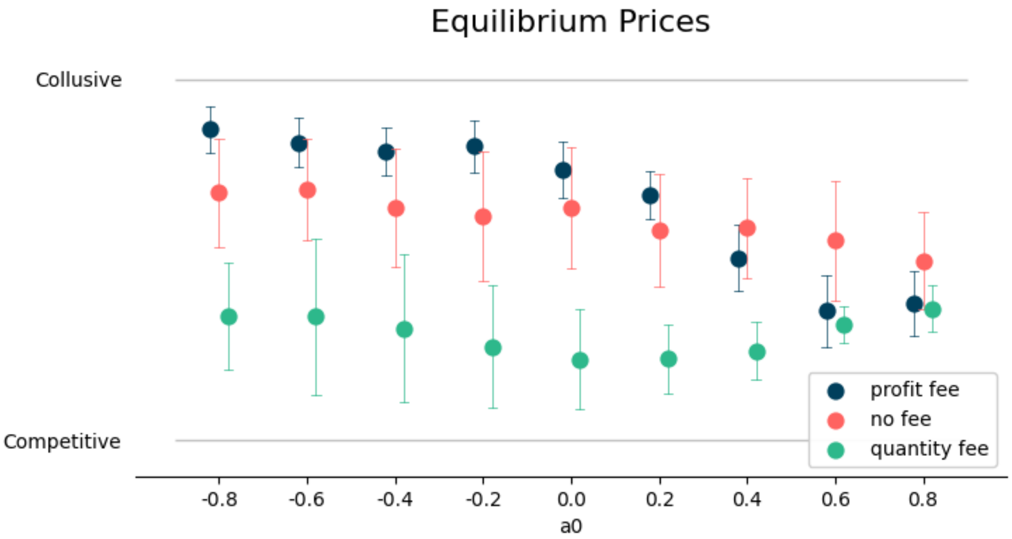


Figure 8: Equilibrium platform's strategy when the platform charges a quantity fee, across different value of the outside option value parameter a_0. On the left, the exclusion frequency of Seller 1, and, on the right, the exclusion frequency of seller 2. Color intensity is proportional to the height of the bars. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

From Figure 8, we see that as market size decreases, the ability of the platform to influence firms' action decreases. In fact, when the market becomes too small and firms' actions become irrelevant, the role of the platform is null.

In Figure 9, I plot the distribution of equilibrium outcomes across different platform fees, for different platform business models. The baseline value is $f = 0.5$.

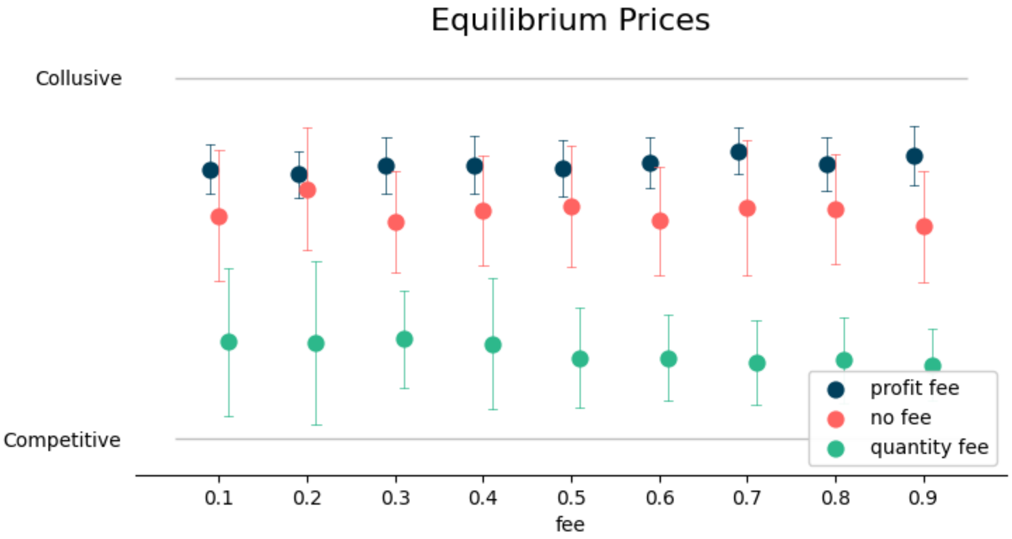


Figure 9: Equilibrium platform's strategy when the platform charges a quantity fee, across different value of the fee parameter f. On the left, the exclusion frequency of Seller 1, and, on the right, the exclusion frequency of seller 2. Color intensity is proportional to the height of the bars. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

Interestingly, it seems that the magnitude of the fee charged by the platform does not play a significant role in determining whether the platform is able to influence algorithmic collusion. Whether the platform extracts 10% or 90% of profits, the outcome is the same.

Lastly, I plot the distribution of equilibrium outcomes across different elasticities of substitution μ, for different platform business models. The baseline value is $\mu = 0.25$.

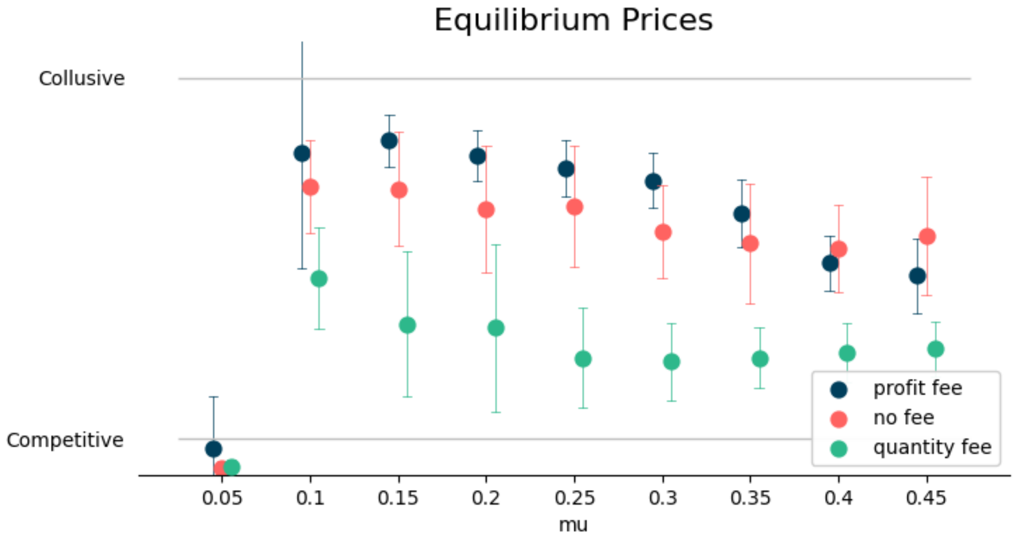


Figure 10: Equilibrium platform's strategy when the platform charges a quantity fee, across different value of the fee parameter μ. On the left, the exclusion frequency of Seller 1, and, on the right, the exclusion frequency of seller 2. Color intensity is proportional to the height of the bars. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

From Figure 10, we observe a different pattern with respect to the two previous figures. As the competition in the market decreases, equilibrium prices decrease. This is a finding consistent with Calvano et al. (2020). If the products are less substitutable, the differences in payoffs are less sharp and algorithms are less likely to learn collusive strategies. At the same time, also the platform is less able to promote collusion, when it is collecting a profit fee. On the other hand, it seems that its ability to discourage collusion when it is collecting a quantity fee remains unchanged.

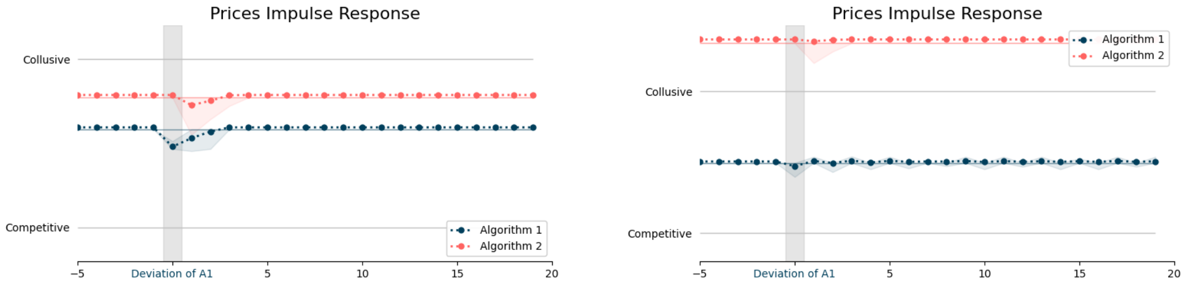
5 Dual Role

In this Section, I investigate what happens when the platform is also active on its own marketplace as a seller. This is the so-called platform "*dual role*". The dual role has been the object of attention of competition authorities recently, because of the conflict of interests that this role might generate.

In this setting, one of the two sellers, Algorithm 1, and the platform are going to maximize a common objective function: the sum of their profits. Both the pricing algorithm and the marketplace's ranking algorithm are going to have the same objective. One major concern with this setting is that, in its baseline formulation, the marketplace would have the incentive to permanently exclude Algorithm 2, the independent seller. In order to avoid this scenario, I am going to always impose a slight malus to the exclusion payoffs so that, for every combination of seller prices, it is never statically more profitable for the platform to exclude any seller. One can interpret this malus as a reputational constraint: even if the platform would gain in the short run from excluding some sellers, it would disrupt its reputation, making it suboptimal in the long run.

5.1 Results

I start by presenting the equilibrum distribution of prices, and the price impulse response to an unilateral deviation of Algorithm 1 in Figure 11.



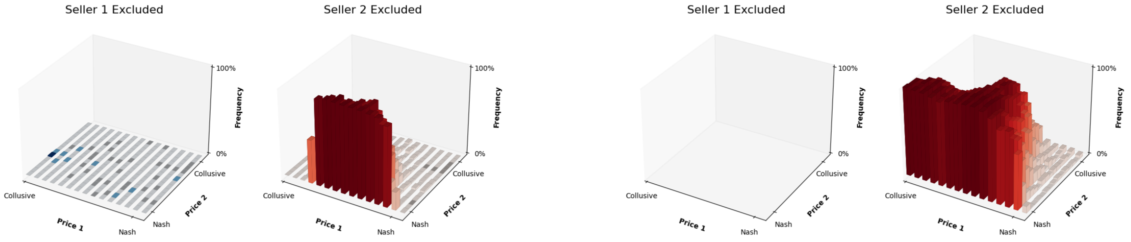
(a) Profit Fee

(b) Quantity Fee

Figure 11: Timelines of prices of both algorithms, with the platform dual role. In period 0, Algorithm 1 is manually forced to set the static best response price to Algorithm 2's price. Afterwards, the algorithms play according to their Q-functions. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2. The actions and payoffs have been normalized to each simulation average. The dots represent the median response and the areas around them represent the 25th to 75th percentiles.

From Figure 11, we see that, when the platform has a dual role, prices are generally higher, with Algorithm 1 slightly undercutting the price of Algorithm 2. This allows the platform to both maximize total profits and to also capture most of them.

But how is the platform able to influence the actions of Algorithm 2 and convince it not to undercut Algorithm 1? The answer is simple: by punishing its every move besides its preferred one. In Figure 12, I plot the strategy of the marketplace, for the two main business models.



(a) Profit Fee

(b) Quantity Fee

Figure 12: Distribution of equilibrium average prices, for different business models of the online marketplace. On the left, the distributions of equilibrium prices in the baseline model, as in Figure **??**. On the right, the distributions of equilibrium prices with the platform dual role, when the objective functions are split. The vertical lines represent the competitive (Nash equilibrium) and the collusive prices. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2.

As we can see from Figure 12, Algorithm 1 is never punished, while Seller 2 is punished whenever it undercuts the price of Seller 1. This strategy allows the marketplace to coerce the independent seller to act according to its preferred strategy.

It is important to notice that in equilibrium one would not observe exclusionary behavior. An observer would only perceive that both sellers charge supracompetitive prices (if the observer has information on the level of competitive prices) with the platform's seller charging a lower price than its competitor. Exclusion only occurs off-path and it is not observed in equilibrium.

5.2 Policy: Split Division

One policy that has been extensively discussed to solve the conflict of interests coming from platform's dual role is the separation of the sale and marketplace divisions (see for example the detailed review by Khan (2019)). A complete separation of the divisions into separate entities would restore the situation described in Section 4. In this Section, I study a different policy: a separation of the two objective functions. I assume that the two entities maximize different objective functions, even though they have the same owner, i.e. the same objective. Importantly, since Algorithm 1 does not pay any fee, the platform would have the incentive to exclude it, since it does not generate profits. In order to avoid this outcome, I am going to assume that the platform acts "*as if*" it was collecting a fee from Algorithm 1 as well, even though it does not.

In Figure **??**, I present the price timeline and impulse response to a unilateral deviation of Algorithm 1, when the platform is charging a profit fee (on the left) and a revenue fee (on the right).

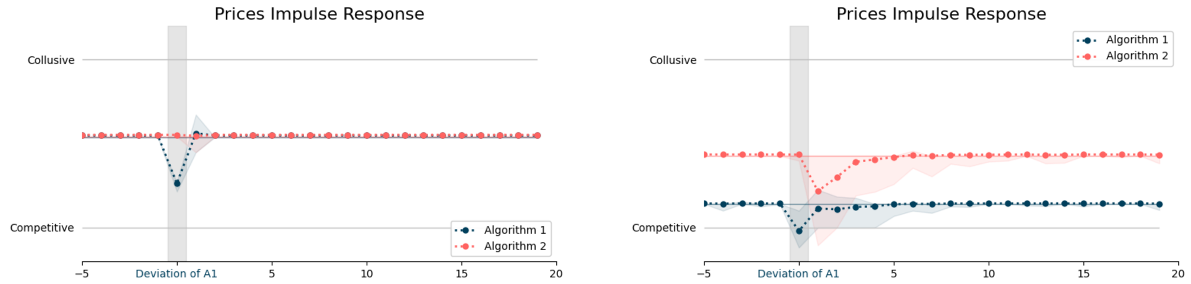


Figure 13: Timelines of equilibrium gross prices (including the platform fee) and impulse response to an unilateral deviation of Algorithm 1, with the platform dual role. In period 0, Algorithm 1 is manually forced to set the static best response price to Algorithm 2's price. Afterwards, the algorithms play according to their Q-functions. The data is produced from 100 simulations of the model described in Section 4 with the parameters reported in Table 2. The actions and payoffs have been normalized to each simulation average. The dots represent the median response and the areas around them represent the 25th to 75th percentiles.

As we can see from Figure 13, when the platform is charging a profit fee, the unbiased but collusive outcome depicted in Figure 2 is restored. However, when the platform is charging a quantity fee, there is an asymmetry between the the two Algorithms' prices. In fact, since the fee is distortive, the platform perceives Algorithm 1 as more efficient and advantages it.

6 Conclusion

In this paper, I have examined the role of online marketplaces in shaping sellers' conduct, when both the platform and the sellers automate their decision making process through the use of reinforcement learning algorithms. The results come from simulations in a controlled environment, therefore they have to be interpreted as possibility results. Nevertheless, they highlight the possibility of platforms being able to prevent or encourage collusive behavior. This is a crucial insight, since it shows that even in automated settings, incentives matter. Even when we are unable to open the black-boxes of modern AI algorithms, understanding their objective functions is crucial. The empirical behavior of reinforcement learning algorithms is often unpredictable *ex-ante*. Also the analysis of their inputs might be misleading in predicting outcomes, but their objective function seems to be the best predictor of their behavior.

Another important insight that comes out of this paper is the danger of having market-makers be active on their own markets, at their own rules. Unless the objective functions are clearly separated, the outcomes can be discriminative towards independent market participants. Moreover, even when the objective functions are separated, one issue remains: the platform's sales division is exempt from fees and this might prevent firms from competing on a level field. This insight calls towards a closer look at the "dual role" of platforms, especially for what concerns incentives. In fact, the observation of equilibrium outcomes might be misleading and hide out-of-equilibrium-path discriminatory behavior.

References

Stephanie Assad, Robert Clark, Daniel Ershov, and Lei Xu. Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. 2020.

Alan W Beggs. On the convergence of reinforcement learning. *Journal of economic theory*, 122(1):1–36, 2005.

Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolo, and Sergio Pastorello. Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10): 3267–97, 2020.

Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolo, and Sergio Pastorello. Algorithmic collusion with imperfect monitoring. *International Journal of Industrial Organization*, 2021.

Alberto Cavallo. More amazon effects: online competition and pricing behaviors. *Harvard Business Review*, 2018.

Le Chen, Alan Mislove, and Christo Wilson. An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1339–1349, 2016.

Australian Competition Consumer Commission. Digital platforms inquiry preliminary report. Technical report, 2019.

Jacques Crémer, Yves-Alexandre de Montjoye, and Heike Schweitzer. Competition policy for the digital era. Technical report, European Commission, 2019.

Mahsa Derakhshan, Negin Golrezaei, Vahideh Manshadi, and Vahab Mirrokni. Product ranking on online platforms. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 459–459, 2020.

Nestor Duch-Brown. The competitive landscape of online platforms. Technical report, JRC Digital Economy Working Paper, 2017.

Jason Furman, Diane Coyle, Amelia Fletcher, Derek McAuley, and Philip Marsden. Unlocking digital competition: Report of the digital competition expert panel. Technical report, 2019.

Mitsuru Igami. Artificial intelligence as structural estimation: Deep blue, bonanza, and alphago. *The Econometrics Journal*, 23(3):S1–S24, 2020.

Justin Johnson, Andrew Rhodes, and Matthijs R Wildenbeest. Platform design when sellers use pricing algorithms. 2020.

Lina M Khan. The separation of platforms and commerce. *Columbia Law Review*, 119(4): 973–1098, 2019.

Timo Klein. Autonomous algorithmic collusion: Q-learning under sequential pricing. *Amsterdam Law School Research Paper*, (2018-15):2018–05, 2019.

OECD. Algorithms and collusion: Competition policy in the digital age, 2017.

Stigler Committee On Digital Platforms. Final report. Technical report, Stigler Center, 2019.

M Schallbruch, H Schweitzer, and A Wambach. A new competition framework for the digital economy. Technical Report 2, 2019.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

UK Competition and Markets Authority. Pricing algorithms, 2018.

UK Competition and Markets Authority. Algorithms: How they can reduce competition and harm consumers, 2021.

Ludo Waltman and Uzay Kaymak. Q-learning agents in a cournot oligopoly model. *Journal of Economic Dynamics and Control*, 32(10):3275–3293, 2008.