Tacit Collusion by Pricing Algorithm with Rule-Based Rivals*

Yongzhi Xu†

Abstract

Pricing algorithms, particularly reinforcement learning algorithms, have been increasingly used by firms in competitive markets, helping them capture more information about the market and their rivals. While prior work has shown that reinforcement learning algorithms can lead to supracompetitive prices in the absence of communication between firms, existing studies largely assume simultaneous adoption by competing firms. Within a framework of price competition between two firms both initially using rule-based strategies, we provide theoretical and simulation evidence that the prices of both firms weakly increase when one firm adopts an algorithm. We also find that the firm using a rule can "free ride" and benefit more from the other firm's adoption. Our findings contribute to the literature by highlighting the importance of the order of algorithm adoption and the transition from rule-based strategies to learning-based algorithms, and demonstrate how tacit collusion can occur in a broader set of circumstances.

Keywords: Algorithmic Pricing, Collusion, Antitrust, Rule-Based Strategy

JEL Classifications: D21, D43, L13, L44

*We are indebted to Yonghong An for his extensive discussions and constant support. We also thank Huiyi Guo, Silvana Krasteva, Jinliang Liu, Yaxin Wang, Hanzhe Zhang, Sijia Zhang and Yu Zhu for helpful discussions. Portions of this research were conducted with the advanced computing resources provided by Texas A&M High Performance Research Computing. All errors are our own.

†Department of Economics, Texas A&M University, College Station, TX, 77843. Email: yz_xu@tamu.edu.

1 Introduction

Pricing algorithms have been increasingly adopted to price goods and services in competitive markets in recent years. Automated software enables firms to adjust prices in real time in response to changes in the market environment and rivals' actions. More recently, reinforcement learning algorithms have been developed and applied to repeated pricing decisions, equipping firms with the ability to learn and extract information about market demand and their competitors' pricing strategies. These learning algorithms facilitate the dynamic updating of both prices and pricing policies.

However, recent empirical and simulation studies have indicated that such learning algorithms may learn to set and sustain supracompetitive prices — prices that are above the competitive level — in the absence of explicit coordination. Due to the fact that such algorithms are not designed to be collusive and do not communicate with each other, current antitrust policy may be insufficient to regulate this form of tacit collusion.

A crucial question related to algorithmic pricing, which has implications for antitrust regulation and remains understudied, concerns the timing of algorithmic adoption. In practice, firms may not adopt pricing algorithms simultaneously. Without direct communication, firms are unable to adopt pricing algorithms at the same time and, to the best of our knowledge, current computational techniques do not allow firms to perfectly discern the type of algorithm their rivals are employing. This raises the question of whether supracompetitive prices will still arise when firms adopt pricing algorithms at different times, as compared to the case when both adopt simultaneously. If the emergence of supracompetitive prices is prevented by sequential adoption, current regulation may be able to address this issue effectively; otherwise, the concern for tacit collusion persists.

This paper examines the outcomes in markets where firms adopt pricing algorithms at different times. We first develop a theoretical model to predict the interactions between pricing algorithms and subsequently introduce Q-learning, a widely used reinforcement learning algorithm, to simulate firms' pricing in practice. The results indicate that market prices

weakly increase after the first algorithm is adopted and supracompetitive prices still arise even when firms adopt pricing algorithms sequentially. This finding suggests that current regulations may be inadequate to prevent such tacit collusion, even when firms are unable to adopt algorithms simultaneously. Moreover, we demonstrate that explicit collusion, where firms share profits, leads to an even higher price level.

We begin by developing an economic model to derive theoretical predictions, focusing on a duopoly market in which firms have limited information about their rivals and the market. For instance, many third-party sellers on platforms such as Amazon typically lack detailed knowledge about other sellers offering similar products; similarly, gasoline stations may not fully capture the market environment or their rivals' strategies. In the absence of learning algorithms, such firms tend to rely on simple rule-based strategies, such as platform presets or endogenous rules with limited information of the market.[1] Traditional competing strategies that are widely used, such as price trigger and grim trigger, can also be considered simple rules, as they are implemented as functions of a rival's price.

We demonstrate that for a broad class of simple rules, prices for both firms weakly increase after one adopts a learning algorithm. A simple rule is defined as a function of the rival's price, and is assumed to be: (1) weakly increasing — so that if one firm raises its price, the rival will not lower its own; (2) weakly below the rival's price — reflecting competitive behavior. The dynamic optimal price for a firm facing a rule-based rival in a repeated simultaneous game is equivalent to the static optimal price in a sequential game where the rule-based rival moves later, as long as the discount factor is sufficiently large. This result holds regardless of the specific structure of the learning algorithm, as long as the algorithm can learn market demand and the rival's strategy and achieves convergence. If a firm is able to completely learn its rival's rule, market prices will not decrease.

[1]For the purpose of this paper, we draw a distinction between a *rule* and an *algorithm*. A rule refers to a static, rule-based pricing strategy, while an algorithm refers to a learning-based algorithm, such as reinforcement learning. Although rule-based strategies are sometimes referred to as "rule-based algorithms" in the literature, we use "rule" to denote these cases and reserve "algorithm" for learning-based strategies throughout this paper.

Our analysis also reveals that the firm using a rule-based strategy may earn higher profits than one employing learning algorithms when competing against each other. However, if neither firm uses an algorithm, their prices will be lower so that both are worse off. Although the equilibrium outcome when both firms use algorithms is undetermined, one can infer that if their profits in this case are lower than in the rule-versus-algorithm scenario, the algorithm adoption game resembles a chicken game with a mixed strategy equilibrium. Therefore, a firm may prefer to continue using a rule if it believes its rival is highly likely to adopt an algorithm, while adoption is preferable if algorithm use by rivals is perceived as unlikely.

The results could also be extended to markets with asymmetric or multiple firms. In markets where there is a price leader and follower firms employing simple rules, the resulting market price corresponds to the outcome where the same leader competes with a representative firm that aggregates all other firms. Such leadership is evident in settings such as the Amazon buy box and gasoline station markets as discussed in Byrne and De Roos (2019). For asymmetric firms, a linear transformation exists between price spaces, so our baseline results continue to apply.

We then provide simulation evidence using Q-learning. Q-learning is designed for Markov decision processes with finite state and action spaces. In a single-agent problem described in our model, the Markov process is stationary and convergence is maintained. Q-learning also links perfectly with the framework of dynamic programming in economics, which provides a natural economic interpretation. It departs from traditional methods, such as value function iteration (VFI) and policy function iteration (PFI), by not requiring full information and permitting policy updates during the learning process.

Simulation results are consistent with our theoretical predictions. We examine four commonly observed rule-based strategies: myopic, undercut, trigger, and ceiling. Adoption of an algorithm by the first firm leads to a weakly increase in prices. If the second firm also adopts, however, the price change becomes ambiguous, while we observe supracompetitive prices. We also extend the time periods to allow the firm that adopts the algorithm later to

revert to use a rule, and the prices converge to the same steady state as before.

To verify the presence of tacit collusion between an algorithm and a rule, we examine the potential circumstance in which one firm deviates by taking a myopic action. We find that firms return to the steady state observed prior to deviation, which ensures the tacit collusion consistent with our theoretical prediction that the optimal price is a steady state such that the algorithm has no incentive to deviate. Such tacit collusion requires no communication between firms, while explicit collusion will raise the price even further. Simulation results confirm this prediction.

We further investigate the robustness of our findings by varying simulation parameters and setups. Starting with less patient firms with lower discount factors, we find that for rules that require higher discount factors, the predicted steady state is no longer observed and the price is lower as expected, while other rules persist the results. We also show that in order to achieve the convergence, some algorithms might require careful parameter and initial value selection.

This paper contributes to three strands of literature. The first is the emerging literature related to algorithmic collusion. Although early work dates back several decades (Sandholm and Crites, 1995; Tesauro and Kephart, 2002; Waltman and Kaymak, 2008), more recent studies have explored algorithmic pricing, focusing on the tacit collusion when algorithms that start at the same time with simulations (Calvano et al., 2020b; Klein, 2021; Johnson et al., 2023; Wang et al., 2023) and experiments (Werner, 2024). Results are also extended to auctions (Banchio and Skrzypacz, 2022) and capital markets (Dou et al., 2025). A noticeable exception is Assad et al. (2024), which studies the effect of hybrid adoption within a market. However, their study focuses on the empirical application of all types of automated pricing software, including both learning algorithms and rule-based strategies. How firms compete when they adopt algorithms gradually remains largely unexplored in the literature. This paper is the first to investigate such sequential adoption and demonstrate that algorithms lead to not only high prices but also price increase in the non-simultaneous adoption setups.

This paper also contributes to the flourishing literature of rule-based strategies. Chen and Tsai (2024), Chen et al. (2016) and Musolff (2024) study automated pricing software on platforms like Amazon. Theoretical work has further examined other characteristics that could lead to tacit and explicit collusion with endogenous rule-based strategies (Salcedo, 2015; Miklós-Thal and Tucker, 2019; Pai and Hansen, 2020; Lamba and Zhuk, 2022; Peiseler et al., 2022; Brown and MacKay, 2023). The only paper that discusses the competition between algorithms and rules is Wang et al. (2023), which compares rule-algorithm versus algorithm-algorithm in a counterfactual, static setting and considers a limited selection of rules. However, their analysis does not address the dynamic process of sequential algorithm adoption, nor does it systematically examine the properties of rule-based strategies. This paper fills this gap by investigating how reinforcement learning algorithms compete with rule-based rivals across a broad class of rules, and demonstrates that such interactions can lead to increases in market prices.

The third strand of literature concerns the regulatory implications of algorithmic collusion. A key prerequisite for designing effective policy interventions is to understand different market scenarios in which supracompetitive pricing may arise. While previous studies discuss the cases where algorithms fail to converge to supracompetitive prices (Asker et al., 2024; Possnig, 2023; Bichler et al., 2024) and propose potential regulations (Harrington, 2018; Calvano et al., 2020a; MacKay and Weinstein, 2022; Leslie, 2023; Johnson et al., 2023; Spann et al., 2025), the literature has largely overlooked the sequential and hybrid adoption of pricing algorithms. By analyzing the pricing effects resulting from sequential adoption and competition with rule-based rivals, this paper establishes a foundation for further research on the assessment and development of effective regulatory policies in markets where algorithms are increasingly prevalent.

The outline of the remaining paper is as follows. Section 2 presents a model to describe competition between firms and theoretical evidence on how algorithms respond to rules. Section 3 summarizes Q-learning, a reinforcement learning method. Section 4 discusses the

simulation results. Section 5 extends the results, and Section 6 concludes.

2 Economic Model

In this section, we develop an economic model to derive theoretical predictions. We focus on a duopoly market in which firms have limited information about both their rivals and the market environment. We first introduce the model setup and key assumptions, and then analyze the steady state and demonstrate the price increase when the first algorithm is adopted. Finally, we discuss the adoption game and extend our analysis to markets with asymmetric or multiple firms.

2.1 Model Setup

We start with the case where no firm uses a reinforcement learning algorithm. Consider an infinitely repeated incomplete information game in which $n = 2$ symmetric firms act simultaneously.[2] Time periods are indexed discretely by $t \in \{1, 2, \dots\}$. In each period t, firm i earns a profit

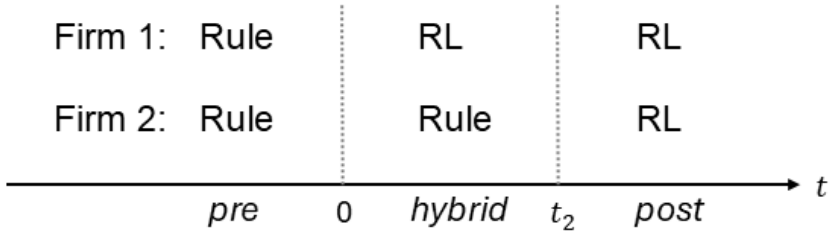
$$\pi_i(p_{i,t}, p_{j,t}) = (p_{i,t} - c) * q_{i,t}(p_{i,t}, p_{j,t}),$$

where $c = 1$ is the constant marginal cost, and $q_{i,t}$ is the demand function. Suppose that firms are not able to respond optimally since they have limited information about their rival and the market without the help of a learning algorithm. Before time 0, firms were competing with a simple rule $p_{i,t} = g_i(p_{j,t-1})$, which is a pure strategy.[3] At time 0, without loss of generality, firm 1 adopts a reinforcement learning algorithm. Firm 2 has no information about this and keeps the same rule until t_2. We call $t < 0$ the pre stage, $0 \leq t < t_2$ the hybrid stage, and $t \geq t_2$ the post stage. The process is shown in Figure 1.

[2] We will discuss the market with asymmetric firms in Section 2.2.

[3] This rule could be exogenous or endogenous.

Figure 1: The timing of firms' adoption of algorithm



Notes: Rule stands for rule-based strategy. RL stands for reinforcement learning algorithm.

Hybrid stage In the absence of learning algorithms, i.e., in the pre stage, suppose that the prices firms charged at time 0 are $s_0 = (p_{1,0}, p_{2,0})$. When $t = 1$, firm 1 starts to use a reinforcement learning algorithm to help it maximize its discounted profit given state variable being the price of both firms in the last period. Suppose that firm 2 has no information about this so its policy function is still g_2. The value function of firm 1 is given by

$$V_1(s) = \max_{p \in \mathcal{P}} \left\{ \pi_1 \left(p, g_2(s) \right) + \delta V_1(s') \right\},$$

where state $s' = (p_1, p_2)$ and δ is the discount factor. We can see from the above equation that if $g_2(p_1, p_2) = g_2(p_1)$ then the state variable could be collapsed to $s' = p_1 = (p_1, g_2(p_1))$. By the contraction mapping theorem, there exists a unique $V^*(s)$ that solves the above equation, and a reinforcement learning algorithm starting with arbitrary V^0 will converge to V^*. Before digging into the V^*, let's introduce some assumptions first.

Assumption 1. *Define p^N to be the static Bertrand Nash equilibrium price, and p^M to be the static monopoly price, i.e.,*

$$p^N = \arg\max_p \pi_1 \left(p, p^N \right)$$

$$p^M = \arg\max_p \pi_1 (p, p)$$

For any $p \in \mathcal{P} = \left[p^N, p^M \right]$

(i). $\pi_1(p_1, p_2)$ *is strictly concave.*

(ii). $\pi_1^M(p) = \pi_1(p, p)$ *is strictly increasing w.r.t.* p.

(iii). $g_2(p) \leq p$.

(iv). $g_2(p)$ *is continuous and weakly increasing w.r.t.* p.

(v). $q_1(p_1, p_2)$ *is weakly increasing w.r.t.* p_2.

The assumptions above are satisfied in most theoretical models. Assumption 1(i). is a typical assumption that states concave profit function, and Assumption 1(ii). is a weaker condition of concave monopoly profit. Both apply to most of the commonly used demand functions. Assumption 1(iii). means that the strategy is "competitive". It's usually irrational to choose a price that is higher than the rival in a steady state if the firm has no information about its rival's pricing strategy. Note that this assumption only applies when the firm has limited information about the rival's strategy. For example, it would happen that $p_1^* > g_2(p_1^*)$ if firm 1 has complete information as shown by results from our simulations. Assumption 1(iv). means that since firms have limited information, it is better to follow the rival and raise price when their increase. Most rule-based strategies adopted by firms satisfy this assumption. We will introduce some of them in Section 4. Assumption 1(v). means that the products are substitutes and therefore there exists competition.

Lemma 1. *Under Assumption 1(iii)., there exists a steady state* $p_1 = p_2 = p_0$.

Lemma 1 ensures that there always exists at least one steady state in the pre stage. Notice that we didn't assume that the rules used by two firms in the pre stage are the same. We instead relax the assumption and show that there exists at least one price such that the two rules interact and are the response to each other. In our assumptions, the price pair (p^N, p^N) is always a steady state such that $p^N = g_2(p^N)$.

2.2 Steady state in The Hybrid Stage

In this subsection, we examine the steady state in the hybrid stage. Let g^* be an optimal policy such that

$$g^*(s) \in \arg\max_{p \in \mathcal{P}} \{\pi_1(p, g_2(s)) + \delta V^*(s')\}$$

We can see that a steady state is when $s = s' = g^*(s) = p_1^*$. In other words, the steady state is the fixed point of the function g^*.

Proposition 1. *Let p^* be the static optimal price of firm 1 given firm 2's strategy g_2, such that*

$$p^* := \arg\max_{p} \pi_1(p, g_2(p))$$

Then there always exists a $\delta_0 \in [0, 1)$ such that $\forall \delta \in (\delta_0, 1)$, p^* is the unique fixed point of g^*, i.e., $p_1^* = p^*$ is the unique steady state.

Proposition 1 links the static optimal price with the dynamic optimal price. It ensures that when δ is large enough, the dynamic optimal price coincides with the static optimal price p^*.[4] In other words, firm 1 has no incentive to deviate from p^* to a lower price, and therefore a steady state exists.

Notice that Proposition 1 only ensures that firm 1 will not deviate from p^* and will always deviate from another steady state, without guaranteeing that the price will also converge to p^*. The algorithm could converge to a policy function that leads to a cycle in general.[5] A sufficient condition of V^* leads to a p^* is that g^* is a contraction, so that by the Banach fixed point theorem, there exists a unique fixed point s^* such that $g^*(s^*) = s^*$ starting from an arbitrary s^0.

[4] Automatic pricing software can update prices very frequently. Therefore, the discount factor δ is typically large.

[5] One example is the zero-sum match pennies game that player 1 loses if the pennies match and wins otherwise. If player 2 always chooses the action played by player 1 in the last period, then the best response of player 1 is to switch action in each period and therefore there's no steady state.

Proposition 2. *A sufficient condition of the reinforcement learning algorithm converges to a stationary point is*

$$\left|\frac{\mathrm{d}g^*}{\mathrm{d}s}\right| = \left|-\frac{\frac{\partial^2\pi}{\partial a\partial p_2}(g^*(s), g_2(s))g_2'(s)}{\frac{\partial^2\pi}{\partial a^2}(g^*(s), g_2(s)) + \delta V''(g^*(s))}\right| < 1$$

for $s \in S^* = \{s|\exists s' \text{ s.t. } g^*(s') = s\}$.

In practice, it's not necessary for researchers to check the condition in Proposition 2, and sometimes it's not possible to check the functional form. Although algorithms could converge to a cycle instead of a steady state, Proposition 1 guarantees that if it does converge to a steady state, it must be p^* because of the uniqueness. Therefore, a more practical way is to check the realized convergence.

With the information that an algorithm will converge to p^*, we can now compare the price in the hybrid stage with that in pre stage and examine the consequence of adopting an algorithm.

Proposition 3. *Under Assumption 1,* $p^* \geq p_0$ *and* $g_2(p^*) \geq p_0$.

At the steady state p^*, for any available price $p' \in \mathcal{P}$, we have

$$\pi_1(p^*, g_2(p^*)) \geq \pi_1(p', g_2(p')),$$

and equality holds if and only if $p^* = p'$. Thus, the optimal price of firm 1 will be p^* instead of p_0 and the new steady state is $s^* = (p^*, g_2(p^*))$.

Therefore, if the algorithm is able to fully learn the rival's strategy, then there will be a price increase in the hybrid stage compared to pre stage. Notice that we do not require a specific algorithm to reach this conclusion. It instead holds for all algorithms that will converge to the optimal policy g^*.[6]

[6]Although not the focus of this paper, it is worth noting that the price increase is driven by the full information of firm 2's rule (or policy function). Therefore, any algorithm that could learn the rule g_2 or the optimal policy function g^* will lead to a price increase. Revealing its own rule may be beneficial, but it could raise potentially antitrust concerns.

With the price increase in the market, both firms benefit from the adoption of the learning algorithm of firm 1. We should see that, however, the two firms do not equally share the market in the steady state. Firm 2 still has more market share than firm 1 due to its lower price. Therefore, even if firm 1 is the firm that adopts the algorithm, firm 2 actually free rides and gets more benefit than firm 1.

So far, we have considered a market in which firms share the same price space. It is more realistic, however, that firms have asymmetric price spaces. For example, firms producing identical goods but in different quantities, such as one wholesale and one retail firm, or firms with asymmetric cost, have distinct profit functions and therefore different price space. Firms producing differentiated products will have asymmetric demand functions and profit functions.

In general, suppose that firms' prices satisfy $p_1 \in \mathcal{P} = [p_1^N, p_1^M], p_2 \in \tilde{\mathcal{P}} = [p_2^N, p_2^M]$. Firm 2 set its price according to a simple rule $g_2(p_1)$. We impose the following assumption in this setting. Let $h : \tilde{\mathcal{P}} \mapsto \mathcal{P}$ defined as

$$h(\tilde{p}) = p_1^N + \frac{p_1^M - p_1^N}{p_2^M - p_2^N}\left(\tilde{p} - p_2^N\right)$$

and $\tilde{g} \coloneqq h \circ g_2$. Then $\tilde{p}_2 = h(p_2) = h(g_2(p_1)) = \tilde{g}(p_1) \in \mathcal{P} = \left[p_1^N, p_1^M\right]$. We can then impose Assumption 1 on $\tilde{q}_i(p_{i,t}, \tilde{p}_{j,t}) = \tilde{q}_i(p_{i,t}, \tilde{g}(p_{i,t})) \coloneqq q_i(p_{i,t}, p_{j,t})$ and $\tilde{\pi}_i(p_{i,t}, \tilde{p}_{j,t}) = \tilde{\pi}_i(p_{i,t}, \tilde{g}(p_{i,t})) \coloneqq \pi_i(p_{i,t}, p_{j,t})$. Therefore, Proposition 3 remains for p_1 and $\tilde{p}_2$ and we will still observe a price increase.

2.3 Adoption Game

We now turn to the adoption game, where firms decide whether to adopt pricing algorithms given the expected outcomes in the hybrid stage. In a market where both firms are using rule-based strategies, the steady state is p_0. Then the profit gain from adopting the algorithm for firm 1 is given by $\pi_1^* \coloneqq \pi_1(p_1^*, g_2(p_1^*)) \geq \pi_0 \coloneqq \pi_1(p_0, p_0)$. The profit gain of firm 2 is $\pi_2^* \coloneqq \pi_2(p_1^*, g_2(p_1^*)) \geq \pi_1(p_1^*, g_2(p_1^*))$, since $g_2(p_1^*) \leq p_1^*$. If we assume that the expected profit

of both firms when they are both adopting the algorithm is $\bar{\pi}$, the payoff matrix is then shown in Table 1.[7]

Table 1: Payoff matrix of the adoption game.

		Firm 2	
		Rule	Algorithm
Firm 1	Rule	(π_0,π_0)	(π_2^*,π_1^*)
	Algorithm	(π_1^*,π_2^*)	$(\bar{\pi},\bar{\pi})$

There exists a pure strategy Nash equilibrium (Algorithm, Algorithm) if $\bar{\pi} \geq \pi_2^*$. Otherwise, there exists a mixed strategy Nash equilibrium in which each firm adopts an algorithm with probability $(\pi_1^* - \pi_0)/(\pi_1^* + \pi_2^* - \pi_0 - \bar{\pi})$ and continues using rule with probability $(\pi_2^* - \bar{\pi})/(\pi_1^* + \pi_2^* - \pi_0 - \bar{\pi})$. In this case, if one firm believes that the rival will adopt an algorithm with high probability, the best response is to keep using the current rule. If instead it believes that the rival is not likely to adopt an algorithm, the best response is to start adopting.

2.4 Multiple Firms

The results in Section 2.2 could be extended to markets with multiple firms. Suppose that there are n firms in the market. Firm 1 is the firm that leads the market, and decides whether to adopt an algorithm.[8] Other firms are using an simple rule $p_i = g_i(p_1)$. We can treat firms 2 to n as a representative firm whose demand function and profit function is defined as

$$q_{-1}(p_1, p_{-1}) = \sum_{i=2}^{n} q_i(p_1, p_2, \ldots, p_n)$$

[7]Here we use the current period profit for simplicity of notation. The discounted profit is given by $1/(1-\delta)$ times the current period profit and therefore would not affect the equilibrium.

[8]The Amazon buy box is one of the examples. Many automated software has an option to target only the buy box price. This price leadership also exists in markets with coordination as in Byrne and De Roos (2019)

$$\pi_{-1}(p_1, p_{-1}) = \sum_{i=2}^{n} p_i * q_i(p_1, p_2, \ldots, p_n)$$

where the notation -1 stands for the representative firm aggregating all firms except firm 1. We can then define the price of the firm -1 as

$$p_{-1} = g_{-1}(p_1, p_2, \ldots, p_n) = \frac{\sum_{i=2}^{n} p_i * q_i(p_1, p_2, \ldots, p_n)}{\sum_{i=2}^{n} q_i(p_1, p_2, \ldots, p_n)}$$

We will extend the assumptions in Assumption 1 to markets with multiple firms.

Assumption 2. *For any* $p \in \mathcal{P} = \left[p^N, p^M\right]$

(i). $\pi_1(p_1, p_2, \ldots, p_n)$ *is strictly concave.*

(ii). $g_i(p) \leq p, \forall i > 1.$

(iii). $\pi_1^M(p) = \pi_1(p, p, \ldots, p)$ *is strictly increasing w.r.t.* p.

(iv). $g_i(p)$ *is continuous and weakly increasing w.r.t.* $p, \forall i > 1.$

(v). $q_i(p_1, p_2, \ldots, p_n)$ *is weakly increasing w.r.t.* p_j $\forall j \neq i.$

It can then be derived that the market with firm 1 and -1 satisfies Assumption 1 when the demand and profit functions are defined as above. Assumption 2(ii). and 2(v). can be extended directly since the sum of concave (resp. increasing) functions preserves concavity (resp. monotonicity), and Assumption 2(iii). follows from the fact that $p_{-1} = p_i$ if g_i is identical across firms. Assumption 2(ii). and 2(v). can be derived by noting that p_{-1} is a weighted average of p_i with positive weights and thus g_{-1} is a finite linear combination of g_i. Therefore, proposition 3 holds in the shadow market. Moreover, the prices of firms 2 to n also increase given $g_i(p^*) \geq g_i(p_0)$.

Proposition 4. *Under Assumption 2,* $p^* \geq p_0$ *and* $g_i(p^*) \geq p_0, \forall i > 1.$

3 Q-learning

The reinforcement learning algorithm we employ is Q-learning. This algorithm targets the Q-function and learns iteratively by realized rewards. For an economic perspective, the Q-function produces the discounted profit given state and action, which can be interpreted as the choice specific value function in dynamic programming. Unlike value function iteration (VFI), policy function iteration (PFI), or other dynamic programming methods that require full information about the environment and rich data, Q-learning can learn while making decisions and updates policies in real time without requiring a closed-form solution to the Bellman Equation.

Consider first a single agent problem with a stationary Markov decision process. In each period $t = 0, 1, 2, \ldots$, an agent observes a state $s_t \in S$ and then chooses an action $a_t \in A$. Both S and A are finite and time-invariant and A is state-independent. Agent receives a payoff $\pi_t = \pi(s_t, a_t)$, which could be stochastic, at each period t, then the system moves on to the next state $s_{t+1} \in S$.

Let $a^*(s)$ represent an optimal policy. The decision-maker's problem is to maximize the expected present value of discounted payoff:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t \pi_t\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t \pi(s_t, a^*(s_t))\right],$$

where $\delta < 1$ represents the discount factor. Let $V(s)$ denote the value of being in state s

$$V(s) = \max_{a \in A} \Big\{ \mathbb{E}[\pi|s, a] + \delta\, \mathbb{E}[V(s')|s, a] \Big\},$$

which represents the maximum discounted payoff in state s, and $Q(s, a)$ be the choice-specific value function

$$Q(s, a) = \mathbb{E}[\pi|s, a] + \delta\, \mathbb{E}\left[\max_{a' \in A} Q(s', a')|s, a\right],$$

which represents the future discounted payoff of taking action a at state s and choosing the optimal policy function $a^*(s)$ in the future. Notice that Q-function is related to the value

function by

$$V(s) = \max_{a \in A} Q(s, a).$$

Q-learning could deal with the case where both state and action are finite. Note that in such case, Q-function collapses to a matrix. If the Q-matrix were known, the optimization problem could be solved by searching the maximizer of the specific row of Q-matrix corresponding to state s, or

$$a^*(s) = \arg\max_{a \in A} Q(s, a).$$

Therefore, as long as the Q-matrix is known, without knowing any underlying model, the agent is able to solve the optimization problem. Q-learning is an algorithm that estimates the Q-matrix using the following iterative procedure without model-based assumptions, i.e., it is model-free.

The Q-learning algorithm proceeds as follows. Starting from an arbitrary initial matrix Q_0, the algorithm chooses an action a_t at state s_t for each time period t. After observing the payoff π_t, the algorithm updates one cell of Q-matrix according to the following learning rule:

$$Q_{t+1}(s, a) = (1 - \alpha) Q_t(s, a) + \alpha \left[\pi_t + \delta \max_{a \in A} Q_t(s', a) \right],$$

where the weight $\alpha \in [0, 1]$ is a step-size parameter, which determines the learning rate. For all other cells $s \neq s_t$ and $a \neq a_t$, the Q-value does not change. Since α is constant, the algorithm puts the same weight on recent observations and the information from early observations diminishes over time.

However, the agent may stuck to a suboptimal policy during the learning process above. For example, if the agent doesn't have a good expectation of Q-matrix, it's very likely that the initial matrix Q_0 is very different from Q. For state s and an action $a' \neq a^*$, such that a' is not the optimal action at state s and $Q(s, a') < Q(s, a^*)$, if $Q_0(s, a^*) < Q(s, a') < Q_0(s, a')$, it is very likely that the algorithm prefers a' to a^* and never gets the chance to update $Q_0(s, a^*)$. If so, the algorithm will be stuck at a' and never learn that a^* is the optimal action at state s.

In order to estimate a^* and Q-matrix from an arbitrary initial matrix Q_0, the algorithm should be allowed to "make mistakes", or to explore non-optimal actions. The method we use in our analysis is the ε-greedy model, which works as follows. The algorithm exploits (chooses the currently optimal action) with probability $1-\varepsilon^t$ and explores (randomizes uniformly across all actions) with probability ε^t. The probability ε^t decays with time and is assumed to be $\varepsilon^t = e^{-\beta t}$, with $\beta > 0$.

In a repeated game described in Section 2, however, stationarity is usually not satisfied. The state transition depends on the previous or current action of all players. Although convergence is not guaranteed ex ante, it can be verified ex post. In our simulation, convergence is achieved if, for all players, their current optimal policy functions do not change for 10^5 consecutive periods. It diverges if it does not converge after 10^8 periods.

To be specific, we started with the initial Q_i^0 set for each firm and an initial state s^1. At the beginning of each period, each firm chooses an action based on its current Q_i^t and state s^t by $a_i^t = \arg\max_a Q_i^t(s^t, a)$ with probability $1-\varepsilon^t$, and chooses a random action uniformly from A with probability ε^t. We then compare the optimal actions with the recorded actions.[9] If they match for 10^5 consecutive periods for all firms, the algorithms converge and we stop the simulation. Otherwise, we assume that firm has only one period of memory and the state in the next period is defined as $s^{t+1} = (a_1^t, \ldots, a_n^t)$. The Q-matrix is then updated by

$$Q_i^{t+1}(s^t, a_i^t) = (1-\alpha)Q_i^t(s^t, a_i^t) + \alpha\left[\pi_i^t + \delta \max_{a\in A} Q_i^t(s^{t+1}, a_i^t)\right]$$

where $\pi_i^t = (a_i^t - c) * q_i^t$ is the profit of firm i at period t. Table 2 provides the visualized procedure of Q-learning in our simulation in a pseudo-code form.

4 Simulation Results

This section presents the simulation setup and main findings. We begin by detailing the parametrization of the model, including key market demand and pricing algorithm param-

[9] We are comparing the optimal actions without exploration.

Table 2: Q-learning: pseudo code

Initialize $Q_i^0(s, a)$

$t = 1, s^1 = \text{random}$

record actions $Opt_i(s)$

while $t < 10^8$

$$a_i^t = \begin{cases} \arg\max_a Q_i^t(s^t, a) & \text{with prob. } 1 - \varepsilon^t \\ \text{random} & \text{with prob. } \varepsilon^t = e^{-\beta t} \end{cases}$$

if $a_i^t == Opt_i(s^t)$ for 10^5 continuous periods

break

else

$Opt_i(s^t) = a_i^t$

end if

$s^{t+1} = (a_1^t, \ldots, a_n^t)$

$Q_i(s^t, a_i^t) = (1 - \alpha)Q_i(s^t, a_i^t) + \alpha \left[\pi_i^t + \delta \max_{a \in A} Q_i(s^{t+1}, a_i^t)\right]$

end while

---

eters. We then present four rule-based strategies — myopic, undercut, trigger, and ceiling — that firms employ before any algorithm is adopted. Finally, we report the main results, organized into four parts: the hybrid stage, the post stage (including the retrieve stage), the deviation analysis that examines firms' responses to exogenous shocks, and the shared profit counterfactual, which explores outcomes under explicit coordination.

4.1 Parametrization

In our simulation, we follow the assumption that the firms are symmetric, and share a constant marginal cost $c = 1$ as in Calvano et al. (2020b). Each firm produces one product that is differentiated and an outside option is available. To be specific, the vertical differentiation

$\gamma_i = 2, \gamma = 0$, and horizontal differentiation $\mu = 1/4$; the logit demand is

$$q_{i,t}(p_{i,t}, p_{j,t}) = \frac{e^{\frac{\gamma_i - p_{i,t}}{\mu}}}{\sum_{j=1}^{n} e^{\frac{\gamma_j - p_{j,t}}{\mu}} + e^{\frac{\gamma_0}{\mu}}}.$$

As for the parameters in the algorithms, the initial Q-matrix is set to the discounted payoff that would accrue to player i if rivals randomized uniformly:

$$Q_{i,0}\left(s, a_i\right) = \frac{\sum_{a_{-i} \in A^{n-1}} \pi_i\left(a_i, a_{-i}\right)}{(1-\delta)|A|^{n-1}}.$$

Notice that the initial value of the algorithm does not consider the fact that the rival's action a_{-i} is affected by the state, and the state transition depends on the actions. That is, the algorithm is not cooperative by design, and will charge low prices to maximize its own profit. As for learning parameters, we focus on $\alpha = 0.05$ and $\beta = 10^{-6}$. For each rule that firm 2 uses, we run 1000 sessions.

We discretize the action set into $A = \{p^1, \ldots, p^{15}\}$ of 15 equally spaced prices, where $p^2 = p^N$ and $p^{14} = p^M$. In this setup, the Bertrand price is about 1.472 and the monopoly price is about 1.925. This is slightly different from the setup in Calvano et al. (2020b), as we want the Bertrand price and monopoly price to be available to the firms. To have a clearer view of the result, we will use the price grid instead of the absolute price value in the rest of this section, i.e., p^2 stands for 1.472 and p^{14} stands for 1.925. We also normalize the profit by

$$\Delta_i := \frac{\pi_i - \pi^N}{\pi^M - \pi^N}$$

where π_i is the average profit of firm i upon convergence, π^N is the profit in the static Bertrand-Nash equilibrium, and π^M is the profit under full collusion (monopoly), so that $\pi_N = \pi_1(p^N, p^N)$ and $\pi_N = \pi_1(p^M, p^M)$.

4.2 Rule-based Strategies

We take four most widely used rules as examples in the simulations: myopic, price undercut, price trigger, and price ceiling. In the results hereafter, we also use rule names to represent

the simulations that firm 2 is using the specific rule in the pre and hybrid stage. The rules are defined as follows:

Myopic The myopic rule is applied when the firm only focuses on the current period profit instead of the discounted profit, defined as follows:

$$p_i(p_{j,t-1}) = \arg\max_p \pi_{i,t}(p, p_{j,t-1})$$

It is a rule that the rival is not seeking cooperation but just maximizing its own profit. In the steady state, the prices are equivalent to those in a sequential game where the algorithm moves first. The firm using rule, which moves later, therefore has a second-mover advantage and will charge lower price as described in Gal-Or (1985) and Amir and Stepanova (2006). The myopic rule requires that the firm knows not only the Bertrand price and monopoly price, but also the market demand, therefore it is not widely seen as preset rules in automated pricing software. However, it is still a good example to show how algorithms could seek tacit collusion with an endogenous rule.

Undercut The undercut rule is the case where the firm knows nothing about the market so it will just follow the rival's price with a minimal price undercut. It can be specified as follows:

$$p_i(p_{j,t-1}) = \begin{cases} p_{j,t-1} - \Delta p & \text{if } p_{j,t-1} > p^N \\ p^N & \text{o.w.} \end{cases}$$

where Δp is the price grid step, and therefore $p_{j,t-1} - \Delta p$ is the highest price that is below $p_{j,t-1}$. In the standard Bertrand setup where the firm with lower price takes the whole market, price undercut is the static optimal strategy. Although it is usually not optimal in a market with continuous demand, this rule is widely used in automatic pricing software in competitive markets, given its simplicity and straightforward intuition.

Trigger The trigger rule is a typical dynamic strategy where a firm charges the monopoly price if the rival also does, and charges the Bertrand price otherwise. It's described as follows:

$$p_i(p_{j,t-1}) = \begin{cases} p^M & \text{if } p_{j,t-1} = p^M \\ p^N & \text{o.w.} \end{cases}$$

Unlike the previous two rules, which are the optimal strategies and best response functions in some static games, the trigger rule is an optimal strategy in some dynamic games. Therefore, there exist circumstances that firms would end up with supracompetitive prices, and that the firm using trigger rule is seeking cooperation through punishment and reward. It's also well known as tit-for-tat since we assume that firms have only one-period memory. Green and Porter (1984) and Abreu (1988) have demonstrated that price trigger strategy could form a tacit collusive equilibrium.

Ceiling Firms using the ceiling rule will choose the same price as their rival with a price ceiling, which is defined as follows:[10]

$$p_i(p_{j,t-1}) = \begin{cases} p_{j,t-1} & \text{if } p_{j,t-1} < p^C \\ p^C & \text{o.w.} \end{cases}$$

The ceiling rule is not commonly seen. The firm using this rule will choose to price the same as the rival if the price is below some price ceiling $p^C < p^M$, and will price at the ceiling otherwise. This happens when the firm is using an automatic pricing software but fails to realize the highest price it can charge. In the simulation, we pick $p^C = p^7$. The ceiling rule is used as a robustness check because it is the rule that firm 1 yields the lowest profit when price is greater than p_0.

All four rules mentioned above can lead to a steady state where both firms price the same as static Bertrand price as discussed in our theoretical model. Trigger could also lead to the static monopoly price, if firms start at some specific state, while ceiling could end up

[10] It collapses to a constant price if $p^C = p^N$.

charging any price smaller than or equal to p^C. These steady states are not guaranteed to be equilibrium, especially with logit demand. Firms, however, are not able to deviate since they can only observe their rival's action but not strategy as assumed.

Besides the timing mentioned in Figure 1, we also include the results when firm 2 retrieves the adoption of the algorithm in the post stage after convergence and uses a rule again. Specifically, we record the Q-matrix of each firm every 10^5 periods. After the first algorithm converges, we record the convergence time and equally draw 5 points from the recorded Q-matrices. We start again from each time period when the Q-matrix was selected and examine the counterfactual in which firm 2 adopts an algorithm at that time. After both algorithms converge, we let the firm 2 to retrieve the adoption of the algorithm and use the same rule again as in the hybrid stage. The time period after the post stage is referred to as the retrieve stage hereafter.

4.3 Results

In this section, we present the simulation results. We first report the results in the hybrid stage and then move to the results in and after the post stage. To examine firms' incentives to deviate, we exogenously force firm 1 to deviate from the converged policy and adopt a myopic rule that maximizes its current period profit after the retrieve stage. Finally, we explore the shared profit counterfactual, where firms coordinate explicitly to maximize joint payoffs.

4.3.1 Hybrid Stage

We first compare the difference between hybrid stage and pre stage, i.e., the price change when firm 1 starts to adopt an algorithm. We can see from Table 3 that all results satisfy Proposition 3 and end up with a price increase. The steady states of myopic and undercut rules are both higher than the initial steady states, while steady states of trigger and ceiling are both the same as the highest possible price in pre stage, respectively. Notice that for the

trigger rule, the pre steady state and hybrid steady state are both the monopoly price. For all four rules, the hybrid steady states observed are the same as theoretical prediction. In

Table 3: Converged results in hybrid stage with $\beta = 10^{-6}$

Rule	Pre s-s	Predicted Hybrid s-s	Observed Probability	Δ
Myopic	(p^2, p^2)	(p^8, p^5)	100%	$(0.18, 0.85)$
Undercut	(p^2, p^2)	(p^{14}, p^{13})	100%	(0.84,1.15)
Trigger	(p^{14}, p^{14}) or (p^2, p^2)	(p^{14}, p^{14})	100%	(1.00,1.00)
Ceiling	any (p, p) s.t. $p \leq p^7$	(p^7, p^7)	100%	(0.61,0.61)

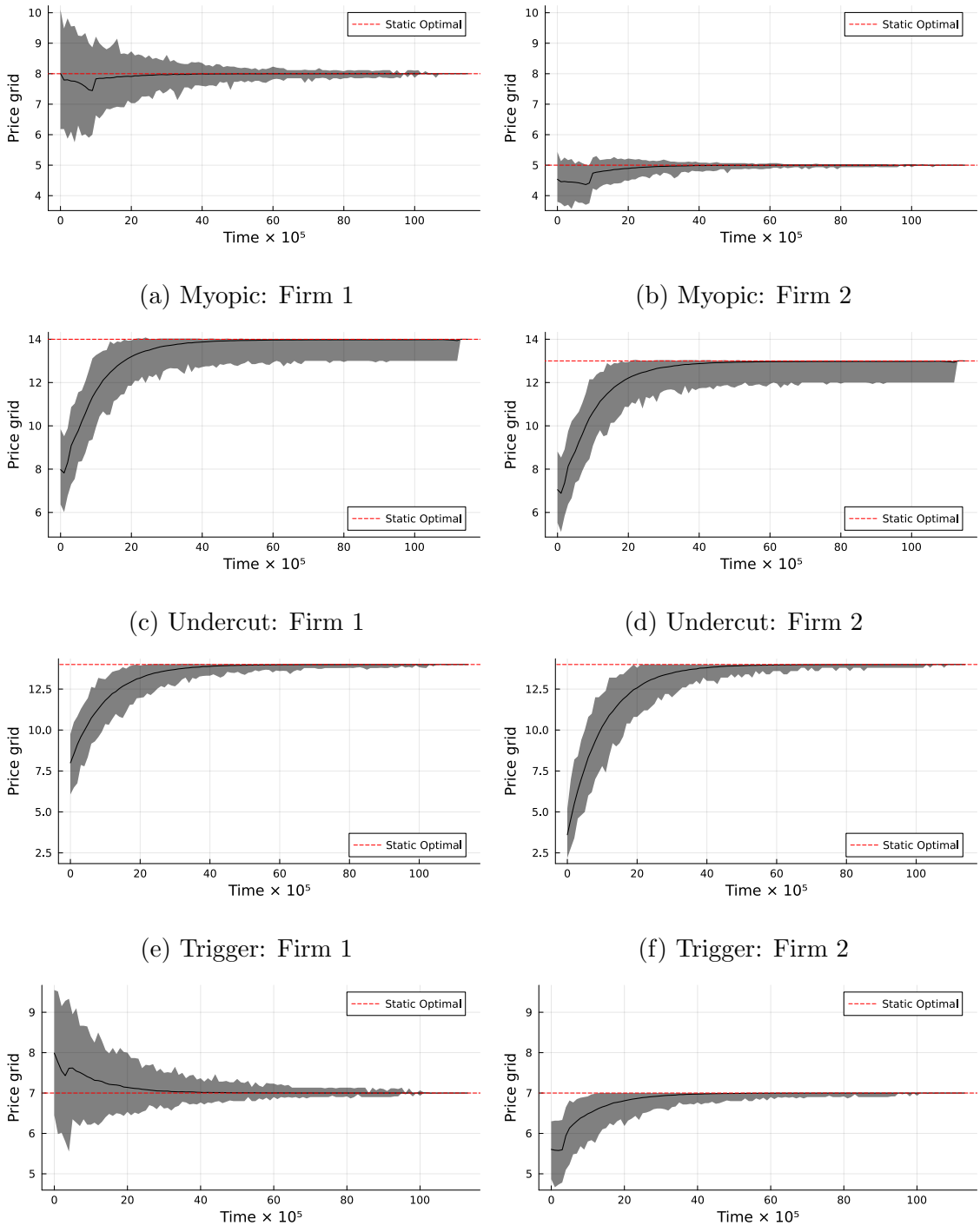
Notes: The observed probability is the probability that prices converge to the predicted steady state in simulations.

all four rules, firm 2 yields a weakly lower price and therefore has higher market share and profit than firm 1. Among them, prices of firm 1 in the undercut and trigger rules converge to monopoly price, while prices of firm 2 in the trigger rule also converges to monopoly price and price in the undercut rule converges to the price smaller than monopoly price due to the price undercut. The profits in these two rules are also relatively high, with $\Delta = 1$ for both firms in trigger (both firms earn monopoly profit). Converged prices in myopic and ceiling are smaller than the monopoly price, while they are higher than the steady state in the pre stage. For the most competitive rule, myopic, the difference between the profits of the two firms is the largest.

The converging paths of all four rules with $\beta = 10^{-6}$ are shown in Figure 2. Each point on the graph is an average of the first 60 periods for every 10^5 periods.[11] We can see that the algorithms tend to reduce the price at the beginning for the myopic, undercut and ceiling rules, which shows that the algorithms are not collusive by design. After some periods, each algorithm learns to cooperate with the rule and starts to increase the price, until it reaches

[11] When the price cycle length is 1,2,..,6, the average price of 60 periods is always numerically the same as the average price of a cycle.

Figure 2: Converging path in hybrid stage



(a) Myopic: Firm 1

(b) Myopic: Firm 2

(c) Undercut: Firm 1

(d) Undercut: Firm 2

(e) Trigger: Firm 1

(f) Trigger: Firm 2

(g) Ceiling: Firm 1

(h) Ceiling: Firm 2

Notes: The red dashed line indicates the predicted steady state. The shaded area represents the range from the minimum to the maximum price grid, and the black line is the average price grid over time.

the theoretically optimal price. Although the learning paths differ across simulations, we can see that they end up converging to the optimal price with 100% probability.

We then switch to the extensions of asymmetric or multiple firms. Table 4 presents the results of three firms, where firm 1 adopts an algorithm and the other two use the rule. Note that the prices in the undercut did not converge to the predicted steady state because it requires $\delta_0 = 0.998$. Results for all other rules is consistent with our prediction. Table 5 presents the result where firm 2 has a higher cost that equals $1.1c$. All results except undercut are the same as predicted. About 25% of the simulations converge to (p^{14}, p^{13}), which yields about 0.56% of profit loss compared to the predict one. This difference is too small and sometimes the algorithm fails to converge to p^*. We will discuss more on the parameter selection in Section 5.2.

Table 4: Converged results in hybrid stage with three firms

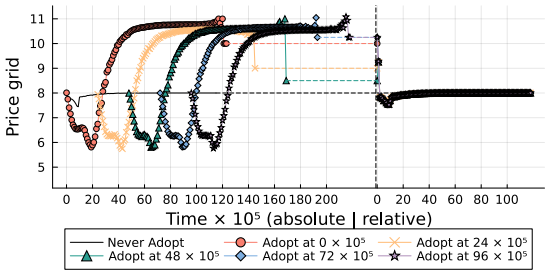
Rule	Pre s-s	Two-Firm Hybrid s-s	Three-Firm Hybrid s-s	Observed Hybrid s-s
Myopic	(p^2, p^2)	(p^8, p^5)	(p^8, p^8, p^8)	(p^8, p^8, p^8)
Undercut	(p^2, p^2)	(p^{14}, p^{13})	(p^{15}, p^{14}, p^{14})	$(p^{12.5}, p^{11.5}, p^{11.5})$
Trigger	(p^{14}, p^{14}) or (p^2, p^2)	(p^{14}, p^{14})	(p^{14}, p^{14}, p^{14})	(p^{14}, p^{14}, p^{14})
Ceiling	any (p, p) s.t. $p \leq p^7$	(p^7, p^7)	(p^7, p^7, p^7)	(p^7, p^7, p^7)

Notes: The Two-Firm Hybrid s-s stands for the predicted hybrid steady state in market with two firms as in the baseline. The Three-Firm hybrid s-s stands for the predicted hybrid steady state in market with three firms.

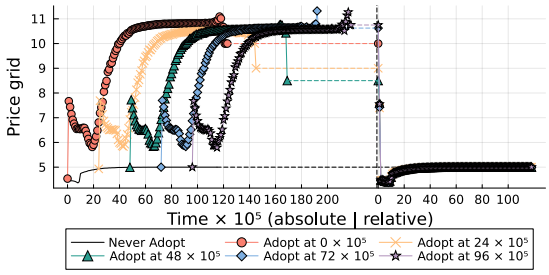
4.3.2 Post stage

To see the results in the post stage, consider that firm 2 adopts an algorithm at t_2 periods when firm 1 did at time 0 as described in Figure 1. We select 5 different t_2 that are equally spaced from $\{0, 1 \times 10^5, 2 \times 10^5, \ldots, T\}$, where T is the time when the algorithm of firm 1 converged. Any $t_2 \geq T$ should yield a very similar result to $t_2 = T$.

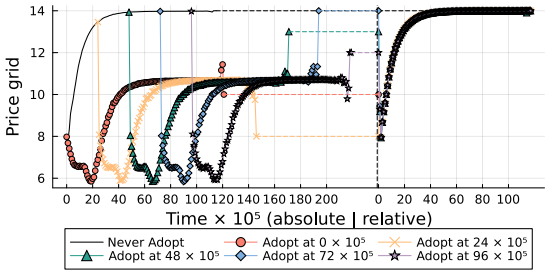
Figure 3: Converging path in post and retrieve stage



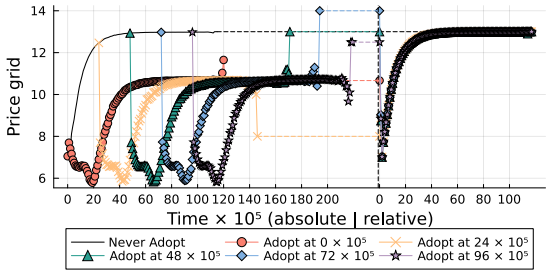
(a) Myopic: Firm 1



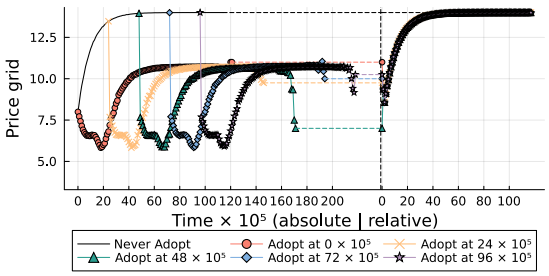
(b) Myopic: Firm 2



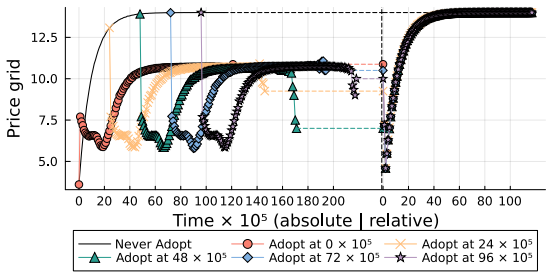
(c) Undercut: Firm 1



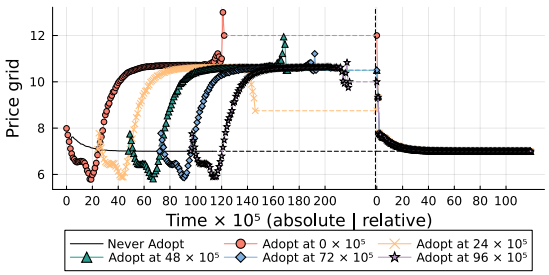
(d) Undercut: Firm 2



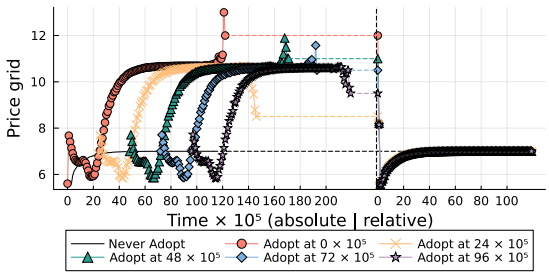
(e) Trigger: Firm 1



(f) Trigger: Firm 2



(g) Ceiling: Firm 1



(h) Ceiling: Firm 2

Notes: Results for different values of t_2 are indicated by different colors and markers. The vertical dashed lines separate the post stage and retrieve stage. Results in post stage are shown in absolute time, while results in retrieve stage are in relative time.

Table 5: Converged results in hybrid stage with asymmetric costs

Rule	Pre s-s	Symmetric Hybrid s-s	Asymmetric Hybrid s-s	Observed Hybrid s-s
Myopic	(p^2, p^2)	(p^8, p^5)	(p^7, p^4)	(p^7, p^4)
Undercut	(p^2, p^2)	(p^{14}, p^{13})	(p^{15}, p^{14})	$(p^{14\sim15}, p^{13\sim14})$
Trigger	(p^{14}, p^{14}) or (p^2, p^2)	(p^{14}, p^{14})	(p^{14}, p^{14})	(p^{14}, p^{14})
Ceiling	any (p, p) s.t. $p \leq p^7$	(p^7, p^7)	(p^7, p^7)	(p^7, p^7)

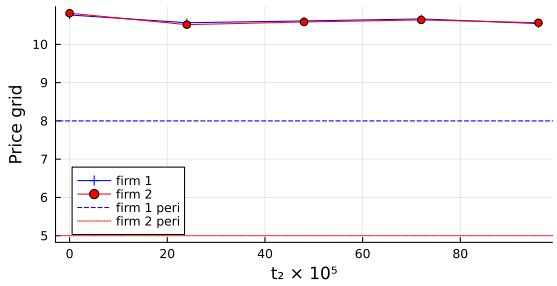
Notes: The Symmetric Hybrid s-s stands for the predicted hybrid steady state when firms are identical. The Asymmetric hybrid s-s stands for the predicted hybrid steady state when firms have asymmetric marginal cost.

Figure 3 shows the results from $t = 0$. The black unmarked line is the converging line in hybrid stage where firm 2 never adopts an algorithm. Since we stop the simulation when we observe empirical convergence, we use dashed lines for prices in periods that are not actually observed to avoid confusion. The x-axis has two parts — absolute time and relative time, separated by a vertical dashed line. The hybrid and post stages are the absolute periods. Since the retrieve stage starts after the post period, we align the relative periods after the vertical dashed line.

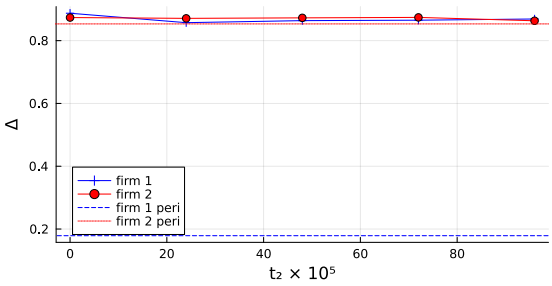
For the post stage, we can observe a clear pattern that the algorithms start to reduce the price at the beginning and learn to cooperate and increase price later. Although we observe price increases for the myopic and ceiling rules compared to hybrid stage, it is difficult to predict their converged prices. The results also depend on the rule as well as the adoption time t_2 with no clear pattern.

We can see a clearer result in Figure 4. The blue dashed and red dotted lines are the prices observed in the hybrid stage for firm 1 and 2, respectively. We can see that under the myopic rule, the prices keep increasing when firm 2 also adopts an algorithm. The profit of firm 1 increases significantly (Δ increases from about 0.2 to about 0.9) so that firm 1 benefits from the adoption of firm 2, but firm 2 itself does not benefit much from its own adoption.

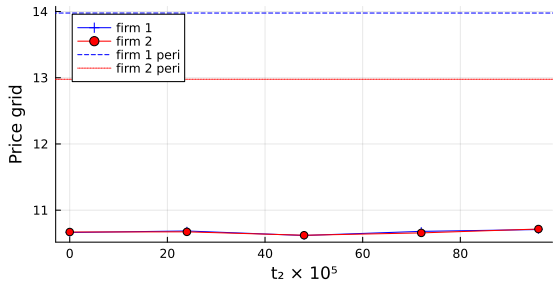
Figure 4: Converged results in post stage



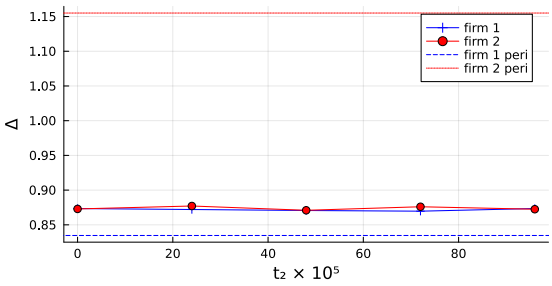
(a) Myopic: Price



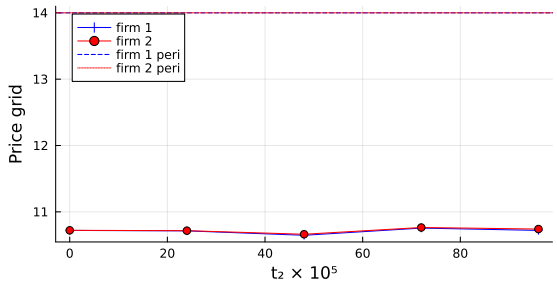
(b) Myopic: Profit



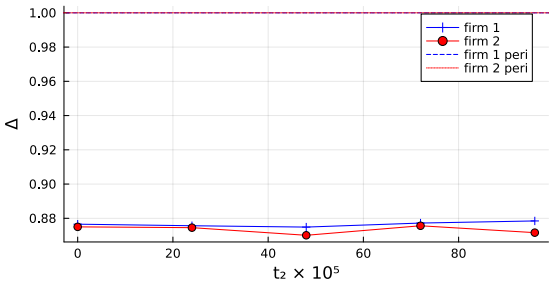
(c) Undercut: Price



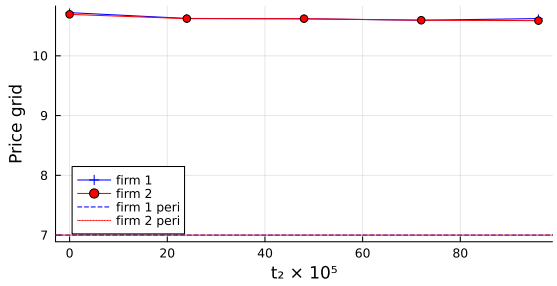
(d) Undercut: Profit



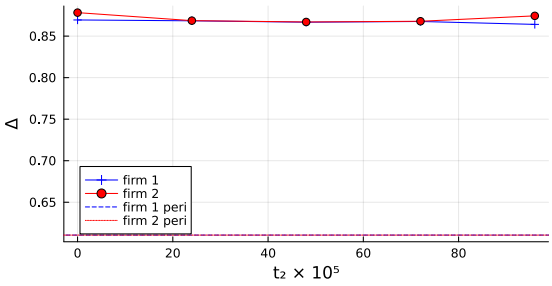
(e) Trigger: Price



(f) Trigger: Profit



(g) Ceiling: Price



(h) Ceiling: Profit

Notes: Each point stands for the average of converged results for t_2. The horizontal lines present the hybrid steady state.

Things are similar with the ceiling rule, while Δ increases from about 0.6 to about 0.9 for both firms.

As for the undercut and trigger rule, where the prices are already close to the monopoly price in the hybrid stage, the algorithm of firm 2, however, fails to continue charging the monopoly price. One reason is that the exploration by algorithms increases the profit of deviation and reduces the profit of charging the monopoly price, causing algorithms to deviate from the monopoly price in early periods.[12] While learning from the environment and also from the rival, the monopoly prices are no longer the best response to the policy of the rival's algorithm. The profit of firm 1 is nearly unaffected in undercut, while Δ of firm 2 is reduced by about 0.3. The Δ of both firms is reduced by about 0.1 in the ceiling rule.

As for the retrieve stage, regardless of t_2, it will converge to the same price as in the hybrid stage. We also notice that here the initial Q value does not affect the converging path much, as the paths in the retrieve stage almost coincide with each other. This is consistent with our theoretical prediction that no matter what initial value is, as long as the algorithm converges, it will converge to the predicted price when only one firm is using the algorithm.

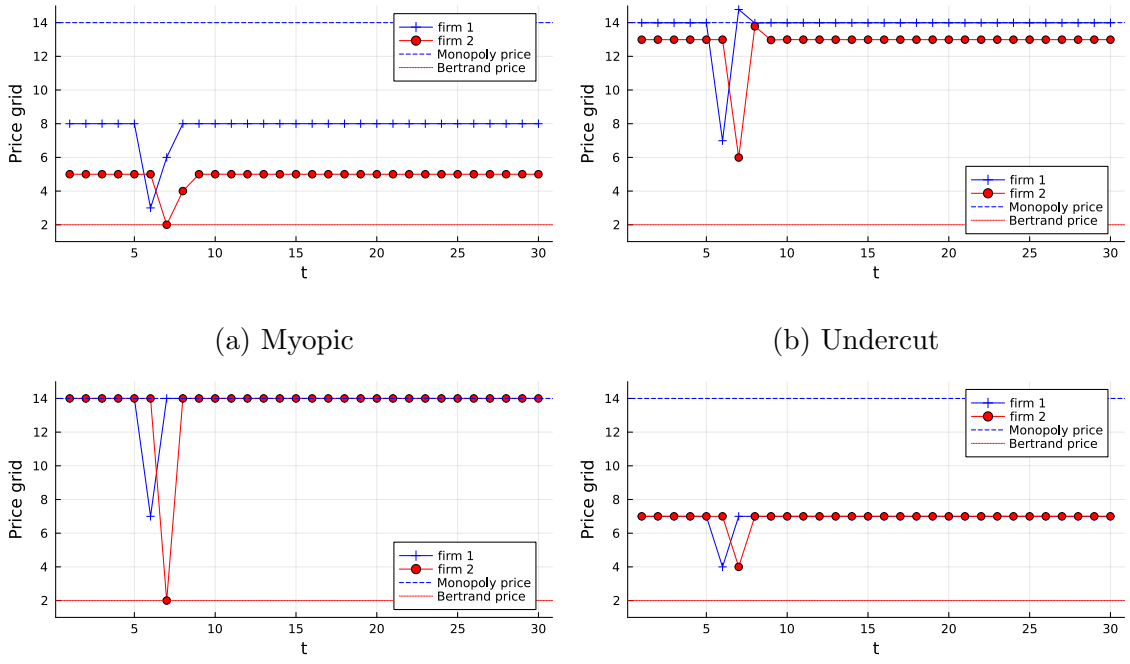
4.3.3 Deviation

To conclude that the algorithms are in tacit collusion, it's not enough just to show their prices in the steady states. We would also want to know how they would act in other circumstances and whether the results are just by chance. However, we are unable to describe the policies given the fact that there are 225 possible states in our simulation.

One key factor of tacit collusion is whether a firm would send a signal to the rival that it will cooperate. Therefore, we test by forcing firm 1 to deviate after firm 2 retrieves. Figure 5 shows how firms would response if firm 1 is forced to deviate at the 6th relative period after

[12]The algorithm will charge randomly with some probability in any state. Therefore it's less likely for an algorithm that's still learning to punish the deviation than a simple predetermined rule.

Figure 5: Deviation results



(a) Myopic

(b) Undercut

(c) Trigger

(d) Ceiling

Notes: Firm 1 deviate to myopic rule at time 6.

it converges.[13] Each point in the figure is the average of 1,000 simulations in that relative period. Firm 1 will use the myopic rule and charge the price that maximizes its current period profit at relative period 6. For any other relative periods, the firms are just pricing with their own policy — firm 1 with the converged algorithm and firm 2 with the simple rule.

We observe that firm 2 punishes firm 1 for this deviation differently according to the rule it uses at relative period 7, the period after the deviation occurs. Firm 1, however, will not continue to charge low prices, but will start to raise prices in that period. This is consistent with our theoretical prediction that the static optimal price p^* is numerically the same as the dynamic optimal price and thus will prevent deviation.

[13]Figure 5 shows the results when $t_2 = T$. The results for different t_2 are very similar to those in Figure 5 and thus omitted.

4.3.4 Shared Profit

So far we have been discussing how algorithms could lead to tacit collusion that is not directly covered by existing antitrust regulations. Prices could increase even further if explicit collusion occurs when firms share their profits. In this subsection, we assume that the firms share profits and they equally split the profit earned in each period. We continue to assume that firms are not communicating, such that firm 1 is using an algorithm while firm 2 is using a rule, because otherwise the optimal price will always be the monopoly price and not worth analyzing.

Table 6: Converged results in hybrid stage of firms with shared profit

Rule	Pre s-s	Separated Hybrid s-s	Shared Hybrid s-s	Observed Hybrid s-s
Myopic	(p^2, p^2)	(p^8, p^5)	(p^{13}, p^7)	(p^{13}, p^7)
Undercut	(p^2, p^2)	(p^{14}, p^{13})	(p^{15}, p^{14})	(p^{15}, p^{14})
Trigger	(p^{14}, p^{14}) or (p^2, p^2)	(p^{14}, p^{14})	(p^{14}, p^{14})	(p^{14}, p^{14})
Ceiling	any (p, p) s.t. $p \leq p^7$	(p^7, p^7)	(p^{12}, p^7)	(p^{12}, p^7)

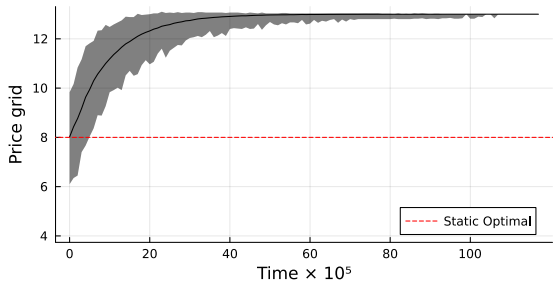
Notes: The Separated Hybrid s-s stands for the predicted hybrid steady state when firms earn their separate profit. The Shared hybrid s-s stands for the predicted hybrid steady state when firms share and equally split the profits they earned.

Table 6 shows the steady states observed in our simulation, and Figure 6 shows the converging path. Prices predicted in tacit and explicit collusion are labeled as "Separated Hybrid s-s" and "Shared Hybrid s-s" respectively. The results in the simulations are in the column "Observed Hybrid s-s".

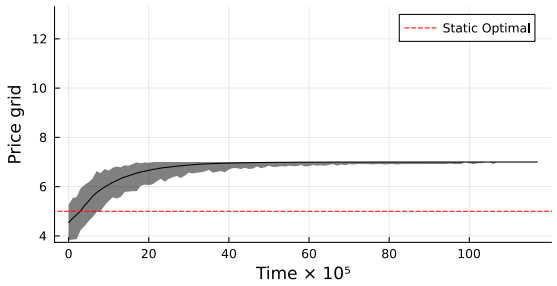
In summary, we do see prices go up in an explicit collusion compared to a tacit collusion. In the myopic and ceiling rules, where the prices were not as high as the monopoly prices in a tacit collusion, the prices increase significantly when firms share their profits. Prices in undercut, however, even exceed the monopoly price.[14] We can see that even if the algorithms

[14]In Assumption 1 we assume that the price is less than or equal to the monopoly price and greater than

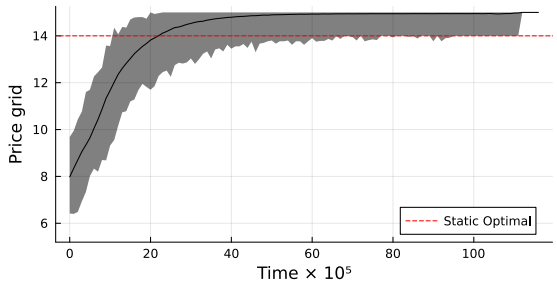
Figure 6: Converging path in hybrid stage: shared profits



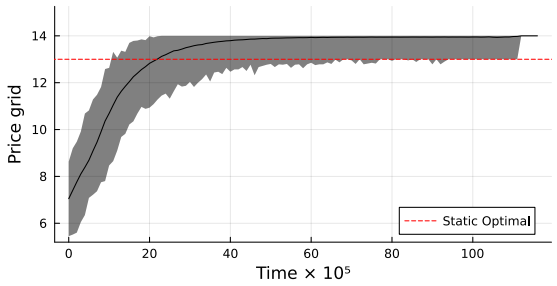
(a) Myopic: Firm 1



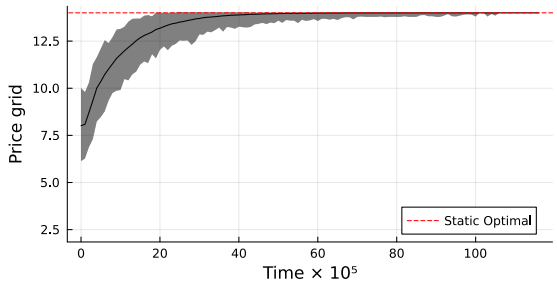
(b) Myopic: Firm 2



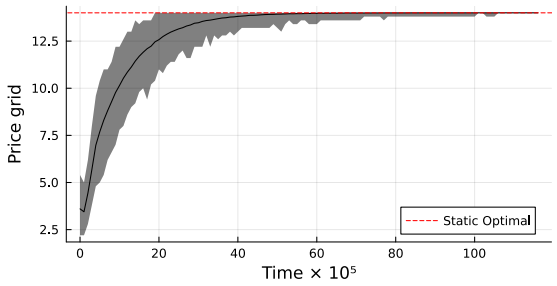
(c) Undercut: Firm 1



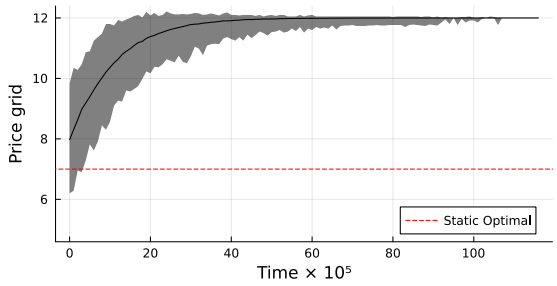
(d) Undercut: Firm 2



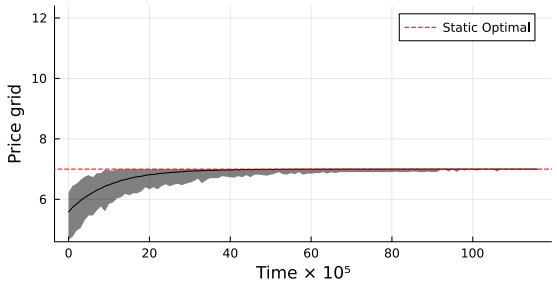
(e) Trigger: Firm 1



(f) Trigger: Firm 2



(g) Ceiling: Firm 1



(h) Ceiling: Firm 2

Notes: The red dashed line indicates the predicted steady state. The shaded area represents the range from the minimum to the maximum price grid, and the black line is the average price grid over time.

are already in a tacit collusion with rules and raise the prices, sharing profits will make the situation even worse.

5 Robustness

In this section, we extend our simulation results in the hybrid stage to alternative parameters and setups. The results show that the baseline findings are robust to parameters.

5.1 Less Patient Firms

As we discussed in Section 2, Proposition 1 assumes that the discount factor δ is greater than some δ_0. Specifically, the δ_0 for the four rules used in our analysis are shown in Table 7. δ is usually assumed large given the fact that algorithms can response to environment changes in real time. However, there are also circumstances in which firms are not able to change prices frequently. This could be due to the menu costs, and also regulations as in Byrne and De Roos (2019). Here we discuss the counterfactual when we have less patient firms that violate the assumption and $\delta = 0.5$.

Table 7: δ_0 for rules

Rule	Myopic	Undercut	Trigger	Ceiling
δ_0	0.945	0.915	0.478	0.380

As seen in table 7, myopic and undercut rules would violate the assumption since $\delta = 0.5 < \delta_0$, and therefore are not predicted to converge to the static optimal price. We run simulations on all four rules, and the results are shown in Table 8. The results are as expected that the converged prices in the myopic and undercut rules are not the same as predicted,

or equal to the Bertrand price when firms do not share their profits.

while they remain the same for the trigger and ceiling rules. The converging path is shown in Figure 7.

Table 8: Converged results in hybrid stage of less patient firms

Rule	Pre s-s	Predicted Hybrid s-s	Observed Hybrid s-s
Myopic	(p^2, p^2)	(p^8, p^5)	(p^4, p^3)
Undercut	(p^2, p^2)	(p^{14}, p^{13})	(p^8, p^7)
Trigger	(p^{14}, p^{14}) or (p^2, p^2)	(p^{14}, p^{14})	(p^{14}, p^{14})
Ceiling	any (p, p) s.t. $p \leq p^7$	(p^7, p^7)	(p^7, p^7)

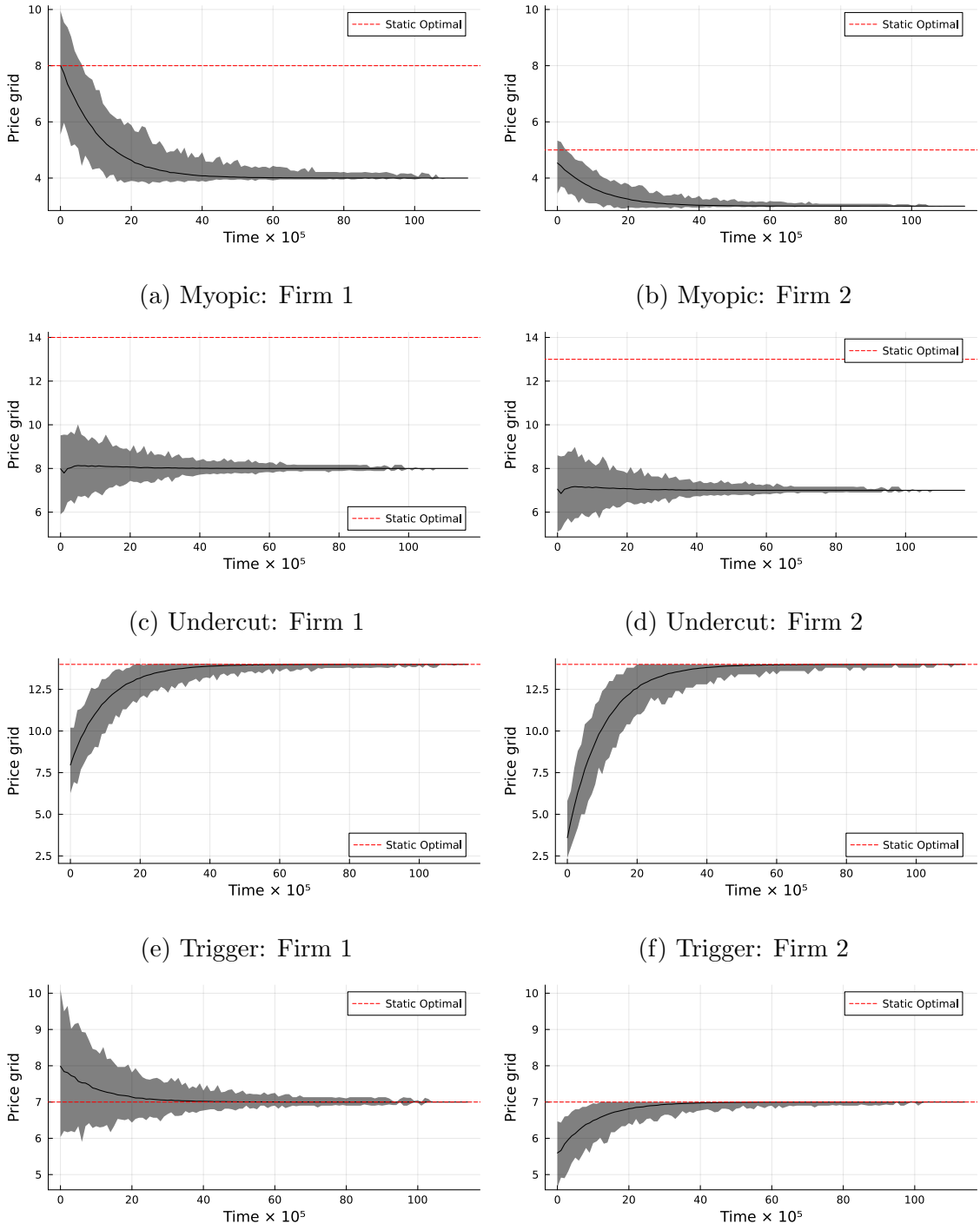
Notes: The predicted Hybrid s-s stands for the predicted hybrid steady state when firms has large $\delta = 0.95$. The observed hybrid s-s stands for the steady state in results when firms are less patient and have $\delta = 0.5$.

It is worth noting that although Proposition 3 does not hold for $\delta < \delta_0$, the optimal policy function g^* still has a fixed point. In Figure 7 we see that the algorithms converge to a steady state in all simulations. The sensitivity of g^* with respect to δ depends on the derivative of g_2, which can be interpreted as the punishment of a strategy. The price trigger strategy has stronger punishment and therefore requires less δ to preserve the monopoly price compared to price undercut strategy, which does not punish deviation and requires a very large δ to keep the steady state of monopoly price.

5.2 Refined Initialization

In Section 4, we discussed the results when we set $\beta = 10^{-6}$. Although the uniqueness of steady state is guaranteed by Proposition 1, it is not always observed that prices converge to p^*. In practice, the convergence of Q-learning depends on the environment as well as parameter selection. Two key parameters are the initial value and the exploration rate. Table 9 and Figure 8 show the results when the algorithm explores less and $\beta = 10^{-5}$, while keeping the initial Q_0 identical to that in Section 4.

Figure 7: Converging path in hybrid stage: less patient firms



(a) Myopic: Firm 1

(b) Myopic: Firm 2

(c) Undercut: Firm 1

(d) Undercut: Firm 2

(e) Trigger: Firm 1

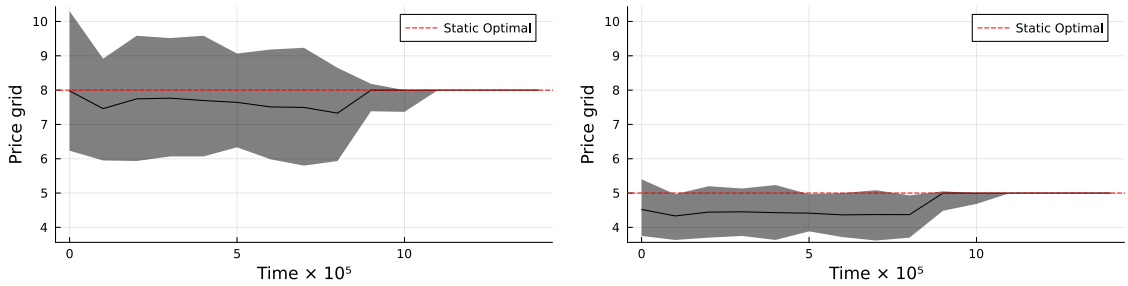
(f) Trigger: Firm 2

(g) Ceiling: Firm 1

(h) Ceiling: Firm 2

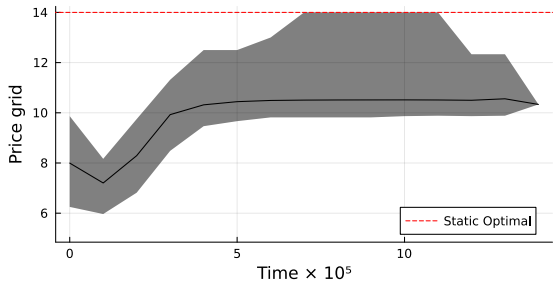
Notes: The red dashed line indicates the predicted steady state. The shaded area represents the range from the minimum to the maximum price grid, and the black line is the average price grid over time.

Figure 8: Converging path in hybrid stage: less exploration

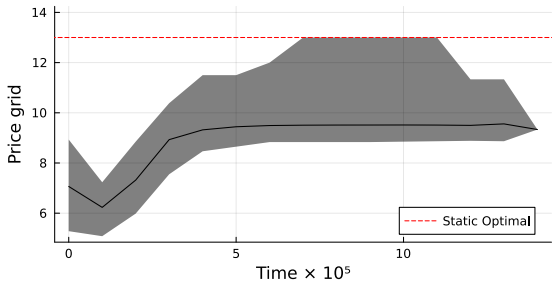


(a) Myopic: Firm 1

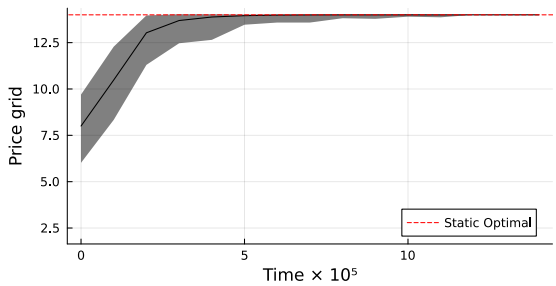
(b) Myopic: Firm 2



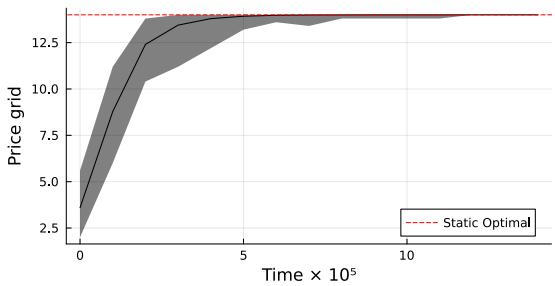
(c) Undercut: Firm 1



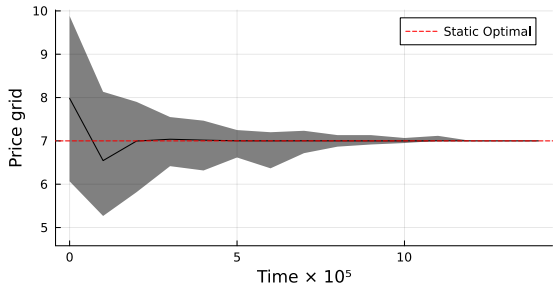
(d) Undercut: Firm 2



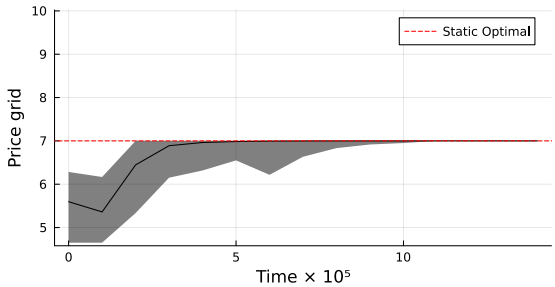
(e) Trigger: Firm 1



(f) Trigger: Firm 2



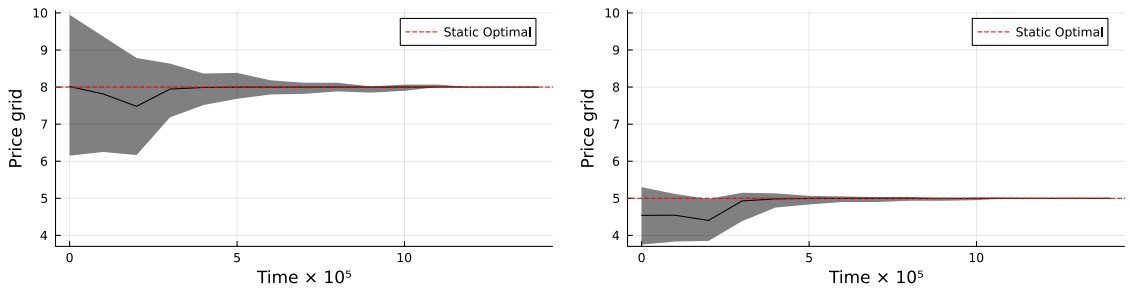
(g) Ceiling: Firm 1



(h) Ceiling: Firm 2

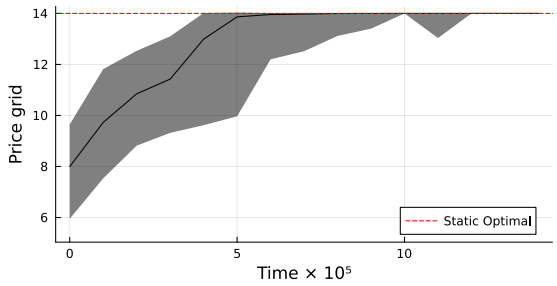
Notes: The red dashed line indicates the predicted steady state. The shaded area represents the range from the minimum to the maximum price grid, and the black line is the average price grid over time.

Figure 9: Converging path in hybrid stage: better initial value

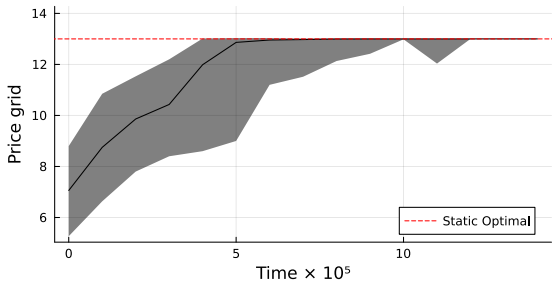


(a) Myopic: Firm 1

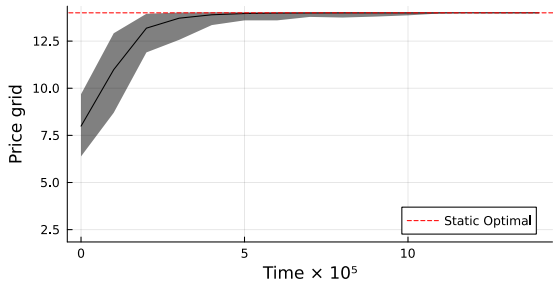
(b) Myopic: Firm 2



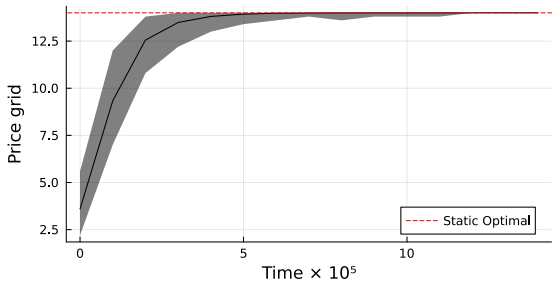
(c) Undercut: Firm 1



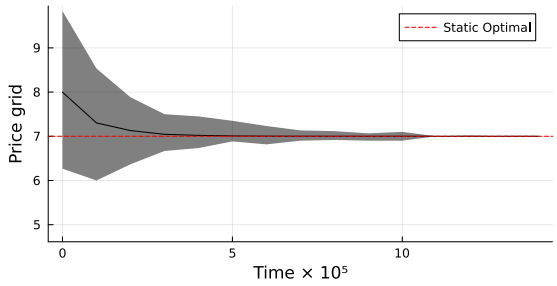
(d) Undercut: Firm 2



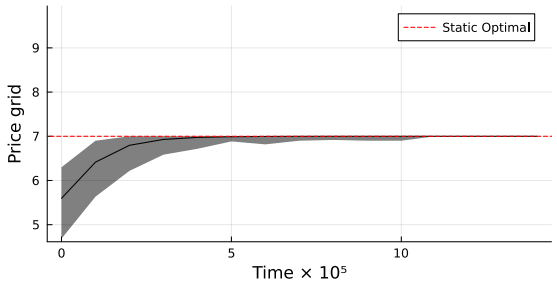
(e) Trigger: Firm 1



(f) Trigger: Firm 2



(g) Ceiling: Firm 1



(h) Ceiling: Firm 2

Notes: The red dashed line indicates the predicted steady state. The shaded area represents the range from the minimum to the maximum price grid, and the black line is the average price grid over time.

Table 9: Converged results in hybrid stage of firms with less exploration

Rule	Pre s-s	Predicted Hybrid s-s	More Exploration Hybrid s-s	Less Exploration Hybrid s-s
Myopic	(p^2, p^2)	(p^8, p^5)	(p^8, p^5)	(p^8, p^5)
Undercut	(p^2, p^2)	(p^{14}, p^{13})	(p^{14}, p^{13})	$(p^{10} \sim p^{11}, p^9 \sim p^{10})$
Trigger	(p^{14}, p^{14}) or (p^2, p^2)	(p^{14}, p^{14})	(p^{14}, p^{14})	(p^{14}, p^{14})
Ceiling	any (p, p) s.t. $p \leq p^7$	(p^7, p^7)	(p^7, p^7)	(p^7, p^7)

Notes: The More Exploration Hybrid s-s stands for the predicted hybrid steady state when $\beta = 10^{-6}$. The Less Exploration hybrid s-s stands for the predicted hybrid steady state when $\beta = 10^{-5}$.

We observed different results for the undercut rule. In particular, the algorithms fail to converge to the predicted steady state, and sometimes even fail to converge to a steady state but rather converge to a cycle. As we discussed in Section 3, the convergence in practice is defined as the policy function not changing for 10^5 periods, while it is not considered convergence in theory if different simulations converge to different prices. This violates the key assumption that the algorithm should converge. However, we are still able to obtain the same results with less exploration. Suppose that we have a smart AI that we can guess the initial value quite well that

$$Q_{i,0}(s, a_i) = \frac{\pi_i(a_i, g_2(a_{-i}))}{(1-\delta)}.$$

Notice that this initial value is not the same as the actual Q-value, because it's just the discounted profit if the firm sets price a_i forever, which is not the optimal action in most states. Even so, we are able to observe the results exactly as predicted as shown in Figure 9.

6 Conclusion

This research sheds light on the impact of pricing algorithm adoption on tacit collusion and its interaction with rule-based strategies in competitive markets. Starting with both firms in the market employing rule-based strategies, we developed a theoretical model that shows a

weak increase in price when the first firm starts to adopt a reinforcement learning algorithm. The firm maintaining a rule-based strategy can "free ride" and benefits more from the other firm's adoption. The analysis relies on broad assumptions about the convergence of the algorithm and the properties of the rules. The results can also be extended to more general markets. We further show that when the likelihood of rivals adopting algorithms is high, it may be optimal for some firms to continue using rule-based strategies.

Simulation results with Q-learning coincide with the theoretical predictions across several widely used rule-based strategies. Our robustness checks, including analyses of less patient firms and different algorithm specifications, affirm that our main findings hold under a variety of market assumptions. Moreover, we find that while standard antitrust policies focused on restricting communication are effective in prohibiting explicit collusion, they have limited effect on tacit collusion. These findings provide a foundation for future research and inform the design of regulatory policies for markets increasingly shaped by pricing algorithms.

There are a few directions for future work. First, we focus on the steady state to which a reinforcement learning algorithm converges within a stationary Markov process in the theoretical model. Future research can extend the analysis to nonstationary environments. Second, because algorithm adoption is not required to be public under current antitrust policy, this paper emphasizes theoretical and simulated results, leaving empirical validation for future work. Both extensions will provide deeper insights into the emerging literature on algorithmic pricing and tacit collusion.

References

Abreu, D. (1988). On the theory of infinitely repeated games with discounting. *Econometrica*, 56(2):383–396.

Amir, R. and Stepanova, A. (2006). Second-mover advantage and price leadership in bertrand duopoly. *Games and Economic Behavior*, 55(1):1–20.

Asker, J., Fershtman, C., and Pakes, A. (2024). The impact of artificial intelligence design on pricing. *Journal of Economics & Management Strategy*, 33(2):276–304.

Assad, S., Clark, R., Ershov, D., and Xu, L. (2024). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy*, 132(3):723–771.

Banchio, M. and Skrzypacz, A. (2022). Artificial intelligence and auction design. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 30–31.

Bichler, M., Durmann, J., and Oberlechner, M. (2024). Online optimization algorithms in repeated price competition: Equilibrium learning and algorithmic collusion. *arXiv preprint arXiv:2412.15707*.

Brown, Z. Y. and MacKay, A. (2023). Competition in pricing algorithms. *American Economic Journal: Microeconomics*, 15(2):109–156.

Byrne, D. P. and De Roos, N. (2019). Learning to coordinate: A study in retail gasoline. *American Economic Review*, 109(2):591–619.

Calvano, E., Calzolari, G., Denicolò, V., Harrington Jr, J. E., and Pastorello, S. (2020a). Protecting consumers from collusive prices due to AI. *Science*, 370(6520):1040–1042.

Calvano, E., Calzolari, G., Denicolo, V., and Pastorello, S. (2020b). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–3297.

Chen, L., Mislove, A., and Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th international conference on World Wide Web*, pages 1339–1349.

Chen, N. and Tsai, H.-T. (2024). Steering via algorithmic recommendations. *The RAND Journal of Economics*, 55(4):501–518.

Dou, W. W., Goldstein, I., and Ji, Y. (2025). AI-powered trading, algorithmic collusion, and price efficiency. *Jacobs Levy Equity Management Center for Quantitative Financial Research Paper, The Wharton School Research Paper*.

Gal-Or, E. (1985). First mover and second mover advantages. *International Economic Review*, 26(3):649–653.

Green, E. J. and Porter, R. H. (1984). Noncooperative collusion under imperfect price information. *Econometrica*, 52(1):87–100.

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

Johnson, J. P., Rhodes, A., and Wildenbeest, M. (2023). Platform design when sellers use pricing algorithms. *Econometrica*, 91(5):1841–1879.

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

Lamba, R. and Zhuk, S. (2022). Pricing with algorithms. *arXiv preprint arXiv:2205.04661*.

Leslie, C. R. (2023). Predatory pricing algorithms. *NYUL Rev.*, 98:49.

MacKay, A. and Weinstein, S. N. (2022). Dynamic pricing algorithms, consumer harm, and regulatory response. *Wash. UL Rev.*, 100:111.

Miklós-Thal, J. and Tucker, C. (2019). Collusion by algorithm: Does better demand prediction facilitate coordination between sellers? *Management Science*, 65(4):1552–1561.

Musolff, L. (2024). Algorithmic pricing, price wars and tacit collusion: Evidence from e-commerce. Technical report, Working Paper, Wharton School of the University of Pennsylvania.

Pai, M. and Hansen, K. (2020). Algorithmic collusion: Supra-competitive prices via independent algorithms. Technical report, CEPR Discussion Papers.

Peiseler, F., Rasch, A., and Shekhar, S. (2022). Imperfect information, algorithmic price discrimination, and collusion. *The Scandinavian Journal of Economics*, 124(2):516–549.

Possnig, C. (2023). *Reinforcement learning and collusion*. Department of Economics, University of Waterloo.

Salcedo, B. (2015). Pricing algorithms and tacit collusion. *Manuscript, Pennsylvania State University*.

Sandholm, T. W. and Crites, R. H. (1995). On multiagent q-learning in a semi-competitive domain. In *International Joint Conference on Artificial Intelligence*, pages 191–205. Springer.

Spann, M., Bertini, M., Koenigsberg, O., Zeithammer, R., Aparicio, D., Chen, Y., Fantini, F., Jin, G. Z., Morwitz, V. G., Leszczyc, P. P., et al. (2025). Algorithmic pricing: Implications for marketing strategy and regulation. *International Journal of Research in Marketing*.

Tesauro, G. and Kephart, J. O. (2002). Pricing in agent economies using multi-agent q-learning. *Autonomous agents and multi-agent systems*, 5:289–304.

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

Wang, Q., Huang, Y., Singh, P. V., and Srinivasan, K. (2023). Algorithms, artificial intelligence and simple rule based pricing. *Available at SSRN 4144905*.

Werner, T. (2024). Algorithmic and human collusion. *Available at SSRN 3960738*.

Appendix

This appendix collect supplementary materials. Appendix A contains detailed proofs of Lemmas and Propositions in Section 2. Appendix B shows the myopic strategy given the logit demand in our simulation in Section 4.

A Proofs

We provides detailed proofs of Lemmas and Propositions in Section 2.

Proof of Lemma 1.

Proof. $p_{1,0} = g_2(p_{2,0}) \leq p_{2,0}$ and $p_{2,0} = g_1(p_{1,0}) \leq p_{1,0}$. So $p_{1,0} = p_{2,0}$. □

Proof of Proposition 1.

Proof. By Brouwer fixed point theorem, g^* always has at least one fixed point. Suppose by contradiction that there exists another price $p' \neq p^*$ that is a fixed point. We will show that firm 1 will always deviate from p' to p^* so that p' is not a steady state, i.e., $p' \neq g^*(p')$ and p' is not a fixed point of g^*. Denote the difference of profit of deviating from p'_1 to p^* as $\Delta\pi$, we know that

$$\begin{aligned}\Delta\pi \geq &\left(\pi(p^*, g_2(p')) + \frac{\delta}{1-\delta}\pi(p^*, g_2(p^*))\right) \\ &- \left(\pi\left(p', g_2(p')\right) + \frac{\delta}{1-\delta}\pi(p', g_2(p'))\right),\end{aligned}$$

since the discount profit of state p^* is greater than or equal to $\frac{1}{1-\delta}\pi(p^*, g_2(p^*))$, which is the profit if firm 1 choose p^* forever.

- When $p' > p^*$, we have $\pi\left(p', g_2(p')\right) < \pi\left(p^*, g_2(p^*)\right) \leq \pi(p^*, g_2(p'))$. Note that π_1 is weakly increasing w.r.t. p_2 because for $\pi_1(p_1, p_2) = (p_1 - c)q_1(p_1, p_2)$, we always have

  $$\frac{\partial}{\partial p_2}\pi_1(p_1, p_2) = (p_1 - c)\frac{\partial q_1}{\partial p_2} \geq 0.$$

  Combined with $\pi(p^*, g_2(p^*)) > \pi(p', g_2(p'))$, we have $\Delta\pi > 0$.

- When $p' < p^*$ and $\pi\left(p', g_2(p')\right) < \pi(p^*, g_2(p'))$, we still have $\Delta\pi > 0$.

- When $p' < p^*$ and $\pi(p', g_2(p')) \geq \pi\left(p^*, g_2(p')\right)$, we can rewrite the right hand side as

$$\begin{aligned}\Delta\pi \geq & \Big(\pi(p^*, g_2(p')) - \pi\left(p', g_2(p')\right)\Big) \\ & + \frac{\delta}{1-\delta}\Big(\pi(p^*, g_2(p^*)) - \pi(p', g_2(p'))\Big) \\ & := A + \frac{\delta}{1-\delta}B,\end{aligned}$$

where $A \leq 0$ and $B > 0$. Let $\delta_0 = -A/(B-A) \in [0,1)$, for any $\delta > \delta_0$, $\Delta\pi > 0$.

Therefore, there always exists a $\delta_0 \in [0,1)$ such that $\forall\delta \in (\delta_0, 1)$, $\Delta\pi > 0$, so that firm 1 will always deviate from p' to p^*, which contradicts with that p' is a fixed point. Therefore p^* is the unique fixed point. □

Proof of Proposition 2. Proposition 2 directly follows by implicit differentiation from the Bellman equation and is therefore omitted.

Proof of Proposition 3.

Proof. Since π_1 is weakly increasing w.r.t. p_2, under Assumption 1(iii). and 1(ii)., for any price $p < p_0$, we have $\pi_1\left(p, g_2(p)\right) \leq \pi_1(p,p) < \pi_1(p_0, p_0)$. Therefore, $p^* \geq p_0$.

Under assumption 1(iv)., we have $g_2(p^*) \geq g_2(p_0) = p_0$. □

Proof of Proposition 4. The proof of Proposition 4 is analogous to that of Proposition 3 and is therefore omitted.

B Myopic Price of Logit demand

We provide in this section the price that firm would charge with myopic rule given the rival's price in a duopoly market. Firms are maximizing the per period profit

$$\pi_i = (p_i - c_i)q_i = (p_i - c_i) \cdot \frac{\exp(\frac{\gamma_i - p_i}{\mu})}{\exp(\frac{\gamma_i - p_i}{\mu}) + \exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})}.$$

The optimal price derived by the first order condition is

$$p_i^* = c_i + \frac{\mu}{1-q_i} = c_i + \mu \frac{\exp(\frac{\gamma_i - p_i}{\mu}) + \exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})}{\exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})}$$
$$= c_i + \mu \left(1 + \frac{\exp(\frac{\gamma_i - p_i}{\mu})}{\exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})}\right)$$

We can then get

$$\frac{\gamma_i - p_i}{\mu} = \frac{\gamma_i - c}{\mu} + 1 + \frac{\exp(\frac{\gamma_i - p_i}{\mu})}{\exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})}$$
$$\frac{\exp\left(\frac{\gamma_i - c_i}{\mu} - 1\right)}{\exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})} = \exp\left(\frac{\exp\left(\frac{\gamma_i - p_i}{\mu}\right)}{\exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})}\right) \cdot \frac{\exp\left(\frac{\gamma_i - p_i}{\mu}\right)}{\exp(\frac{\gamma_j - p_j}{\mu}) + \exp(\frac{\gamma_0}{\mu})}$$

The solution is given by

$$\frac{\exp\left(\frac{\gamma_i - p_i}{\mu}\right)}{\exp\left(\frac{\gamma_j - p_j}{\mu}\right) + \exp\left(\frac{\gamma_0}{\mu}\right)} = W\left(\frac{\exp\left(\frac{\gamma_i - c_i}{\mu} - 1\right)}{\exp\left(\frac{\gamma_j - p_j}{\mu}\right) + \exp\left(\frac{\gamma_0}{\mu}\right)}\right)$$

where W is the LambertW function. Using the logarithmic property of the LambertW function $ln(W(x)) = ln(x) - W(x)$, we get

$$\frac{\gamma_i - p_i}{\mu} = \frac{\gamma_i - c_i}{\mu} - 1 - W\left(\frac{\exp\left(\frac{\gamma_i - c_i}{\mu} - 1\right)}{\exp\left(\frac{\gamma_j - p_j}{\mu}\right) + \exp\left(\frac{\gamma_0}{\mu}\right)}\right)$$

$$p_i^* = c_i + \mu + \mu W\left(\frac{\exp\left(\frac{\gamma_i - c_i}{\mu} - 1\right)}{\exp\left(\frac{\gamma_j - p_j}{\mu}\right) + \exp\left(\frac{\gamma_0}{\mu}\right)}\right)$$