RAND Journal of Economics
Vol. 52, No. 3, Fall 2021
pp. 538–599

Autonomous algorithmic collusion: Q-learning under sequential pricing

Timo Klein*

Prices are increasingly set by algorithms. One concern is that intelligent algorithms may learn to collude on higher prices even in the absence of the kind of coordination necessary to establish an antitrust infringement. However, exactly how this may happen is an open question. I show how in simulated sequential competition, competing reinforcement learning algorithms can indeed learn to converge to collusive equilibria when the set of discrete prices is limited. When this set increases, the algorithm considered increasingly converges to supra-competitive asymmetric cycles. I show that results are robust to various extensions and discuss practical limitations and policy implications.

■

> *"It's true that the idea of automated systems getting together and reaching a meeting of minds is still science fiction. (…) But we do need to keep a close eye on how algorithms are developing. (…) So that when science fiction becomes reality, we're ready to deal with it."*
> —EU Competition Commissioner Margrethe Vestager (2017)

1. Introduction

■ More and more, prices are set by algorithms rather than humans. One prominent concern is that intelligent, self-learning pricing algorithms may work out by themselves how to ensure high prices (Mehra, 2016; Ezrachi and Stucke, 2016, 2017). Such an outcome would be the same as

* Oxera Consulting LLP and Utrecht University School of Economics; t.klein@uu.nl.
I am grateful for valuable discussions with and comments and support from Arnoud den Boer, Emilio Calvano, Ariel Ezrachi, Joe Harrington, Harold Houba, Laura van der Kall, Dávid Kopányi, Kai-Uwe Kühn, Xavier Lambin, Leonardo Madio, Alexander Rasch, Magda Rola-Janicka, Maarten Pieter Schinkel, Ulrich Schwalbe, and Leonard Treuren, as well as to Gary Biglaiser as editor and two anonymous referees for their constructive comments. Errors remain my own. This article has benefited from presentations at the ESWM 2018 in Naples, SMYE 2019 in Brussels, RES Annual Conference 2019 in Coventry, BECCLE Competition Policy Conference 2019 in Bergen, CCP Annual Conference 2019 in London, EARIE 2019 in Barcelona and JEI 2019 in Madrid, as well as seminars at the UK's Competition and Markets Authority, DG Competition, the Joint Research Centre of the European Commission, the University of Amsterdam, Oxera Consulting LLP, the University of Hohenheim, and the Düsseldorf Institute for Competition Economics.

[Correction added on August 14, 2021 after first Online publication: Author's correct email ID and copyright line was updated].

[Article updated on September 15, 2021 after first online publication: The fourth paragraph in section 2 was missing in initial publication and is now inserted].

in a price cartel, but without any overt act of communication or agreement required to establish a competition law infringement (Harrington, 2018). The debate has received extensive press coverage and increasing interest from authorities.[1] Beyond hypothetical concerns, recent empirical research already suggests that the adoption of self-learning pricing algorithms can indeed have negative effects on competition (Assad et al., 2020). However, exactly how algorithms may lead to autonomous collusion is an open research question.

To show more formally whether and how autonomous algorithms can collude, I investigate the collusive capacity of reinforcement learning. Reinforcement learning is the type of machine learning in which the algorithm learns by itself through autonomous trial-and-error experimentation. More specifically, I investigate the collusive capacity of Q-learning, which is a foundational reinforcement learning algorithm upon which many of the recent breakthroughs in artificial intelligence are based.[2] I use Q-learning as a proof of concept in this case: Autonomous learning is used in real-world pricing applications but it is unlikely that pricing algorithms in use are completely and fully based on Q-learning, as a result of several practical limitations.[3] However, in the final section, I discuss how these practical limitations may be dealt with and performance improved using more advanced machine-learning techniques.

Q-learning is based on the theory of dynamic programming and uses recursive value-function estimation to maximize the net present value of future rewards. The general approach (discussed in more detail in Section 3) is that the algorithm learns iteratively what the long-run value is of taking a certain action in a certain state of the world, taking into account how its action is likely to affect the future state of the world. In picking an action, it continuously balances the need for exploration (picking different actions in order to learn) with the need for exploitation (picking the perceived optimal action to maximize some reward function). In single-agent environments, Q-learning is theoretically guaranteed to converge to optimal behavior under mild conditions. However, this theoretical guarantee is absent when multiple interacting Q-learning algorithms are learning simultaneously. In the absence of theoretical guarantees, I therefore provide an empirical understanding through simulations.

Self-learning algorithms are programmed in discrete time. It may be very unlikely, however, that competing algorithms update their prices at exactly the same time (or, alternatively, that they are unaware of the current competitor prices and hence act "as if" prices are set simultaneously). We therefore deviate from the conventional infinitely repeated simultaneous move framework and use the sequential move framework of Maskin and Tirole (1988) instead, in which firms take turns setting prices. A recent article by Calvano, Calzolari, Denicolò, and Pastorello (2020) (discussed in more detail in Section 2) similarly shows through simulations how Q-learning can lead to collusive strategies. However, they do assume the conventional infinitely repeated simultaneous move framework. Moreover, they require prices to condition on past prices that are no longer payoff-relevant. A meaningful distinction in my sequential setting is that this is no longer required.

[1] Press coverage includes for instance the Financial Times (2017), Frankfurther Allgemeine Zeitung (2018), Harvard Business Review (2016), Politico (2018), The Economist (2017), the The New Yorker (2015) and the The Wall Street Journal (2017). The increased interest from authorities becomes clear from speeches (Delrahim, 2018; Ohlhausen, 2017; Powers, 2020; Vestager, 2017), hearings (FTC, 2018) and reports (Autoridade de Concurrencia, 2019; Autorité de la Concurrence Bundeskartellamt, 2019; Competition and Markets Authority, 2018, 2021; OECD, 2017).

[2] This includes in particular the self-learning of superhuman play in complex board games like Go and chess (Silver et al., 2016, 2017, 2018) and Atari video games (Mnih et al., 2015). See Kohs (2017) for the Netflix documentary on the breakthrough by Google DeepMind in achieving superhuman play in the board game Go using their AlphaGo reinforcement learning algorithm.

[3] Companies offering pricing software that uses autonomous learning include a2i systems (which optimizes fuel pricing, www.a2isystems.com), Eversight Labs (which helps consumer goods companies optimize their pricing, www.eversightlabs.com) and RepricerExpress (which helps third-party Amazon sellers optimize their pricing, www.repricerexpress.com). Assad et al. (2020) provide a discussion of such real-world algorithms and Den Boer (2015) a review on dynamic pricing in the operations research literature.

My main finding is that when the number of discrete prices is limited, competing Q-learning algorithms indeed often coordinate on collusive equilibria. The intuition behind this outcome is that Q-learning first learns the short-run benefit of slightly undercutting its competitor, but then also learns through experimentation the longer-run benefit of "resetting" this gradual price decline with a large price increase once prices become low. When the number of discrete prices is limited, I find that competing Q-learning algorithms often identify a stable high price as mutually optimal, with the gradual price decline functioning as temporary off-equilibrium punishment. When the number of discrete prices increases, Q-learning increasingly converges to a stable supra-competitive Edgeworth price cycling pattern (in which periodic price jumps reset a gradual price decline). This pricing pattern is similar to that regularly observed in other markets often suspected of tacit collusion—in particular gasoline markets (Eckert, 2013; Byrne and de Roos, 2019), which are also the subject of the recent empirical study of Assad et al. (2020). Although generally not an equilibrium outcome, I find that these asymmetric cycles do push average prices above their (Markov-perfect) competitive level. Coordination on collusion or cycles occurs even though the algorithm does not communicate and is only instructed to maximize its own profits. I show that results are robust to reasonable changes to the learning parameters and discuss how more advanced algorithms may improve results and deal with less stylized environments.

The remainder of this article is organized as follows. Section 2 provides a review of the literature so far on the broader question of how pricing algorithms may undermine competition. Section 3 defines the competitive environment, the algorithm and the performance metrics used in this article. Section 4 discusses the baseline empirical results and shows the collusive reward–punishment strategies learned. Finally, Section 5 discusses several comparative statics and robustness checks and Section 6 concludes with a discussion on the practical limitations and policy implications.

2. Literature review

■ The inception of the academic debate around pricing algorithms and collusion is generally ascribed to the legal work of Mehra (2016) and Ezrachi and Stucke (2016, 2017). These works raise two legal concerns. First, algorithms may make it easier to implement explicit or tacit collusive agreements—driven by better price monitoring and quicker retaliation in case of defection. Second, if algorithms learn by themselves to adopt collusive strategies, this would not be illegal under current competition laws.[4]

However, these concerns are not universally shared. Although algorithms may help to implement or stabilize collusion, Kuhn and Tadelis (2017) and Schwalbe (2018) point out that it is not clear how algorithms can resolve the coordination problem: Competitors still need to agree on any one particular collusive outcome and the associated pricing strategy necessary to stabilize this outcome. These authors argue, based to a large extent on experimental economic evidence, that solving the coordination problem realistically requires some form of illegal communication. Moreover, truly autonomous—and hence potentially legal—algorithmic collusion is yet to be observed.[5] However, the absence of observed cases is no proof of impossibility or even improbability.

So what does the economic and computer science literature say? In this section, I review the literature on the broader question of how pricing algorithms undermine competition. Although there is some overlap, this literature can generally be divided into four strands, based on the type of algorithm investigated: static optimization algorithms, algorithms involving a credible commitment, demand-prediction algorithms and dynamic optimization algorithms. This article

[4] Harrington (2018) provides additional important legal contributions on this.

[5] A well-known illustration on pricing algorithms is *The Making of a Fly*, a biology textbook sold on Amazon. One algorithm each day priced 25% above its competitor, whereas the competitor automatically price-matched. This caused prices to escalate (up to $23 million per copy). This was of course not collusion, but simply a bug. In *Topkins* and *GB Eye-Trod*, online poster retailers did use algorithms to coordinate prices, but these were not autonomous.

belongs to the last strand. Below, each strand is covered in turn and contrasted with this article, followed by a brief discussion on the existing empirical evidence and frontier computer science literature.

In operations research and management science, pricing algorithms are often used to estimate and optimize finite-horizon or even one-period, static profit functions (Den Boer, 2015). In theory, finite-horizon optimization cannot lead to stable collusion, as it does not allow for strategies necessary to overcome the prisoner's dilemma dynamics (Milgrom and Roberts, 1990). However, Cooper, Homem-de-Mello and Kleywegt (2015) show that the practice of estimating monopoly models in operations research (ignoring competitor pricing) may still lead to an underestimation of the price elasticity of demand and hence inadvertent cooperative pricing.[6] Hansen, Misra and Pai (2020) and Huck, Normann and Oechssler (2003, 2004) show in different simulation settings that such cooperative pricing can also be driven by an inadvertent correlation in price experimentation.[7] The intuition behind this outcome is that as mutually incognizant competitors start to price similarly, they are no longer able to identify undercutting as profitable. This result may however not be robust to minor fluctuations in the payoff function (Izquierdo and Izquierdo, 2015). The algorithm that I consider does not ignore competitor prices and hence forecloses these misspecification mechanisms. Additionally, I look at dynamic, multi-period optimization and check whether the supra-competitive pricing is even an equilibrium

Algorithms can also lead to higher prices when they involve some form of a credible short-run commitment to pricing strategies. Brown and MacKay (2020) show theoretically that when algorithms can be used to implement a continent pricing strategy, firms will adopt different pricing frequencies under which the lowest equilibrium price increases relative to static Nash. Brown and MacKay also provide high-frequency online retail data consistent with this. Comparably, Salcedo (2015) shows theoretically that under certain sufficient conditions collusion between learning algorithms is inevitable, provided firms adopt a short-run fixed-strategy pricing algorithm that periodically "decodes" the other algorithm and subsequently adjusts. However, these articles do not look at whether algorithms can learn to collude, but whether their use changes the pricing game such that the unique static equilibrium involves higher prices (the relevance of this notwithstanding). They also assume common knowledge of the strategies used by the competition, which I do not.

As opposed to learning or implementing pricing strategies, algorithms may also be used to better forecast current or future demand, which in turn can affect equilibrium behavior. Miklós-Thal and Tucker (2019) and O'Connor and Wilson (2021) show that there are theoretically ambiguous effects on cartel stability and welfare in the presence of better demand forecasting—relying respectively on the observed stochastic demand framework of Rotemberg and Saloner (1986) and the unobserved stochastic demand framework of Green and Porter (1984). Harrington (2020) shows theoretically how the existence of a third-party pricing algorithm that can predict demand better leads to higher market prices, as the third party takes into account that the algorithm may face itself. In contrast to these articles, I do not consider demand prediction algorithms.

The fourth strand of the literature—to which this article belongs—focuses on algorithms designed to optimize some infinite-horizon objective function. Tesauro and Kephart (2002) and Noel (2008), for instance, show how dynamic programming can be used to simulate supra-competitive equilibria in an infinitely repeated sequential pricing game—although assuming full knowledge and ignoring the coordination problem. In a more realistic informational environment,

[6] Meylahn and Den Boer (2021) show how the independent adoption of an equivalent pricing algorithm can guarantee stable collusive prices when the algorithm is explicitly designed to maximize joint profit conditional on mutual adoption.

[7] Hansen, Misra and Pai look at Upper Confidence Bound (UCB) algorithms, which explores actions (prices) that have the highest potential of having an optimal value, and Huck, Normann and Oechssler implement a 'win-continue, lose-reverse' rule under quantity competition.

Xie and Chen (2004) show through simulations that when competitors simultaneously set inventory and prices under stochastic demand, Q-learning converges to a stable Nash equilibrium when ignoring competitors. However, using the same approach, Dogan and Güner (2013) find supra-competitive profits when the algorithms can also condition on own and competitor past prices. Additionally, Waltman and Kaymak (2008) show that Q-learning leads to supra-competitive outcomes in infinitely repeated Cournot competition. However, these two articles do not test for equilibrium behavior.[8]

My article is closest to that of Calvano, Calzolari, Denicolò, and Pastorello (2020). These authors show how Q-learning is able to learn collusive strategies when competing algorithms update their prices at exactly the same time. Their results generally align with what I find. The main difference is that they use the conventional model of simultaneous competition. However, it may be very unlikely that competing pricing algorithms update their prices simultaneously (or have to act "as if"). Additionally, for collusion to occur in their simultaneous move setting, they require that algorithms can condition on own and competitor past prices (which are no longer payoff-relevant). Modeling the problem sequentially no longer requires conditioning on own past prices and provides a natural rationale for why current opponent prices should be taken into account. Moreover, the conditioning by Calvano et al. on own as well as competitor past prices increases the state space at least quadratically and greatly increases the required learning duration.

Abada and Lambin (2020) take the same approach as Calvano et al. and myself and apply it to a dynamic arbitrage problem motivated by energy markets. They also find that Q-learning converges to supra-competitive outcomes, and discuss ways in which regulators may (partially) frustrate the collusive learning processes. Similarly, Johnson, Rhodes, and Wildenbeest (2020) simulate competing Q-learning algorithms to show how competition can be improved by providing longer prominence to sellers that display behavior consistent with deviation from a collusive agreement—supporting their theoretical results on demand-steering policies. As with Calvano et al., these two articles rely on an environment of simultaneous move—which I depart from.

Empirical evidence on the causal effect of pricing algorithms, based on real-world data, remains limited.[9] Uniquely, Assad et al. (2020) proxy which and when retail gasoline stations in Germany adopted algorithmic pricing technology and show that competitive pressures decrease following adoption (correcting for the endogeneity of adoption). When two competing stations both adopted algorithms, margins even increase on average by 28%. Interestingly, they find that these higher margins occur gradually, consistent with a gradual learning process as in Calvano et al. (2020) and this article. The authors do treat the algorithms as black boxes. In this article, I look at the inner workings of reinforcement learning applied to a controlled pricing environment.

Looking instead at the theoretical computer science literature, there is a strand that deals with so-called multi-agent reinforcement learning algorithms—which combines (evolutionary) game theory with reinforcement learning. Q-learning is often used as a starting point here.[10] However, strong theoretical results from this literature have been limited and convergence has generally not been shown beyond very simple matrix games. It therefore seems unlikely that answers on autonomous algorithmic collusion are readily to be found on the current frontier of

[8] In fact, Waltman and Kaymak also find supra-competitive outcomes under conditions in which reward–punishment strategies are not even possible (no memory), which suggests that the higher prices are not actually supported by equilibrium behavior.

[9] A 2017 e-commerce sector inquiry by the European Commission does show that many retailers automatically adjust prices in response to competitors. Similarly, Chen, Mislove, and Wilson (2016) show that in 2015 more than one-third of best-selling Amazon products adopted algorithmic pricing, and these tended to have higher price (and sales). However, this ignores causation.

[10] For instance, Nash-Q maintains Q-functions over joint actions and performs updates based on assuming equilibrium behavior (Hu and Wellman, 2003) and Hyper-Q chooses between mixing strategies and uses estimated opponent strategies as state variables (Tesauro, 2003).

theoretical computer science.[11] The empirical or simulation-based computer science literature has been more successful at providing recent breakthroughs in reinforcement learning though. This is for instance the case with the recent superhuman performance in high-dimensional singe-player environments like Atari video games (Mnih et al., 2015) or complex zero-sum board games like Go and chess (Silver et al., 2016, 2017, 2018; see also Kohs, 2017). Yet, it is relevant to note that breakthroughs are still generally in the context of single-player games, zero-sum games or simple matrix games.[12] Given that the frontier computer science literature does not seem to consider oligopoly environments yet, I simply focus on Q-learning as a simple but foundational reinforcement learning algorithm and do not consider more state-of-the-art algorithms.

3. Environment and learning algorithm

■ This article investigates the collusive capacity of reinforcement learning in an environment of sequential competition—which I have argued reflects more naturally the setting of algorithmic price competition than simultaneous competition. The particular algorithm that I look at is Q-learning, which is a straightforward and foundational reinforcement learning algorithm. This section discusses the pricing environment of Maskin and Tirole (1988) as used in the simulations, the Q-learning algorithm as adapted to this environment and the performance metrics considered.

□ **Sequential pricing duopoly.** To capture the dynamics of sequential pricing, I take the infinitely repeated sequential move pricing duopoly environment of Maskin and Tirole (1988). Below, I describe this environment as applied here and its equilibrium behavior.

Competition between two firms $i \in \{1, 2\}$ takes place in infinitely repeated discrete time indexed by $t \in \{0, 1, 2, \ldots\}$. Adjustments in price occur sequentially: only in odd-numbered periods, firm 1 adjusts its price $p_{1t} \in P$ and only in even-numbered periods, firm 2 adjusts its price $p_{2t} \in P$. Price is a discrete variable scaled between 0 and 1 and with k equally sized intervals—so prices are taken from a discrete set $P = \{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\}$. Assuming no marginal or fixed cost, firm i profit at time t is simply derived as

$$\pi_i(p_{it}, p_{jt}) = p_{it} D_i(p_{it}, p_{jt}), \tag{1}$$

where $D_i(p_{it}, p_{jt})$ its demand as function of own price p_{it} and competitor price p_{jt}, with $j \in \{1, 2\} \setminus i$. Firms discount future profits with a discount factor $\delta \in [0, 1)$, where each firm has as its objective to maximize at time t its cumulative stream of discounted future profits, so

$$\max \sum_{s=0}^{\infty} \delta^s \pi_i(p_{i,t+s}, p_{j,t+s}). \tag{2}$$

In showing whether and to what degree autonomous collusion using Q-learning is possible, I restrict myself to the simple setting of homogeneous goods with linear demand, which is also the baseline case of Maskin and Tirole. Demand has an intercept and slope equal to 1 such that

$$D_i(p_{it}, p_{jt}) = \begin{cases} 1 - p_{it} & \text{if } p_{it} < p_{jt} \\ 0.5(1 - p_{it}) & \text{if } p_{it} = p_{jt} \\ 0 & \text{if } p_{it} > p_{jt} \end{cases}. \tag{3}$$

This provides as monopoly or joint-profit maximizing collusive price $p^C = 0.5$, with an associated per-firm profit of $\pi_i = 0.125$. Note that this simple demand function is for expositional

[11] Introductions to and reviews of the literature on multi-agent reinforcement learning are provided in particular by Bloembergen et al. (2015), Busoniu, Babuska, and De Schutter (2008), Hernandez-Leal et al. (2017), Shoham, Powers, and Grenager (2007), and Tuyls and Weiss (2012).

[12] For instance, Crandall et al. (2018) show how state-of-the-art reinforcement learning is capable of cooperating in simple repeated matrix games, allowing also for some form of signalling. It is unclear how results hold up in an oligopoly setting and absent illegal signalling.

purposes and is in fact unknown to the algorithm. I also follow Maskin and Tirole in imposing the Markov assumption: strategies only depend on variables that are directly payoff-relevant, which in this case is limited to the previous competitor price $p_{j,t-1}$ and does not include, for instance, communication or the history of prices. The strategy of firm i is therefore a dynamic reaction function $R_i(\cdot)$, where in its turn $p_{it} = R_i(p_{j,t-1})$.

The equilibrium outcomes in this setting can be described as follows: A (possibly randomizing) strategy pair (R_1, R_2) is a Nash equilibrium if for all prices along the equilibrium path the following value-function condition holds for both firms:

$$V_i(p_{jt}) = \max_p \left[\pi_i(p, p_{jt}) + E_{p_{j,t+1}}\left[\delta\pi_i(p, p_{j,t+1}) + \delta^2 V_i(p_{j,t+1})\right]\right], \tag{4}$$

where reaction function $R_i(p_j)$ is a maximizing choice of firm i and the expectation over competitor response $p_{j,t+1}$ is taken with respect to the distribution of $R_j(p)$.

One Nash equilibrium here is the static Nash outcome in which firms always price at or one increment above marginal cost, although more equilibria exist for a sufficiently high discount factor.[13] As a refinement of the Nash equilibrium, Maskin and Tirole define the concept of a Markov-perfect equilibrium (MPE), which is a subgame perfect Nash equilibrium under the Markov assumption. A strategy pair (R_1, R_2) is an MPE if Condition (4) holds for both firms and for all prices, including off-equilibrium prices. They show that if firms value future profits sufficiently highly there are two sets of MPE: focal price equilibria and Edgeworth price cycle equilibria. First, in focal price equilibria, both firms sustain a fixed price with the common belief that the other firm would undercut if it were to decrease its price and not follow if it were to increase it. Such beliefs are sustained by off-equilibrium price wars in case any firm undercuts, in which case prices drop and firms mix between staying at lower prices and returning to the fixed price. Second, in Edgeworth price cycle, equilibria firms gradually undercut each other. When further price cuts become too costly, both firms have an incentive to raise their price and reset the gradual downward spiral but prefer the other firm to do so. They therefore mix between maintaining lower prices to punish the other firm for not resetting the price cycle, and resetting themselves.

□ **Sequential Q-learning.** The learning algorithm applied here is an adaptation of Q-learning to sequential interaction. Q-learning is a simple and foundational reinforcement learning algorithm that aims to maximize the net present value of expected future rewards for unknown environments with repeated interaction—where actions affect both the immediate payoff and future states of the world. It was originally proposed by Watkins (1989) to solve unknown Markov decision processes, which are discrete time stochastic processes in which actions affect both current reward and the next state in an otherwise stationary environment. Below, the sequential-move adaptation as used in this article is discussed in detail.[14] Calvano et al. (2020) and Johnson, Rhodes, and Wildenbeest (2020) also provide a general primer on Q-learning.

Q-learning, like any reinforcement learning algorithm, consists of two interacting modules: a *learning module* that processes the observed information and an *action-selection module* that balances exploitation (choosing the currently perceived optimal action) with exploration (choosing perhaps another action, to learn what happens). Each module as it is applied in this setting is discussed in turn.

Learning module. Q-learning estimates a Q-function $Q_i(p_{it}, s_t)$, which maps for firm $i \in \{1, 2\}$ action p_{it} (new own price at time t) into its estimated optimal long-run value given current state $s_t \in S$. Because the choice set is discrete, Q_i is a $|P| \times |S|$ matrix in this case. After observing

[13] To see that one increment above marginal cost is a Nash equilibrium, assume that $R_2(p_1) = \frac{1}{k}$, such that firm 2 always prices one increment above zero marginal cost. Condition (4) then simplifies to $V_1(\frac{1}{k}) = \frac{1+\delta}{1-\delta^2}\max_p \pi_1(p, \frac{1}{k})$. A maximizing choice of firm 1 is then similarly $R_1(p_2) = \frac{1}{k}$, which, by symmetry, is a Nash equilibrium.

[14] For a textbook treatment on Q-learning, see Sutton and Barto (2018).

own profits and new state s_{t+1}, the algorithm updates entry $Q_i(p_{it}, s_t)$ according to the following recursive relationship:

$$Q_i(p_{it}, s_t) \leftarrow (1-\alpha) \cdot \text{previous estimate} + \alpha \cdot \text{new estimate}, \tag{5}$$

$$\text{previous estimate} = Q_i(p_{it}, s_t)$$

$$\text{new estimate} = \pi(p_{it}, s_t) + \delta\pi(p_{it}, s_{t+1}) + \delta^2 \max_p Q_i(p, s_{t+1}),$$

where $\alpha \in (0, 1)$ is a step-size parameter that regulates how quickly new information replaces old information and $\delta \in [0, 1)$ is again a discount factor.

Note that the new estimate of the optimal long-run value given state s_t consists of three components: direct profit $\pi(p_{it}, s_t)$, next period profit $\pi(p_{it}, s_{t+1})$ when new state s_{t+1} realizes but the price has not changed (discounted for one period), and the highest possible Q-value $\max_p Q_i(p, s_{t+1})$ in this new state s_{t+1} (discounted for two periods). This enables a recursive value-function approximation in which initially the Q-values are imprecise, but over time, they become better estimates of the long-run consequences of choosing p_{it} in state s_t, allowing for convergence.

There are three more things worth noting about this learning module. First, under the Markov assumption current and new states s_t and s_{t+1} are equivalent to current and new competitor prices p_{jt} and $p_{j,t+1}$. Second, note the parallel between Condition (4) and Equation (5). This comes from the fact that through recursive updating, Q-learning aims to solve for a dynamic programming condition, and third, in the learning module each time only one entry within the Q-matrix is updated. Such tabular learning leads to a slow learning process. To speed up learning, and allow for continuous state and action spaces, function approximations could be used. This would however increase the amount of parameters and modeling assumptions and is left for future research.

Action-selection module. In balancing exploration and exploitation, the algorithm adopts a probabilistic action-selection policy. I simply use a straightforward procedure called ε-greedy exploration: With probability $\varepsilon_t \in [0, 1]$, it selects a price randomly (exploration) and with probability $1 - \varepsilon_t$, it selects the currently perceived optimal price (exploitation), so

$$p_{it} \begin{cases} \sim U\{P\} & \text{with probability } \varepsilon_t \\ = \text{argmax}_p Q_i(p, s_t) & \text{with probability } 1 - \varepsilon_t, \end{cases} \tag{6}$$

where $U\{P\}$ is a discrete uniform distribution over action set P. In case of ties under exploitation, the algorithm randomizes over all perceived optimal actions.

Note that ε-greedy exploration is very untargeted: When exploring, it selects any price randomly. As with the learning module, the action-selection module could be improved by using more sophisticated techniques but this is outside the scope of this article. A pseudocode of the entire algorithm as used in the simulations is provided below.

Pseudocode Sequential Q-Learning (Simulation)

```
Set demand and learning parameters; Initiate Q-functions
Initialize {p_1t, p_2t} for t = {1, 2} randomly
Initialize t = 3, i = 1 and j = 2
Loop over each period
|  Update Q_i(p_i,t−2, p_j,t−2) according to (5)
|  Set p_it according to (6) and set p_jt = p_j,t−1
|  Update t ← t + 1 and {i ← j, j ← i}
Until t = T (specified number of periods)
```

□ **Theoretical limitations.** There are two theoretical limitations in the above specification that justify my empirical approach through simulations. First, in a multi-agent setting, there are no theoretical convergence guarantees for Q-learning. When a single Q-learning agent faces a fixed-strategy competitor, it is guaranteed to converge to the optimal (rational, best-response) strategy, given mild conditions on step-size parameter α and the rate of exploration ε_t (Watkins, 1992). However, in our setting, Q-learning remains vulnerable to adaptation and experimentation by its opponent. More generally, agents that are simultaneously adapting to the behavior of others face a moving-target learning problem (Busoniu, Babuska, and De Schutter, 2008; Tuyls and Weiss, 2012), in which their best response changes as others change their strategies. Convergence guarantees that exist for single-agent reinforcement learning algorithms then no longer hold.

Second, Q-learning is restricted to playing pure strategies whereas the different MPE identified by Maskin and Tirole require mixing strategies—either off-equilibrium (in case of the focal price) or along the equilibrium path (in case of the Edgeworth price cycles). Although it is incapable of learning subgame perfect equilibria or equilibria which require mixing strategies on the equilibrium path, subgame imperfect Nash equilibria do remain possible. Despite this and the previous limitation, however, the algorithm does not have to perform badly in practice. It only means that theory is unable to say how well it is expected to behave. In the absence of theoretical guarantees, I therefore provide an empirical understanding through simulations.

□ **Performance metrics.** In assessing the performance of the algorithm, I look at how profitable it is at the end of the simulations, how optimal it is relative to best-response behavior and whether it has converged to a Nash equilibrium. I do this for many different runs, in order to assess the average and distribution of performance.

Profitability. I evaluate the final profitability of any one run lasting T periods by looking at the average profit in the final 1,000 periods of this run, so

$$\text{Profitability: } \Pi_i \doteq \frac{1}{1{,}000} \sum_{t=T-1{,}000}^{T} \pi_i(p_{it}, p_{jt}), \tag{7}$$

where I omit a subscript indicating the specific run. The average is taken because pricing can be dynamic and profits can fluctuate such that a low profit in any one period may be offset by a higher profit in another period and vice versa. Looking only at final-period profit fails to capture this. I average over the final 1000 periods as pricing cycles are always significantly shorter than that—as discussed in Section 5.

I compare profitability against two benchmarks: the joint-profit maximizing benchmark of 0.125 (which is the profit that occurs where both firms set $p_i = 0.5$) and a competitive benchmark. The competitive benchmark is not trivial, however. An obvious candidate may seem to be the static Nash outcome of prices equal to (or one increment above) marginal cost. However, the sequential environment makes pricing at or one increment above marginal cost not subgame optimal (for a sufficiently high discount factor). Although marginal cost is still an interesting benchmark for practical purposes, I take the more conservative (higher) competitive benchmark that approximates the most competitive Edgeworth price cycle MPE identified by Maskin and Tirole (1988): Firms undercut each other by one increment until prices reach their lower bound, after which one firm resets prices to one increment above monopoly price and the cycle restarts. It is taken that the first firm that observes the lower-bound price resets the price cycle. This provides in this case an average per-period profit of approximately 0.0611 for $k = 6$ (which increases in the limit of k to approximately 0.0833).

Optimality and Nash equilibrium. To capture a degree of optimality, I define $\Gamma_i(p_i, p_j)$ as the ratio of estimated and best-response discounted future profits at the end of the simulation (as

captured by the associated Q-values), so

$$\text{Optimality:}\Gamma_i(p_i, p_j) \doteq \frac{Q_i(p_i, p_j)}{\max_p Q_i^*(p, p_j)}, \tag{8}$$

where Q_i^* is the optimal Q-function given current competitor strategy (and I again omit a subscript indicating the specific run). Q_i^* is not observed by the algorithm, but can be computed exactly by keeping the competitor Q-function fixed and looping over all action-state pairs until Equation (5) converges. $\Gamma_i(p_i, p_j)$ is evaluated at the prevailing p_i and p_j at the end of the simulation and hence we omit the function notation from now on.

Γ_i has the following interpretation: it shows in percentage terms how much the estimated discounted future profits are below the discounted future profits under best-response behavior given current competitor strategy. When the algorithm learns a best-response strategy, it produces $\Gamma_i = 1$. An outcome is therefore a Nash equilibrium if and only if $\Gamma_i = 1$ holds for both algorithms. In evaluating the performance of the algorithm, I also look at the share of Nash equilibria over all runs (where I allow for a tolerance of 0.00001).

Note finally that Γ_i does not only take into account the next period best-response behavior, but also possible off-equilibrium exploitation of its competitor in states that are otherwise never visited, and note that for Γ_i to be reliable, step-size parameter α or the rate of exploration ε_t has to decrease sufficiently. This allows Q_i to converge and become a reliable estimate of actual discounted future profits. In the simulations, I impose that ε_t goes toward zero toward the end.

Collusive equilibrium. I consider the outcome of a run a collusive equilibrium when profitability is above the competitive benchmark and the algorithms have adopted strategies such that neither can improve given the strategy of the other algorithm (i.e., they are in a Nash equilibrium and $\Gamma_i = 1$ for both firms).

The key characteristic of a collusive equilibrium is the use of a "reward-punishment scheme which rewards a firm for abiding by the supra-competitive outcome and punishes it for departing from it" (Harrington, 2018). Similarly to Calvano et al. (2020), I test in the results section for the existence of reward–punishment strategies. This is done by forcing a deviation by one of the firms at the end of the simulation and observing subsequent responses. Xie and Chen (2004) use a similar approach to test for convergence to a steady Nash equilibrium, which they call a "Nash test".

4. Results

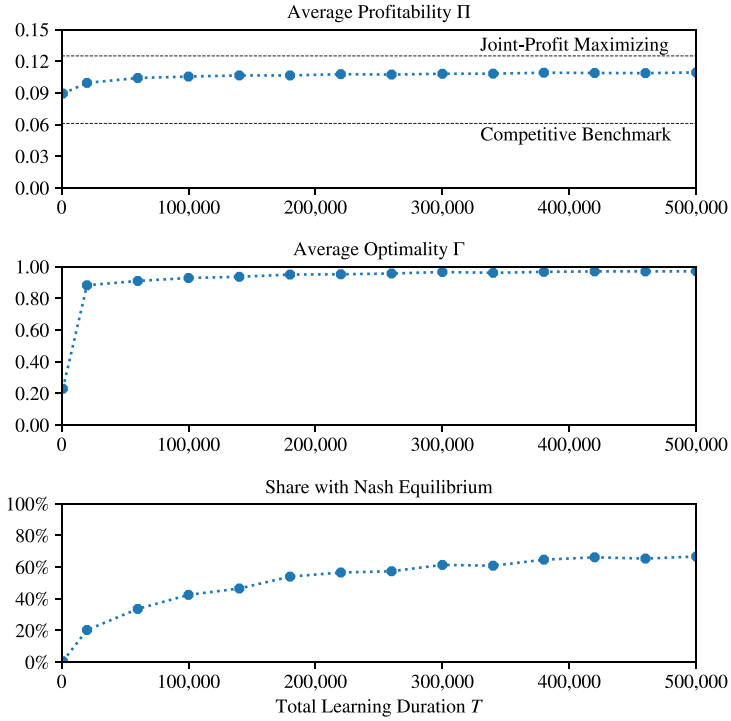
■ For the baseline simulation, I look at $k = 6$ price intervals between 0 and 1, which is the illustrating example in Maskin and Tirole (1988) and the lowest amount of price intervals at which both fixed-price and price cycle MPE exist. To assess the average and distribution of performance, I simulate 1000 runs. In the baseline simulation, I set step-size parameter $\alpha = 0.3$ as a reasonable compromise between the need to ensure learning is not too slow (α too close to 0) and the need to ensure it does not forget too rapidly what it has learned in the past (α too close to 1). I set discount factor $\delta = 0.95$ reasonably close to 1 as periods are generally small. I vary $\{k, \alpha, \delta\}$ in the next section.

I evaluate the average profitability, average optimality and the share of Nash equilibria at the end of the simulations, where I vary each time the total amount of learning periods T. I set the probability of exploration as $\varepsilon_t = (1 - \theta)^t$, where decay parameter θ is set such that the probability of exploration gradually decreases from 100% at the beginning to 0.1% halfway through the run, reaching 0.0001% at the end (so $\varepsilon_{0.5T} = 0.001$ and $\varepsilon_T = 0.000001$). Finally, the Q-values are initiated with all zeros, although results are not sensitive to initialization.

Figure 1 shows that when two Q-learning algorithms face each other sequentially, they manage to converge to profits that are on average supra-competitive, although below the joint-profit maximizing level. When the total amount of learning periods increases, the average optimality is around 97% and the share of Nash equilibria around 67%. The left-hand panel in Figure 2

FIGURE 1

BASELINE PERFORMANCE UNDER DIFFERENT LEARNING DURATIONS T
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: Results are in case of amount of price intervals $k = 6$, step-size parameter $\alpha = 0.3$, and discount factor $\delta = 0.95$

illustrates for $T = 500{,}000$ that even though most runs are symmetric and the algorithms converge to profitability levels at or just below the joint-profit maximizing rate, this is not always the case. In a minority of 230 runs, one of the two algorithms ends up with a lower payoff.

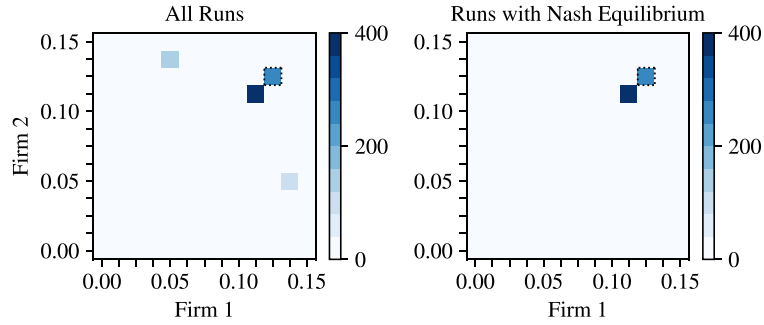
Although average performance is clearly supra-competitive, the key question is whether the algorithms are able to coordinate on collusive equilibria, where profitability is above the competitive level and strategies constitute a Nash equilibrium. The right-hand panel in Figure 2 illustrates that for 667 runs the algorithms indeed managed to coordinate on a collusive equilibrium, 241 of which are on the joint-profit maximizing level. For those runs with a Nash equilibrium outcome, the market price is fixed. For those runs without a Nash equilibrium outcome, the market price generally displays an asymmetric pricing pattern, where prices gradually decrease followed by a sharp increase. This pattern is discussed in more detail in the next section.

The key characteristic of a collusive equilibrium is the existence of reward–punishment strategies, where a firm is rewarded with higher profit by sticking to the collusive outcome and being punished if it deviates. Following the approach in Calvano et al. (2020) of forcing a deviation and observing behavior, Figure 3 shows for those runs where the algorithms managed to coordinate on a joint-profit maximizing Nash equilibrium (241 runs) that the algorithms indeed learn strategies that have the effect of reward–punishment: A deviation by firm 1 triggers a downward price spiral that leads to a net profit loss for firm 1, despite the one-period higher deviation profit. Figure 3 also shows that this punishment effect is temporary, with prices getting back to the monopoly level after a few periods.

Note that although Figure 3 shows that on average prices gradually return to the collusive level, this is actually the consequence of the different runs jumping up in price in different

FIGURE 2

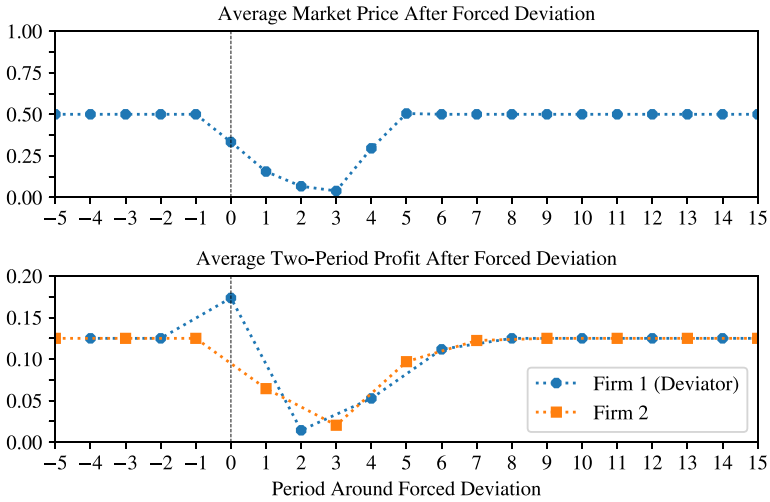
BASELINE JOINT DISTRIBUTION OF PROFITABILITY Π_i
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: Right panel considers only those runs that led to a Nash equilibrium (667 out of 1000 runs). Dotted squares indicate joint-profit maximizing profitability. Results are in case of amount of price intervals $k = 6$, learning duration $T = 500{,}000$, step-size parameter $\alpha = 0.3$, and discount factor $\delta = 0.95$

FIGURE 3

AVERAGE MARKET PRICE AND PROFIT AFTER A FORCED DEVIATION
[Color figure can be viewed at wileyonlinelibrary.com]



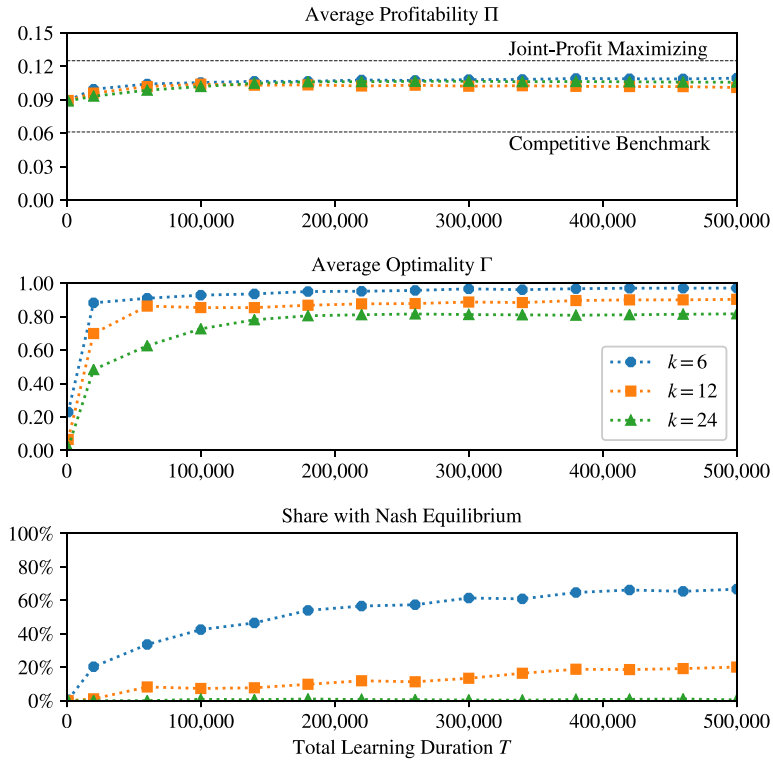
Notes: Results are for those runs that led to a Nash equilibrium outcome on the joint-profit maximizing price. Dotted line indicates moment of deviation. Results are in case of amount of price intervals $k = 6$, learning duration $T = 500{,}000$, step-size parameter $\alpha = 0.3$, and discount factor $\delta = 0.95$

periods. In each of the individual runs, prices shoot back up to the monopoly level after several periods of lower prices and profits (i.e., display one-off asymmetric price cycles).

Interestingly, this off-equilibrium punishment strategy also provides an insight into the intuition behind the learning process that leads to collusive behavior: Q-learning first learns the short-run benefit of slightly undercutting its competitor, but then also learns through experimentation the longer-run benefit of "resetting" this gradual price decline with a large price increase once prices become low. Once the competing Q-learning algorithms have learned the price-cycling behavior, they appear capable of identifying a stable high price as mutually optimal—with the gradual price decline functioning as temporary off-equilibrium punishment.

FIGURE 4

PERFORMANCE UNDER DIFFERENT LEARNING DURATIONS T AND AMOUNT OF PRICING INTERVALS k
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: Results are in case of different amounts of price intervals k, with step-size parameter $\alpha = 0.3$ and discount factor $\delta = 0.95$

5. Comparative statics

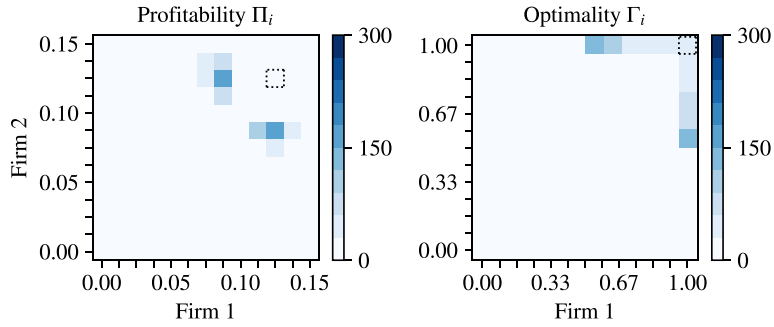
■ The above results show that autonomous algorithmic collusion is possible. In this section, I discuss how results change when I increase the amount of discrete prices the algorithm can choose from. I also show how results are generally robust to changes in step-size parameter $\alpha = 0.3$, discount factor $\delta = 0.95$, and whether the algorithms can also condition on their own past price.

□ **Edgeworth price cycles under more prices.** Figure 4 shows that when the amount of pricing intervals k increases, average profitability remains above the competitive benchmark for the different total learning durations T. However, Q-learning appears to have increasing difficulty in learning strategies that are best responses to its competitor—with lower average optimality when k increases and nearly no Nash equilibria when $k = 24$. The left panel in Figure 5 shows a clear dichotomy that underlies the profitability results: Generally, only one of the two algorithms has a profitability that is around (or even above) the joint-profit maximizing level, whereas the other algorithm has a lower profitability. The right panel in Figure 5 shows a similar dichotomy for optimality. It is the case that the algorithm with the higher profitability is also the one with an optimality equal to one.

So what underlies these different outcomes when the algorithms have more prices to choose from? Whereas under $k = 6$ the algorithms often converge to a collusive equilibrium with a fixed

FIGURE 5

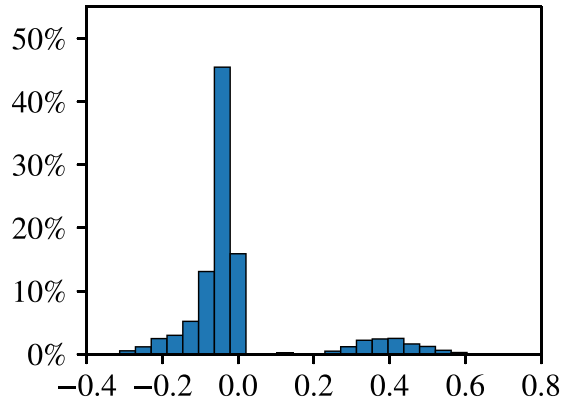
JOINT DISTRIBUTION OF FINAL PROFITABILITY Π_i AND OPTIMALITY Γ_i FOR $k = 24$
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: Dotted squares indicate joint-profit maximizing profit and Nash equilibrium respectively. Results are in case of amount of price intervals $k = 24$, learning duration $T = 500{,}000$, step-size parameter $\alpha = 0.3$, and discount factor $\delta = 0.95$

FIGURE 6

DISTRIBUTION OF CHANGES IN MARKET PRICE
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: This figure looks at all the changes in market price that occur during the final 100 periods of all runs put together, where each bar represents a possible price change. Results are in case of amount of price intervals $k = 24$, learning duration $T = 500{,}000$, step-size parameter $\alpha = 0.3$, and discount factor $\delta = 0.95$

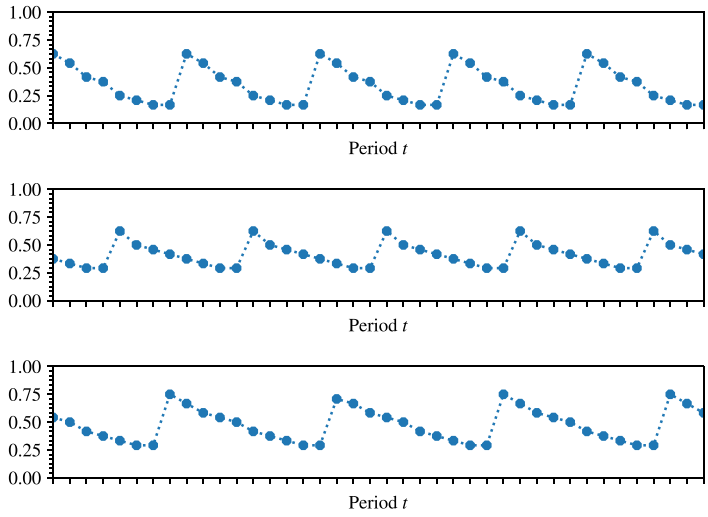
market price, Figure 6 shows that under $k = 24$ final market prices display a clear asymmetric dynamic pattern: Looking at the final 100 periods of all runs, the majority of periods observe a very small price decrease, whereas in a small minority of periods the market price suddenly jumps up by a relatively large amount.

Underlying this asymmetric dynamic pattern is the convergence to deterministic asymmetric price cycles—or Edgeworth price cycles. These Edgeworth price cycles are illustrated in Figure 7, which shows the market price in the final 40 periods of the first three runs of the 1000 runs simulated. It illustrates that when prices have decreased too much, one of the two algorithms has learned to shoot up in price and enable a new gradual price decrease, pushing up average prices and profits and hence keeping average profitability high.

Note that unlike in Maskin and Tirole (1988), these price cycles are deterministic: It is always the same firm that undertakes the costly action of "resetting" the price cycle by jumping

FIGURE 7

ILLUSTRATION OF FINAL MARKET PRICES FOR $k = 24$
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: This figure looks at the market price during the final 40 periods of the first three individual runs. Results are in case of amount of price intervals $k = 24$, learning duration $T = 500{,}000$, step-size parameter $\alpha = 0.3$, and discount factor $\delta = 0.95$

up in price rather than undercutting, with the other firm able to free-ride on this. This difference relative to the stochastic Edgeworth price cycles described theoretically by Maskin and Tirole comes from the fact that Q-learning is a pure-strategy algorithm that cannot learn the mixed-strategy behavior on the equilibrium path that underlies the Edgeworth price cycles of Maskin and Tirole (as also discussed in the Theoretical Limitations subsection of Section 3).

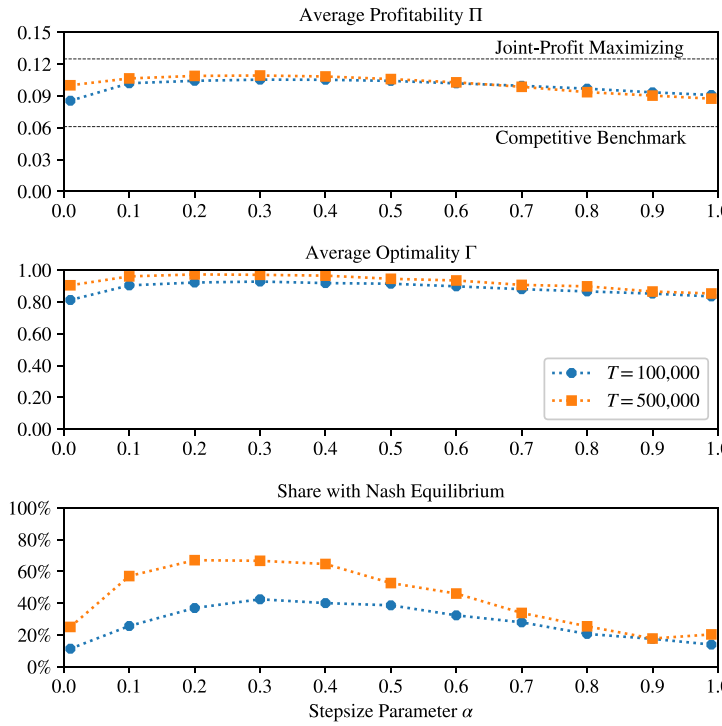
□ **Different step-size parameters.** In the baseline simulation, I have set step-size parameter $\alpha = 0.3$ as a reasonable compromise between the need to ensure learning is not too slow (α too close to 0) and the need to ensure it does not forget too rapidly what it has learned in the past (α too close to 1). Figure 8 confirms that 0.3 is indeed a good compromise, with average profitability, average optimality, and the share of Nash equilibrium runs generally decreasing for step-size parameters that are closer to zero or close to one. Although not shown here, similar results apply when keeping the step-size parameter of one of the two firms fixed (i.e., allowing for asymmetric step-size parameters).

□ **Different discount factor.** I have set discount factor $\delta = 0.95$ reasonably close to 1 as periods are generally small. In case of very short periods, the actual discount factor of a firm would be much closer to 1. However, when setting δ very close to 1, sufficient learning may fail because old Q-value estimates will get too much weight. It may then be required to set a lower δ. Figure 9 shows this. It shows that when δ is low, it consistently learns to coordinate on a static Nash equilibrium outcome. When δ increases, average profitability increases whereas average optimality and the share of Nash equilibrium runs decreases. When δ is set too close to 1, it indeed fails to learn properly and performance collapses. Although not shown here, the same result occurs when setting different step-size parameters α.

□ **Self-reactive conditioning.** Similarly to Maskin and Tirole (1988), I have imposed the Markov assumption, under which the state variable is defined as current competitor price only. In

FIGURE 8

PERFORMANCE UNDER DIFFERENT STEP-SIZE PARAMETERS α
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: Results are in case of amount of price intervals $k = 6$ and discount factor $\delta = 0.95$

other words, the algorithm is not allowed to condition its prices on any history of past prices that are no longer relevant for its current profit. However, in their setting of simultaneous competition, Calvano et al. (2020) consider a Q-learning algorithm that allows for and requires (at least) one-period memory, such that state $s_t = \{p_{i,t-1}, p_{j,t-1}\}$. The cost of this is that it increases the state-space and hence the amount of unique action-state pairs over which the Q-learning algorithm has to learn to optimize.

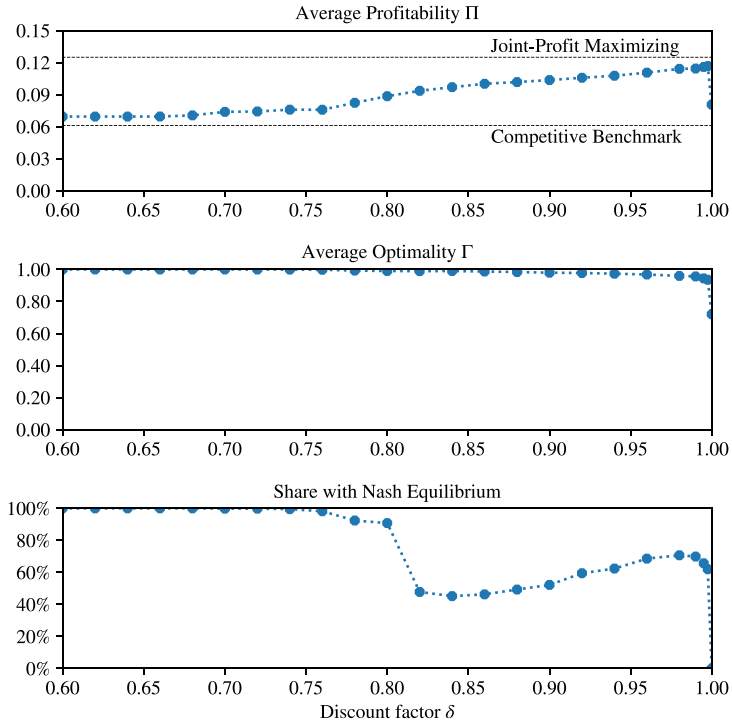
Figure 10 shows that in this setting of sequential competition, also allowing for conditioning on own past price does increase average profitability moderately. However, this comes at the cost of longer learning and less optimality (in terms of approaching best-response behavior). It also has more difficulty into converging to Nash equilibrium behavior—which is not unexpected, given the much larger state-space. Overall performance therefore does not seem to improve in the presence of self-reactive conditioning. The reason why overall performance does not improve in this setting relative to Calvano et al. is that knowing the history of prices does not help the algorithm in learning strategies that involve much more effective reward–punishment effects. The sequential nature of price setting already enables the asymmetric pricing cycles as off-equilibrium punishment strategies.

6. Concluding remarks

■ This article shows that competing pricing algorithms powered by reinforcement learning can learn collusive strategies. This occurs even though the algorithms do not communicate with each other and are only instructed to maximize own profits (i.e., do not receive any instructions

FIGURE 9

PERFORMANCE UNDER DIFFERENT DISCOUNT FACTORS δ
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: Results are in case of amount of price intervals $k = 6$, learning duration $T = 500{,}000$ and parameter $\alpha = 0.3$

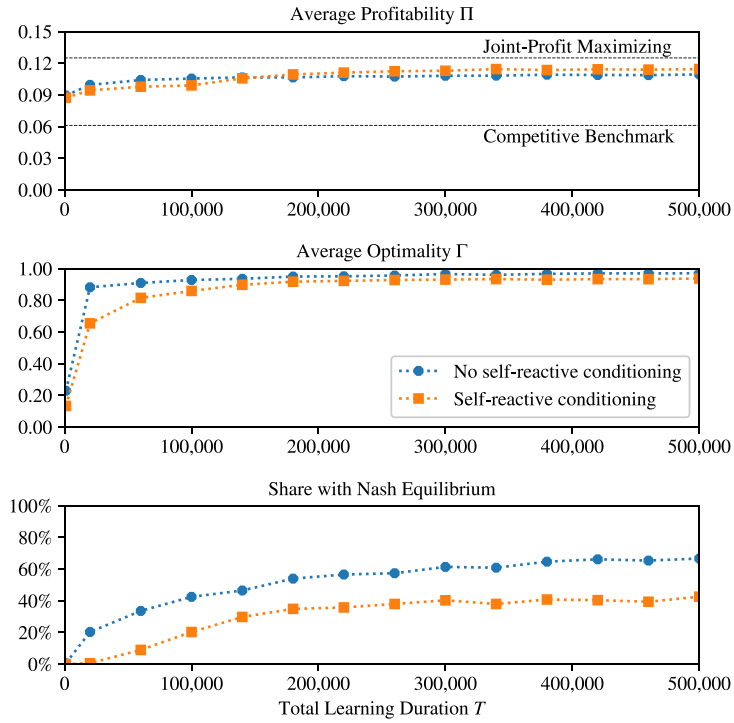
to collude). In this final section, I discuss the practical limitations of Q-learning and how more advanced algorithms may deal with these. I close with several comments on policy implications.

□ **Limitations and future research.** Reinforcement learning techniques are used in pricing applications—in particular, the autonomous exploration of optimal prices. However, as noted in the introduction, I use Q-learning as a proof of concept only: It is unlikely that pricing algorithms observed "in the wild" are completely and only based on Q-learning. This is because Q-learning suffers from three key limitations: It requires many periods of costly experimentation, it may need to adapt its learned behavior when there are structural changes in the environment (such as entry, exit or shifts in cost or demand), and it is not guaranteed to converge to one specific outcome. Additionally, this article only considers a very stylized competitive environment, for expositional purposes. Q-learning is likely to have increasing difficulty in finding optimal strategies under increasingly complex environments.

There are different potential means with which to deal with these limitations in practice. For instance, Calvano et al. (2020) already discuss how pricing algorithms powered by reinforcement learning may be trained in an offline, simulated environment before being put to use in the real world (see also Wang et al., 2018). In fact, this is how reinforcement learning algorithms are trained in board games (Silver et al., 2018) and autonomous driving (Kiran et al., 2002). More importantly, however, a solution to the practical limitations may be to impose more structure on the learning algorithm. The sequential Q-learning algorithm discussed here learns very slowly by design: It only updates one entry in its Q-matrix at a time. Imposing more structure by, for

FIGURE 10

PERFORMANCE UNDER SELF-REACTIVE CONDITIONING
[Color figure can be viewed at wileyonlinelibrary.com]



Notes: Results are in case of amount of price intervals $k = 6$, step-size parameter $\alpha = 0.3$ and discount factor $\delta = 0.95$

instance, modeling a demand function or competitor learning is a very obvious next step to improve the learning process. These two avenues are left for future research.[15]

□ **Policy implications.** The main conclusion of this article is that autonomous algorithmic collusion is in principle possible. This leads to three concrete policy implications. First, we need a better empirical understanding of whether this also occurs in reality. As this article and the literature review show, there are different theoretical competition concerns when it comes to the use of pricing algorithms. Recent empirical evidence on the German retail gasoline market supports in particular the concern around self-learning algorithms (Assad et al., 2020). However, at this stage, it is still unclear what the scope of the concerns are in practice, nor exactly what kind of pricing algorithms are used. This warrants a push for a better empirical understanding. One particularly valuable tool here may be a comprehensive market investigation by authorities into the use of pricing algorithms. Several competition authorities already have the ability to initiate such investigations. Moreover, the European Commission has recently launched a consultation to develop a new competition tool with similar capabilities—and already identifies "the risk of tacit collusion […] due to algorithm-based technological solutions" as a potential topic for investigation (European Commission, 2020).

Second and relatedly, the possibility of autonomous algorithmic collusion raises interesting regulatory questions. Pricing algorithms can involve many pro-competitive effects and

[15] Another valuable avenue for future research may lie with experimental economics, as also discussed by Schwalbe (2018). In our environment of a sequential pricing duopoly, basic Q-learning in any case does not outperform humans as benchmarked in Leufkens and Peeters (2011).

prohibiting their use is sure to be excessive. A more tailored response could for instance restrict what goes into the algorithm. In particular, in my environment, autonomous collusion would be avoided by imposing a rule that firms update prices at the exact same time rather than sequentially (and cannot condition on the history of prices, as in Calvano et al.). It would be valuable to consider how this conclusion applies more broadly to different competitive environments and whether such a restriction does not involve excessive efficiency costs. Additionally, autonomous collusion would be avoided by prohibiting firms from taking into account competitor prices. However, this may be unique to the dynamic optimization environment considered. As discussed in the literature review, opposite results are found in the case of static optimization algorithms, where ignoring competitors may lead to an underestimation of the own price elasticity of demand or inadvertent correlated pricing. In addition to regulating the input to the algorithm, various market design features may also prevent autonomous collusion, without impeding efficiency benefits. These could, for instance, relate to demand-steering policies (Johnson, Rhodes, and Wildenbeest, 2020), or involve forcing a disaggregation of decision-makers or introducing an additional algorithm that aims to maximize social or consumer welfare (Abada and Lambin, 2020). It would be valuable to explore such options further.

Finally, we may need to rethink the basis of our antitrust laws when it comes to algorithms and collusion. As discussed by Harrington (2018) and Calvano et al. (2020), collusion between humans on higher prices involves a three-step process: (1) communication between competitors on the collusive intent and conduct, (2) the mutual adoption of the collusive conduct, and (3) the higher prices as a consequence of the collusive conduct. In prosecuting cartels, antitrust laws have focused on the first stage (communication between competitors). This is because the second stage (the collusive conduct) is generally latent (i.e., occurring only in the heads of the managers) and the third stage (higher prices) is difficult to ascribe definitively to collusive conduct (as opposed to other, innocuous explanations such as changes in demand, cost or other market conditions). This antitrust practice of focusing on communication may be problematic in the case of the autonomous algorithmic collusion shown in this article and in Calvano, Calzolari, Denicolò, and Pastorello (2020), as communication is absent. Collusion is tacit, but whereas authorities generally cannot observe the second stage (the underlying collusive conduct) in the case of human collusion, pricing algorithms can be audited and tested to see whether they employ the kind of strategies that support a collusive equilibrium outcome. The forced deviation as shown in Section 4 is an example of such an approach.

There is actually a more general principle here, which is that algorithms require a far greater level of specificity than human decision-making and this specificity can be probed. This principle provides novel possibilities in detecting and prosecuting unwanted behavior. This goes beyond competition concerns, applying also to concerns around algorithmic bias and discrimination (Kleinberg et al., 2020), for instance. The big challenge, however, will be to translate the principle of probing algorithmic decision making to practical policy.

References

Abada, I. and Lambin, X. "Artificial Intelligence: Can Seemingly Collusive Outcomes be Avoided?" Working Paper, SSRN 3559308, 2020.

Assad, C., Clark, R., Ershov, D., and Xu, L. "Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market." CESifo Working Paper No. 8521, 2020.

Autoridade de Concurrência. "Digital Ecosystems, Big Data and Algorithms." Issues Paper, 2019.

Autorité de la Concurrence and Bundeskartellamt. "Algorithms and Competition." Working Paper, 2019.

Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. "Evolutionary Dynamics of Multi-Agent Learning: A Survey." *Journal of Artificial Intelligence Research*, Vol. 53 (2015), pp. 659–697.

Brown, Z. and MacKay, A. "Competition in Pricing Algorithms." Harvard Business School Working Paper No. 20-067, 2020.

Busoniu, L., Babuŝka, R., and De Schutter, B. "A Comprehensive Survey of Multiagent Reinforcement Learning." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 38 (2008), pp. 156–172.

Byrne, D.P. and de Roos, N. "Learning to Collude: A Study in Retail Gasoline." *American Economic Review*, Vol. 109 (2019), pp. 591–619.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. "Artificial Intelligence, Algorithmic Pricing and Collusion." *American Economic Review*, Vol. 110 (2020), pp. 3267–3297.

Calvano, E., Calzolari, G., Denicolò, V., Harrington, J., and Pastorello, S. "Protecting Consumers from Collusive Prices due to AI." *Science*, Vol. 370 (2020), pp. 1040–1042.

Chen, L., Mislove, A., and Wilson, C. "An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace." In *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 1339–1349.

Competition and Markets Authority. "Pricing Algorithms. Economic Working Paper on the Use of Algorithms to Facilitate Collusion and Personalised Pricing." Working Paper, 2018.

Competition and Markets Authority. "Algorithms: How They Can Reduce Competition and Harm Consumers." Research and Analysis Paper, 2021.

Cooper, W.L., Homem-de-Mello, T., and Kleywegt, A.J. "Learning and Pricing with Models that do not Explicitly Incorporate Competition." *Operations Research*, Vol. 63 (2015), pp. 86–103.

Crandall, J.W., Oudah, M., Ishowo-Oloko, F., Abdallah, S., Bonnefon, J.F., Cebrian, M., Shariff, A., Goodrich, M.A., and Rahwan, I. "Cooperating with Machines." *Nature Communications*, Vol. 9 (2018), p. 233.

Delrahim, M. "Remarks at the Federal Telecommunications Institute's Conference in Mexico City." November 7, 2018.

Den Boer, A.V. "Dynamic Pricing and Learning: Historical Origins, Current Research, and New Directions." *Surveys in Operations Research and Management Science*, Vol. 20 (2015), pp. 1–18.

Dogan, I. and Güner, A.R. "A Reinforcement Learning Approach to Competitive Ordering and Pricing Problem." *Expert Systems*, Vol. 32 (2013), pp. 39–48.

Eckert, A. "Empirical Studies of Gasoline Retailing: A Guide to the Literature." *Journal of Economic Surveys*, Vol. 27 (2013), pp. 140–166.

Ezrachi, A. and Stucke, M.E. *Virtual Competition: The Promise and Perils of the Algorithm-Driven Economy*. Harvard University Press, Cambridge, Massachusetts, 2016.

Ezrachi, A. and Stucke, M.E. "Artificial Intelligence & Collusion: When Computers Inhibit Competition." *University of Illinois Law Review*, Vol. 2017 (2017), pp. 1775–1810.

European Commission. 2020. Antitrust: Commission consults stakeholders on a possible new competition tool, press release, 2 June 2020.

Financial Times. "Policing the Digital Cartels." January 8, 2017.

Frankfurther Allgemeine Zeitung. "Kartellbildung Durch Lernende Algorithmen?" July 13, 2018.

FTC. "The Competition and Consumer Protection Issues of Algorithm, Artificial Intelligence, and Predictive Analytics." Hearing on Competition and Consumer Protection in the 21st Century, November 13-14, 2018.

Green, E.J. and Porter, R.H. "Noncooperative Collusion Under Imperfect Price Information." *Econometrica*, Vol. 52 (1984), pp. 87–100.

Hansen, K., Misra, K., and Pai, M. "Algorithmic Collusion: Supra-Competitive Prices via Independent Algorithms." CEPR Discussion Paper No. DP14372, 2020.

Harrington, J.E. "Developing Competition Law for Collusion by Autonomous Price-Setting Agents." *Journal of Competition Law and Economics*, Vol. 14 (2018), pp. 331–363.

Harrington, J.E. "Third Party Pricing Algorithms and the Intensity of Competition." Unpublished Working Paper, 2020.

Harvard Business Review. "How Pricing Bots Could Form Cartels and Make Things More Expensive." October 27, 2016.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., and Munoz de Cote, E. "A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity." Working Paper, arXiv 1707.09183, 2017.

Hu, J. and Wellman, M.P. "Nash Q-Learning for General-Sum Stochastic Games." *Journal of Machine Learning Research*, Vol. 4 (2003), pp. 1039–1069.

Huck, S., Normann, H.T., and Oechssler, J. "Zero-Knowledge Cooperation in Dilemma Games." *Journal of Theoretical Biology*, Vol. 220 (2003), pp. 47–54.

Huck, S., Normann, H.T., and Oechssler, J. "Two are Few and Four are Many: Number Effects in Experimental Oligopolies." *Journal of Economic Behavior & Organization*, Vol. 53 (2004), pp. 435–446.

Izquierdo, S.S. and Izquierdo, L.R. "The "Win-Continue, Lose-Reverse." In *Lecture Notes in Economics and Mathematical Systems*, Vol. 676. Cham: Springer, 2015.

Johnson, J., Rhodes, A., and Wildenbeest, M.R. "Platform Design When Sellers Use Pricing Algorithms." Working Paper, SSRN 3691621, 2020.

Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A.A.A., Yogamani, S., and Pérez, P. "Deep reinforcement learning for autonomous driving: A survey." Working Paper, arXiv 2002.00444, 2020.

Kleinberg, J., Ludwig, J., Mullainathan, S., and Sunstein, C.R. "Algorithms as Discrimination Detectors." *Proceedings of the National Academy of Sciences*, 117 (2020), pp. 30096–30100.

Kohs, G. *AlphaGo*. Netflix Documentary, 2017.

Kühn, K.U. and Tadelis, S. "Algorithmic Collusion." Presentation Prepared for CRESSE 2017, 2017.

Leufkens, K. and Peeters, R. "Price Dynamics and Collusion Under Short-Run Price Commitments." *International Journal of Industrial Organization*, Vol. 29 (2011), pp. 134–153.

Maskin, E. and Tirole, J. "A Theory of Dynamic Oligopoly II: Price Competition, Kinked Demand Curves and Edgeworth Cycles." *Econometrica*, Vol. 56 (1988), pp. 571–599.

Mehra, S. "Antitrust and the Robo-Seller: Competition in the Time of Algorithms." *Minnesota Law Review*, Vol. 100 (2016), pp. 1323–1375.

Meylahn, J.M. and Den Boer, A. "Learning to Collude in a Pricing Duopoly." Working Paper, SSRN 3741385, 2021.

Miklós-Thal, J. and Tucker, C. "Collusion by Algorithm: Does Better Demand Prediction Facilitate Coordination Between Sellers?" *Management Science*, Vol. 65 (2019), pp. 1552–1561.

Milgrom, P. and Roberts, J. "Rationalizability, Learning, and Equilibrium in Games with Strategic Complementarities." *Econometrica*, Vol. 58 (1990), pp. 1255–1277.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. "Human-Level Control Through Deep Reinforcement Learning." *Nature*, Vol. 518 (2015), pp. 529–533.

Noel, M.D. "Edgeworth Price Cycles and Focal Prices: Computational Dynamic Markov Equilibria." *Journal of Economics & Management Strategy*, Vol. 17 (2008), pp. 345–377.

O'Connor, J. and Wilson, N.E. "Reduced Demand Uncertainty and the Sustainability of Collusion: How AI Could Affect Competition." *Information Economics and Policy*, Vol. 54 (2021), p. 100882.

OECD. "Algorithms and Collusion: Competition Policy in the Digital Age." Report, November, 2017.

Ohlhausen, M.K. "Should We Fear the Things That Go Beep in the Night? Some Initial Thoughts on the Intersection of Antitrust Law and Algorithmic Pricing." Remarks for the Concurrences Antitrust in the Financial Sector Conference, May 23, 2017.

Politico. "When Margrethe Vestager Takes Antitrust Battle to Robots." February 28, 2018.

Powers, R.A. "Remarks at Cartel Working Group Plenary: Big Data and Cartelization, 2020 International Competition Network Annual Conference." September 17, 2020.

Rotemberg, J.J. and Saloner, G. "A Supergame-Theoretic Model of Price Wars During Booms." *American Economic Review*, Vol. 76 (1986), pp. 390–407.

Salcedo, B. "Pricing Algorithms and Tacit Collusion." PhD Manuscript, Pennsylvania State University, 2015.

Schwalbe, U. "Algorithms, Machine Learning, and Collusion." *Journal of Competition Law and Economics*, Vol. 14 (2018), pp. 568–607.

Shoham, Y., Powers, R., and Grenager, T. "If Multi-Agent Learning is the Answer, What is the Question?" *Artificial Intelligence*, Vol. 171 (2007), pp. 365–377.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. "Mastering the Game of Go with Deep Neural Networks and Tree Search." *Nature*, Vol. 529 (2016), pp. 484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. "Mastering the Game of Go Without Human Knowledge." *Nature*, Vol. 550 (2017), pp. 354–359.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go Through Self-Play." *Science*, Vol. 362 (2018), pp. 1140–1144.

Sutton, R.S. and Barto, A.G. *Reinforcement Learning: An Introduction*, 2d ed. Cambridge, MA: The MIT Press, 2018.

Tesauro, G. "Extending Q-Learning to General Adaptive Multi-Agent Systems." In *Advances in Neural Information Processing Systems*, 871–878. Cambridge, MA: MIT Press, 2003.

Tesauro, G. and Kephart, J.O. "Pricing in Agent Economics Using Multi-Agent Q-Learning." *Autonomous Agents and Multi-Agent Systems*, Vol. 5 (2002), pp. 289–304.

The Economist. "Price-Bots Can Collude Against Consumers." May 6, 2017.

The New Yorker. "When Bots Collude." April 25, 2015.

The Wall Street Journal. (2017) "Why Do Gas Station Prices Constantly Change? Blame the Algorithm." May 8, 2017.

Tuyls, K. and Weiss, G. "Multiagent Learning: Basics, Challenges, and Prospects." *AI Magazine*, Vol. 33 (2012), pp. 41–52.

Vestager, M. "Algorithms and Competition." Speech at the Bundeskartellamt 18th Conference on Competition, March 16, 2017.

Waltman, L. and Kaymak, U. "Q-Learning Agents in a Cournot Oligopoly Model." *Journal of Economic Dynamics & Control*, Vol. 32 (2008), pp. 3275–3293.

Wang, W., Hao, J., Wang, Y., and Taylor, M. "Towards Cooperation in Sequential Prisoner's Dilemmas: A Deep Multiagent Reinforcement Learning Approach", Working Paper, arXiv 1803.00162, 2018.

Watkins, C.J.C.H. "Learning from Delayed Rewards." PhD Manuscript, University of Cambridge, 1989.

Watkins, C.J.C.H. and Dayan, P. "Q-Learning." *Machine Learning*, Vol. 8 (1992), pp. 279–292.

Xie, M. and Chen, J. "Studies on Horizontal Competition Among Homogeneous Retailers Through Agent-Based Simulation." *Journal of Systems Science and Systems Engineering*, Vol. 13 (2004), pp. 490–505.