TI 2018-056/VII
Tinbergen Institute Discussion Paper

Autonomous Algorithmic Collusion: Q-Learning Under Sequential Pricing

Revision: November 2020

Timo Klein[1]

[1] University of Amsterdam

Autonomous Algorithmic Collusion: Q-Learning Under Sequential Pricing[*]

Timo Klein[†]

November 2020

Abstract

Prices are increasingly set by algorithms. One concern is that intelligent algorithms may learn to collude on higher prices even in absence of the kind of communication or agreement necessary to establish an antitrust infringement. However, exactly how this may happen is an open question. I show in a simulated environment of sequential competition that competing reinforcement learning algorithms can indeed learn to converge to collusive equilibria. When the set of discrete prices 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.

JEL Codes: D43, D83, L13, L41
Keywords: Algorithmic Collusion, Pricing Algorithms, Machine Learning, Reinforcement Learning, Q-Learning

[*] I am grateful for valuable discussions with and comments from Arnoud den Boer, Emilio Calvano, Ariel Ezrachi, Joe Harrington, Harold Houba, Dávid Kopányi, Kai-Uwe Kühn, Xavier Lambin, Leonardo Madio, Alexander Rasch, Magda Rola-Janicka, Maarten Pieter Schinkel, Ulrich Schwalbe and Leonard Treuren. 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 and seminars at the UK's Competition & 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.

[†] Utrecht University School of Economics and Oxera Consulting LLP, email t.klein@uu.nl. Any views expressed in this article are my own.

"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 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, Clark, Ershov and Xu, 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]

[1] Press coverage includes for instance Financial Times (2017), Frankfurter Allgemeine Zeitung (2018), Harvard Business Review (2016), Politico (2018), The Economist (2017), The New Yorker (2015) and 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 Concurrência, 2019; Autorité de la Concurrence and Bundeskartellamt, 2019; Competition & Markets Authority, 2018; OECD, 2017).

[2] This includes in particular the self-learning of superhuman play in complex board games like

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 to observe pricing algorithms that are completely and fully based on Q-learning, because 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 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.

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

A recent article by Calvano, Calzolari, Denicolò and Pastorello (2020b) (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. Unlike Calvano et al., I also do not require the algorithm to condition its prices on payoff-irrelevant past prices in order to collude.

My main finding is that when the number of discrete prices that the algorithm can choose from is limited, competing Q-learning algorithms indeed often coordinate on collusive equilibria. I show that these equilibria are sustained through the kind of reward-punishment strategies that are defined by standard cartel theory. When the number of discrete prices increases, Q-learning increasingly converges to supra-competitive Edgeworth price cycles, in which periodic upward 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 (Noel, 2011; Eckert, 2013; Byrne and de Roos, 2019), which is 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 monitoring of competitor prices and quicker retaliation in case of defection. Second, if algorithms learn by themselves to adopt strategies that result in supra-competitive outcomes, this would not be illegal under current competition laws.[4]

However, the concerns around algorithmic collusion are not universally shared. Although algorithms may help to implement or stabilize a collusive agreement, Kühn and Tadelis (2017a, 2017b) and Schwalbe (2019) 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 extend on experimental economic evidence, that solving the coordination problem realistically requires some form of illegal communication. Moreover, it is often pointed out that all known cases of collusion involving algorithms also involved some degree of illegal human behavior.[5] Truly autonomous—and hence potentially legal—algorithmic collusion is yet to be observed. Although true, the absence of observed cases is no guarantee.

In this section, I review the literature on the broader question of how pricing algorithms undermine competition. Although there is some overlap, this literature

[4] Harrington (2018) and Gal (2019a) provide additional important legal contributions on this.

[5] One prominent cartel involving algorithms are the *Topkins* (US) and *GB Eye-Trod* (UK) cases in 2015 and 2016, where online poster retailers used algorithms to coordinate differentiated product prices. Another is the allegation that Accenture (a management consultancy) provided competing car manufacturers in the EU with an algorithm that allowed them to coordinate prices of spare parts (Reuters, 2018). Neither is a case of algorithms learning autonomously to collude, however. Another illustration is “The Making of a Fly”, a biology textbook sold on Amazon. In 2011, one seller used an algorithm that each day priced 25% above its competitor, while its competitor used a price-matching algorithm. This caused prices to escalate (up to 23 million dollar per copy). This is again no autonomous collusion. A review of recent case law around algorithmic collusion is provided for instance by O’Kane and Kokkoris (2020).

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 belongs to the last strand. Below, each strand is covered in turn and contrasted to this article, followed by a brief discussion on the existing empirical evidence and some of the frontier computer science literature.[6]

Static Optimization In operations research and management science, pricing algorithms are often used to estimate and optimize static, one-period profit functions—see Den Boer (2015) for a survey. In theory, such static optimization cannot lead to stable collusion, as it does not allow for the kind of reward-punishment strategies that are necessary to solve the prisoner's dilemma dynamics (Milgrom and Roberts, 1990). However, the practice in operations research is often to simply estimate monopoly models and ignore competitors and strategic considerations. Cooper, Homem-de-Mello and Kleywegt (2015) show that this practice may lead to an underestimation of the price elasticity of demand, with the inadvertent effect that the algorithms learn to price cooperatively rather than competitively.

Hansen, Misra and Pai (2020) and Huck, Normann and Oechssler (2003, 2004) come to the same conclusion in two very different simulation settings. Their results are driven by an inadvertent correlation in experimentation. Hansen, Misra and Pai model firms as experimenting with a set of discrete prices using the so-called Upper Confidence Bound (UCB) algorithm (which is a basic reinforcement learning algorithm that explores those actions that have the highest potential of having an

[6] Relevant articles written on the topic that are more policy-oriented (and not discussed in further detail here) include Ballard and Naik (2017), Capobianco and Gonzaga (2017), Calvano, Calzolari, Denicolò and Pastorello (2020a), Deng (2018), Ezrachi and Stucke (2020), Gal (2017, 2019b), Johnson, Rhodes and Wildenbeest (2020a), Klein (2020), Klein, van der Noll and Sviták (2020), McSweeny and O'Dea (2017), Moore, Pfister and Piffaut (2020), Okuliar and Kamenir (2017), Oxera (2018, 2020) and Sviták and van der Noll (2019), as well as reports by different competition authorities—including in particular Competition & Markets Authority (2018), Autoridade de Concurrência (2019) and Autorité de la Concurrence and Bundeskartellamt (2019). Calvano, Calzolari, Denicolò, Harrington and Pastorello (2020) provide a valuable recent addition to this policy-oriented literature and is discussed in more detail in the concluding section of this article.

optimal value). They show that when there is not too much noise in the demand signal, UCB causes competitors to perceive the same options as potentially optimal and inadvertently start to correlate their experimentation. As they start setting equivalent prices, they fail to learn to profitably undercut and end up with supra-competitive prices. A similar mechanism of correlated experimentation is found by Huck, Normann and Oechssler (2003, 2004) in a simple "win-continue, lose-reverse" rule in simulated quantity competition. However, Izquierdo and Izquierdo (2015) show that the correlated experimentation and joint-profit maximizing convergence in the case of Huck, Normann and Oechssler breaks down when there are minor changes in the payoff function. A similar result may apply to Hansen, Misra and Pai.

My article differs from this strand of the literature in two ways. First, the algorithm that I consider does not ignore competitor prices and hence forecloses the mechanism of inadvertent price correlation or underestimation of price elasticity. Second, this article looks at dynamic, multi-period optimization, which is a theoretically necessary condition for a collusive equilibrium to arise autonomously in the presence of prisoner's dilemma dynamics. I also check whether the supra-competitive pricing is even an equilibrium.

Algorithmic Commitment Related to the previous strand of the literature, there are two articles in particular that show how 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 pricing algorithms lead to higher prices in a model where firms use algorithms to implement a contingent pricing strategy and prices are updated at different frequencies. They show that competing firms will endogenously self-select into asymmetric pricing frequencies, which effectively results in a mutually profitable leader-follower relationship that pushes up equilibrium prices. They 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.

The main contribution of these articles is that they argue how the reasonably realistic delegation of pricing strategies to algorithms can change the pricing game in such a way that the most competitive equilibrium is no longer the Bertrand-Nash equilibrium where prices are equal to marginal cost. Interestingly, this does not require an extension to dynamic, multi-period optimization function. Their relevance notwithstanding, this does mean that 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. Additionally, Salcedo assumes that the algorithms can periodically 'decode' each other's contingent pricing strategies—which may be interpreted as illegal communication (Schwalbe, 2019). Moreover, Brown and MacKay have complete information by assumption. In my article, I neither assume a pre-set commitment to a pricing strategy or knowledge on the strategies used by the competition.

Demand Prediction 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 (2020) show that there are theoretically ambiguous effects on cartel stability and welfare in the presence of better demand forecasting. Miklós-Thal and Tucker rely on the framework of Rotemberg and Saloner (1986), in which future demand is stochastic (and collusion is continuous but the cartel may need to price below joint-profit maximization to ensure incentive compatibility). O'Connor and Wilson instead rely on the framework of Green and Porter (1984), in which there are stochastic demand shocks and competitor prices are not observed. Both articles show that better demand prediction increases the expected payoff under both collusion and deviation, which may actually (though not necessarily) decrease cartel stability and increase welfare in equilibrium.

Better demand prediction algorithms may also reside with third-party pricing software providers that may supply their services to competing firms. Harrington (2020) show theoretically how such situations can lead to price increases, as a third-party pricing algorithm takes into account that it may face itself. Interestingly, higher prices already occur when only a single firm ends up adopting the third-party pricing algorithm. The model does exogenously assume an absence of competition between different third-party pricing algorithms. Moreover, all these articles treat algorithms as exogenous and black box improvements of demand predictability. I instead explicitly models algorithms as autonomous price setting agents.

Dynamic Optimization The fourth strand of the literature focuses on autonomous algorithms designed to optimize some long-run or multi-period objective function. In theory, multi-period optimization is required to learn the reward-punishment strategies necessary to stabilize a collusive equilibrium. Looking at some form of sequential price competition with no-standard demand function, Tesauro and Kephart (2002) show though simulations how dynamic programming techniques can converge to profitable asymmetric price cycles—with cycles becoming shorter and profits increasing if products are more differentiated or consumers less informed. Noel (2008) similarly uses dynamic programming in simulated competition to identify and analyze likely Markov perfect equilibria in various extensions to the more accepted sequential-pricing framework of Maskin and Tirole (1988). However, both articles assume full knowledge of the environment and only provide an equilibrium identification analysis, ignoring the whole coordination problem.

In a more realistic informational environment, Xie and Chen (2004) show through simulations that when competing algorithms simultaneously set their inventory and prices in an environment of stochastic demand, a Q-learning algorithm that simply ignores competitors converges to a stable Nash equilibrium. Dogan and Güner (2013) build on this by extending the state space to include current and previous prices and inventories of itself and its competitors and find positive profits in case of competing Q-learning algorithms. Finally, Waltman and Kaymak (2008) are the first to show that Q-learning leads to supra-competitive outcomes in a conventional Cournot oligopoly environment that is infinitely repeated. However, while the last two papers find supra-competitive outcomes, they do not test for equilibrium behavior or collusive reward-punishment strategies. In fact, Waltman and Kaymak also find supra-competitive outcomes under conditions in which reward-punishment strategies are not theoretically possible (no memory), which suggests that supra-competitive outcomes may be spurious rather than collusive.

My article is most similar to that of Calvano, Calzolari, Denicolò and Pastorello (2020b) and Abada and Lambin (2020). Calvano et al. 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, they require conditioning on own and competitor past prices for collusion to occur, which 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 competitive environment that is instead motivated by energy markets. Instead of competing only for a per-period market demand, competitors also face a dynamic arbitrage problem where they sell or buy an inventory at a prevailing market price (taking into account capacity constraints on inventory and the amount that can be bought or sold in any one period). The authors find that in this environment, competing Q-learning algorithms again learn strategies that sustain supra-competitive prices. They also discuss how regulators can (partially) frustrate the collusive learning processes by disaggregating the amount of agents that are optimizing their dynamic capacity or by introducing an agent that aims to maximize social or consumer welfare. Similarly as Calvano et al., Abada and Lambin take an environment of simultaneous move—which I depart from.

Finally, a interesting recent article by Johnson, Rhodes and Wildenbeest (2020b) shows theoretically how market designers can induce sellers towards more competitive behavior by steering demand in an environment of dynamic optimization. They apply this reasoning to the context of e-commerce pricing algorithms. Using simulations with competing Q-learning algorithms, they show how competition on an e-commerce platform can be improved by basically providing longer prominence to Q-learning sellers that display behavior consistent with deviation from a collusive agreement. Interestingly, the authors show that in their environment, such demand-steering policies raise both platform profit and consumer surplus and is therefore a particularly interesting market design mechanism to consider.

Empirical Literature In each of the above four strands of the literature, the articles generally rely on theoretical models or computer simulations to investigate the effect of pricing algorithms on competition (with the exception of Brown and MacKay (2020), as discussed). Empirical evidence based on real-world data remains limited and difficult to obtain. A 2017 e-commerce sector inquiry by the European Commission does provide survey data that shows that a majority of retailers track online prices of competitors and two-thirds of them automatically adjust prices in response, but this inquiry does not look at any relation to for instance markups. Chen, Mislove and Wilson (2016) show that of 1,600 best-selling products on Amazon, in 2015 more than one-third adopted algorithmic pricing strategies and that these did tend to have higher price (and sales volumes), but this does not show any chain of causation.

However, there is one important recent empirical contribution that does look at the causal effect of pricing algorithms. Assad, Clark, Ershov and Xu (2020) are able to proxy which and when retail gasoline stations in German adopted algorithmic pricing technology, between 2016 and 2019, based on observed changes in pricing behavior. Correcting for the endogeneity of station-level adoption, they show that margins increase in non-monopoly markets. This is in particular the case when two competing stations adopted algorithms, in which case margins increase on average by 28%. Interestingly, they find that these higher margins occur gradually. This appears consistent with some form of learning to coordinate on higher prices—as in Calvano et al. (2020b) and my article. The authors do treat the algorithms as black boxes. In my article, I look at the inner workings of reinforcement learning applied to a controlled pricing environment.

Frontier Computer Science The above literature mostly comes from economics. There are also several articles on the academic frontier in computer science that look more generally at cooperation between reinforcement learning algorithms. From the theory perspective, there is a strand of the computer science literature that looks at so-called multi-agent reinforcement learning algorithms, which combines (evolutionary) game theory with reinforcement learning—often building on Q-learning as a baseline specification.

For instance, Hu and Wellman (2003) propose Nash-Q learning, which maintains Q-functions over joint actions and performs updates based on assuming Nash equilibrium behavior given current Q-values. This work is extended further by Greenwald and Hall (2003) by looking for correlated equilibria and Könönen (2003) by looking at leader-follower stage games. Tesauro (2003) proposes Hyper-Q learning, which learns Q-values associated with mixed strategies and uses estimated opponent strategies as additional state variables. And Singh, Kearns and Mansour (2000) allow for learning optimal mixing strategies in multi-agent games by using an infinitesimal gradient ascent (IGA) algorithm—later extended or generalized by in particular Zinkevich

(2003), Bowling and Veloso (2002), Abdallah and Lesser (2008) and Awheda and Schwartz (2013).

However, strong theoretical results from the multi-agent reinforcement learning literature have been relatively limited and key results that are there (for instance on convergence guarantees to Nash equilibria under reasonable informational assumptions) have not been shown beyond even simple simulated matrix games. Moreover, the theoretical results often have high informational requirements (such as observing opponent rewards). It therefore seems unlikely that answers on autonomous algorithmic collusion are readily found on the current frontier of theoretical multi-agent reinforcement learning.[7]

However, most of the recent developments in reinforcement learning do not come from theoretical computer science. Instead, recent breakthroughs in reinforcement learning (and artificial intelligence more generally) come from empirical or simulation-based investigations. 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 these high-profile breakthroughs are still generally in the context of single-player or zero-sum games. One article that does look at multi-agent environments that are not zero-sum is Crandall et al. (2018), which shows how state-of-the-art reinforcement learning algorithms are capable of cooperating both with other algorithms and with humans in very simple repeated matrix games. However, it is unclear how results hold up in a oligopoly setting or pricing application. Additionally, Crandall et al. allow for certain signalling mechanisms—which in the context of competition law may be seen as illegal

[7] For a general introduction to the theory of multi-agent reinforcement learning see Shoham, Powers and Grenager (2007) and Tuyls and Weiss (2012) and for an overview of the literature see in particular Busoniu, Babuŝka and De Schutter (2008), Hernandez-Leal, Kaisers, Baarslag and Munoz de Cote (2017) and Albrecht and Stone (2018). A textbook-style treatment is provided by Schwartz (2014) and for a survey on the literature linking multi-agent reinforcement learning with evolutionary game theory, see Bloembergen, Tuyls, Hennes and Kaisers (2015).

communication. Other articles discussing frontier reinforcement learning algorithms in cooperative matrix games include Leibo, Zambaldi, Lanctot, Marecki and Graepel (2017), Lerer and Peysakhovich (2018), Romero and Rosakha (2019) and Wang, Hao, Wang and Taylor (2018). 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 algorithms 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 for instance 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 motivated in the introduction, 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 $t \in \{0, 1, 2, ...\}$. Adjustments in price occur sequentially: in turn, each firm adjusts its price $p_{it} \in P$, where in odd-numbered periods firm 1 adjusts its price and in even-numbered periods firm 2. Price is a discrete variable scaled between 0 and 1 and with k equally sized intervals—so prices are taken from a discrete set

$P = \{0, \frac{1}{k}, \frac{2}{k}, ..., 1\}$. 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 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 exposition 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.[8] 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 a 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 high 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 itself.

[8] 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.

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. For a textbook treatment on Q-learning, see Sutton and Barto (2018). Calvano et al. (2020b) and Johnson, Rhodes and Wildenbeest (2020b) 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 $s_t \in S$. Assuming a discrete state set, Q_i is a $|P| \times |S|$ matrix in this case. After observing 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 stepsize 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 enable 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 state s_t and s_{t+1} are equivalent to current and new competitor price 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 modelling 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

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

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 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 $t \leftarrow t + 1$ and $\{i \leftarrow j, j \leftarrow i\}$
**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 stepsize parameter α and the rate of exploration ε_t (Watkins and Dayan, 1992; Tsitsiklis, 1994). 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, Babuŝka 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 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 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

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

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 compare profitability against two benchmarks: the joint-profit maximizing benchmark of 0.125 (which 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 Γ_i 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

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

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.

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 learned a best-response strategy it therefore 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, stepsize 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 towards zero towards 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).

The key characteristic of a collusive equilibrium is the use of a "reward–punishment scheme which rewards a firm for abiding by the supracompetitive outcome and punishes it for departing from it" (Harrington, 2018). Similar as in Calvano et al. (2020b), I test for the existence reward-punishment strategies 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 1,000 runs. In the baseline simulation, I set stepsize 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 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 percent and the share of Nash equilibria around 67 percent. The left-hand panel in Figure 2 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 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.

[FIGURES 1 AND 2 AROUND HERE]

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 while punished if it deviates. Following the approach in Calvano et al. (2020b) 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, this is actually the consequence of the different runs jumping up in price in different 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).

[FIGURE 3 AROUND HERE]

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 stepsize 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 to learn 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, while 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.

[FIGURES 4 AND 5 AROUND HERE]

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 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 1,000 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.

[FIGURES 6 AND 7 AROUND HERE]

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 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 Stepsize Parameters

In the baseline simulation, I have set stepsize 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 stepsize parameters that are closer to zero or close to one. Although not shown here, similar results apply when keeping the stepsize parameter of one of the two firms fixed (i.e. allow for asymmetric stepsize parameters).

[FIGURE 8 AROUND HERE]

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 while 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 stepsize parameters α.

[FIGURE 9 AROUND HERE]

Self-Reactive Conditioning

Similarly as Maskin and Tirole (1988) I have imposed the Markov assumption, under which the state variable is defined as current competitor price only. In other words, the algorithm is not allowed to condition its prices on any history of past prices that are not longer relevant for its current profit. In their setting of simultaneous competition, Calvano et al. (2020b) however 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. It also has more difficulty to converge 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.

[FIGURE 10 AROUND HERE]

6 Concuding 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 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 off 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 to observe pricing algorithms 'in the wild' that 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. Relatedly, this article only considers a very stylized competitive environment, for exposition purposes. Q-learning is likely to have increasingly more difficulty in finding optimal strategies under increasingly more complex environments.

There are different avenues with which to deal with these limitations in practice. For instance, Calvano et al. (2020b) 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, Hao, Wang and Taylor, 2018). In fact, this is how reinforcement learning algorithms are trained in for instance board games (Silver et al., 2018) and autonomous driving (Kiran et al., 2020). 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 instance modelling a demand function or competitor learning is a very obvious next step to improve the learning process (Romero and Rosakha, 2019; Schwartz, 2014; Sutton and Barto, 2018). These two avenues are left for future research.[9]

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 on 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).[10]

[9] Another valuable avenue for future research may lie with experimental economics, as also discussed by Schwalbe (2019). In our environment of a sequential pricing duopoly, Q-learning does not outperform the human performance benchmark as provided by Leufkens and Peeters (2011). For future autonomous pricing algorithms, it may be interesting to see whether they are capable of outperforming human subjects in controlled laboratory settings.

[10] Note that this focuses specifically on the concerns around algorithmic collusion. Other competition concerns in the use of pricing algorithms can for instance relate to the increased use of differentiated prices (Competition and Markets Authority, 2018), which may be seen as unfair, or

Second and relatedly, the possibility of autonomous algorithmic collusion raises interesting regulatory questions. Pricing algorithms can involve many pro-competitive effects and 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 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 when prohibiting firms from taking into account competitor prices. However, this may be unique to 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 can for instance relate to demand-steering policies (Johnson, Rhodes and Wildenbeest, 2020b), or 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, Calzolari, Denicolò, Harrington and Pastorello (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,

the increase of barriers to entry when the algorithms are driven by proprietary data (Autorité de la Concurrence and Bundeskartellamt, 2019; OECD, 2016). Additionally, any investigation needs to similarly look at the pro-competitive effects of algorithms—such as lower costs, better market clearing and lower entry barriers (Oxera, 2020).

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 head 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 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 just competition concerns, applying for instance also to concerns around algorithmic bias and discrimination (Kleinberg, Ludwig, Mullainathan and Sunstein, 2020). 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.

Abdallah, S. and Lesser, V. "A Multiagent Reinforcement Learning Algorithm with Non-Linear Dynamics." *Journal of Artificial Intelligence Research*, Vol. 33 (2008), 521-549.

Albrecht, S.V. and Stone, P. "Autonomous Agents Modelling Other Agents: A Comprehensive Survey and Open Problems." Working Paper, arXiv 1709.08071, 2018.

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.

Awheda, M. and Schwartz, H.M. "Exponential Moving Average Q-Learning Algorithm." *Proceedings of the IEEE Symposium Series on Computational Intelligence*, (2013).

Ballard, D.I. and Naik, A.S. "Algorithms, Artificial Intelligence, and Joint Conduct." *Antitrust Chronicle*, Vol. 1 (2017).

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), 659-697.

Bowling, M. and Veloso, M. "Multiagent Learning Using a Variable Learning Rate." *Artificial Intelligence*, Vol. 136 (2002), 215-250

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).

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

Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. "Algorithmic Collusion: A Real Problem for Competition Policy?" *Antitrust Chronicle*, Vol. 1 (2020a).

Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. "Artificial Intelligence, Algorithmic Pricing and Collusion." *American Economic Review*, Vol. 110 (2020b), 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), 1040-1042.

Capobianco, A. and Gonzaga, P. “Algorithms and Competition: Friends or Foes?” *Antitrust Chronicle*, Vol. 1 (2017).

Chen, L., Mislove, A. and Wilson, C. “An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace.” *Proceedings of the 25th International Conference on World Wide Web*, (2016), 1339-1349.

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

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), 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), 233.

Delrahim, M. “Remarks at the Federal Telecommunications Institute’s Conference in Mexico City.” 7 November, 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), 1-18.

Deng, A. “What Do We Know About Algorithmic Tacit Collusion?” *Antitrust*, Vol. 33 (2018), 88-95.

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

Eckert, A. “Empirical Studies of Gasoline Retailing: A Guide to the Literature.” *Journal of Economic Surveys*, Vol. 27 (2013), 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), 1775-1810.

Ezrachi, A. and Stucke, M.E. “Sustainable and Unchallenged Algorithmic Tacit Collusion.” *Northwestern Journal of Technology and Intellectual Property*, Vol. 17 (2020), 217-260.

Financial Times. “Policing the Digital Cartels.” 8 January, 2017.

Frankfurther Allgemeine Zeitung. “Kartellbildung Durch Lernende Algorithmen?” 13 July, 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, 13-14 November, 2018.

Gal, M.S. “Algorithmic-Facilitated Coordination: Market And Legal Solutions.” *Antitrust Chronicle*, Vol. 1 (2017).

Gal, M.S. “Algorithms as Illegal Agreements.” *Berkeley Technology Law Journal*, Vol. 34 (2019a), 67-118.

Gal, M.S. “Illegal Pricing Algorithms.” *Communications of the ACM*, Vol. 62 (2019b), 18-20.

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

Greenwald, A. and Hall, K. “Correlated Q-Learning.” *Proceedings of the 22nd Conference on Artificial Intelligence*, (2003), 242-249.

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), 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.” 27 October, 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), 1039-1069.

Huck, S., Normann, H.T. and Oechssler, J. “Zero-Knowledge Cooperation in Dilemma Games.” *Journal of Theoretical Biology*, Vol. 220 (2003), 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), 435-446.

Izquierdo, S.S. and Izquierdo, L.R. “The “Win-Continue, Lose-Reverse.” *Lecture*

Notes in Economics and Mathematical Systems, Vol. 676 (2015).

Johnson, J., Rhodes, A. and Wildenbeest, M.R. "Combating Anti-Competitive Behavior Involving Algorithms: Platform Design and Organizational Process." *Antitrust Chronicle*, Vol. 1 (2020a).

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

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.

Klein, T. "(Mis)understanding Algorithmic Collusion." *Antitrust Chronicle*, Vol. 1 (2020).

Klein, T., van der Noll, R. and Sviták, J. "Prijsalgoritmes, Machine Learning en Mededinging." *KVS Preadviezen 2020*, (2020).

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

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

Könönen, V. "Asymmetric Multiagent Reinforcement Learning." *Proceedings IEEE/WIC International Conference on Intelligent Agent Technology*, (2003), 336-342.

Kühn, K.U. and Tadelis, S. "The Economics of Algorithmic Pricing: Is Collusion Really Inevitable?" Unpublished Working Paper, 2017a.

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

Leibo, J.Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. "Multi-Agent Reinforcement Learning in Sequential Social Dilemmas." *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*, (2017).

Lerer, A. and Peysakhovich, A. "Maintaining Cooperation in Complex Social Dilemmas Using Deep Reinforcement Learning." Working Paper, arXiv 1707.01068, 2018.

Leufkens, K. and Peeters, R. "Price Dynamics and Collusion Under Short-Run Price Commitments." *International Journal of Industrial Organization*, Vol. 29 (2011), 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), 571-599.

McSweeny, T. and O'Dea, B. "The Implications of Algorithmic Pricing for Coordinated Effects Analysis and Price Discrimination Markets in Antitrust Enforcement." *Antitrust*, Vol. 32 (2017).

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

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

Milgrom, P. and Roberts, J. “Rationalizability, learning, and equilibrium in games with strategic complementarities.” *Econometrica*, Vol. 58 (1990), 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), 529-533.

Moore, J., Pfister, E. and Piffaut, H. “Some Reflections on Algorithms, Tacit Collusion, and the Regulatory Framework.” *Antitrust Chronicle*, Vol. 1 (2020).

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

Noel, M.D. “Edgeworth Price Cycles.” *The New Palgrave Dictionary of Economics*, (2011).

O’Connor, J. and Wilson, N.E. “Reduced Demand Uncertainty and the Sustainability of Collusion: How AI Could Affect Competition.” *Information Economics and Policy*, (2020).

OECD. “Big Data: Bringing Competition Policy to the Digital Era.” Report, November, 2017.

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, 23 May, 2017.

O’Kane, C.P. and Kokkoris, I. “A Few Reflections on the Recent Case Law on Algorithmic Collusion.” *Antitrust Chronicle*, Vol. 1 (2020).

Okuliar, A. and Kamenir, E. “Pricing Algorithms: Conscious Parallelism or Conscious Commitment?” *Antitrust Chronicle*, Vol. 1 (2017).

Oxera. “Algorithmic Competition.” Contribution to European Commission Panel on Shaping Competition Policy in the Era of Digitisation, September, 2018.

Oxera. “The Risks of Using Algorithms in Business: Artificial Price Collusion.”

Agenda, (2020).

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

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

Reuters. "Software and Stealth: How Carmakers Hike Spare Parts Prices." 3 June, 2018.

Romero, J. and Rosokha, Y. "A Model of Adaptive Reinforcement Learning." Working Paper, SSRN 3350711, 2019..

Rotemberg, J.J. and Saloner, G. "A Supergame-Theoretic Model of Price Wars During Booms." *American Economic Review*, Vol. 76 (1986), 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*, (2019), Online.

Schwartz, H.M. *Multi-Agent Machine Learning: A Reinforcement Approach.* Wiley, Hoboken, New Jersey, 2014.

Shoham, Y., Powers, R. and Grenager, T. "If Multi-Agent Learning is the Answer, What is the Question?" *Artificial Intelligence*, Vol. 171 (2007), 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), 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), 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), 1140-1144.

Singh, S., Kearns, M. and Mansour, Y. "Nash Convergence of Gradient Dynamics in General-Sum Games.", Uncertainty in Artificial Intelligence Proceedings, (2000), 541-548.

Sutton, R.S. and Barto, A.G. *Reinforcement Learning: An Introduction.* 2nd Edition, The MIT Press, Cambridge, Massachusetts, 2018.

Sviták, J. and van der Noll, R. “De Mechanismes van Algoritmische Collusie.” *Tijdschrift van Toezicht*, Vol. 1 (2019), 103-116.

Tesauro, G. “Extending Q-Learning to General Adaptive Multi-Agent Systems.” *Advances in Neural Information Processing Systems*, (2003), 871-878

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

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

The New Yorker. “When Bots Collude.” 25 April, 2015.

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

Tuyls, K. and Weiss, G. “Multiagent Learning: Basics, Challenges, and Prospects.” *AI Magazine*, Vol. 33 (2012), 41-52.

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

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

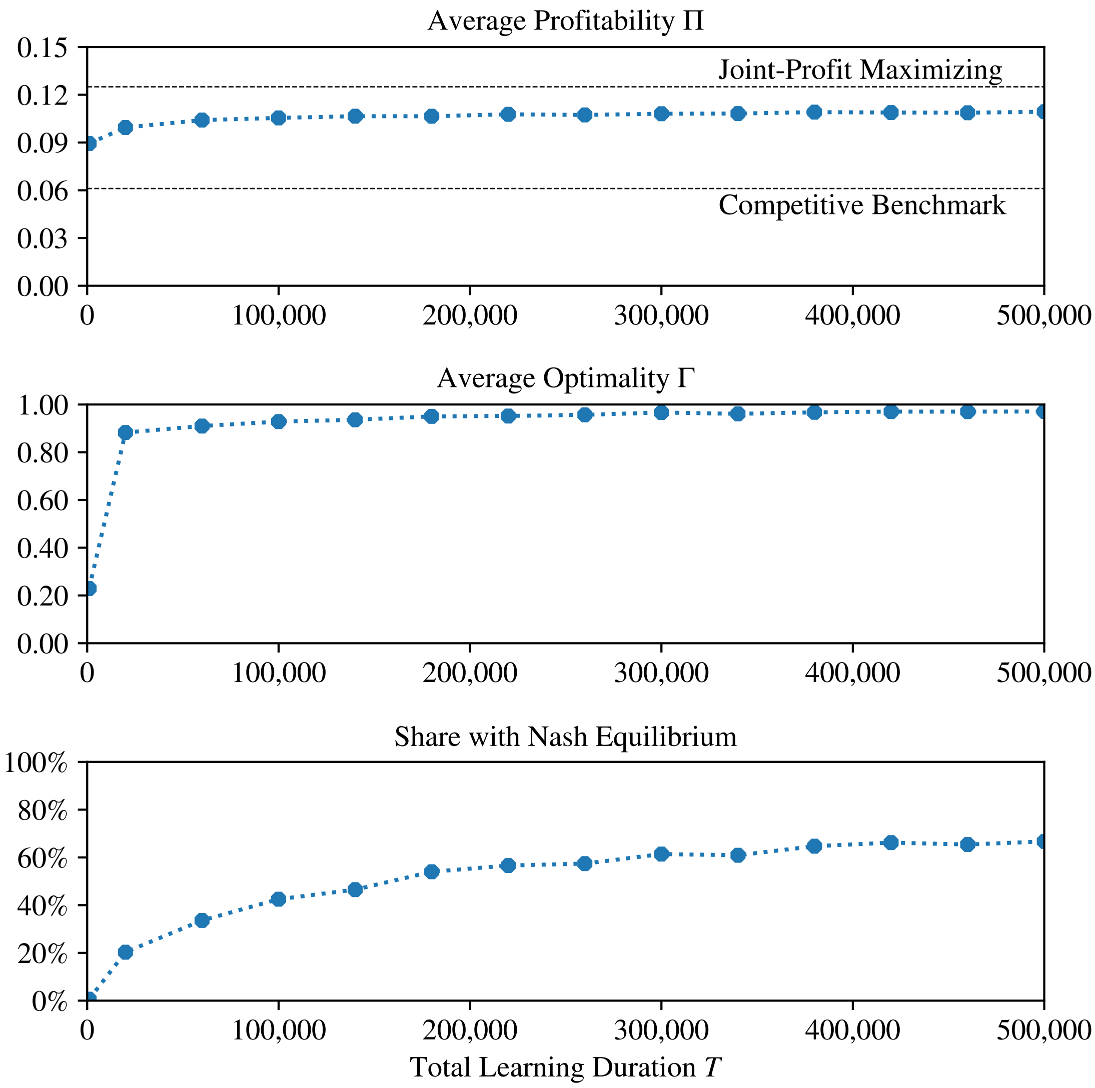
Watkins, C.J.C.H. “Learning from Delayed Rewards.” PhD Manuscript, University of Cambridge, 1989.

Watkins, C.J.C.H. and Dayan, P. “Q-Learning.” *Machine Learning*, Vol. 8 (1992), 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), 490-505.

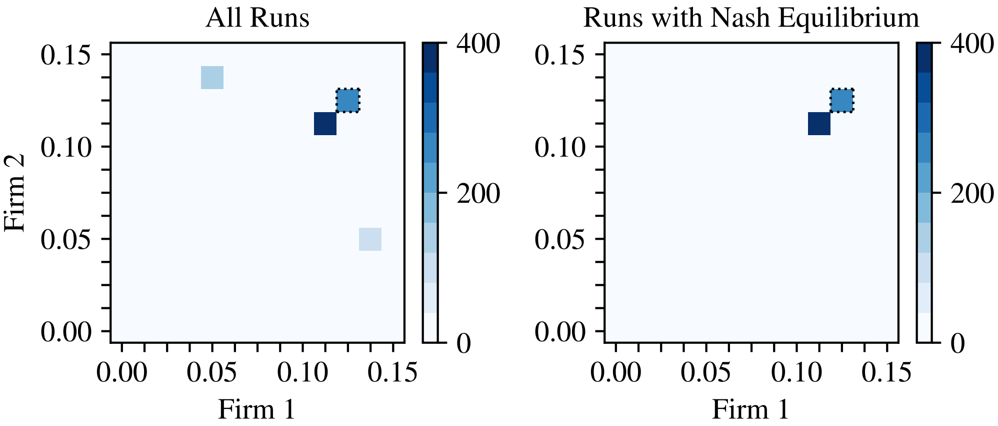
Zinkevich, M. “Online Convex Programming and Generalized Infinitesimal Gradient Ascent.” *Proceedings 20th International Conference on Machine Learning*, (2003), 928-936.

Figure 1: Baseline Performance Under Different Learning Durations T



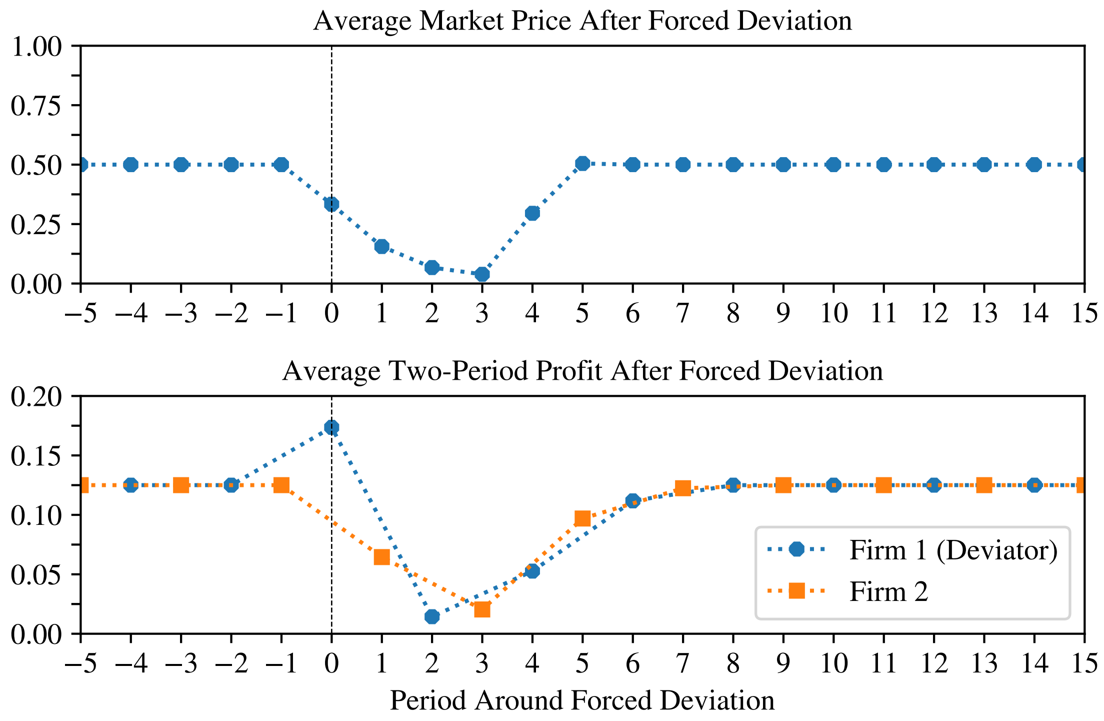
Notes: Results are in case of amount of price intervals $k = 6$, stepsize parameter $\alpha = 0.3$ and discount factor $\delta = 0.95$.

Figure 2: Baseline Joint Distribution of Profitability Π_i



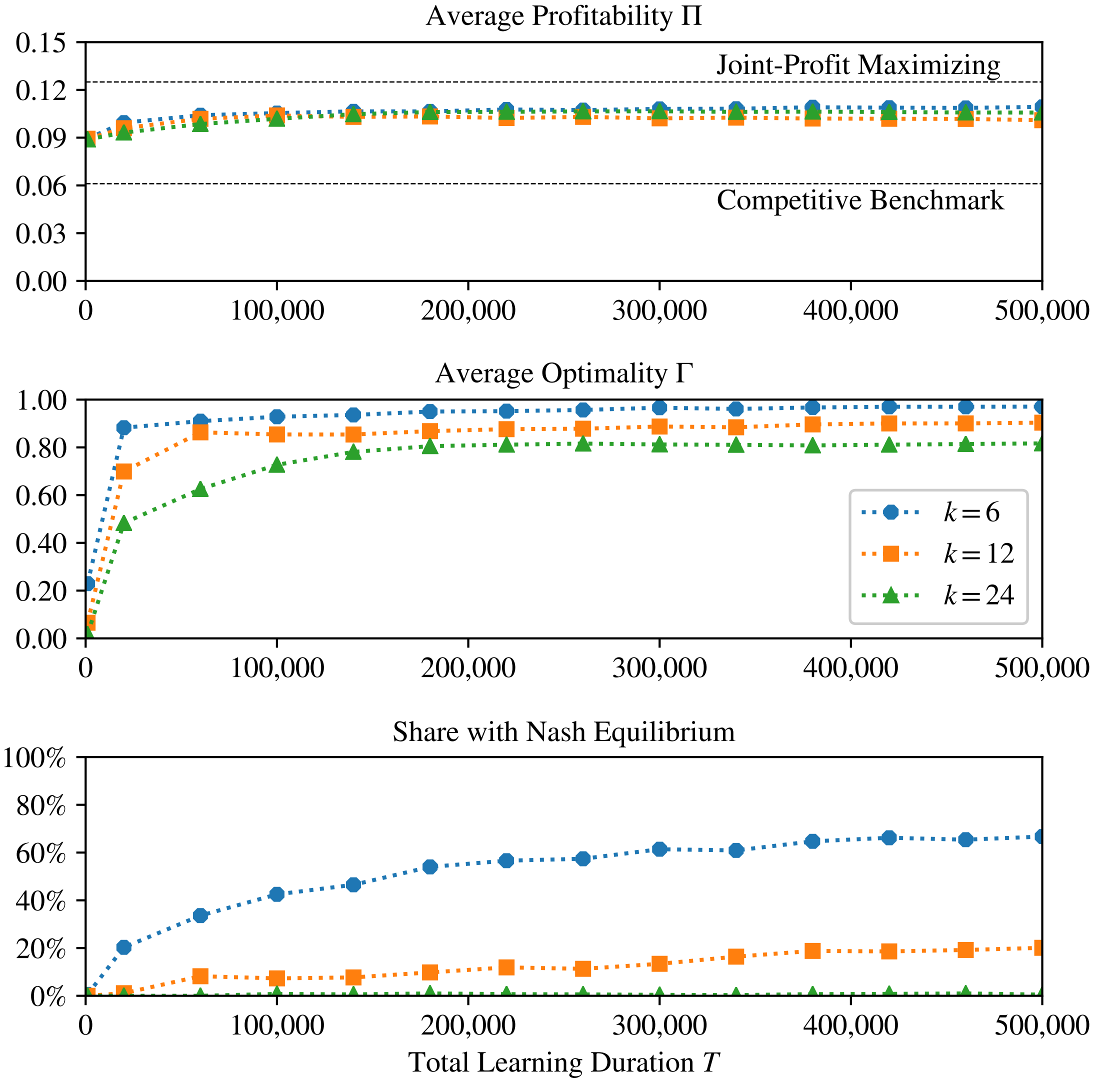
Notes: Right panel considers only those runs that led to a Nash equilibrium (667 out of 1,000 runs). Dotted squares indicate joint-profit maximizing profitability. Results are in case of amount of price intervals $k = 6$, learning duration $T = 500{,}000$, stepsize parameter $\alpha = 0.3$ and discount factor $\delta = 0.95$.

Figure 3: Average Market Price and Profit After a Forced Deviation



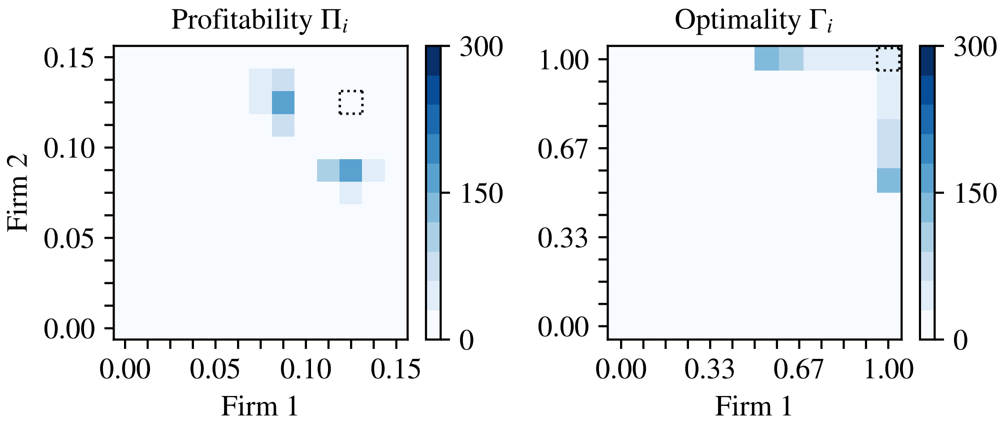
Notes: Results are for those runs that led to a Nash equilibrium outcome on the joint-profit maximizing price. Dotted line indicates moment of deviation. Results are in case of amount of price intervals $k = 6$, learning duration $T = 500{,}000$, stepsize parameter $\alpha = 0.3$ and discount factor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

Figure 4: Performance Under Different Learning Durations T and Amount of Pricing Intervals k



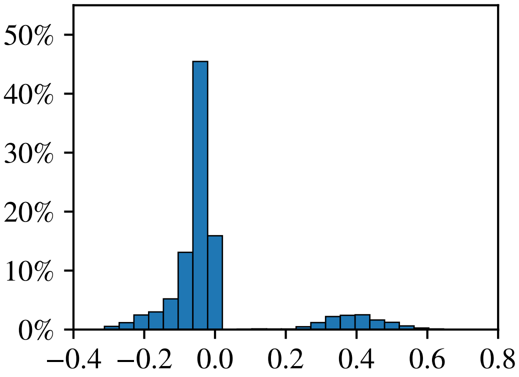
Notes: Results are in case of different amounts of price intervals k, with stepsize parameter $\alpha = 0.3$ and discount factor $\delta = 0.95$.

Figure 5: Joint Distribution of Final Profitability Π_i and Optimality Γ_i for $k = 24$



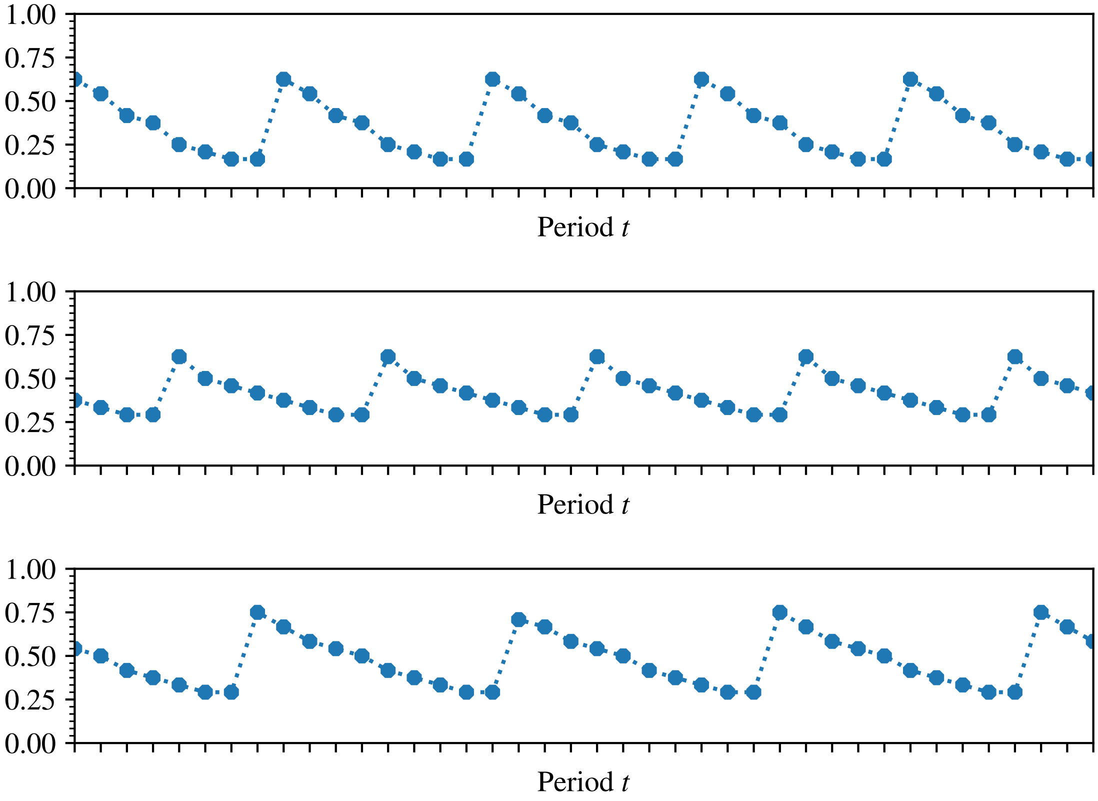
Notes: Dotted squares indicate joint-profit maximizing profit and Nash equilibrium respectively. Results are in case of amount of price intervals $k = 24$, learning duration $T = 500{,}000$, stepsize parameter $\alpha = 0.3$ and discount factor $\delta = 0.95$.

Figure 6: Distribution of Changes in Market Price



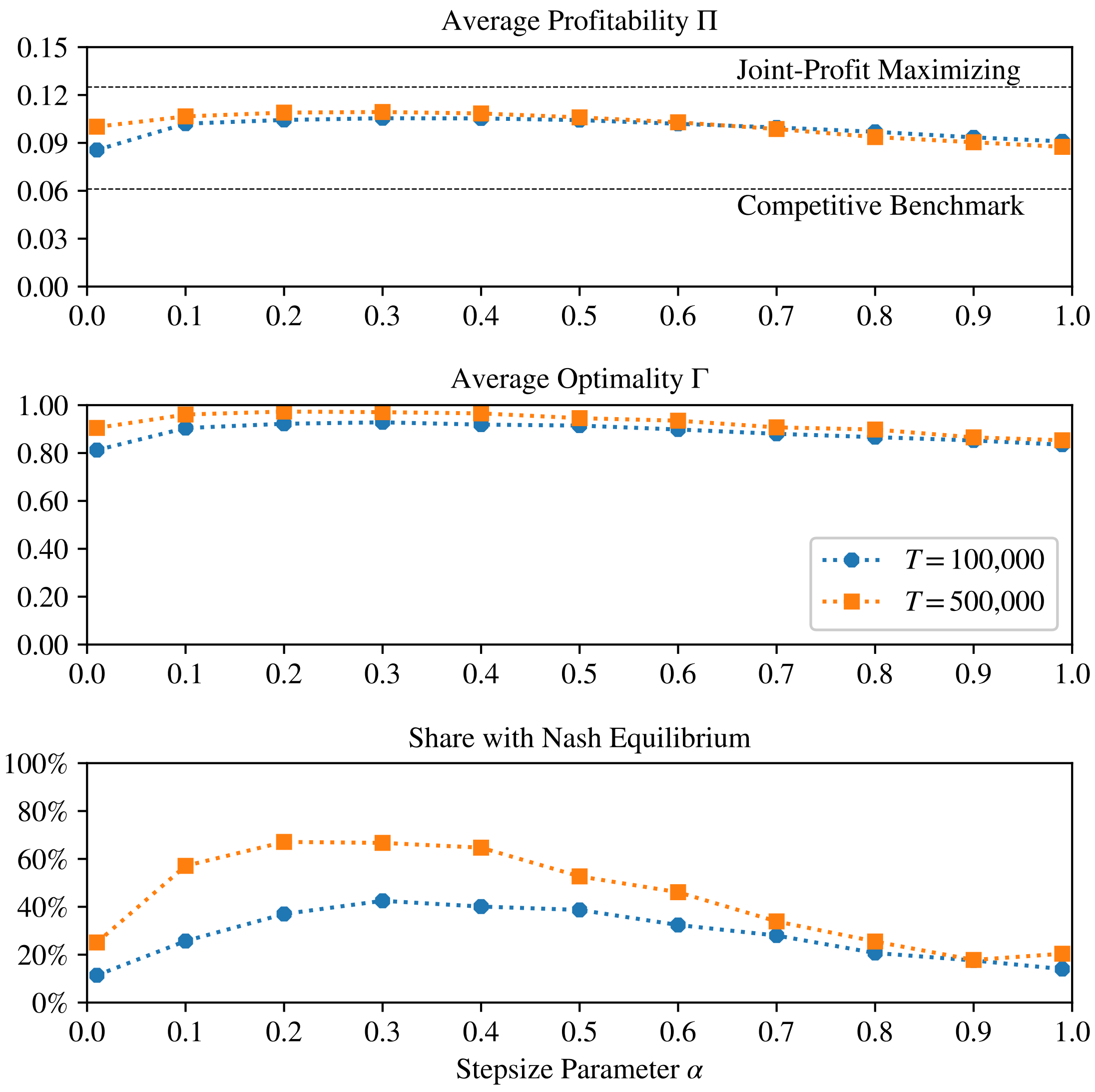
Notes: This figure looks at all the changes in market price that occur during the final 100 periods of all runs put together, where each bar represents a possible price change. Results are in case of amount of price intervals $k = 24$, learning duration $T = 500{,}000$, stepsize parameter $\alpha = 0.3$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

Figure 7: Illustration of Final Market Prices for $k = 24$



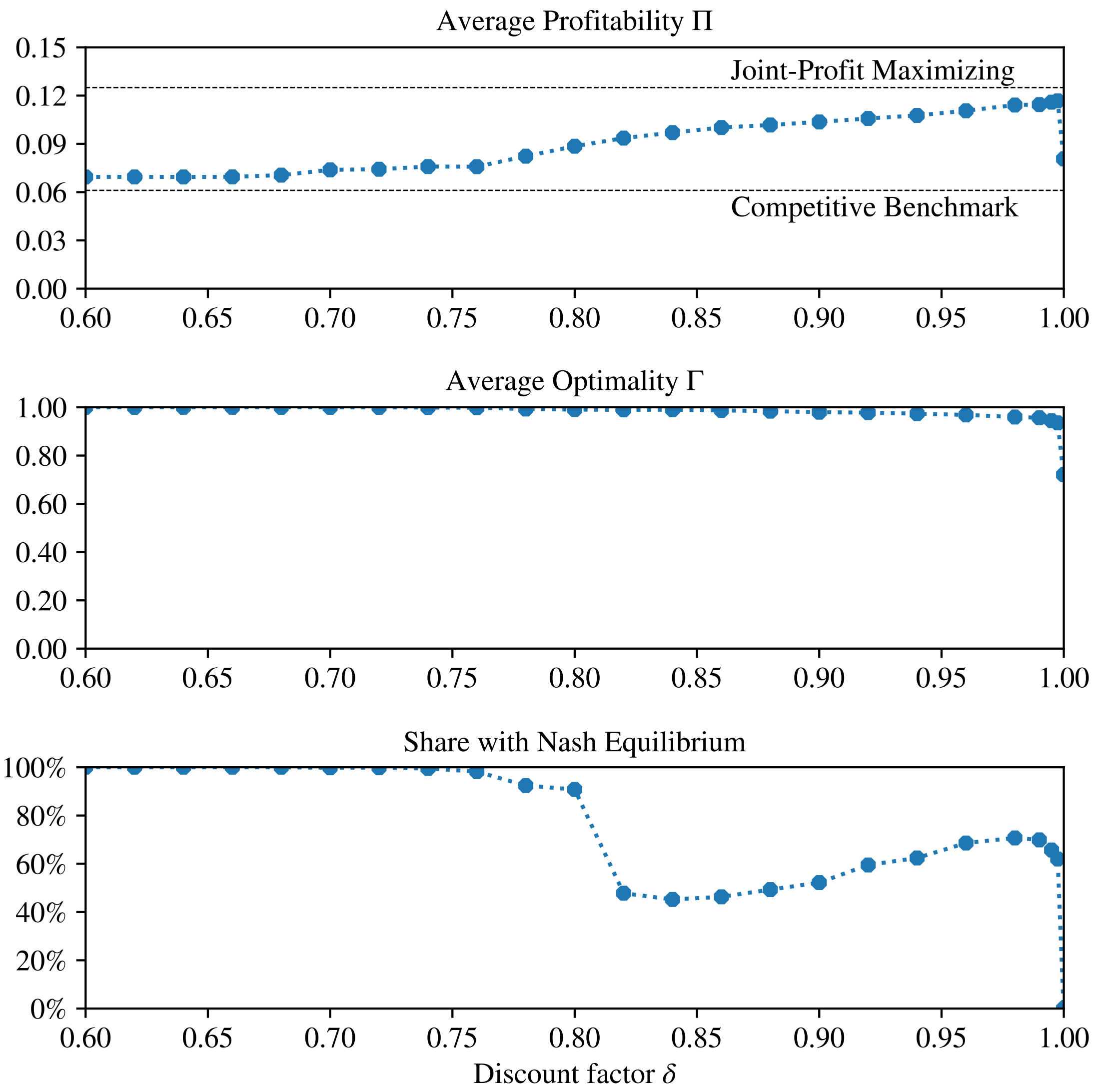
Notes: This figure looks at the market price during the final 40 periods of the first three individual runs. Results are in case of amount of price intervals $k = 24$, learning duration $T = 500{,}000$, stepsize parameter $\alpha = 0.3$ and discount factor $\delta = 0.95$.

Figure 8: Performance Under Different Stepsize Parameters α



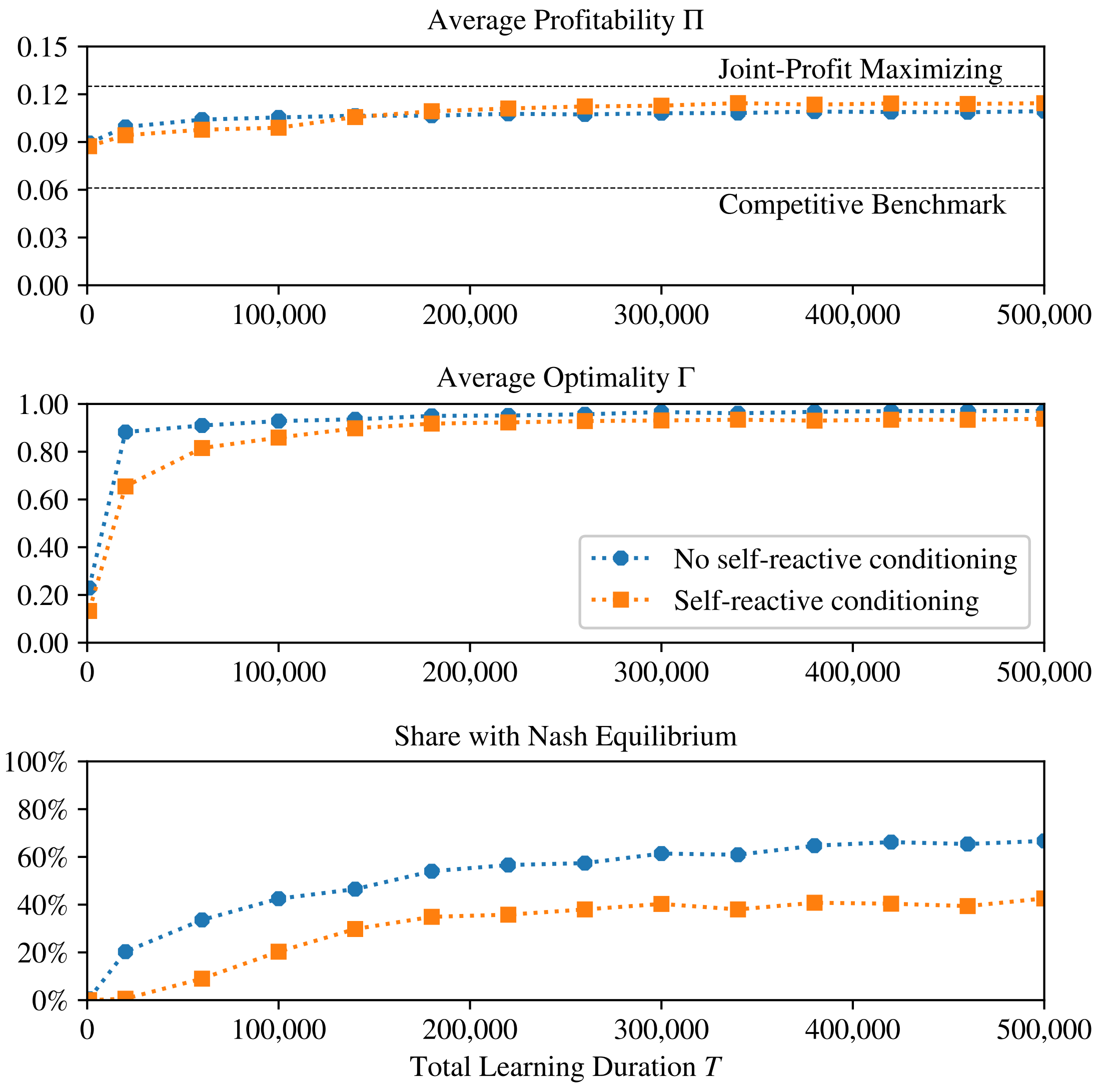
Notes: Results are in case of amount of price intervals $k = 6$ and discount factor $\delta = 0.95$.

Figure 9: Performance Under Different Discount Factors δ



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

Figure 10: Performance Under Self-Reactive Conditioning



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