Algorithmic Pricing: What Implications for Competition Policy?[1]

Emilio Calvano*°, Giacomo Calzolari*^°, Vincenzo Denicolò*^ and Sergio Pastorello*

*University of Bologna

^CEPR, °Toulouse School of Economics

June 27th, 2018

Abstract. Pricing decisions are increasingly in the "hands" of artificial algorithms. Scholars and competition authorities have voiced concerns that those algorithms are capable of sustaining collusive outcomes more effectively than human decision makers. If this is so, then our traditional policy tools for fighting collusion may have to be reconsidered. We discuss these issues by critically surveying the relevant law, economics and computer science literatures.

I Introduction

The geographical fragmentation of markets has dramatically fallen in Europe in the last decades. The European Common Market has removed artificial barriers to the free movement of goods, technological progress has significantly decreased transportation costs, and the advent of electronic commerce has greatly enlarged the set of potential suppliers which a typical buyer has access to. As a result, for many types of goods Europe is today a single geographical market.

This process has brought about substantial benefits but has also created new challenges. In this paper, we focus on one side-effect of electronic commerce, namely, the diffusion of algorithmic pricing. Firms' pricing decisions are increasingly delegated to software programs that incorporate the latest developments of Artificial Intelligence. While pricing algorithms have been used by airline companies for decades,[2] recently they have been adopted in sectors such as financial markets and the hotel and insurance industries. Still more recently, their diffusion has further extended beyond these domains. Today, Algorithmic Pricing (algorithmic pricing) has become

[1] We are grateful to Patrick Legros, two anonymous referees, the editors and participants at the RIO conference Cambridge Judge Business School 2018.

[2] British Airways seems to have been the first company to use pricing algorithms in the '70s.

widespread. For example, Chen et al. (2016) document that a significant fraction of sellers in a large online marketplace (Amazon US), where many different types of goods are traded, adopted algorithmic pricing in 2015.[3]

If anything, the prevalence of algorithmic pricing seems destined to increase. The demand for algorithmic pricing is likely to keep growing as more and more transactions take place in digital environments, and the software technology further improves. The supply of algorithmic pricing will presumably keep up with the demand. In fact, algorithmic pricing has become affordable even for small businesses, as off-the-shelf machine learning solutions and computing capability are now being supplied by tech giants such as Amazon, Google and Microsoft. More entry in this new industry might likely take place in the future. Recent developments, such as the Distributed Digital Ledgers and the Internet of Things, may further fuel the growth in the demand for and supply of algorithmic pricing.

The diffusion of algorithmic pricing raises various concerns for competition policy and regulation. One concern is that algorithmic pricing tremendously enlarges the scope for price discrimination. [4] While the competitive effects of price discrimination are generally uncertain, with algorithmic pricing prices may be conditioned not only on relatively innocent information such as the timing of the purchase, or the firm's residual capacity, but also on the buyer's entire past purchasing history. Such conditional pricing may lead to consumer poaching, or to the use of exclusivity or market-share discounts, both of which may have anti-competitive effects. Furthermore, the commercial exploitation of information that should arguably remain private may raise issues of privacy.

Another source of concern is the possibility that algorithmic pricing may facilitate collusion. This concern has been repeatedly voiced in the last years, both in the popular press[5] and the academic literature. In particular, Ezrachi and Stucke (2015) and Mehra (2016) explicitly point to the risk that algorithmic pricing may inhibit competition and effectively sustain collusion with no need of human intervention. The issue is now on the radars of various antitrust agencies, such as the Federal Trade Commission and the European Commission.[6]

---

[3] European Commission's 2017 *Final report on the E-commerce Sector Inquiry* concludes that "A majority of retailers track the online prices of competitors. Two thirds of them use software programs that autonomously adjust their own prices based on the observed prices of competitors."

[4] algorithmic pricing could provide a competing explanation, and thus an identification challenge, for the evidence of higher online prices in some markets relative to offline – the prevalent one being that of increase in the match quality (Ellison and Ellison, 2018). algorithmic pricing could also speak to the question of what is causing online price dispersion, both in cross-section and time-series, in seemingly-homogenous product markets (Chen et al., 2016).

[5] For example, the *New Yorker* asked what happens "*When bots collude*" (Algorithmic pricingril 25th, 2015), and the *Financial Times* talked of "*Digital cartels*" (January 8th, 2017).

[6] See, for instance, the remarks by the Acting Chairman of the U.S. Federal Trade Commission, M. Ohlhausen, at the *Concurrences Antitrust in the Financial Sector Conference*, New York, May 23, 2017 ("Should We Fear the Things That Go Beep in the Night? Some Initial Thoughts on the Intersection of Antitrust Law and Algorithmic Pricing"); the OECD Roundtable on Algorithms and Collusion of June 2017, and the remarks of European Commissioner for Competition, M. Vestager, at the *Bundeskarellamt 18th Conference on Competition*, Berlin, March 16, 2017 ("Algorithms and Competition").

These concerns may seem somewhat speculative as, so far, the evidence of digital cartels seems limited. To the best of our knowledge, the only antitrust case involving algorithmic pricing is the US's agencies successful challenge of a pricing software allegedly designed to coordinate the price of posters by a number of online sellers.[7] Commentators have also pointed to a few specific examples, such as that of software programs that appear to have escalated the price of a second-hand book to the millions of dollars,[8] or, perhaps more importantly, the use by several car manufacturers of the same pricing algorithm for the pricing of car parts, which allegedly resulted in billions of extra profits in the European market.[9] However, we have not seen much real action on the antitrust front so far.

In the light of this evidence, or lack thereof, the optimistic view of algorithmic pricing is that algorithms are not really more conducive to collusive outcomes than humans. According to this view, the concerns mentioned above are exaggerated. The pessimistic view, in contrast, maintains that the anticompetitive potential of algorithmic pricing has not fully materialized yet, that algorithmic pricing is still in its infancy, but things can get worse as it enters maturity, and that antitrust authorities may have refrained from intervening because they are not well equipped to cope with this new form of collusion.

In this paper, we contribute to this debate, which so far has involved for the most part law scholars and, especially, computer scientists.[10] Bringing an economic perspective into the debate may be useful, as algorithmic pricing raises in fact a number of important economic questions. Can "smart" pricing algorithms learn to collude? Is collusion among algorithms any different from collusion among humans? In particular, is algorithmic pricing conducive to collusion more often than what humans could do? If the answers to these questions are affirmative, further issues will arise. How can we detect algorithmic collusion? What are the appropriate new standards for competition policy?

In the next sections, we shall briefly address these questions. Since economic research on algorithmic pricing is still limited, we are not in a position to provide answers. Our aim is, more modestly, to clarify the terms of the debate.

II Adaptive and learning algorithms

At the cost of oversimplifying, one can distinguish between two classes of algorithms for pricing, which for the sake of brevity we shall call "adaptive" and "learning" algorithms. We discuss them separately as the competitive concerns they raise may be different.

[7] See *U.S. v. Topkins*, 2015.

[8] See Olivia Solon, 2015, "How a book about flies came to be priced $24 million on Amazon."

[9] See https://theblacksea.eu/stories/article/en/car-parts-probe.

[10] Harrington (2017) develops a legal approach for collusion with algorithmic pricing that is grounded in economic analysis.

II.a Adaptive algorithms

First-generation pricing algorithms were adaptive in nature. These algorithms incorporated a model of the market and sought to maximize the firm's profit based on the model. Dynamic pricing for revenue management, which has been used for some time in hotel booking and airline services, belongs to this class.

Adaptive pricing algorithms typically perform two activities: estimation and optimization. Accordingly, they may be viewed as comprising an estimation module and an optimization module. The estimation module estimates market demand using past volumes and prices, and possibly other control variables. The optimization module then chooses the optimal price given the demand estimate and observed past behavior of rivals.

When market conditions are known, so that the estimation function is idle, adaptive algorithms essentially set a firm's price as a function of rival's past prices. This adaptive behavior may be more or less sophisticated. In some cases, the "optimization" module actually boils down to a fixed and perhaps somewhat arbitrary adjustment rule. For example, an algorithm may set the own price as a fraction (or multiple) of the rival's price. In other cases, the optimizing behavior is more sophisticated. For example, the algorithm may calculate a best response to rivals' strategies. Examples include "best response dynamics" (where the firm's price is the best response to the competitors' last period prices), "fictitious play" (where the firm plays a best response to a fictitious mixed strategy which is taken to be the past price distribution) and what is called, perhaps imprecisely, "Bayesian learning" (where the firm plays a best response to a weighted average of past previous prices with exponentially declining weights).[11]

These forms of adaptive behavior, where a player plays a static best response to some combination of the rivals' past strategies, have been theoretically analyzed by Milgrom and Roberts (1990). They show that in supermodular games (a class of games with strategic complementarities that includes the typical Bertrand pricing game), such adaptive behavior generally converges to outcomes that do not exhibit collusion. For example, in pricing games the system converges to prices that are no higher than the Nash equilibrium prices of the one-shot pricing game.[12]

This result has been taken to suggest that algorithmic pricing does not really make collusion any easier to achieve. To produce collusive outcomes, the programs must not play static best responses; rather, they must be instructed to condition their actions on the rivals' past behavior in a collusive fashion.

Of course, not every such conditioning leads to collusive outcomes. For example, it may be easy to design rules that mechanically lead to high prices. However, collusion requires that the prices be *profitably* high.[13] The problem is, prices may be profitably high in many different ways, which may

[11] The term may be imprecise as these algorithms learn only in a very limited sense.

[12] Whether this result survives when market conditions are not stationary, and hence the estimation function of the pricing algorithm is active, is an open question.

[13] For example, the second-hand book episode seems to have been generated by adaptive algorithms which would set the own price as a multiple of the rival's price. If two firms adopt a pricing rule of the type $p_i = a_i p_j$, the system

benefit different firms to different degrees. Adaptive algorithms must therefore be instructed to coordinate on one of many possible outcomes. Furthermore, adaptive algorithms must be instructed to support such coordination by means of a system of punishments in case the rivals defect.

Importantly, both sets of instructions must be fed into the software: adaptive algorithms cannot collude unless they are designed by their programmers to do so. But if this is so, then the programmers must solve exactly the same coordination problems as human price makers.

From this observation, skeptics of algorithmic pricing collusion draw two conclusions. The first one is that it is unlikely that separate programmers can achieve any significant degree of coordination without explicitly communicating. This implies that algorithmic pricing collusion may be proved by exactly the same type of evidence as traditional collusion: for example, minutes of meetings, phone calls, e-mails etc. The only difference is that the search for a smoking gun is moved one stage ahead, to the design of the software rather than the actual pricing.

In fact, and this is the second conclusion, collusive software must include lines of coding that reveal the programmers' (or the managers') collusive intent (as in *U.S. v. Topkins*). This contributes to generate the "hard" evidence that antitrust authorities look for. In sum, according to the skeptics algorithmic pricing collusion is harder to achieve and, if anything, easier to detect than human collusion.

Yet, adaptive algorithms differ from humans in another important way, namely, the frequency of interactions. algorithmic pricing may react to rivals' actions much more quickly than human beings. This property of algorithmic pricing has been emphasized by Ezrachi and Stucke (2015) and Mehra (2016). As is well known, simple models of collusion predict that more frequent interaction makes collusion easier to sustain, as defection is punished more promptly and hence the gains from defection are reaped for a shorter time. From this, Ezrachi and Stucke (2015) and Mehra (2016) conclude that collusion is more likely with algorithmic pricing.[14]

In any case, with adaptive algorithms there seems to be no reason to change the traditional policy approach. Antitrust authorities should look for the usual type of evidence; the only difference is that they should focus not on managers but on codes and programmers as witness or complicit.

II.b Learning algorithms

Second-generation algorithmic pricing is based on more recent developments in computer science, which belong to the field of Machine Learning (machine learning). Rather than specifying

---

explodes whenever $a_i a_j > 1$. As a results prices may get very high indeed, but in terms of profit maximization the outcome will be poor.

[14] In fact, the result that more rapid responses facilitate collusion has been questioned in more recent theoretical contributions. Sannikov and Skrzypacz (2007) argue that faster interaction may actually impede collusion under imperfect observability of the rivals' actions. The reason for this is that responding too quickly to noisy information may unravel the collusive scheme. Sannikov and Skrzypacz derive this result by assuming that agents optimally extract the signal from the noisy information they receive. Whether the result continues to be true also for algorithmic pricing remains to be investigated.

a pricing problem and instructing the software to solve it, with machine learning the software learns how to solve the task from experience.[15]

To gain such experience, machine learning algorithms experiment adopting strategies that would be sub-optimal according to their current knowledge. Experimentation is costly in that it entails, in expectation, a short-run sacrifice of profits. However, it is valuable as it allows to learn from more diverse situations.[16]

In the terminology of Fudenberg and Levine (2016), whereas adaptive algorithms learn in a passive way, what we call "learning" algorithms learn more actively. They define learning as "passive" when (p. 157)

> "players have no incentive to change their actions to gain additional information."

This is precisely what adaptive algorithms do. They do get additional information as time passes, so their estimation of market demand may improve, but they do not intentionally change their behavior in order to acquire more information. machine learning pricing algorithms, in contrast, exhibit a more active type of learning: they are willing to adopt strategies that may be suboptimal so as to learn from experience.

Even more importantly, machine learning algorithms are to a large extent model free. There is no need of specifying a model of the market, estimating the model, and solving for the optimal strategy. The programmer chooses just which variables the strategy should be conditioned on, how frequently the program must experiment, and how much weight should be given to the more recent experience relative to the cumulated stock of knowledge. The algorithm starts from an arbitrary assessment of the value of the feasible strategies, and then updates these values on the basis of its realized payoffs. In this fashion, the algorithm learns to play optimally from experience.

For example, a machine learning program designed to play chess need not be fed with any notion of chess strategy. All the program needs to know are the legal moves. In addition, the program needs an initial assessment of the value of each possible position. This can be arbitrary, or extremely simple (for example, the difference between the total value of own pieces and the opponent's pieces). The program learns from experience how to assess every possible position, and hence how to play optimally.

As mentioned above, during the learning phase suboptimal decisions may be taken frequently. This is costly, and the learning phase may last for quite a while. (In practice, the problem can be alleviated by letting the program learn in a simulated environment before using it for making real pricing choices.) After the initial phase, however, being model-free gives machine learning pricing algorithms a great advantage over adaptive algorithms, especially in complex environments.

[15] Machine learning techniques have been developed and are currently adopted for a large number of applications. By far the most popular in the social sciences are those of classifying tasks (with supervised learning) and those meant to uncover hidden structure in big datasets (with unsupervised learning).

[16] If the environment is stationary, experimentation is particularly valuable initially. As time passes and the algorithm learns, however, experimentation may become less crucial and thus it may optimally vanish, eventually. In a stochastic dynamic environment, in contrast, it may be optimal to keep experimenting forever.

From the viewpoint of competition policy, however, the concern is that algorithms which learn from experience "too well" may actually learn to collude, even if they have not been specifically designed to do so.

This last point is indeed crucial. Differently from adaptive algorithms, learning algorithms may come to solve the coordination problem even if they have been designed innocently. Programmers, and the supervising managers, need not instruct the program to coordinate on a specific outcome, nor to adopt a specific system of punishments. Therefore, they need not communicate to achieve efficient coordination. This implies that to the extent that learning algorithms do collude, then this type of collusion poses new challenges to competition policy.

However, sustaining collusion is a daunting task, because the algorithms must come to coordinate on both a collusive outcome and a punishments mechanism. Thanks to their attitude to explore, machine learning algorithms might in principle solve both coordination problems quite effectively. However, the process of learning by each algorithm may be disrupted by the rival's experimentation.

The problem of whether, and to what extent, learning algorithms may actually learn to collude is still open.[17] Previous literature[18] has found that collusion among algorithms is possible but rather unlikely. That is, collusion is typically either partial, or it occurs only with a relatively low probability, in the range of 30%. That literature therefore suggests that algorithmic pricing does not significantly increase the risk of collusion as compared to humans, who also exhibit in experiments a tendency to collude of comparable magnitude.[19]

In what follows, we shall report some very preliminary results from our own research on the topic, which suggest that in fact algorithmic collusion may be much more prevalent. With reasonable parameter values, we have observed collusion to emerge in more than 60% of the cases (more than 80% after enough repetitions that the algorithmic learning may be regarded as completed). In the next sections, we shall briefly describe our experiments with pricing algorithms. We then discuss the implications of these preliminary findings for competition policy.

III Q-learning in a simple pricing game

We focus on a specific class of algorithms, namely, Q-Learning algorithms. The reason for this is that Q-learning algorithms are well understood and relatively simple, and they form the basis for more sophisticated algorithms.

[17] Salcedo (2015) presents a theoretical model showing that collusion with algorithms is not only possible, but actually inevitable. However, this result relies on strong assumptions. First, Salcedo assumes that algorithms are able to read into other algorithms, thereby learning their "intentions." Second, he posits that programmers can commit not to revise the algorithms in use.

[18] See, for instance, Tesauro and Kephart (2002), Xie and Chen (2004), Waltman and Kaymak (2008) and Dogan and Guner (2015).

[19] See, for instance, the recent survey by Dal Bo and Frechette (2018).

Generally speaking, Q-learning tools tackle the problem of finding an optimal policy in Markov Decision Problems or problems alike.[20] A Markov Decision Problem is a formal framework that allows the analysis of repeated decision making in dynamic stochastic environments. To be concrete, consider the problem of a price setting firm in oligopoly. Every period, the firm i) observes relevant information such as the price charged by its rivals in previous periods or the state of demand, ii) sets its own price and iii) collects the resulting profits. The firm's problem is that of finding the pricing policy that maximizes the present value of its profits. A policy is a mapping from what it observes, the "state", to its control variable, the price. The Q-Learning algorithm is a tool designed to "crack" this decision problem through a process of experimentation. Experimenting allows to learn the policy that maximizes long-run profits.[21]

Q-learning algorithms are particularly appealing for a number of reasons. First, they learn the optimal policy while playing and thus do not need to be trained with data (in contrast with supervised learning). Second, they do not require any *a priori* knowledge of the effect of one's actions on the environment. For example, in the pricing task they do not require the programmer to specify consumers' demand, or rivals' future reaction to one's price. Finally, they can confront a great deal of uncertainty.

IV.a Experimentation and learning

How does the algorithm learn? A key ingredient is the Q-matrix, which stores an estimate of the present value of choosing any given action in any given state. In our simplified setting, the task for firm i is to choose, in every period, one of two exogenously given prices, say $p_{\mathrm{i}} = p^H$ or $p_{\mathrm{i}} = p^L$. The firm observes the prices charged by its rival, $p_{\mathrm{j}t}$, and keeps track of its own past price, $p_{\mathrm{i}t}$. Assuming a one-period memory,[22] the current strategy may be conditioned only on the last period prices. These prices then constitute the present state $s_t = (p_{\mathrm{i}t}, p_{\mathrm{j}t})$. With two possible prices for each firm, there are 4 possible states. Thus, in this simple example the Q-matrix is a 4x2 matrix:

$Q_{\mathrm{i}}(s,p_{\mathrm{i}})$	$p_{\mathrm{i}} = p^L$	$p_{\mathrm{i}} = p^H$
$s_1=(p^L, p^L)$		
$s_2=(p^L, p^H)$		
$s_3=(p^H, p^L)$		
$s_4=(p^H, p^H)$		

[20] For a textbook introduction to Q-learning see Sutton and Barto (1998).

[21] It is important to note that Q-learning algorithms differ from numerical methods for equilibrium identification in the analysis of Markov Perfect industry dynamics, as in Doraszelski and Pakes (2007). The crucial difference is that algorithms experiment and learn from experimentation.

[22] Some variants of Q-learning assume that the decision is memoryless. In that case, the discount factor γ becomes irrelevant.

The rows indicate the four possible states. As said above, each entry of the matrix can be interpreted as an assessment of the present value of the stream of profits obtained by following the choice of price p_i in state s.

The matrix is initialized with some arbitrarily assigned values[23] and is then updated on the basis of experience. The updating takes places as follows. Let π_i denote the observed profit of firm i resulting from charging price p_i in state s. At the end of the current period, the new value of the Q-matrix at the $(s,\ p_i)$ cell is updated as follows (we denote the present state by s and the future state by s'):

$$Q_i^{new}(s, p_i) = (1-\alpha)Q_i(s, p_i) + \alpha[\pi_i + \gamma \max_{p_i} Q_i(s', p_i)] \qquad (1)$$

where the positive parameter $\alpha < 1$ is the *learning rate* and $\gamma < 1$ may be interpreted as the discount factor.[24]

The Q-learning equation is reminiscent of a Bellman equation in dynamic programming.[25] Analytically, learning is reflected by the term inside square brackets in (1). The updating takes into account not only the current realized profit π_i, but also the future payoff that can be obtained once the system moves to the new state s'.

Only the cell of the matrix that corresponds to the state which has been visited is updated. For all other cells, the Q-values do not change. Strategies that perform well are reinforced, as their Q-value increases. This is why Q-learning may be viewed as an instance of Reinforcement Learning.[26]

We now describe more precisely how Q-values translates into actual choices. The choice of a price level balances two needs, that of gathering new information (exploring) and that of reaping profits (exploiting). In other words, exploitation means that the information already gathered is used to choose the action that corresponds to the higher estimate of the present value. This is the price (action) corresponding to the larger value of $Q_i(s, p_i)$ in state s.

The balancing between exploitation and exploration may be described by two alternative models, the ε-*greedy* model and the *Boltzmann* exploration model. In the ε-greedy model, in every period the algorithm explores with a given probability ε and exploits with the complementary probability

[23] In our experiments, we have initialized the Q-matrix assuming that the Q-values are the discounted profits under the conservative assumption that the rival always sets the low price.

[24] Note that equation (1) implicitly assumes what in the computer science literature has come to be known as "independent learning." In the alternative case of "joint learning," the updating involves a notion of equilibrium play given the new state s'.

[25] Generally speaking, in "reinforcement learning" models an agent learns by trials and errors. Those actions associated with better consequences (rewards) are reinforced and thus have higher chances of being chosen in the future. See, for instance, Roth and Erev (1995), Erev and Roth (1998) and Sarin and Vahid (2001).

$1 - s$. When it explores, the algorithm chooses from all feasible prices (including those with a low Q-value) with the same probability.

The probability of exploration s can be time invariant, or it may decline over time. In our experiments, we have taken the probability s to be time invariant.

A common alternative to the ε-greedy model is the *Boltzmann exploration model* where the probability of selecting price p_i at time *t* is

$$\frac{e^{Q_i(s,p_i)/\beta}}{\sum_p e^{Q_i(s,p)/\beta}} \qquad (2)$$

where β is the so-called "temperature" of the system. Like s, β may be time invariant or monotonically decreasing with time. The Boltzmann exploration model always favors actions with higher Q-values, the more so the lower is β. Thus, letting the "temperature" decrease over time favors exploration at early stages while facilitating the eventual convergence of the system.

Does the algorithm converge to the "optimal policy?" For optimization problems that do not involve strategic interaction, Watkins and Dayan (1992) showed that it does, provided that certain technical conditions are met. With many players and strategic interaction, there exist no general convergence result. The difficulty is that all algorithms simultaneously explore and thus each affects the learning environment of the rivals.[27]

IV.b An example

We now present some preliminary results obtained from a simple example, where the per-period profits are given by the following table:

	$p_i = p^L$	$p_i = p^H$
$p_j = p^L$	0.5	0.25
$p_j = p^H$	1.25	1

The stage game has therefore a unique Nash equilibrium, which is to choose the low price. Aggregate profits, in contrast, are maximized when prices are high. This environment captures in a simple way the idea that in order for a collusive agreement to be sustained, firms need to overcome the temptation to maximize their short run profits by undercutting the price set by their rivals.

Now consider an infinitely repeated game, where profits are discounted by the common discount factor $\gamma = 0.995$. With one-period memory, when the stage game is infinitely repeated there are three Subgame Perfect Nash Equilibria, one competitive and the others collusive. In particular, we shall refer to the collusive equilibria as "Grim Trigger" and "One Period Punishment." In the first,

[27] Formally, in a strategic environment it is no longer true that the state representation of the system contains all the relevant information (the Markov property) and the environment is not stationary. For a systematic treatment of the convergence of Q-learning algorithms converge also in strategic environments, see for instance Busoniu et al. (2008).

competitive equilibrium firms always choose the low price, irrespective of the past history, as in the static Nash equilibrium. In the Grim Trigger equilibrium firms charge a high price initially, and then charge the high price if both have chosen the high price in the last period, and the low price otherwise. [28] In the One Period Punishment equilibrium, in contrast, firms choose the high price if they chose the same price (high or low) in the last period, the low price if they chose different prices.

In our experiment, the repeated game was played by two identical Q-Learning algorithms with a learning rate $\alpha = 0.15$ and a constant experimentation rate of $\mathrm{s} = 0.04$.

With $\mathrm{s} = 0.04$, each firm chooses the price that it regards as suboptimal with a probability of 2%. Since firms starts from an assessment that regards high prices as suboptimal, this means that collusion occurs by chance, on average, four times every 10,000 repetitions. This suggests that a large number of repetitions may be necessary for the firms to learn to collude.

We have therefore had the stage game played one million of times.[29] Since firms' experimentation creates noise, we have run 1,000 experiments for the same set of parameter values. We have then calculated the average profit of the firms, both over the last 100,000 repetitions (where the learning process is presumably complete) and over the entire set of repetitions.[30]

It is convenient to express the payoffs in terms of the percentage gain with respect to the static Nash equilibrium. The gain would be 100% if firms always chose the high price, 0 if they always chose the low price. This is a synthetic measure of the degree of collusion achieved by the firms.

Averaging across the 1,000 simulations, the profit gain is almost 70% of the maximum potential gain. In the last 100,000 repetitions, the gain is more than 85%.

How is such high level of collusion achieved? To provide some insights, Figure 1 shows, for a representative simulation, the gain from defection (i.e., the difference $Q_{\mathrm{i}}(s_4,\, p^L) - Q_{\mathrm{i}}(s_4,\, p^H)$ in the "collusive" state s_4 where both prices are high. A positive value of the difference means that, for that particular state, the algorithm would prefer to set a low price if it did not have to experiment. Collusion is sustained if the difference is negative, so that, assuming that the algorithm exploits and does not explore, it sets a high price.

Figure 1 shows that it takes around 70,000 repetitions[31] for the algorithms to realize that collusion may indeed be profitable. (Remember that the Q-matrix is initialized assuming that the rival does not cooperate.) After that learning phase, the Q-value is negative most of the times. This means that most of the times, when in a high-price state, the algorithms keep setting a collusive price

[28] Our notion of "Grim Trigger" is reminiscent of the standard Grim Trigger strategy, which requires an infinitely long memory, and exhibits the same incentive compatibility condition for the strategy to be an equilibrium.

[29] Roughly speaking, assuming that prices are changed every 5 minutes, this corresponds to a time horizon of ten years. (This time-scale would imply that initially, both firms set collusive prices by chance once per week.) With so short a period, even a value of the discount factor of 0.995 may seem too low (the implied interest rate being extremely high). However, such a low discount factor is in fact a conservative assumption that reduces the likelihood of collusion.

[30] Even if the game is symmetric and the algorithms are identical, the payoffs of the two firms need not be the same as the algorithms experiment randomly. However, the actual differences turn out to be nominal.

[31] In the time-scale of footnote 28, this roughly corresponds to half an year.

(unless they are experimenting). However, since firms do keep experimenting, every now and then a "price war" occurs. We conjecture that if the rate of experimentation decreased over time, such episodes would be less and less likely.

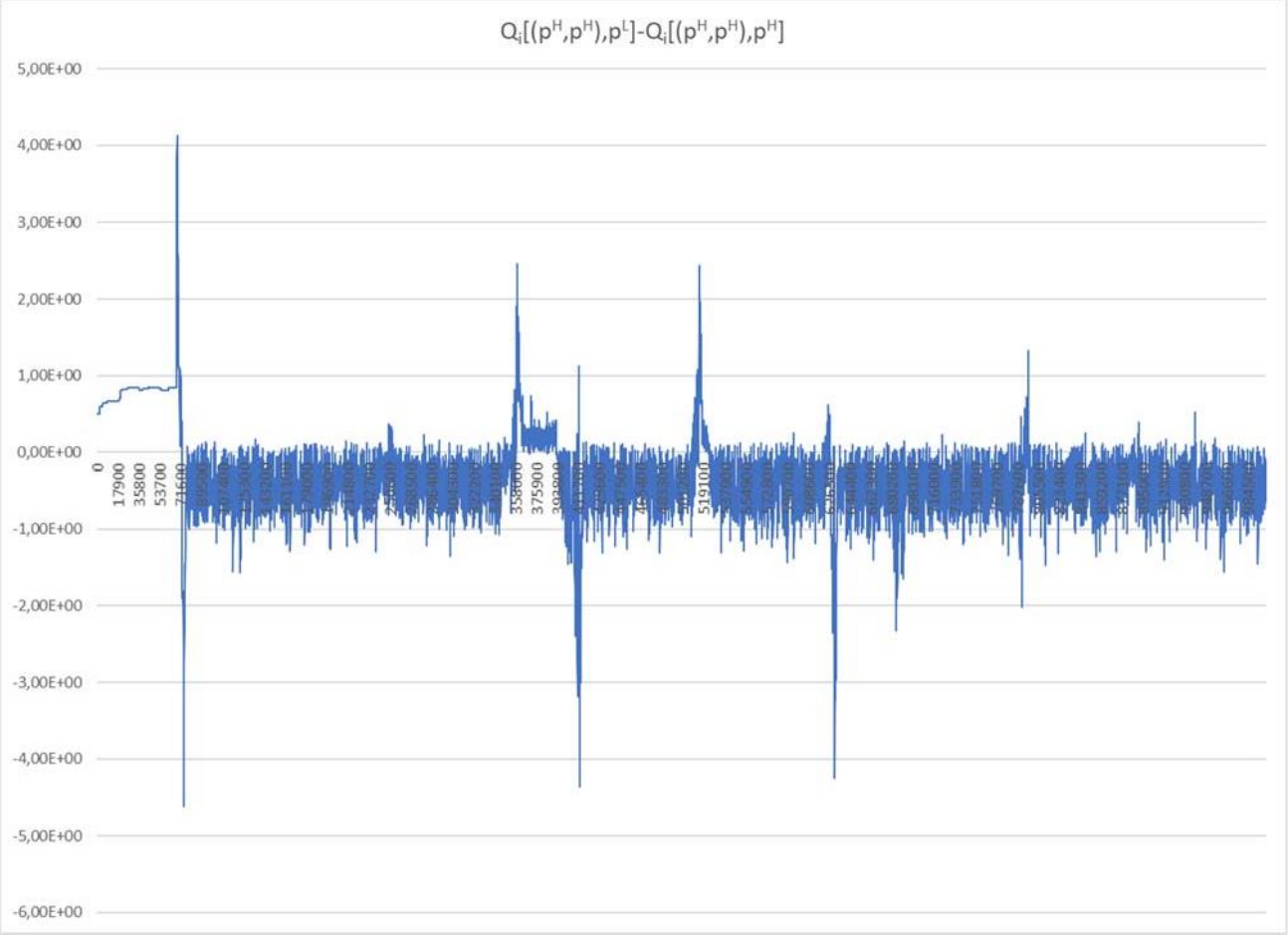

Figure 1: Difference of the Q-values in the "collusive" state with high prices: a negative value means that the algorithm sets a high collusive price when it does not explore.

The selected simulation is representative of the 1000 we have run. Averaging across simulations, the competitive state (both prices low) occurred in approximately 24% of the repetitions, mixed states (a low and a high price) in 13% of the repetitions, and the collusive state in approximatively 63% of the repetitions. (For the last 100,000 repetitions, the corresponding figures are 7%, 11% and 82%, respectively.)

The evidence presented in this section represents just one example. However, the results are quite impressive and warrant, in our opinion, a more thorough investigation of the possibility of collusion in algorithmic pricing.

IV An agenda for future research

While our preliminary results may be suggestive, there is still a lot of research to be done in order to understand to what extent and under what conditions machine learning pricing algorithms may

learn to collude. In this section, we briefly discuss a number of robustness checks and challenges that should be addressed before drawing reliable conclusions.

First, with only two possible price levels to choose from, the pricing game effectively becomes a simple prisoner's dilemma. Prisoner's dilemma games do indeed contain important elements of price competition, such as the mutual interest to set high prices and the unilateral incentive to undercut. However, they are probably simplistic description of actual interactions in markets. This raises the issue of whether pricing algorithms are able to cooperate in more complex and realistic environments. For example, just enlarging the set of possible prices would make the coordination problem much more difficult.

Second, theory suggests that introducing asymmetries and increasing the number of firms would make collusion less likely. It would be interesting to check whether these predictions, which have been obtained for perfectly rational agents, hold also for Q-learning algorithms.

Third, in our simple setting we have ruled out intrinsic uncertainty, the only uncertainty being that generated by the algorithms' experimentation. Theory suggests that uncertainty may make cooperation more difficult, especially if rivals' prices are not observable. With imperfect monitoring, players might misinterpret low payoffs as due to rivals' defection rather than bad market conditions. Is this the case also for algorithms?

Fourth, the analysis should be extended to more sophisticated algorithms than Q-learning. In fact, it seems likely that pricing algorithms adopted in practice incorporate the latest developments of Artificial Intelligence and hence are "smarter" than those used in our experiments.

Using more sophisticated algorithms may also serve to address some of the extensions mentioned above. For example, enlarging the set of strategies increases the computational complexity of the problem. However, recent advances in Neural Networks and Deep Learning allow to cope with high-dimensional state spaces by giving the algorithms the ability to "generalize." That is, the algorithms update their assessment of the Q-value of states that have not been visited by looking at what happens for "close enough" states.[32] For example, Tampuu et al. (2017) use a Deep Q-network that is similar to our simple Q-learning, adding an approximation of the Q matrices with neural networks.[33]

More sophisticated algorithms may also learn how rivals learn. For example, second-generation Q-Learning algorithms rely on estimates of the other players' Q-matrices. This requires that not only all actions are publicly observable, but also all players' payoffs. Other algorithms, called "joint-action learners," estimate the rivals' strategies. One may wonder whether these more recent approaches may be even better at reaching cooperation.

[32] In a celebrated experiment Mnih et al (2015) show how a deep reinforcement machine learning algorithm learned to solve complex tasks such as playing classic Atari videogames better than humans using as state the color of 210 x 160 pixels on the screen with a 128-colour palette and the scoreplay.

[33] With these sophisticated algorithms, observed behavior ranges from very competitive in zero-sum games, to highly cooperative when less confrontational payoffs appear in the stage game. Other recent and promising examples about cooperation in multi-agent setting is the work at Facebook Artificial Intelligence Research, e.g. by Lerer and Peysakhovich (2018).

Last but not least, we have ruled out communication among the algorithms. A vast theoretical and experimental literature, starting from Cooper et al. (1990), shows that that communication (even "cheap talk") may play a key role in sustaining cooperation among humans. In this respect, the lack of communication among pricing algorithms may limit the emergence of cooperation.[34]

This limitation could be overcome by studying algorithms that develop a language to communicate. Here again recent developments in Artificial Intelligence may be useful. For example, Sukhbaatar et al. (2016) have analyzed algorithms that, in cooperative tasks, have the ability to learn to communicate amongst themselves. In recent work, Crandall et al. (2018) have observed a state-of-the-art machine learning algorithm engaging in a form of signaling. They showed that this allows to sustain cooperation in a variety of different environments, and even when algorithms interact with humans. [35]

V Conclusions and implications for policy

Our understanding of whether and how pricing algorithms may collude, and if collusion among algorithms is easier than among humans, is still very limited. Yet, our preliminary results suggest that the risk of algorithmic collusion may be real and pose new challenges to competition policy. It is therefore worth discussing the proposed policy approaches.

Broadly speaking, one may distinguish three possible approaches. The first one, based on the optimistic view that algorithmic pricing does not really pose any new problem, is to stick to current policy. The second approach is to regulate the introduction of pricing algorithms *ex ante*, pretty much in the same way as the commercialization of new drugs is currently regulated. That is, any new pricing algorithm should be tested by a regulatory agency to ascertain whether it exhibits a tendency to collude (in which case it would be prohibited) or not (in which case it would be approved). Finally, the third approach is to regulate *ex post*, as competition policy typically does, but using different legal standards from the current ones.

In principle, a fourth possible option is an outright prohibition of algorithmic pricing. However, there is a wide consensus that algorithms may deliver big efficiency gains by allowing more efficient pricing. A *per se* prohibition of algorithmic pricing is therefore unlikely to be optimal, even setting aside the enormous problems of implementing such prohibition in practice.

Let us start then from current policy. Economists generally define collusion as a reward-punishment scheme that leads to prices and profits above some competitive benchmark (see, for instance, Harrington, 2017). This scheme may be agreed by the parties explicitly or tacitly. The

[34] Cooper and Kühn (2014) have shown that communication between humans helps cooperation by clarifying how individuals think about the environment(e.g., whether they really mean punishing deviations) and by making social punishments and rewards explicit.

[35] Studying environments where humans interact with machines is another interesting challenge for future research. Some of the more spectacular successes of Artificial Intelligence are precisely that algorithms were able to beat human champions at such games as chess, Go, checkers and poker.

effects of tacit collusion are not different from those of explicit collusion; the difference between the two lies mainly in the greater difficulty of achieving coordination without communication.

The current legal standard for collusion, however, requires not only coordination on supra-competitive prices, but also some conscious and mutually accepted agreement among firms – a "meeting of minds" – to restrain competition. In other words, current policy prohibits explicit but not tacit collusion. For example, parallelism in pricing is not enough to prove collusion, as it can be the outcome, for instance, of independent reactions to common shocks.

From an economic point of view, this policy can be rationalized on the basis of a particular assessment of the likelihood and the costs of making mistakes – false positives and false negatives. The implicit presumption underlying current policy must be that it is quite unlikely that coordination is reached without an explicit agreement (so false negative are rare), and that there are no precise methods for inferring collusion from observed price movements (so false positives are frequent). If this presumption is correct, then it makes sense to require direct rather than circumstantial evidence of an agreement among the parties.

Those who claim that algorithmic pricing does not call for any change in current policy explicitly or implicitly argue that algorithmic pricing does not radically modify the assessment of the likelihood of false positives and false negatives. In other words, they explicitly or implicitly argue that even with algorithmic pricing communication (among programmers rather than final decision makers) is still crucial for collusion, and that detecting implicit collusion remains extremely difficult. Our discussion above suggests that these claims are probably true for first-generation, adaptive algorithms but may not be true for learning algorithms. Pricing algorithms that learn from experience do not need to communicate in order to collude (in fact, at least in their first incarnations, they are not capable of communicating). Furthermore, these algorithms may learn to cooperate without any explicit collusive intent on the part of their programmers. Furthermore, the levels of collusion achieved may be considerable. Therefore, algorithmic pricing may significantly increase the risk of false negatives.

The second policy approach, *ex ante* regulation, has been proposed both by lawyers such as Ezrachi and Stucke (2016) and economists such as Harrington (2017). Specifically, Ezrachi and Stucke (2016) proposed a "sand-box" approach (as the one currently adopted for fintech firms in the UK) where one tries to nurture algorithms in virtual markets and monitor the association between their properties and the observed outcomes. Harrington (2017) proposes to check the behavior of specific algorithmic pricing and define a black list of algorithms that would become unlawful *per se*.

This regulatory approach would represent a form of public intervention much more intrusive than competition policy, which acts *ex post* rather than *ex ante*. Normally, such intrusive intervention is reserved for cases where market failure is evident and costly, and the efficiency losses due to the regulation are limited. It is unclear whether these conditions are met in the case of algorithmic pricing.

Another problem is that the collusive properties of pricing algorithms may depend on which other algorithms they interact with. Suppose that algorithm A has been approved on the basis of evidence that it does not tend to collude with existing algorithms B, C and D. Suppose, however,

that a new, superior algorithm E is subsequently developed. The new algorithm E tends to collude with A, but not with B, C and D. Which algorithm should be prohibited? E is better than A, so on efficiency grounds E should be approved and A prohibited. But A was approved at the outset, and it may be costly to outlaw it at a later stage (for example, firms may have sunk investments in technology A).

The difficulties of pursuing the first two approaches may suggest taking seriously the third one, *ex post* intervention. As discussed above, if algorithmic pricing indeed makes collusion easier, and, in particular, if it dispenses with the need for direct communication among the parties, then the likelihood that current policy may lead to false negatives may be significantly higher. If this is so, then the balance between explicit and tacit collusion which underlies current policy may have to be reconsidered.

This brings to the forefront the problem of detecting tacit collusion. Future research should therefore focus not only on the possibility that algorithmic pricing may facilitate collusion, but also on the possible new features that tacit collusion may exhibit under algorithmic pricing.

Much research remains to be done. For now, we simply do not know enough about algorithmic pricing to make definitive policy recommendations.

References

Bloembergen D., K. Tuyls, D. Hennes, and M. Kaisers. 2015. “Evolutionary dynamics of multi-agent learning: A survey.” Journal of Artificial Intelligence Research, 53:659–697.

Busoniu, Lucian, Robert Babuska, and Bart De Schutter. 2008.. "A comprehensive survey of multiagent reinforcement learning." IEEE Transactions on Systems, Man, And Cybernetics-Part C: Algorithmic pricingplications and Reviews, 38 (2)., 2008(2008).

Chen, L., A. Mislove, and C. Wilson, 2016, “An empirical analysis of algorithmic pricing on Amazon marketplace." Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2016.

Cooper, D. J. and KU Kühn. 2014. “Communication, Renegotiation, and the Scope for Collusion,” American Economic Journal: Microeconomics, 6(2): 247-78.

Cooper, Russell W., Douglas V. DeJong, Robert Forsythe, and Thomas W. Ross. 1990. “Selection Criteria in Coordination Games: Some Experimental Results.” The American Economic Review, 80(1): 218-33.

Crandall, J.W, M. Oudah, Tennom, F. Ishowo-Oloko, S. Abdallah, J. Bonnefon, M. Cebrian, A. Shariff, M.A. Goodrich, and I. Rahwan, “Cooperating with Machines”, Nature Communications, vol. 9, n. 233, 2018.

Dogan, I. and A. R. Guner, 2015, "A Reinforcement Learning Algorithmic pricingproach to Competitive Ordering and Pricing Problem," Expert Systems, 32, 39-47.

Doraszelski U. and A. Pakes (2007): “A Framework for Algorithmic pricingplied Dynamic Analysis in IO,” in Handbook of Industrial Organization, Vol. 3, ed. by M. Armstrong and R. H. Porter. Amsterdam: Elsevier Science, Chapter 4.

Ellison, G. and Ellison, S.F., 2018. Match quality, search, and the Internet market for used books (No. w24197). National Bureau of Economic Research.

Erev, I., Roth, A.E., 1998. Predicting how people play games: reinforcement learning in experimental games with unique, mixed strategy equilibria. American Economic Review 88, 848–881.

Ezrachi, A. and M. E. Stucke, 2015, "Artificial Intelligence and Collusion: When Computers Inhibit Competition," Oxford Legal Studies Research Paper No. 18/2015, University of Tennessee Legal Studies Research Paper No. 267.

Fudenberg D., and D. K. Levine, 2016, “Whither Game Theory? Towards a Theory of Learning in Games,” The Journal of Economic Perspectives, Vol. 30, No. 4: 151-169.

Harrington J. E. 2017, “Developing Competition Law for Collusion by Autonomous Agents,” working paper, The Wharton School, University of Pennsylvania.

Hu, Junling, and Michael P. Wellman. "Nash Q-learning for general-sum stochastic games." Journal of machine learning research 4.Nov (2003): 1039-1069.

Leibo, J.Z., Zambaldi, V., Lanctot, M., Marecki, J. and Graepel, T. (2017), “Multi-agent Reinforcement Learning in Sequential Social Dilemmas,” Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AA-MAS 2017), São Paulo, Brazil.

Lerer, A., and Peysakhovich, A. 2018 “Maintaining cooperation in complex social dilemmas using deep reinforcement learning.” arXiv preprint arXiv:1707.01068

Mehra, S. 2016, "Antitrust and the Robo-Seller: Competition in the Time of Algorithms", Minnesota Law Review 1323.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015; 518(7540):529–533. https://doi.org/10.1038/nature14236 PMID: 25719670

Roth, A.E., Erev, I., 1995. Learning in extensive form games: experimental data and simple dynamic models in the intermediate term. Games and Economic Behavior 8, 164–212.

Salcedo, B., 2015, “Pricing Algorithms and Tacit Collusion,” working paper Pennsylvania State University.

Sandholm, T.W., Crites, R.H., 1996. Multiagent reinforcement learning in the iterated prisoner’s dilemma. Biosystems 37, 147–166.

Sannikov, Y., and A. Skrzypacz (2007), "Impossibility of Collusion Under Imperfect Monitoring With Flexible Production," American Economic Review.

Sarin, R., Vahid, F., 2001. Predicting how people play games: a simple dynamic model of choice. Games and Economic Behavior 34, 104–122.

Sukhbaatar,S., A. Szlam, and R. Fergus. 2016 Learning multiagent communication with backpropagation. arXiv, preprint arXiv:1605.07736.

Sutton, R. S. and A.G. Barto, 1998, Reinforcement learning: An introduction, MIT press Cambridge.

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J., and Vicente, R. 2017, “Multiagent cooperation and competition with deep reinforcement learning,” PLoS ONE 12(4): e0172395.

Tesauro, G. and Je§J O. Kephart, 2002, "Pricing in Agent Economics Using Multi-Agent Q-Learning," Autonomous Agents and Multi-Agent Systems, 5, 289-304.

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

Waltman, L. and U. Kaymak, 2008, "Q-learning Agents in a Cournot Oligopoly Model," Journal of Economic Dynamics & Control, 32, 3275-3293.

Watkins C. J. C. H. and P. Dayan, (1992). \Q-learning." Machine Learning 8, 279, 292.

Wunder, M., Littman, M. L., & Babes, M. (2010). Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 1167–1174.

Xie, M. and J Chen, 2004, "Studies on Horizontal Competition Among Homogeneous Retailers through Agent-based Simulations," Journal of Systems Science and Systems Engineering, 13, 490-505.