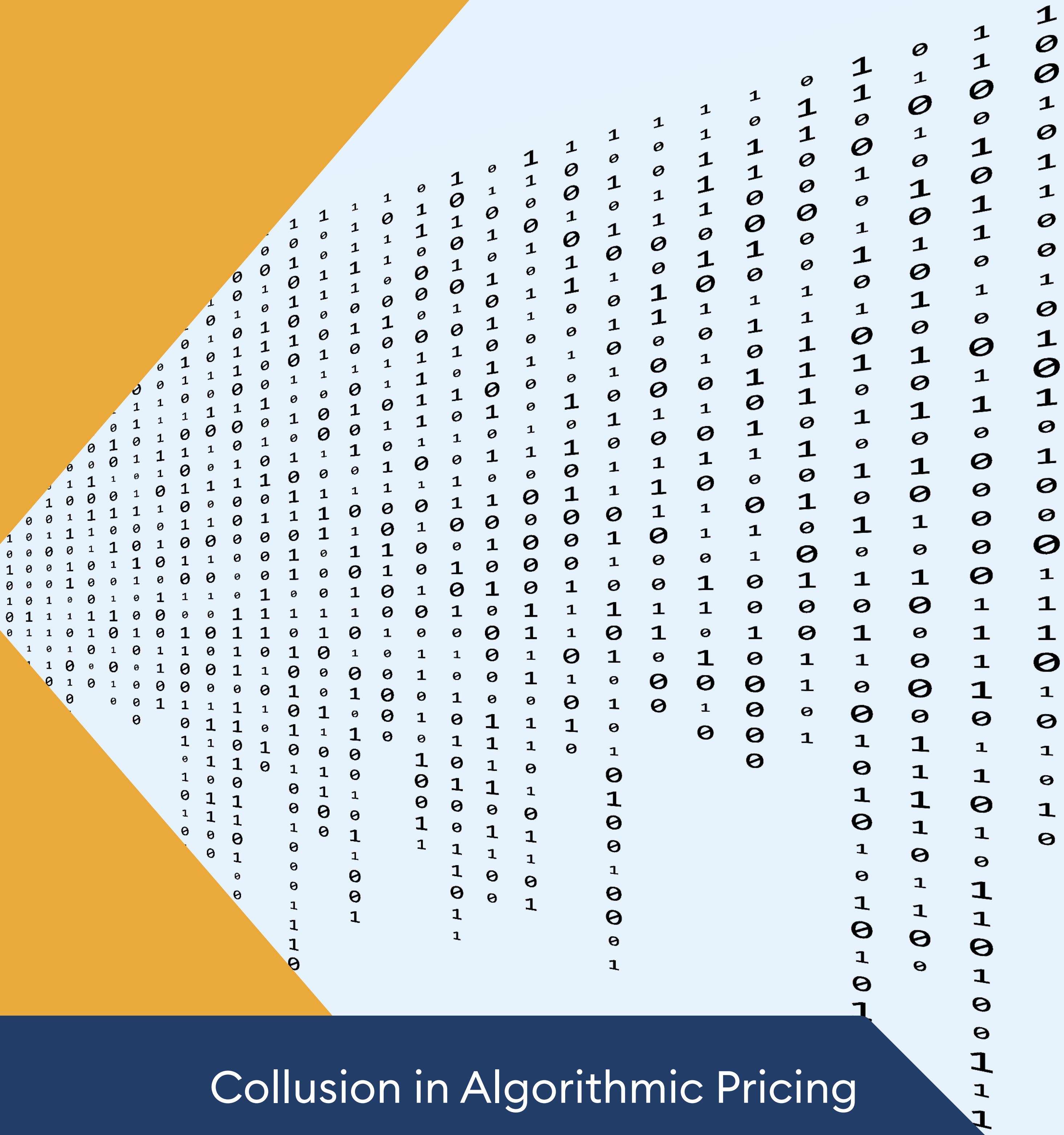
KONKURRENSVERKET
Swedish Competition Authority



Collusion in Algorithmic Pricing

Tuwe Löfström, Hilda Ralsmark, Ulf Johansson
på uppdrag av Konkurrensverket

UPPDRAGSFORSKNINGSRAPPORT 2021:3

Konkurrensverket november 2021
Utredare: Tuwe Löfström, Hilda Ralsmark, Ulf Johansson

Förord

I Konkurrensverkets uppdrag ingår att främja forskning på konkurrens- och upphandlingsområdet. Konkurrensverket har därför gett universitetslektor Tuwe Löfström och professor Ulf Johansson vid Tekniska högskolan i Jönköping i uppdrag att, inom ramen för Konkurrensverkets uppdragsforskning, undersöka om självständiga algoritmer kan lära sig att utveckla strategier som är liknande det vid prissamarbete. Medförfattare är ekon.dr. Hilda Ralsmark.

Digitaliseringen har radikalt förändrat mötesplatserna för köpare och säljare och medfört att konsumenter numera i högre grad jämför priser och produkter samt köper produkterna online. På samma sätt som digitaliseringen har förändrat mötesplatserna för både köpare och säljare, har den tekniska utvecklingen förändrat både köp- och handelsmönster. Många företag som säljer via e-handelsplattformar och egna webbplatser använder sig i dag av algoritmer som styr prissättningen. Beroende på hur algoritmen är programmerad att agera kan den påverka konkurrensen antingen positivt eller negativt. En möjlig konkurrenshämmande effekt är att algoritmer som möter varandra lär sig sätta priser som är högre än vid normal konkurrens.

Utifrån ett konkurrens- och tillsynsperspektiv har den här rapporten gett en ökad kunskap och förståelse om hur självständiga algoritmer påverkar priskonkurrensen. Tidigare studier och teser har pekat på att användandet av självständiga algoritmer kan leda till högre priser och vinster för företag. Författarna i denna studie pekar i sina resultat på att detta även kan ske när företag använder olika typer av självständiga algoritmer eller algoritmer vars inlärning sker vid olika tillfällen. Dessutom kan företag uppnå konkurrensfördelar genom att förbättra sina algoritmer. Slutligen visar resultaten även att hotet att nya företag ger sig in på marknaden hämmar prisutvecklingen och leder till lägre prisnivåer.

En referensgrupp har deltagit i projektet. I gruppen ingick Andrea Enache (Handelshögskolan i Stockholm), Tony Lindgren (Stockholms universitet) och Andreas Stephan (Jönköping International Business School). Från Konkurrensverket har Björn Axelsson, Arvid Fredenberg, Stefan Jönsson, David Nordström och Joakim Wallenklint deltagit.

Författarna ansvarar själva för alla bedömningar och slutsatser i rapporten.

Stockholm, november 2021

Rikard Jermsten
Generaldirektör

Innehåll

Sammanfattning

De senaste årens teknologiska utveckling har möjliggjort för autonoma agenter att utnyttja artificiell intelligens för att lära sig optimerade prispolicyer genom att interagera med marknaden. Både konkurrensmyndigheter och forskare har uttryckt oro för att autonoma prisoptimerande agenter som oberoende av varandra agerar på samma marknad kan lära sig varandras policyer genom den implicita interaktionen. Deras oro är att agenterna på så sätt når ett prisutfall som liknar regelrätt prissamarbete.

Den här rapporten avser att: förklara hur algoritmer som används för automatiserad prisoptimering fungerar; presentera en litteraturgenomgång kring automatiserad prisoptimering och prissamarbete; genomföra och presentera resultaten från en samling av experiment som utvärderar flera tidigare outforskade aspekter kring självlärande och autonoma prisoptimerande agenter; och, slutligen, presentera slutsatser som dragits utifrån litteraturgenomgången och de genomförda experimenten.

De genomförda empiriska experimenten kompletterar befintlig forskning genom att utvärdera agenter som är olika varandra eller som agerar asynkront, vilket efterliknar den situation som råder när inget explicit samarbete existerar.

Våra resultat tillsammans med tidigare forskning implicerar att prisoptimerande agenter kan förväntas nå prisutfall i närheten av dem som råder vid regelrätt prissamarbete, oavsett hur starka agenter som används. Vidare så implicerar den empiriska undersökningen att företag kan uppnå konkurrensfördelar genom att kontinuerligt förbättra sina prisoptimerande agenter. Slutligen så visar våra resultat även att hotet att nya aktörer ger sig in på marknaden hämmar prisutvecklingen och resulterar i reducerade prisnivåer.

Summary

In recent years, technological developments based on artificial intelligence have allowed autonomous agents to interact with the market and learn optimised price policies through experience. Both competition authorities and researchers have raised concerns that autonomous price optimising agents operating independently of each other in the same market may learn each other's policies from their implicit interaction through the shared market. They fear that it will result in collusive outcomes with prices above competitive price levels.

This report aims to: explain how algorithms suitable for algorithmic pricing work; present a literature review of algorithmic pricing and collusion; perform and present the results from a comprehensive set of experiments investigating several novel aspects of self-learning and autonomous price-setting agents; and, finally, to present conclusions drawn from the literature review and the experimental study.

The empirical experiments complement existing research by evaluating dissimilar or asynchronous agents, thus mimicking the situation when no explicit collaboration exists.

Our empirical results, together with previous work, imply that price optimising agents can be expected to reach collusive outcomes no matter how strong the agents are. The empirical investigation further implies that firms are incentivised to compete through continuous improvement of their price optimising agents since firms gaining a competitive edge can increase their share of the profit. Lastly, we also find that the threat of entry by another agent appears to have a disciplinary effect and lowers the collusive outcome.

1 Introduction

Technical development has dramatically changed how sellers and consumers interact. When consumers plan to buy a new product, they often compare prices and services across multiple stores and platforms to find the best offer. Pricing is a crucial success factor in many domains, while at the same time, the task of optimising prices to gain a competitive edge has become increasingly difficult. Today, many firms use algorithms to automatically set prices to help them stay competitive, often referred to as algorithmic pricing. Many platforms also offer solutions for price automation to the sellers listed on the platform. The algorithms' knowledge base can easily incorporate price information about competitors due to the price transparency online and price changes being updated almost instantly on websites. A study showed that nearly one-third of Amazon's 1641 best-selling products in 2015 used some algorithmic price-setting strategy.[1] The European Commission wrote in 2017 that "[a] majority of retailers track the online prices of competitors. Two-thirds of them use software programs that autonomously adjust their prices based on the observed prices of competitors".[2]

The term algorithmic pricing refers to the use of algorithms performing autonomous price adjustments. It was initially introduced in the 1980s when hotels and airline firms began setting prices mechanically based on season, location etc. Consequently, the term algorithmic pricing does not connote any reference to intelligent algorithms in the sense that they are self-adapting. In recent years, however, this has changed dramatically due to several factors. Two necessary infrastructural factors are computational power and digitalisation. The development within these fields has provided firms with easy and rich access to data suitable for analysis. In parallel, research in artificial intelligence and machine learning has produced an extensive toolbox of algorithms capable of learning autonomously from experience. Contrary to the first strand of algorithms, the new algorithms are not hardwired to behave in explicit ways, but will learn strategies for optimising a pre-decided outcome (such as profit) autonomously by actively experimenting and adapting to changes in their environment.

Businesses' increased use of big data and algorithms to make well-informed strategic decisions has undoubtedly come with great benefits to both consumers and firms. For example, the increased price transparency through price comparison sites that often use algorithms makes it easier for customers to make well-informed decisions based on price and product characteristics, while lowering switching costs. It also makes firms more innovative and efficient. However, these benefits of increased data, transparency and AI is sometimes described as a double-edged sword, due to an increased risk that market competition is distorted. This concern does not only apply to markets that are usually considered to be prone to collusion, such as concentrated markets with homogenous

[1] Le Chen, Alan Mislove, and Christo Wilson. 2016. An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace. In Proceedings of the 25th International Conference on World Wide Web (WWW '16), International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1339–1349. DOI:https://doi.org/10.1145/2872427.2883089

[2] COM (2017) Report from the Commission to the council and the European Parliament. Final report on the E-commerce Sector Inquiry.

products and high barriers to entry, but also extends to markets that do not display these characteristics.[3]

Well-functioning markets are crucial to ensure that customers get the best possible products and services at the lowest possible prices. The European Commission for Competition, national competition authorities, organisations such as the OECD, and scholars are giving more attention to the question whether autonomous price-setting algorithms, learning from experience, will lead to prices above the competitive level.[4] A number of reports have been published on the subject of price algorithms, such as those by the OECD[5], the UK Competition and Markets Authority (CMA)[6] and a joint report by the French and German competition authorities.[7] A key concern often discussed in these reports is the potential limitations of Article 101 TFEU and national competition laws in addressing this outcome. These laws prohibit, amongst other things,

> *"all agreements between undertakings, decisions by associations of undertakings and concerted practices which may affect trade between Member States and which have as their object or effect the prevention, restriction or distortion of competition within the internal market."*

Thus, Article 101 TFEU and national legal counterparts only forbid agreements and concerted practices. Consequently, firms can independently adjust their behaviour to competitors' current or anticipated future behaviour without violating the competition law, as long as no communication between undertakings has occurred. In other words, price algorithms that lead to collusive outcomes and have serious adverse effects on consumer welfare may not be unlawful as long as colluding intent or communication is absent.

[3]Antonio Capobianco, *Digital cartels & algoritms*, ICN Webinar – 16 January, 2019. URL: https://ec.europa.eu/competition/cartels/icn/capobianco.pdf

[4] The competitive level is the prices that would be if there were no price algorithms in use.

[5] OECD (2017), *Algorithms and Collusion: Competition Policy in the Digital Age* www.oecd.org/competition/algorithms-collusion-competition-policy-in-the-digital-age.htm

[6] Competition and Markets Authority. 2021 *Algorithms: How they can reduce competition and harm consumers*. URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/954331/Algorithms_++.pdf

[7] French Autorité de la concurrence and the German Bundeskartellamt (2019). *Algorithms and competition*. URL: https://www.bundeskartellamt.de/SharedDocs/Publikation/EN/Berichte/Algorithms_and_Competition_Working-Paper.pdf?__blob=publicationFile&v=5

Ezrachi and Stucke[8] were among the first to point out that algorithms may learn to collude. The authors write that AI "can expand tacit collusion beyond price, beyond oligopolistic markets, and beyond easy detection." The authors distinguish between two scenarios: the predictable agent scenario, where "*humans unilaterally design the machine to deliver predictable outcomes and react in a given way to changing market conditions*"; and the *digital eye scenario*, where an algorithm is given a goal such as profit maximisation, and the algorithm can act autonomously to achieve that goal. These two scenarios differ significantly regarding the responsibility that the person designing the algorithm has on its outcome. In the digital eye scenario, "*tacit coordination – when executed – is not the fruit of explicit human design but rather the outcome of evolution, self-learning, and independent machine execution*", according to Ezrachi and Stucke.

Gal also raises such concerns in her discussion about tacit collusion by algorithms without the need for agreement.[9] Ballad and Naik argue that 'Joint conduct by robots is likely to be *different*—harder to detect, more effective, more stable and persistent.'[10] In a 2017 speech, then European Commissioner for Competition, Margrethe Vestager, highlighted the potential risks to consumers with increased use of more or less sophisticated algorithms.[11] A 2017 background paper by Organization for

Economic Cooperation and Development (OECD) brought up whether algorithms can make tacit collusion easier not only in an oligopolistic market but also in markets that do not have the structural features that one would typically associate with an increased risk of collusion. António Gomes, Head of the OECD, has stated in an interview that AI and machine learning that enable algorithms to achieve a collusive outcome efficiently is:

> *"the most complex and subtle way for companies to collude, without explicitly programming algorithms to do so."*[12]

Harrington points out that collusive outcomes that arise from producers that apply autonomous learning algorithms are not unlawful and that a case based on Article 101 TFEU will be challenging to win in court with this type of tacit collusion.[13] To tackle this, one suggestion is to bring a case, based on the algorithms' owners' negligence to understand what their algorithm was doing, to court. However, with the black-box nature of sophisticated AI models, the owners cannot understand the mechanism behind its decisions. Thus, it will not be easy to hold them accountable.[14]

[8] Ezrachi, A. and Stucke, M.E. (2016), *Virtual Competition: The Promise and Perils of the Algorithm-Driven Economy*, Harvard University Press.

[9] Michal S. Gal (2019) *Algorithms as illegal agreements*, Berkeley technology law journal. Vol 34, pp. 67-118.

[10] Ballard, D.I and Amar S. Naik. *Algorithms, artificial intelligence, and joint conduct*. Competition Policy International, CPI Antitrust Chronicle May 2017.

[11] Margrethe Vestager (2017), European Commissioner for Competition. https://wayback.archive-it.org/12090/20191129221651/https://ec.europa.eu/commission/commissioners/2014-2019/vestager/announcements/bundeskartellamt-18th-conference-competition-berlin-16-march-2017_en

[12] CPI Talks, Interview with Antonio Gomes of the OECD, Antitrust Chronicles. (May 2017).

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

[14] Yavar Bathaee (2018), *The artificial intelligence black box and the failure of intent and causation*, Harvard Journal of Law & Technology Vol. 31(2).

Whether or not the concern expressed by scholars as well as national and international competition authorities is valid is still debatable. As the OECD background paper conclude, it is not yet clear to which degree collusion among algorithms is likely to happen. In 2018, the Competition Bureau of Canada pointed out the lack of evidence of autonomous algorithmic collusion.[15] Further, several researchers and professionals have stated that this worry is severely overstated and represents a dystopian view close to science fiction. Petit wrote that:

> *"AAI [Antitrust and Artificial Intelligence] literature is the closest ever our field came to science-fiction."*[16]

Along the same lines, Mehra reflects that:

> *"The possibility of enhanced tacit collusion . . . remains theoretical."*[17]

Reasons for this reluctance a few years back include that most empirical work showing the potential for algorithmic collusion thus far had been done with two players, a limited number of strategies available to the algorithms and a static market environment. This type of setup is of course not a realistic model of the real-world markets. In addition, since the markets that algorithms operate in are very complex, it is also challenging to prove that the algorithms cause the price level to be higher than the competitive level.

Lastly, it is not known how widespread the use of different algorithms is, as firms do not disclose this information publicly. Recent surveys from national competition authorities are starting to fill this knowledge gap, however. A recent survey from Norway shows that more and more firms are using monitoring and pricing algorithms. Fifty-five per cent of the surveyed firms use so-called surveying algorithms, and twenty per cent of the surveyed firms use so-called pricing algorithms, which automatically change prices.[18] Most interestingly, for this report, only a limited number of the surveyed firms state that they use self-learning algorithms. Other examples of market surveys on the use of algorithms include, but are not limited to,

- the European Commission, who in a 2017 report found that around one third used monitoring algorithms.[19]

[15] Competition Bureau of Canada (2017) *Big Data and Innovation: Key Themes for Competition Policy in Canada* http://www.competitionbureau.gc.ca/eic/site/cb-bc.

[16] Nicolas Petit (2017) *Antitrust and Artificial Intelligence: A Research Agenda*, 8 J. Eur. Competition L. & Pract. 361, 361–362.

[17] Salil K. Mehra (May 2017) *Robo-Seller Prosecutions and Antitrust's Error-Cost Frame - work*, CPI Antitrust Chronicles. 37.

[18] The Norwegian Competition Authority's market survey on the use of monitoring and pricing algorithms. Report 2021, *What effect can algorithms have on competition?*

[19] EU-Commission (2017) "Final report on the e-commerce Sector Inquiry - Accompanying Staff Working Document", Page 175.

- the Portuguese Competition Authority, who in a 2019 report found that 37 percent use monitoring algorithms.[20]

More recent economics research has attempted to answer whether algorithms can learn to collude by turning to computer-simulated markets. Calvano et al. show in a seminal paper that price-setting algorithms based on machine learning indeed reach outcomes above the competitive level, without being programmed to collude or communicate. In the investigated setting, the algorithms also develop collusive strategies that include temporary punishments of defections.[21] The results of Calvano et al. thus highlight the potential limitations of the current legislation of competition law as well as the toolbox of competition commissions by bringing the focus beyond conceptual concerns.[22]

These recent developments have intensified the debate among antitrust agencies on whether the current legislation and antitrust tools are still applicable, or if there is a need for alterations or even the introduction of new legislation and tools to tackle algorithmic collusion. There is also a discussion on whether algorithms should be regulated and the effect of this on innovation.[23]

The question for national and international competition authorities is no longer whether algorithms may learn to collude or if this is "science fiction". Instead, the question has evolved into in which settings they may learn to collude. Consequently, recent research makes the settings more realistic to the real-life environment in which firms set prices today. The empirical investigation performed in this report is following that same principle. By making the agents and environments more realistic, we hope to bring new evidence regarding the potential for algorithms to collude, and the extent of the resulting harm to competition and, ultimately, consumers.

1.1 Purpose

The purpose of this report is consequently to

- Present and explain how algorithms suitable for algorithmic pricing work
- Present a literature review of algorithmic pricing and collusion
- Perform and present the results from a comprehensive set of experiments investigating several novel aspects of self-learning and autonomous price-setting agents. Specifically, this report investigates the pricing policies developed and the resulting price levels
- Present conclusions drawn from the literature review and the experimental study

[20] Autoridade da Concorrência (2019), "Digital ecosystems, Big Data and Algorithms - issues paper", page 44.

[21] Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello (2020). *Artificial Intelligence, Algorithmic Pricing, and Collusion*. American Economic Review, 110 (10): 3267-97.

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

[23] Algorithms, O. E. C. D. (2017) *Collusion: Competition Policy in the Digital Age*.

In particular, our main research interest is how the risk of tacit collusion changes based on an aspect not previously investigated; namely the use of agents that differ. We have also investigated how the threat of additional agents entering the market affects the results.

The main experiments deal with agents that are different from each other.

Experiment 1: There are agents with different algorithms and settings in the same market.

Experiment 2: The agents in the market observe and update their prices with different and varying intervals.

Motivation for experiment 1 and 2: The digital sector consists of tech giants such as Google and Amazon, who have developed very sophisticated algorithms to analyse everything from consumer behaviour to prices, and smaller businesses that have not nearly the capacity to develop or even use similar algorithms. There will be businesses with varying degrees of sophistication in almost all markets when it comes to algorithms. Two important questions related to this are how the market is affected in general, but also if certain types of agents are able to outperform others. Specifically, we investigate whether a mix of agent types on the market will prevent collusive outcomes, creating instead a competition between the agents. To answer these questions we test whether the composition of algorithms matters. If it does, there may in fact be less need to worry about collusive outcomes in markets with a mix of big and small actors, or where market entry is easy and requires a low level of R&D to develop solutions based on machine learning. In other words, markets that resemble competitive markets.

In order to investigate how the threat of additional agents entering the market, as profits rise, affect agents price policies, we also ran an additional experiment.

Experiment 3: The algorithms' goal – profit maximisation – is adjusted to take into account the threat of the entry/exit of new competing agents in the market.

Motivation for experiment 3: Economic theory suggest that firms in competitive markets can only sustain an abnormal profit in the short run. The reason is that higher profits in a market will lead to market entry by additional firms that want to reap some of the abnormal profit. Entry barrier models of imperfect competition argue that actual entry affects competition. In contrast, models of imperfect competition with contestable markets argue that the threat of entry is enough to curb market power. Several theories lie in between these two extremes regarding the role of potential and actual entry in the market.[24]

[24] Bresnahan, Timothy F., and Peter C. Reiss. *Entry and Competition in Concentrated Markets*, Journal of Political Economy, vol. 99, no. 5, 1991, pp. 977–1009. *JSTOR*, www.jstor.org/stable/2937655. Accessed 22 June 2020.

1.2 Outline

The report has the following outline. The next section presents an overview of algorithms used for learning. Section 3 introduces collusion, including a discussion about economic theory. The section also includes a subsection on game theory and possible methods for detecting collusion. A literature review is presented in section 4, whereas section 5 presents the empirical investigation, including experimental setup and results. Section 6, finally, contains the concluding discussion.

2 Techniques used for learning

Machine learning is the term generally used when talking about algorithms used for learning. Machine learning can broadly be divided into algorithms for supervised learning, unsupervised learning and reinforcement learning. In supervised learning, historical data is used to train a model, which is eventually used to predict new instances. In reality, the model is actually a function from a number of input attribute values to one specific output attribute value, called the target. If the target is restricted to a set of pre-defined discrete values (“labels”), the task is classification, if the target is continuous, it is called regression. The learning of the relationship between input and output values is called supervised since the algorithm uses examples, consisting of both input and target values, to minimise the errors made by the model. Unsupervised algorithms, on the other hand, are not provided with target values but try to find and highlight naturally occurring patterns or groups in the data. Whether the patterns found are valuable remains for an end-user to decide. Reinforcement learning, finally, is finding and optimising a strategy (called a policy) for solving a specific task through interaction with the environment.

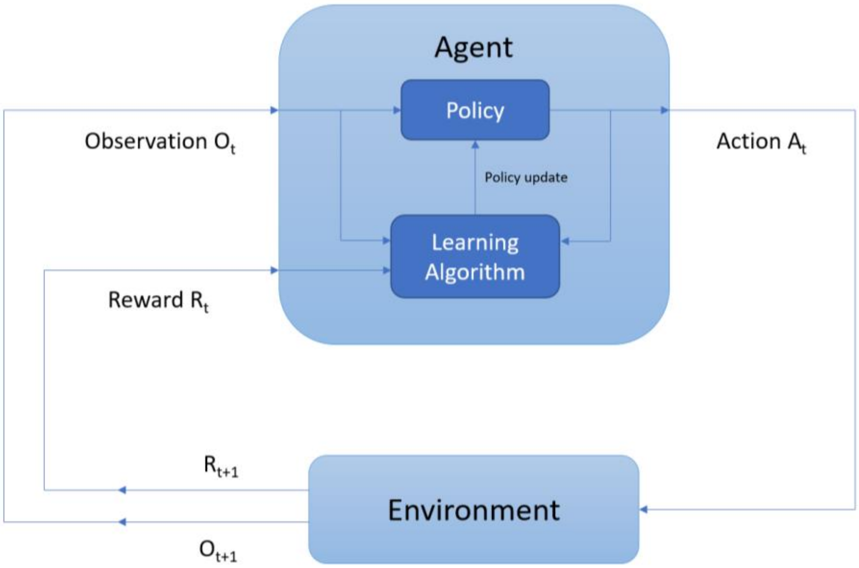
2.1 Reinforcement Learning

The two core components in reinforcement learning (RL) are the *agent* and the *environment*. The agent has an assigned task that it tries to learn to solve in the environment, normally by repeatedly performing the task. The environment consists of *states*, and the agent takes *actions* to fulfil the task, where an action results either in a move from one state to another, or that the agent remains in the same state. The *policy* is a description of which action to take for every possible state. For the agent to learn anything from the environment, it receives *rewards* from the environment and uses this information to improve the policy.

The properties of the environment determine which RL-agents that can be used. While a comprehensive description of the different environments is outside the scope of this text, the environment later used in the experimentation can be described as having a *fully observable* and *discrete* state space, while simulating a *continuing* task. Starting with the fully observable discrete state space, this means that the environment consists of a limited number of states, and that the agent always knows exactly which state it is in, i.e., it doesn’t need to learn the environment. The fact that the task is continuing means that there is no natural end to the interaction, instead the agent will continue to interact with the environment “for ever”.

This interaction between an agent and the environment is depicted in figure 1.

Figure 1. The agent-environment interaction with agent building blocks



Here, the agent and the environment interact in a sequence of time steps $t = 0,1,2,3, \dots$ [25]. In each step, the agent performs an action, based on the current policy and the current state. The action moves the agent to a new state, which is observed, and the agent may or may not receive a reward. In fully observable environments, the observation is simply which state the agent is in, while the reward is a numeric measure of how successful the action is, with respect to finishing the task. The overall goal for reinforcement learning is to find a policy that maximises the total reward for the agent's interaction with the environment. Consequently, if the rewards in the environment correlate well with the ability to solve the task, the agent will improve, just by this trial-and-error interaction with the environment.

Internally, the agent consists of two components: the *policy* and the *learning algorithm*. As described above, the policy determines which action to take by the agent in any possible situation. The learning algorithm continuously updates the policy based on actions, observations, and rewards. The goal of the learning algorithm is to find the optimal policy, i.e., the policy that results in the highest cumulative rewards for the agent when trying to solve the task.

In order for an agent to learn not just what is best in the short term, but also what is most beneficial in the long run, the agent needs to keep track of how good certain states and actions are. This can be done in some slightly different ways. The first alternative is to simply record the *value* of each state, V(s), which is an estimation of the long-term return from that state. If all states have correct values, finding an optimal policy is of course trivial, it will just need a one-step look-ahead from each state picking the action that results in the highest sum of the immediate reward and the value of the next state. Since we are targeting a continuing task, the value of a state is actually a discounted sum of future rewards, thus prioritising immediate returns over more future. So, contrasting rewards and value, the reward is the immediate signal received in a given state, whereas

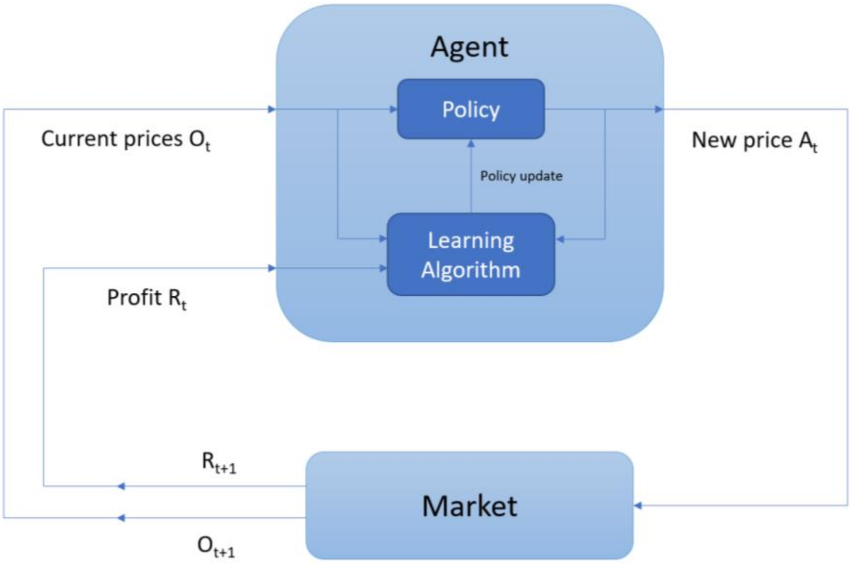
[25] It is worth mentioning that interaction can also be continuous, but we will only consider discrete time steps.

the value summarises all rewards you can anticipate from that state, if following a certain policy. Instead of just state values V(S), the *Q-value*, often written as Q(s, a) can be used. The only difference is that Q-values are recorded for state-action pairs instead of just states, i.e., Q(s, a) for the state action pair (s, a) is the estimated long-term return of performing the specific action a in the state s, and then following a certain policy from the resulting next state. If the estimated Q-values are correct, finding the optimal policy is again trivial, it is just to pick the action with the highest Q(s, a) in each state. Consequently, the overall goal of many RL-algorithms is actually to learn either V(s) or Q(s, a).

Exactly how and when the agent updates state values or state-action values, and how the policy is improved, varies between different learning algorithms. Often, however, the values are actually updated for every step taken. We describe this in some more detail for the algorithms used in the study in section 2.1.1.

In our case, where we investigate agents' capability to learn tacit collusion, the environment encapsulates the market. A state (or observation) would, in this context, typically be the current prices of all active competitors, whose price information is freely available. Making an action would be to set a price, and the reward corresponds to the immediate profit gained by using that price. A state value V(s) would be the cumulative profits from a specific state, and Q(s, a) the cumulative profits from taking the action a (setting a specific new price) in that state. Obviously, a policy would, in this scenario, dictate exactly which action (price level) to use in every possible situation (state), i.e., the current price levels of all competitors. See figure 2 for an illustration.

Figure 2. The agent-environment interaction in the price optimisation context



2.1.1 Algorithms

There are many different learning algorithms for reinforcement learning. Providing a detailed description of these are outside the scope of this report, but we will present some of the core building blocks, while giving an overview of the algorithms used in the report.

Initially, in a RL setting, the environment is often unknown to the agent, in the sense that the agent doesn't know the rewards associated with states and actions. The agent therefore alternates between *exploitation*, in which it pursues the currently best policy, and *exploration*, in which it selects a random action to explore the environment in order to find new and better opportunities to improve the policy and solve the task. So, when the agent is exploiting, it will use the action dictated by the policy, while it may deviate from the policy when exploring. While exploration is costly, as it is suboptimal, it is still necessary since it enables the agent to explore the environment in order to improve the policy. This ability to mix exploitation and exploration is a trademark of RL, not used by other learning approaches.

For the agent to find the best action to perform in each state, it must ensure that all state-action pairs are visited. This is normally accomplished by using what is called an ε-greedy policy, meaning that the agent follows the policy (exploits) with probability 1- ε, and picks a random action (explores) with probability ε. Normally, ε is continuously (but very slowly) lowered, making the agent exploit more and more.

In the experimentation, we use two fundamental RL learning algorithms called Q-learning and SARSA. These two algorithms are, in fact, very similar, and they belong to the family of action-value methods; i.e., they learn, as described above, the value of different actions, and select actions based on these estimates. More specifically, both Q-learning and SARSA use Q-values as their representation of the environment, and they update the estimated value for Q(s, a) in every step. Actually, this update is also performed almost identically by the two algorithms. Since both algorithms are based on what is called *temporal-difference learning* (TD-learning) the overall idea is that the value of the current state (or state-action pair) should be equal to the reward given when moving to the next state, plus the value of the next state. While this may sound obvious, it is in fact this very local relationship that is utilised by TD-learning, making it possible to solve RL-problems, i.e., finding globally optimal policies, in reasonable times. Especially in episodic tasks, i.e., where an interaction is finished when the agent reaches a goal state, the agent first learns the values of states close to the goal state, and this knowledge is then propagated, using the TD-learning principle, to states further from the goal states.

SARSA is an *on-policy* method, meaning that it evaluates and improves on the actual policy used by the agent. If the agent has at the time step *t* performed action *a* in state *s*, which has lead to a reward *r* and the agent ending up in a new state (in time step t+1) the update performed is:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

where α is a parameter called the learning rate and γ is the discount factor. Looking first at the expression inside the square brackets, it is simply the immediate reward given for performing action *a* in state *s*, plus the difference between the (discounted) estimated value of following the policy from the resulting state in the next time step, and the current estimation of taking action *a* in state *s*. The result is multiplied with the learning rate, and added to the current estimation of *Q(s, a)*, resulting in the updated estimation of *Q(s, a)*. Here, we would like to point out how natural this update rule really is. It simply says that we first estimate the value of taking action *a* in the current state *s* as the immediate reward plus the value of following the policy from the resulting state, and then compare this to our current estimation of *Q(s, a)*, resulting in an updated estimation.

Q-learning, on the other hand is said to be an *off-policy* method, i.e., it can in principle use any policy to estimate *Q*. In practice, however, both SARSA and Q-learning normally use an ε-greedy policy, i.e., take what they consider to be the best action (exploit) a majority of the time, but pick a random action (explore) sometimes. If this is the case, the only minor difference between the algorithms is how the update is performed. In Q-learning, when estimating the value of the next state, it is always assumed to take the best action from there, as expressed by the *Q(s, a)* values, while SARSA, as described above, follows the policy and uses the resulting value $Q(s_{t+1}, a_{t+1})$, even if that move was exploratory. The update function for Q-learning is consequently:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

If the observation space, i.e., the set of possible states is small, the Q-values used by Q-learning and SARSA are normally simply stored in a lookup-table. For larger state spaces, however, some kind of function approximator has to be used instead. The consequence is that a number of states become indistinguishable to the agent, meaning that they are treated as identical. Typical function approximators are machine learning models, like (deep) neural networks, that are trained to fit the environment and task.

The third learning algorithm used in the experimentation is a policy gradient (PG) method meaning it directly optimise the policy in action space. Here, a value function can still be used to learn the policy, but it is not required for the action selection. While a detailed description of PG-methods is outside the scope of this report, the main idea is that each possible action has a probability associated with a representation of the state. After learning, the final policy will then be an optimal *stochastic* policy. This means that while the final policy could actually be deterministic for a certain state (have probability 1.0 for one action) it could also end up with a probability distribution over the different possible actions. This is in sharp contrast to the action-value methods, where the policy always consists of exactly one action for each state. Here it must be noted that for some problems, like playing games with imperfect information (e.g., Poker) optimal policies in different situations *are* stochastic. For such problems, the action-value methods have no natural way of finding these optimal stochastic policies, while they are directly approximated by the PG-variants.

Another, quite straightforward advantage for PG-variants is that the policy may actually be a simpler function to approximate than the state-value.

Finally, it may be noted that PG-variants have another theoretical advantage over the Ɛ-greedy methods. In PG, the action probabilities change smoothly, but in Ɛ-greedy methods, the change is very abrupt; from one action to another. Because of this "continuity" there are actually stronger convergence guarantees for the PG-methods.

2.1.2 Single-Agent Reinforcement Learning

From the perspective of the price-competing firm, the price optimising agent will be implemented as a single-agent solution, where the environment includes competitors' prices. Implementation of a single-agent RL for price optimisation must take several aspects into account. The firm needs to select the algorithm to use, and its specific learning parameters, what data to collect about the environment, how frequently it should collect information about the environment and how often its own prices should be allowed to change. The data collected about the environment will generally include data about competitor prices but can also be enriched with all kinds of additional information. In the domain of algorithmic trading, where reinforcement learning is often used to trade on the financial market, it is common to include, e.g., market data, news, sentiment, and economic indicators in the data collected from the environment.

For firms that does not have any explicit communication about its implementation details, it is extremely unlikely that two firms would develop solutions with identical behaviour. However, since many firms rely on external consultants and software vendors for this kind of solutions, two firms selecting the same software supplier and using the default settings, could end up using identical agents if they were to include the same description of the environment.

2.1.3 Multi-Agent Reinforcement Learning

This report is investigating what can happen when multiple firms, independently of each other, use price-optimising agents to set their prices. Consequently, the experiments in our study could not rely on single-agent reinforcement learning. When using multiple agents, there are several different scenarios that RL can be used for.

Multiple agents can co-operate in order to solve a shared problem. In these situations, the agents can share information freely among each other, including learnt policies.

Another scenario is when multiple agents team up against each other, where the goal is for one team to win over the other team. This scenario could have many different variations. However, in a similar way as the situation where all agents co-operate, the agents within each team are free to share information and learnt policies.

Finally, all agents can also compete against each other or simply be unaware of other active agents. In this situation, the only shared information is the environment, which, in its simplest form, is perceived in the same way by all agents. The agents will not share any information in this scenario. This is the scenario which can be used to emulate a situation where different firms, independently from each other, optimise their prices in a shared market.

There are a few aspects which distinguish reality, where firms use single-agent RL to optimise their own prices, with multi-agent RL emulating simultaneous optimisation of multiple single-agents used by multiple firms. First of all, in a multi-agent emulation of the real situation, all agents will normally perceive the environment in exactly the same way. Secondly, all agents are normally beginning to explore the environment simultaneously without any prior knowledge. Thirdly, in the multi-agent emulation, agents act, collect information and learn at the same time, and with the same frequency. Fourthly, all agents in the multi-agent emulation will generally use the same or very similar reward functions, basically optimising policies targeting the same goal. Each of these aspects are, in themselves, less likely to occur in reality. Specifically, that all these aspects should be in play simultaneously in reality with independent firms is highly unlikely. We will address some of these observations in our experiments, but will leave some of them as future work for others to study further.

3 Collusion

Collusion can be explicit, tacit or a combination of the two. Explicit collusion is the coordinated behaviour by firms for the purpose of obtaining a price level that is higher than would occur in equilibrium if the firms would compete. Such a formal and organised scheme is often called a cartel. It often involves a reward-punishment scheme. Tacit collusion is a market conduct that also enables firms to obtain a higher price level than would otherwise occur in equilibrium with competition, but it does not involve an explicit exchange of information or agreements between firms to raise prices. Tacit collusion also often involves a reward-punishment scheme. Tacit collusion is not illegal as there is no explicit agreement, but still harmful in practice as it affects the consumers negatively.

The model used in this report is the traditional Bertrand model of oligopoly competition where firms compete on prices. In the standard model, goods are assumed to be homogenous, firms set prices simultaneously, costs are symmetric and there is perfect information (zero search costs). If prices are identical, consumers split their demand evenly between the firms. Price competition between the firms will lead to prices falling to marginal cost and a normal profit. This is the only Nash equilibrium in the Bertrand model.

The Bertrand model has several strong assumptions that may not hold in reality for many markets. For example, price may not be the main factor on which firms compete, products can be differentiated, customers may not always choose to buy from the firm with the lowest price since they may incur costs from searching for the lowest price, and firms can have asymmetric costs. When some of these assumptions are relaxed, the results of the model may no longer hold. For example, when products are differentiated the result that price equals marginal costs no longer holds. Whether customers will actively search for the lowest priced product also most likely depends on whether the product is relatively cheap or expensive.

Given the assumptions behind the model used in this report, the results more likely hold for specific types of markets and not for others. For example, it is more likely to hold for electronics and appliances where there are a number of retailers selling the same product, products can be expensive and seldom purchased, and it is commonplace for customers to use price comparison services to find the lowest price. On the other hand, it is less likely to hold for fashion items or interior design products where each retailer may exclusively sell their own brand and price is not the most important factor on which competition occurs.

3.1 How to identify collusion

3.1.1 The traditional approach

Traditionally, the work of competition authorities when it comes to identifying collusion essentially means the work of identifying cartels. The reason for this is that explicit collusion is illegal, whereas tacit collusion is not.

Cartels undermine competition which harms consumers and, in the long run, destroys the competitive process that leads to innovation and efficiencies. It is therefore natural that cartel detection is one of the most important areas of activity of competition authorities and a priority of the European Commission.[26]

The gains from a cartel can be very large. Connor finds that median increase in price attributable to collusion is around 25%.[27] An OECD report finds that the average increase in the selling price is 10% and reduction in output as high as 20%.[28] It is therefore important that competition enforcement makes the expected punishment higher than the expected profit gain from collusion.

The work of competition authorities when it comes to combating cartels can be divided into cartel detection and punishment, and a deterrence effect that prevents cartels from arising or continuing.

When it comes to the detection and punishment, Harrington classifies cartel detection into two broad categories: structural and behavioural. [29] A structural approach means identifying markets that have traits that are normally considered conducive to collusion, such as few firms, homogenous products and more stable demand and market conditions. Steel and cement industries are examples of two such sectors. A behavioural approach means either observing the ways in which firms coordinate, such as some form of direct communication, or observing the end result of that coordination, such as suspicious patterns of factors such as prices or quantities. Harrington argues that the risk of false positives, where firms are incorrectly classified as a cartel, is higher with the structural approach than the behavioural approach.

Cartel detection work at competition authorities often starts with a process of *screening*. The purpose is to identify suspicious behaviour. This can involve observing price patterns or bidding patterns when it comes to public procurement. It can be a data- and time-intensive process but not necessarily, as much of the initial screening comes through buyer complaints, competitors that are upset, or the leniency program. The first-to-report leniency programs, have been particularly successful in combating cartels.[30]

[26] Monti, Mario in the *3rd Nordic Competition Policy Conference on Fighting Cartels - Why and How?* Chapter 1. The Swedish Competition Authority (2001).

[27] Connor, John M., *How High Do Cartels Raise Prices? Implications for Reform of the Antitrust Sentencing Guidelines*, American Antitrust Institute, Working Paper 04-01, August 2004.

[28] OECD, *Report on hard core cartels*, (2000).

[29] Harrington, Joe (2005) *Detecting Cartels*. Conference paper for "Advances in the Economics of Competition Law".

[30] Harrington, Joe (2005) *Detecting Cartels*. Conference paper for "Advances in the Economics of Competition Law".

Today, it is commonplace for competition authorities to use data analysis to identify suspicious patterns. Here, the goal must be to have an up-to-date model that evolves in parallel with how cartelists behave in order to correctly flag suspicious behaviour. It is important to have a good method for screening as the following processes takes time and resources from the authority that could be used elsewhere, and time and resources from suspected firms. This is, of course, not an easy task.

Screening is followed by *verification*. Here, authorities attempt to distinguish collusion as the explanation for the observed behaviour from competition as the explanation. The third and final process is *prosecution*, where economic evidence is gathered to persuade the court or another administrative body that there has been a violation of the law.[31] A successful competition authority has to have an effective leniency programme, effective enforcement powers and sanctions, and close co-operation among competition authorities.[32]

3.1.2 The digital era

The development of algorithmic pricing has received a lot of attention from national and international competition authorities. This has resulted in competition authorities broadening their investigative tools and employing data scientists. For example, the UK Competition and Market Authority has set up a New Data Unit that explores how firms use algorithms in their business models and how this affects customers. The German Bundeskartellamt has set up a Digital Economy unit focusing on all aspects of the digital sector. Similarly, the Danish Competition and Consumer Authority (DCCA) has created the Digital Market Division, which among other things, coordinate the DCCA's studies of machine learning, artificial intelligence, big data and the use of algorithms. Lastly, the Swedish Competition Authority has recently hired a data scientist.

The debate regarding AI and competition law can be divided into three strands. The first one includes experts that question whether the legal framework needs to change. The second one includes experts wondering whether it is necessary to develop a clearer definition of agreement for antitrust purposes. Thirdly, experts question the liability of the creators, users, or benefiters of the algorithms.[33] The answers to these questions have implications for how competition authorities approach the concept of cartel detection and prevention in today's digital era.

[31] Harrington, Joe (2005) *Detecting Cartels*. Conference paper for "Advances in the Economics of Competition Law".

[32] Monti, Mario in the *3rd Nordic Competition Policy Conference on Fighting Cartels - Why and How?* Chapter 1. The Swedish Competition Authority (2001).

[33] OECD (2017), *Algorithms and Collusion: Competition Policy in the Digital Age*.

It has been suggested that competition authorities may have to improve its merger control regimes as a preventative measure if the long-term result is that AI raises the risk of tacit collusion. The purpose of these measures would be to avoid market conditions where AI might form tacit collusion.[34] However, it is difficult to say whether algorithms change the importance of traditional structural characteristics of markets that make them more prone to collusion, such as the number of competing firms. Thus, such efforts by competition authorities may prove to be fruitless. Other approaches may be to use a) market study and investigation tools to identify markets where algorithms may pose a more serious concern and select the best regulatory or enforcement solutions, b) use ex-ante merger control to review coordinate effects in markets that are not too concentrated but where algorithms are common, or c) use commitments and remedies to prevent the use of algorithms as a facilitating practice.[35]

When it comes to liability, an AI who merely executes the firms' agreement to collude is not treated any differently by competition authorities than a traditional cartel. Whether the competition authority can hold the firm liable when it comes to tacit collusion, however, the view on liability differs between authorities. The European Commission considers the use of an AI to be no different from any other price setting tool. The commission expects firms to take steps in the design of the AI that ensures that it does not lead to anti-competitive outcomes. A firm or individual can (and should) always be held accountable for the actions of the algorithm. (Then) European Commissioner for Competition Margrethe Vestager said that:

> *…businesses also need to know that when they decide to use an automated system, they will be held responsible for what it does. So they had better know how that system works.*[36]

This is a similar view to the one expressed by the German Bundeskartellamt's president Mundt when saying in the context of *Lufthansa* that Lufthansa could 'not hide behind an algorithm'. On the other hand, the UK Competition authority CMA's view is that an AI may be intelligent enough to circumvent the compliance protocols that firms put into its design to ensure that it does not lead to anti-competitive outcomes. These examples illustrate that competition authorities do not necessarily have the same stance on AI and liability.

The other side of the coin is that AI may also be used to detect cartels. If this is the case, then AI can be used both by competition authorities to screen for cartels and as part of a defence brought forward by firms accused of engaging in a cartel.

[34] Garyali, K. *Is the Competition Regime Ready to Take on the AI Decision Maker*. CMS London. Available online: https://cms. law/en/gbr/publication/is-the-competition-regime-ready-to-take-on-the-ai-decision-maker (accessed on 21 August 2020).

[35] OECD (2017) Executive Summary of the Roundtable on Algorithms and Collusion Annex to the Summary Record of the 127th meeting of the Competition Committee.

[36] Margrethe Vestager (2017) European Commission.

One example of how AI can be used to detect cartels is cartel screening tools. Sanchez-Graells shows that AI, combined with traditional screening methods used by competition authorities, is a powerful tool to detect bid-rigging in tenders.[37] The author finds that two screens are particularly important, namely the ratio of the price difference between the second and winning first lowest bid to the average price difference among all losing bids and the coefficient of variation of bids in a tender.

It is not only competition authorities that are interested in AI and cartel detection. DLA Piper has launched Aiscension, an AI-based service that helps firms to detect and prevent cartels to ensure that they fulfil their compliance system, in collaboration with tech company Reveal.[38]

3.2 Related Work

As shown above, the question of whether algorithmic pricing using reinforcement learning may result in tacit collusion has attracted a lot of interest in recent years, both by regulatory authorities and within the scientific community. However, not many studies have been presented where empirical investigations into the problem have been conducted. Within this section, we will look closer at the studies that have been performed and summarise their findings.

A study by Klein[39] uses Q-learning to assess whether autonomous machine learning algorithms may learn to collude on prices. In the paper, two different scenarios are evaluated: Q-learning vs fixed-strategy behaviour and Q-learning vs Q-learning. When Q-learning competes against a fixed-strategy competitor with monopoly fixed price behaviour, the agent also converges to monopoly prices, which makes sense from a game-theoretic perspective. When two Q-learning agents are competing against each other, they converge to prices above competitive price levels. The larger the number of price levels available, the higher the price levels the agents converge to.

A seminal work recently presented is the study by Calvano et al.[40] The paper was made public as a working paper in several iterations before being published and has stirred up a lively interest in academia and competition authorities.

Calvano et al. use a simple baseline environment with two agents, fifteen price levels (actions) and the Q-learning algorithm. They run a very large number of experiments, varying many different parameters in a structured way. The parameters being varied are related both to the algorithm, such as learning and exploration parameters, and to the environment, such as the number of agents, the amount of information they use when learning, the number of possible price levels, etc.

[37] Sanchez-Graells, A. (2019) *Data-driven and digital procurement governance: Revisiting two well-known elephant tales*. Available at SSRN 3440552.

[38] Aiscension: https://www.dlapiper.com/en/europe/focus/aiscension/overview/

[39] Klein, T. (2018) *Assessing Autonomous Algorithmic Collusion: Q-Learning Under Short-Run Price Commitments* (No. TI 2018-056/VII). Tinbergen Institute Discussion Paper.

[40] Calvano, E., Calzolari, G., Denicolo, V., & Pastorello, S. (2020) *Artificial intelligence, algorithmic pricing, and collusion*. American Economic Review, 110(10), 3267-97.

The overall conclusion drawn by Calvano et al. is that Q-learning algorithms systematically learn to collude, i.e., that the resulting price levels are higher than for an efficient market and that the agents share the profit. However, one very important observation from their experiments is that asymmetry in the environment significantly decreases the tendency to collude. Asymmetries introduced relate to the cost and demand between firms. As part of our experimentation, we have repeated some of the experiments performed by Calvano et al. which confirm their results. These results have not been included in the report.

In a follow-up paper[41], Calvano et al. investigate algorithmic collusion with imperfect monitoring where competitive actions are more difficult to observe, again using Q-learning and synthetic environments based on a model of Curnout competition. Here, they are able to show that with perfect monitoring, the behaviour is very similar to the Bertrand setting used in the above-mentioned work. However, with imperfect monitoring, the level of collusion decreases, even if the agents are still capable of reaching collusive outcomes. The agents enter into "price wars" in situations where the competitor deviates and after a demand shock. However, the price war is temporary, and the agents return to above competitive price levels after some time.

Another study evaluating tacit collusion empirically is a Master thesis by Mellgren[42]. The main contribution in his thesis, compared to Calvano et al., is that he is using deep Q-learning in his experiments. Deep Q-learning uses a deep neural network as a function approximator, instead of a lookup table, which could make the agent capable of handling more complex environments. The experiments, however, include only two agents and a relatively small number of price levels. The results show that the agents achieve higher than competitive prices. Furthermore, the results also indicate that the addition of neural networks, as probably should be expected for such small state space, did not make learning fundamentally different from tabular Q-learning.

A recent paper is addressing the situation where the pricing algorithms are not aware of (or at least does not use) competitors' prices. The paper, by Hansen et al.[43], shows that collusive outcomes can be achieved even under these circumstances. In their experiment, the algorithm indirectly perceives information about competitor prices through the reward function, which consists of the true profit, but with a random noise element added. By varying the noise distribution, i.e., the level of informativeness of the reward function, they are able to show that informativeness is a critical element in explaining collusive outcomes.

[41] Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2021) *Algorithmic collusion with imperfect monitoring*. International Journal of Industrial Organization, 102712.

[42] Mellgren, F. *Tacit collusion with deep multi-agent reinforcement learning*. Available at: https://www.konkurrensverket.se/globalassets/dokument/kunskap-och-forskning/uppsatstavling/uppsatser/uppsats2020_filip-mellgren.pdf

[43] Hansen, K., Misra, K., & Pai, M. (2020) *Algorithmic collusion: Supra-competitive prices via independent algorithms.*

An important observation regarding all the studies mentioned above is that algorithmic parameters are identical between the agents. This means that agents are always identical in all experiments. As previously discussed, the likelihood for individual firms defining solutions having identical agents in isolation from each other is extremely unlikely. However, this kind of identical implementation is possible without explicit collusion if an external factor is shared, like in a hub-and-spoke scenario, such as when the same software vendor or consultancy (the hub) is setting up their standard solution for multiple competing firms. Obviously, firms may use algorithmic pricing without being part of a hub-and-spoke scenario and even when there exists a shared hub, different firms may still require the provided solution to be different from the solutions implemented by competitors. Consequently, in our opinion, all studies performed so far are missing an essential aspect of how pricing algorithms can be expected to work in reality. The essential aspect is the fact that agents operating on a market would most likely be different, thus more or less limiting the insights into how collusion is affected in a hub-and-spoke scenario.

4 Empirical investigation

In this section, we present the method and the results for the three experiments conducted.

1) The first experiment examined different algorithms.
2) Experiment two investigated asynchronous and asymmetric update speeds of the agents.
3) The third experiment examined how the possibility of new agents/actors entering the market affects the tendency to implicit collusion.

4.1 Experimental setup

General setup

In our experimentation, we have made a series of design choices. The first choice was to only consider discrete price levels. The rationale behind this choice is that prices typically follow a certain logic. Even though all prices are theoretically possible, it is unusual to assign prices violating the logic. For example, if something is to be priced at about 100 Swedish Krona (SEK), it would be more natural to set the price to 95, 99 or 100 rather than 101, 97 or even 105. The price levels used are not set up to mirror prices used in reality, but are instead simply equidistant, they still mimic the general logic of price levels generally being discrete.

The general setup of the experiments is heavily influenced by the setup used by Calvano et al.[44]. The reasons for using the same setup have been:

1. By using a setup that had been peer-reviewed, our method can be considered scrutinised by experts.

2. It makes it easier to compare results with previous work in the field.

For transparency of the setup used in the report, a summary of the implementation details and how it deviates from Calvano et al. follows. The economic environment is represented as a canonical model of collusion. This means that we are simulating an infinitely repeating pricing game with actions taken simultaneously by all firms[45], conditioning their actions on historical actions and consequences. The model of price competition is using logit demand and constant marginal costs.

With n firms, each with their own product, and an outside good representing an inverse index of aggregate demand, the demand for product $i = 1, \dots, n$ for each period t is:

$$q_{i,t} = \frac{e^{\frac{a_i - p_{i,t}}{\mu}}}{\sum_{j=1}^{n} e^{\frac{a_j - p_{j,t}}{\mu}} + e^{\frac{a_0}{\mu}}}$$

[44] Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2018*) Artificial intelligence, algorithmic pricing and collusion*. CEPR Discussion Paper No. DP13405.

[45] The second experiment deviates in that actions are not always taken simultaneously.

where a_i are product quality indexes capturing vertical differentiation, product a_0 is the outside good, and μ is an index of horizontal differentiation. The reward per firm i and period t is

$$\pi_{i,t} = (p_{i,t} - c_i)q_{i,t},$$

where c_i is the marginal cost. As long as firms stay active, fixed costs can be ignored.

We have already motivated using discrete and equidistant price levels. Furthermore, it is reasonable to assume that prices are finite. To calculate the possible prices, which also defines the set of actions A, we used Bertrand-Nash equilibrium of the one-shot games, denoted $\mathbf{p}^N$, and the monopoly price, denoted $\mathbf{p}^M$, calculated for the parameters. The prices are ranging from below Bertrand and above monopoly and are given by m equally spaced points in the interval

$$[\mathbf{p}^N - \boldsymbol{\xi}(\mathbf{p}^N - \mathbf{p}^M), \mathbf{p}^M + \xi(\mathbf{p}^N - \mathbf{p}^M)],$$

where $\xi > 0$ is a parameter.
It is possible to let the agents have a memory, including past prices in its state description. However, the size of the observation space increases exponentially with the length of the history. The number of actions is $|A| = m$ and the size of the observation space is $|O| = m^{nk}$. As the history was kept constant in all experiments, setting $k = 1$, i.e., the agents only consider the current price levels set by all competitors, it follows that the size of our observation space is $|O| = m^n$.

Exploration used by Q-learning was a simple ϵ-greedy model with time-declining exploration rate. Initially, the algorithm chooses actions randomly and as the learning progressed, the algorithm gradually makes more and more policy-guided actions. The time-declining exploration rate was $\epsilon_t = 1 - e^{-\beta t}$, with the parameter $\beta > 0$. The exploration declines faster the greater β is.

We have opted to define convergence, in accordance with Calvano et al.[46], as a constant strategy play by all agents during a sufficient number of iterations. More specifically, if for each agent i and each observed state s, the action $a_{i,t}(s) = arg\ max\ Q_{i,t}(a, s)$ does not change for any of the visited states during q consecutive iterations, that episode is considered to have converged. This will either happen when the policies have converged to a single state, or when the agents have created a sequence which is repeated over and over.

The baseline setup has been the basis for most experiments, with some slight variations described below. All experiments assigned $c_i = 1, a_i = 2, a_0 = 1, \mu = \frac{1}{2}, \delta = 0.95$ and a one-period history $k = 1$. In tabel 1 , all parameters with settings varying across the experiments are listed.

[46] Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2018) *Artificial intelligence, algorithmic pricing and collusion*. CEPR Discussion Paper No. DP13405.

Tabel 1. Parameter settings used

Description	Parameter	Exp1	Exp2	Exp3
Number of agents	n	2, 3	2	2
Price range	ξ	0.1	0.1	0.1
Algorithm		Q, PG, SARSA	Q	Q
Type of game		Sync	Async	Sync
Convergence	q	50	25K	250
Exploration decay	β	Fixed	Fixed	Varied

Initially, the settings regarding convergence and maximum number of iterations were similar to the ones used by Calvano et al. However, after a lot of experimentation, using different values for both maximum number of iterations and convergence, we saw that the large values used by Calvano et al. were not really necessary. Furthermore, as only results from experiments that converged have been included, the maximum number of iterations is not as relevant. The motivation for this is simply that it is very hard to draw useful conclusions based on results that are, in some sense, random. Since we also saw that very long sequences never occurred in the exploratory experiments, we eventually also decreased the convergence parameter without any noticeable difference in the results. Experiments were repeated until at least 10 runs had converged.

Evaluation metrics

Several different evaluation metrics can be used for evaluation, including prices, profits etc. One metric used in the literature is the economic indicator *average profit gain* Δ which has the attractive property of being normalised in relation to the Nash equilibrium ($\Delta = 0$) and the monopoly price ($\Delta = 1$). It is defined as

$$\Delta = \frac{\bar{\pi} - \pi^N}{\pi^M - \pi^N},$$

where π^N is the profit achieved per firm in the Bertrand-Nash equilibrium, π^M is the profit achieved when all firms collude fully, and $\bar{\pi}$ is the average profit achieved after convergence. By substituting $\bar{\pi}$ for the profit achieved by each agent, the individual contribution to the average profit gain can be calculated which can be used to deduce the contribution by each agent.

A drawback of profit gain is that it is somewhat abstract and may be hard to relate to. To complement this metric, we also look at a metric that indicates how much higher than Nash prices or lower than monopoly prices, the converged price levels are in per cents. The *per cent of distance*, provide a linear estimate of how far from the Nash or the monopoly prices the converged price levels are. This measure is similar to profit gain but is defined using the prices rather than using the profit.

The purpose of the first two experiments has been to evaluate what happens when agents somehow differ from each other. The motivation for this is that it is unlikely that two different firms, by themselves, can create two identical price optimising agents if we assume no explicit collusion. The final experiment aims to investigate how the threat of additional agents entering the market affects the tendency to develop tacit collusion.

Experiment 1 – Different algorithms

The first experiment evaluates different learning algorithms. The implementation was done using the Matlab Reinforcement Learning Toolbox. However, since multiagent RL was not supported, a protocol for sharing the current state was implemented, making it possible for each agent to inform all other agents of its current action while at the same time synchronising the updates. The `spmd` function was used to run the experiments in parallel and the `labSendRecieve` function was used to share state information among all the agents.

Three different algorithms were evaluated, and they were competing against each of the other algorithms both pairwise and all three together. For comparison, both internally and with previous results, each algorithm was also competing with itself, using identical setups. The algorithms were implemented using default settings (as described in the documentation) and parameters were not optimised. Q-learning and SARSA used a kind of lookup-table called Q-tables as value functions, whereas PG used a neural network function approximator. Here it must be noted that a Q-table explicitly stores estimated values for all state-action pairs (here observed prices and the setting of a new price) whereas a neural network creates its own black-box representation of the state-action space.

Experiment 2 – Different update frequency of the agents

In this and the remaining two experiments, Q-learning was used exclusively. The purpose of this experiment was to evaluate what happens when two agents are not synchronised, and in particular, when one agent can update the policy more often than the other.

In the experiment, one agent was allowed to be more agile, with more frequent price changes. The more active agent changed its prices 3-6 times as often as the slow changing agent. Both agents were able to observe the environment and learn in every iteration, even if only one agent had been allowed to act.

Experiment 3 – Threat of entry of new competitors

The final experiment intended to investigate the effect of letting the agents incorporate an anticipated cost if new agents were to enter the market. The reward function was altered by adding two additional parts: 1) a likelihood of a new agent entering, and 2) a weighted reward component representing the expected reward that the agent would have got with an additional agent competing.

The likelihood of a new firm entering the market is the profit gain *pf* achieved by using the prices of the agents already in the market. A higher profit gain indicates a larger likelihood for a new firm to enter, trying to gain some of the profit.

The reward for firm *i* is:

$$\pi_{i,t} = pf\left(p_{i,t} - c_i\right)\frac{e^{\frac{a_i - p_{i,t}}{\mu}}}{\sum_{j=1}^{n+1} e^{\frac{a_j - p_{j,t}}{\mu}} + e^{\frac{a_0}{\mu}}} + (1 - pf)\left(p_{i,t} - c_i\right)\frac{e^{\frac{a_i - p_{i,t}}{\mu}}}{\sum_{j=1}^{n} e^{\frac{a_j - p_{j,t}}{\mu}} + e^{\frac{a_0}{\mu}}}$$

where the main difference between the two parts is the inclusion of $p_{n+1,t}$ (in the first summation) which is defined as:

$$p_{n+1,t} = \mathbf{p}^N + \gamma \left(\frac{1}{n} \sum_{j=1}^{n} p_{j,t} - \mathbf{p}^N \right)$$

The *aggressiveness* factor $\gamma \in [0,1]$ is the expected aggressiveness of the new agent. This means that the expected price of a new entrant is the Nash-price if $\gamma = 0$ and the average price of existing agents if $\gamma = 1$. The reward is consequently the combined weighted reward the agent would normally have got plus the weighted reward the agent would have got with an additional agent setting prices based on γ competing with the existing agents. The weights are determined by *pf*.

4.2 Results

The main focus of the experiments is on evaluating what happens when the competing agents are different from each other, either because of the algorithms used or the updating frequency.

Experiment 1 – Different algorithms

Experiment 1 has two parts, one part in which the price optimising agents are using the same algorithm and one part in which they are using different algorithms. The results from Experiment 1 will be presented and analysed from several different perspectives. All the results presented are from experiments in which the agents have converged.

The purpose of the first part is 1) to establish a baseline indicating how identical agents behave which we can compare with when evaluating agents that are not identical and 2) to compare with published previous results. The results of these experiments are shown in tabel 2. The overall profit gain is the average profit gain achieved from the experiment and *% Dist* is the resulting price levels as a per cent of the distance between Nash and monopoly prices. This means that a Nash price would result in 0% of the distance and a monopoly price in 100%.

Tabel 2. Results from experiments with the same algorithms competing against each other

	ProfitGain	% Dist
Q-Q	0.62	50.0%
SARSA-SARSA	0.66	51.4%
PG-PG	0.59	48.8%
Mean	**0.63**	**50.1%**

First of all, there is very little difference between the different algorithms when both agents are using the same algorithm. Furthermore, it is evident that the algorithms have converged to price levels clearly above competitive prices, i.e., exhibiting tacit collusion. In addition, when two identical agents compete against each other, they seem to converge to approximately the same level of tacit collusion, regardless of which algorithm was used. These results when using identical agents are in line with the findings seen in exiating research. Consequently, our results indicate that previous results, obtained using only Q-learning, generalise to other kinds of algorithms.

In the second part of experiment 1, the price optimising agents are using different algorithms. First, in tabel 3, we show the profit gain for the four combinations of algorithms evaluated. The overall profit gain is the average profit gain achieved from the experiment. The Q-, SARSA- and PG-profit gains are the average individual contributions of the overall profit gain from each of the algorithms.

Tabel 3. Profit gain from experiments with different algorithms

	Profit Gain			
Experiment	**Q**	**SARSA**	**PG**	**Overall**
Q-PG	30.1%		69.9%	*0.72*
Q-SARSA	50.7%	49.3%		*0.64*
SARSA-PG		37.2%	62.8%	*0.87*
Q-SARSA-PG	29.5%	33.0%	37.6%	*0.49*
Mean per algorithm	**22.9%**	**29.8%**	**47.3%**	***0.70***

There are several noteworthy things that we can see from tabel 3. First of all, the overall profit gain levels are high, compared to the Nash-level of 0, in all experiments, showing that the agents indeed reach states with tacit collusion after convergence. However, even more interesting is that the individual algorithms' contribution to the profit gain differs quite substantially and that the difference is systematic in favour of the PG algorithm. These results indicate that the PG algorithm is more capable of adapting to the problem and exploiting the other algorithms' weaknesses. When two algorithms that are similar (such as Q and SARSA) are evaluated against each other, there is no difference in how much they contribute, and the overall profit gain is similar to the levels reported in tabel 2. On the other hand, when SARSA and Q compete against PG, the overall profit gain is higher than when similar agents compete. In the Q-PG trials, the overall profit gain is high but Q's contribution is rather low. In the SARSA-PG trials, the overall profit gain is even higher and SARSAs contribution is higher compared to Qs contribution in the Q-PG trials.

When three agents with different algorithms were evaluated, the individual contributions by the different algorithms are more even, but follow the same pattern as in the pairwise experiments: PG gains the highest profit, followed by SARSA and Q. Finally, it seems like the overall level of tacit collusion is affected by the number of agents, with a clearly lower profit gain with three agents.

Profit gain is not a very transparent measure, so it is worth looking into what these values actually mean in terms of price levels compared to the Nash price as well as the monopoly price. Tabel 4 shows the resulting price levels as a per cent of the distance between Nash and monopoly prices. This means that a Nash price would result in 0% of the distance and a monopoly price in 100%.

Tabel 4. **Price levels achieved as per cent of the distance between Nash and monopoly prices (0% = Nash price and 100% = monopoly price)**

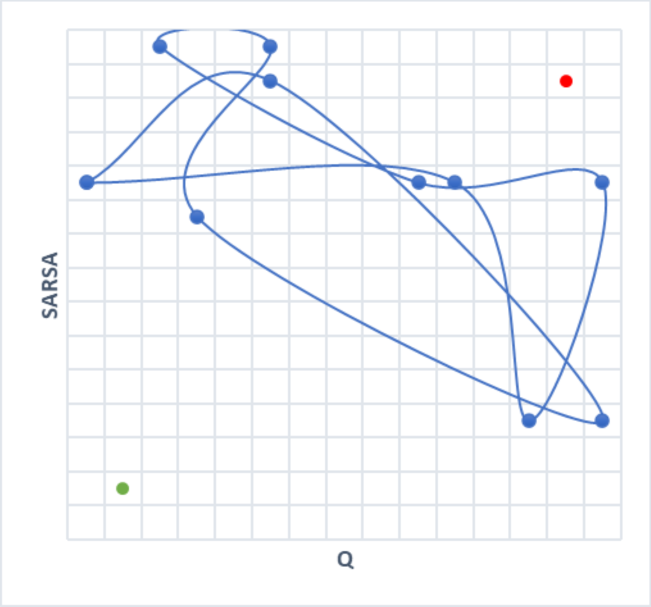
	% of Distance			
Experiment	**Q**	**SARSA**	**PG**	**Overall**
Q-PG	70.2%		49.2%	*59.7%*
Q-SARSA	51.4%	51.6%		*51.5%*
SARSA-PG		83.1%	68.8%	*75.9%*
Q-SARSA-PG	49.7%	45.6%	40.5%	*45.3%*
Mean	**62.3%**	**67.1%**	**54.4%**	***60.3%***

The most interestring result from these comparisons is that the prices achieved by the PG algorithm, despite its higher profit gain, are lower than its opponent(s). When the two weaker algorithms, Q and SARSA, compete, they achieve the lowest price levels. When, on the other hand, the two algorithms achieving the highest profit, PG and SARSA, are competing, they end up with prices rather close to monopoly prices. When agent asymmetry result in one agent having lower prices and higher profit, it also follows that those lower prices are the price levels available to the customers. This means that even if the average price levels are higher between PG and Q than between SARSA and Q, the lowest prices (on average) are still provided by the former pair, even if the difference is marginal in this comparison. However, the SARSA-PG trial still results in the highest price levels, both overall and as the (average) lowest prices, so competition among the agents is not guaranteeing lower prices to customers.

There are some patterns that we have been able to identify among the policies developed after convergence. One such pattern is that the agents converge to a single state, with constant prices for both agents. This happened most often with PG and SARSA, but also with PG and Q. It did not, however, happen when agents using Q and SARSA were competing. With only a few exceptions, the general pattern is that PG converges to a much lower price, which is still higher than the Nash price though, achieving a higher market share and higher profit gain. The prices sat by the algorithms opposing PG were generally very high, close to or at the Monopoly level. Single state convergence also happened when agents using the same algorithms were competing against each other. However, it was not possible to detect any systematic behaviour in these cases.

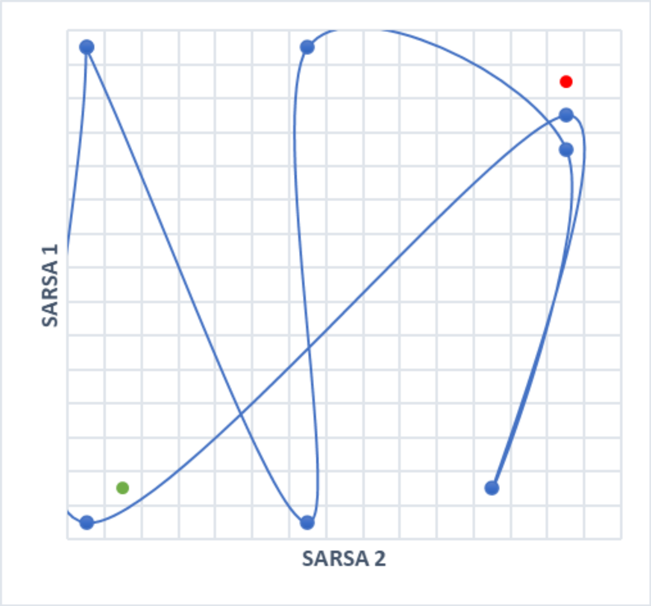
When the agents did not converge to a single state, they instead converged to a state-loop. Thus, the agents revisit the same states in the same order over and over again, reusing the same sequence of prices again and again. Depending on which algorithms were competing, the length of the loops varied. When the experiment involved PG, one of two situations appeared; either the converged policies involved very few states (typically one to three) or longer sequences, i.e., over twenty steps. When Q and SARSA competed, the loops' length was generally between 6 and 12 states long. With these two algorithms, the dynamic that tends to re-appear is that the agents interact in such a way that when one agent raises or lowers its price, the other will follow. Below, figure 3, shows an example of policies developed between Q and SARSA after convergence. The boxes represent the different states, having the price levels of the agents on the axes, with the Nash state in green, where both agents have the second lowest price level, and the monopoly state in red, where both agents have the second highest price level. No information about prices or price levels are shown as it would not provide additional insights. This is a typical example of how agents are reacting to each other. The sequence involves both high and low prices for both agents, but the general pattern is that when one agent sets a high price, the other has a relatively low price and vice versa. Consequently, these two agents tend to explore the diagonal from the upper left to the lower right, avoiding getting close to the Monopoly (red) or the Nash (green) states.

Figure 3. Example of policies developed by agents using Q and SARSA



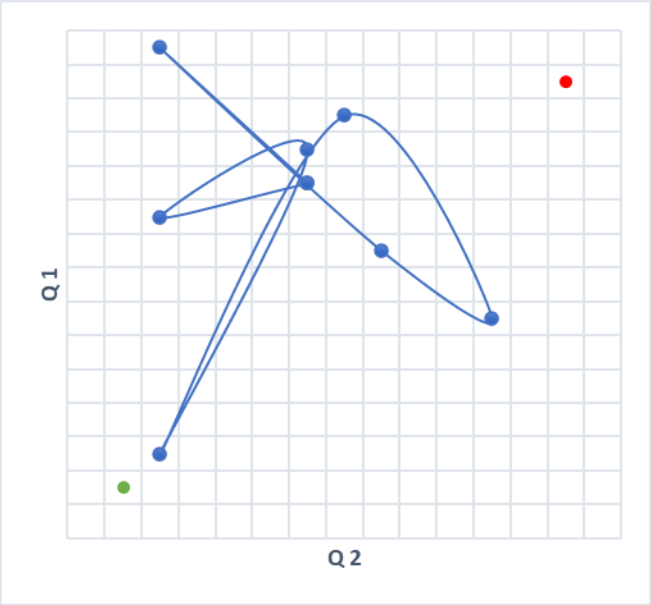
The previous example shows that the developed policies never visited states particularly close to the Nash or Monopoly states. However, this is not always the case, as can be seen in the following two examples where the agents competed using the same algorithms. In the next example, presented in figure 4, the policies move from a near Monopoly state to a near Nash state and then the agents try to react using different policies. The policy of the first agent (SARSA 1) is to oscillate between high and low prices, whereas the policy of the second (SARSA 2) is to gradually increase its price, with a small oscillation at the top.

Figure 4. Example of policies developed by two agents using SARSA



In figure 5, a pair of policies similar to the one presented in figure 3 can be seen, with the difference that both agents simultaneously decrease their prices to almost Nash levels at one point.

Figure 5. Example of policies developed by two agents using Q



Experiment 2 – Different update frequency of the agents

In the second experiment, the two agents used the same algorithm (Q-learning) but was changing action with different frequencies. One agent was allowed to be fast and change its price in every timestep, whereas the other had to wait a number of timesteps. The number of timesteps in which the slow agent waits until changing its price was randomly set to be between 4-6 timesteps. The frequency was randomly re-assigned every time a new price had been decided. Both agents were, however, allowed to learn from the environment in every timestep. This setup was intended to mimic an asynchronous situation, where one agent could not affect the environment as frequently as the other but where both agents could register changes to the environment. This limitation could, e.g., stem from a pricing policy limiting the frequency of price changes.

The results of the second experiment are presented in tabel 5 below.

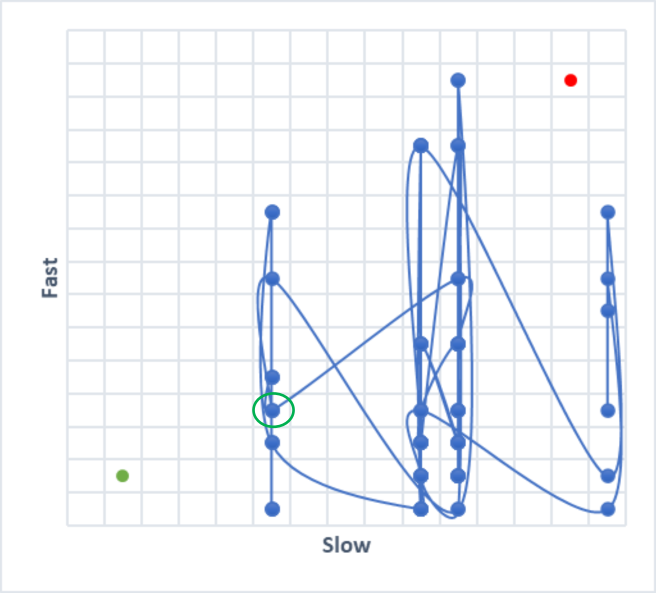
Tabel 5. Results from experiments with one fast and one slow agent competing against each other using Q-learning

	ProfitGain	% Dist
Slow agent	-0.1%	73.3%
Fast agent	100.1%	30.8%
Mean	**0.62**	**52.1%**

When looking at the profit gain and the price levels achieved in the experiment, the average (Mean) results are very similar to the results achieved by Q-Q in tabel 2. Here, however, the profit is not split between the agents, instead the faster agent obtains all the profit. The contribution to the average profit gain comes entirely from the fast agent. Interestingly enough, the fast agent achieves this advantage through prices that are much lower, and closer to the Nash price, than in any of the previous experiments. In this experiment, the lowest (average) price provided to the customers is low, but still above the Nash price.

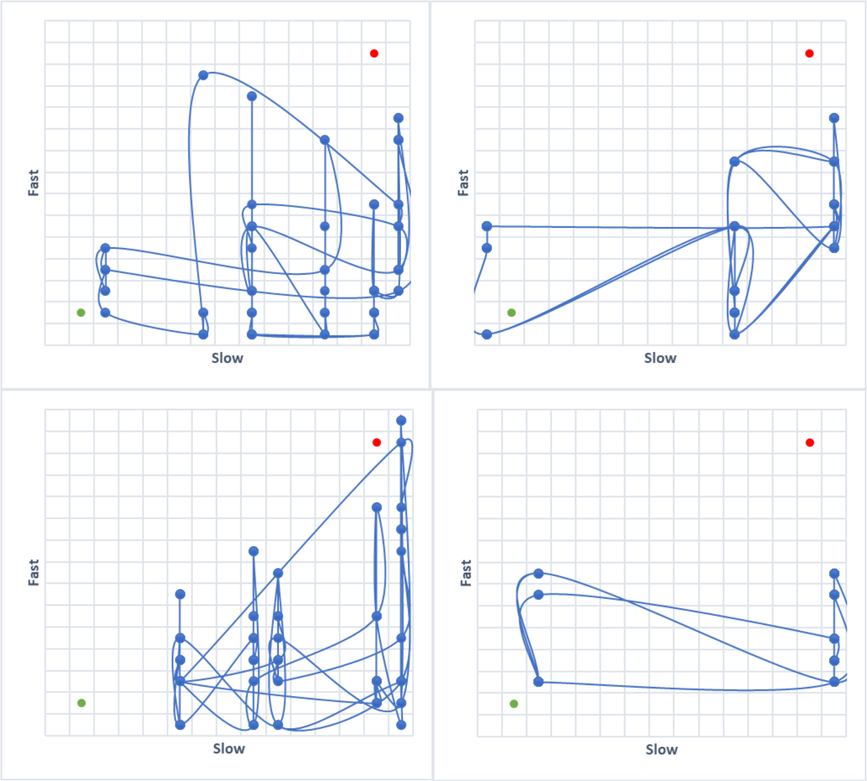
The policy developed by the fast agent has some interesting characteristics. As the slow agent changes its behaviour with random intervals, the fast agent develops different policies depending on the price assigned by the slow agent. And conversely, depending on which price the fast agent has, the slow agent will adopt different policies. These two characteristics in combination results in complex, but predictable, behaviours by the agents. In the following example, this is illustrated and explained.

Figure 6. Example policies developed by a fast and a slow agent using Q



Looking at the example in figure 6 , we see the four different price levels adopted by the slow agent and the different price levels chosen by the fast agent for each of these. As an example of how the policies of the fast and slow agents work, we can look at what happens with the different agents in the green encircled state (with price levels 4/6 for the fast/slow agents). In this state, the slow agent is either not acting, keeping its current price, or may change price depending on how long the delay is. Since the fast agent does not know whether the slow agent is going to keep its price constant one more iteration or change its price, it will always take the same action, namely setting the price to level eight. Consequently, this position has two possible outcomes: 8/6 (if the slow agent keeps its price) or 8/11 (if it changes its price). Similarly, when the state 3/11 is reached, the fast agent will always choose price level 6, but the slow agent will either stay with price level 11 or lower the price to level 10, resulting in 6/11 or 6/10. In a similar way, there are some other states in which the same kind of pattern appears. Even if the policies of the agents are more complex due to the randomised delay of the slow agent, they follow consistent policies. As the complementary examples presented in figure 7 show, all of the policies primarily explore the lower right half, with lower prices for the fast agent and higher prices for the slow agent. This is, of course, corresponding to the results we could see in tabel 5.

Figure 7. Example policies developed by a fast and a slow agent using Q



Experiment 3 – Threat of entry of new competitors

The third experiment intended to see how agents already competing in a market are affected by the threat of new agents entering the market. We have run several experiments, varying the assumed aggressiveness of a new agent entering. A higher aggressiveness factor means the assumed price of a new agent is closer to the Nash price, whereas a lower aggressiveness factor means a price closer to the prices used by the agents already in the market.

Tabel 6. **Results from experiments investigating the risk of a new agent entering a market with two agents using Q-learning**

Aggresiveness	ProfitGain	% Dist
0	0.48	35.1%
0.1	0.47	34.4%
0.2	0.44	32.8%
0.3	0.48	36.0%
0.4	0.44	33.3%
0.5	0.46	33.8%
0.6	0.42	30.9%
0.7	0.42	31.4%
0.8	0.41	29.7%
0.9	0.52	39.1%
1	0.40	31.2%

There are two takeaways from this experiment: First of all, the profit gain levels are slightly lower than in most previous examples, indicating that a built-in awareness of the threat of entry by new actors have the predictable effect of holding the agents back, not pushing the prices as high as they would otherwise do. In fact, the prices achieved in these experiments are generally lower than in other experiments. Furthermore, the aggressiveness factor is also having the expected effect, with higher anticipated aggressiveness resulting in lower prices in general. So, when there is a threat of entry by a competitor, this lowers the price levels and profits of the algorithms. The more aggressive the potential entrant is, the larger this effect is.

This highlights the disciplinary effect that threat of entry by a competitor might have on a firm's price decisions, even when an algorithm makes these decisions.

Summary

Summarising the findings from the experiments, a key result is of course that the price levels are significantly higher than the Nash level. Specifically, if two identical agents compete, they share the profit, thus clearly showing that tacit collusion takes place. This is in accordance with previous research. On the other hand, our experiments also show that when different agents compete, the agents no longer share the profit more or less equally, but the stronger agent obtains a larger part of the profit gain. Here, the relative strength of the algorithms come from the algorithm design (experiment 1) or an ability to react to the market more frequently. Interestingly enough, the results show that the stronger agents are primarily winning by setting lower prices, thus gaining a larger market share. Still, even if the winning agent competes with lower prices, the prices are generally still higher on average than when two identical agents are competing, i.e., of course well above the Nash level.

5 Discussion

We have performed three experiments to study the effects of sophisticated self-learning algorithms autonomously setting price levels. Our results have several implications for the general view on the use of price-setting autonomous algorithms and the risk of collusive outcomes. This adds to the academic debate and provides insights on how competition authorities may best approach this issue.

We first show that when pairs of identical agents, using the same algorithm and settings, are used to set prices, the algorithms level of sophistication does not matter for the average profit gain. For all types of identical pairs that we investigated, the average profit gain is more or less the same, and each algorithm gets the same profit gain. Therefore, our first result is that when two identical agents are competing against each other, they achieve tacit collusion, i.e., they converge to price levels above Nash and share the profit in the long run. Furthermore, they appear to converge to approximately the same level of tacit collusion, regardless of which algorithm is used. This is in agreement with, and therefore confirms, existing related work.

However, we obtain very different results when we move away from pairs of identical agents to agents having different capabilities, either due to agents using different types of algorithms or due to agents having different update intervals. The overall profit gain levels remain high in all experiments, indicating that the agents reach states with prices that are higher than the competitive level after convergence, even when the agents have different capabilities. However, when we look at the agents' individual contribution to the profit gain, we find that the stronger agents gain substantially more profit on average than the weaker agent. These results can be seen both when the agents' capabilities differ due to algorithmic differences and when they differ due to differences in update frequency. Worth pointing out is that the stronger agents are able to dominate the competitor by setting lower prices, thus gaining a larger market share and a higher profit. Interestingly, we note that in some cases we observe lower price levels by the stronger agent but higher overall price level compared to a setting with homogenous pairs of agents. Although we have not studied the reason for this, or potential implications of this outcome in detail, we believe that it points towards the complex nature of the topic of algorithms and collusive outcomes and why it is difficult to draw generalised conclusions on the matter.

What are the implications of these results? To the best of our knowledge, this study is the first to systematically investigate the more realistic situation when the agents are not identical. Our results complement previous research by indicating that agent asymmetry has a substantial impact on the outcome. The most important implication is, consequently, that agent asymmetry must be taken into consideration in both future research and policy decision-making. Furthermore, the results imply that there is a clear incentive for firms to have the strongest agent, as a stronger agent will either share the profit equally with a comparable agent, or gain a larger profit if the other firm has a weaker agent. Similarly, if both firms have equally strong agents, there is, of course, no incentive for any firm to change to a weaker agent, as this will result in decreased profit. When both firms strive to have the strongest agent, the larger gain only occurs in the time period during which one firm has a stronger agent than the other firm. Once the competitor has an equally strong agent, the profit gains will most likely, after some time of adjustments, yet again be split evenly. Finally, and in line with previous research, firms have a clear incentive to use price optimising agents as this leads to higher expected profit than they would obtain without an agent.

Another result from our experiments is that in a two-agent setting, the two algorithms can both end up in steady states where neither of the algorithms change their price; or in price loops, where the prices follow a fixed sequence that repeats itself indefinitely. Interestingly, the likelihood for steady states seems to differ depending on the algorithms used. As we have only experimented with three algorithms under some specific limitations (such as a small number of discrete prices, a small subset of possible parameters, etc.), it is not possible to generalise patterns regarding what to expect from specific algorithms when applied to real markets. The results imply that stable price loops indicate convergence among the firms active on the market. Consequently, competition authorities may observe price patterns to identify the existence of steady states or price loops to infer whether firms are using price optimisation and whether their policies have converged. As such, the analysis of price patterns may act as a screening tool. However, developing reliable guidelines for identifying which algorithms an individual firm is using is likely not feasible, as the possible variations in parameters are very large and the expected behaviours by different algorithms are most certainly overlapping substantially. Furthermore, as indicated by the results with asynchronous agents, the price loops may be extremely complex when the agents are not synchronised, which would most likely not be the case unless some form of collaboration existed between the firms.

Entry barriers, factors that can prevent or impede a firm from entering the market, is relevant in all competition cases except for per se violations such as participating in hard-core cartels. The reason is that entry barriers may soften or remove the usual mechanism for checking market power: the attraction and arrival of new competitors. The results regarding how the threat of entry by new firms may affect price levels indicate that when the agents consider this, it has a clear impact on the price policies. The implication, which is in line with conventional theory, is that the threat of entry by new firms is a key aspect in determining the risk of collusive outcomes, but this must be taken into account by the algorithm. Thus, the results suggest that entry barriers will continue to play an important role in competition matters, even when looking at algorithms' price policies.

Our results also show that in a setting where each firm has an agent with a different type of algorithm, the stronger agent tends to gain the highest profit. However, the gain from having the strongest agent is significantly smaller in a competition between three agents, compared to pair-wise competitions. Consequently, the market price falls when having three agents. In summary, these results indicate that the average profit gain is reduced as more agents compete.

It is worth noting that an agent's capability can be increased in more ways than by just using an algorithm better adapted to the problem or through more frequent actions. For example, by including more relevant data about competitors and the market, an agent may be able to gain a competitive edge even towards other firms with agents otherwise similar. Examples of relevant data to use include data capturing sentiments about products and firms, news, economic indicators, behavioural data about customers, events, or seasonality etc. Anything that would be considered important to predict user behaviour could potentially be useful for the price optimising agent as well.

This report was motivated by an increased worry among competition authorities and academia that firms' use of self-learning algorithms may lead to higher price levels than the competitive outcome that would otherwise occur. The research presented here and in previous work is only the beginning of a growing strand of literature. It is too soon to give well-developed policy recommendations. However, we believe that our experimental results show the complex nature of algorithms and the risk of collusive outcomes, the sensitivity of the models chosen in research, and the importance of studying the effect of algorithms in more realistic settings.

5.1 Conclusions

The results presented and analysed above, as well as related work, show that price optimising agents can be expected to learn to reach collusive outcomes. A novel result, and a key takeaway from this report, is the fact that as soon as the agents are different in some aspect, e.g., their underlying algorithm or their updating frequency, they will no longer share the profit equally. Instead, one algorithm (the stronger algorithm) will dominate the other, and obtain a clear majority of the profit gain. Unfortunately, the prices are still clearly higher than the Nash level in these scenarios, often actually higher than for markets created by two identical agents. With this in mind, there is a clear incentive for firms to continuously improve their solutions. In fact, an individual firm can only expect a profit gain if their agent is at the very least comparable to other agents operating on the market. So, while two identical agents result in tacit collusion, i.e., prices above the Nash level and the profit share equally between the agents, a scenario including agents with different capabilities is rather a competition between the agents about how the profit gain should be divided.

5.2 Future work

We have in our experiments looked at two possible ways that competing agents can be different. There are many other options available for making agents differ. Combining the ideas in Hansen et al., with agents unaware of or unable to utilise market information, with agents utilising information about competitor prices (and more) could be interesting. This can be developed even further by letting each agent use different perceptions of the environment, so that an agent adds additional information capturing e.g., news, whereas another agent captures some other set of additional information etc. This would better emulate a realistic situation where each firm will try to include as much useful information as possible for their agents to learn from, but where each firm can be expected to have different notions of what is relevant. In all experiments we have seen in the literature, the reward function has been more or less identical between agents (even if adjusting some parameters have been evaluated). In a real-world scenario, different firms could likely end up with different ways of defining the reward function and exploring the effect of this in experiments would be interesting. Building upon the study by Hansen et al., constructing an environment in which many agents are run as single-agents and interacting with the environment entirely asynchronous could be an interesting option, opening up many different paths to explore.

Another area we have only touched upon, but which would be interesting to explore further, is how to detect active price optimising agents and tacit collusion in markets.

Finally, our results are inconclusive regarding how asymmetric agents affect the prices available to consumers. On the one hand, the stronger agent got higher profit through lower prices than its competitor. On the other hand, both agents ended up with rather high prices in some of our experiments. Further experimentation, clarifying what to expect in general regarding consumer prices, is called for.

References

Aiscension: https://www.dlapiper.com/en/europe/focus/aiscension/overview/

Ballard, D.I and Amar S. Naik. *Algorithms, artificial intelligence, and joint conduct*. Competition Policy International, CPI Antitrust Chronicle May 2017.

Bresnahan, Timothy F., and Peter C. Reiss. *Entry and Competition in Concentrated Markets.* Journal of Political Economy, vol. 99, no. 5, 1991, pp. 977–1009. JSTOR, www.jstor.org/stable/2937655. Accessed 22 June 2020.

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2018) *Artificial intelligence, algorithmic pricing and collusion*. CEPR Discussion Paper No. DP13405.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello (2020) *Artificial Intelligence, Algorithmic Pricing, and Collusion.* American Economic Review, 110 (10): 3267-97.

Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2021) *Algorithmic collusion with imperfect monitoring.* International Journal of Industrial Organization, 102712.

Le Chen, Alan Mislove, and Christo Wilson. 2016. A*n Empirical Analysis of Algorithmic Pricing on Amazon Marketplace*. In Proceedings of the 25th International Conference on World Wide Web (WWW '16). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1339–1349.

DOI:https://doi.org/10.1145/2872427.2883089

COM (2017) Report from the Commission to the council and the European Parliament. Final report on the E-commerce Sector Inquiry.

Competition and Markets Authority. 2021 *Algorithms: How they can reduce competition and harm consumers*. URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/954331/Algorithms_++.pdf

Competition Bureau of Canada*, Big Data and Innovation: Key Themes for Competition Policy in Canada* (2017), http://www.competitionbureau.gc.ca/eic/site/cb-bc.

Connor, John M., *How High Do Cartels Raise Prices? Implications for Reform of the Antitrust Sentencing Guidelines*, American Antitrust Institute, Working Paper 04-01, August 2004.

CPI Talks. Interview with Antonio Gomes of the OECD, Antitrust Chronicles. (May 2017).

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

EU-Commission (2017) *Final report on the e-commerce Sector Inquiry - Accompanying Staff Working Document*, Page 175.

French Autorité de la concurrence and the German Bundeskartellamt (2019). *Algorithms and competition*.

Michal S. Gal (2019) *Algorithms as illegal agreements*. Berkeley technology law journal. Vol 34, pp. 67-118.

Garyali, K. *Is the Competition Regime Ready to Take on the AI Decision Maker*. CMS London. Available online:

https://cms.law/en/gbr/publication/is-the-competition-regime-ready-to-take-on-the-ai-decision-maker (accessed on 21 August 2020).

Hansen, K., Misra, K., & Pai, M. (2020). *Algorithmic collusion: Supra-competitive prices via independent algorithms*.

Harrington, Joe (2005) Detecting Cartels. Conference paper for *Advances in the Economics of Competition Law*.

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

Klein, T. (2018) Assessing Autonomous Algorithmic Collusion: Q-Learning Under Short-Run Price Commitments (No. TI 2018-056/VII). Tinbergen Institute Discussion Paper.

Salil K. Mehra, *Robo-Seller Prosecutions and Antitrust's Error-Cost Framework*, CPI Antitrust Chronicles. 37 (May 2017).

Mellgren, F. *Tacit collusion with deep multi-agent reinforcement learning*. Available at: https://www.konkurrensverket.se/globalassets/dokument/kunskap-och-forskning/uppsatstavling/uppsatser/uppsats2020_filip-mellgren.pdf

Monti, Mario in the 3rd Nordic Competition Policy Conference on *Fighting Cartels - Why and How?* Chapter 1. The Swedish Competition Authority (2001)

Nicolas Petit (2017) *Antitrust and Artificial Intelligence: A Research Agenda*, 8 J. Eur. Competition L. & Pract. 361, 361–362.

OECD (2017) *Algorithms and Collusion: Competition Policy in the Digital Age*, www.oecd.org/competition/algorithms-collusion-competition-policy-in-the-digital-age.htm

OECD (2017) *Algorithms and Collusion - Background Note by the Secretariat*, 9 June, DAF/COMP(2017).

OECD (2017) *Algorithms and Collusion: Competition Policy in the Digital Age*, http://www.oecd.org/daf/competition/Algorithms-and-colllusion-competition-policy-in-the-digital-age.pdf

OECD (2017) *Algorithms and Collusion - Background Note by the Secretariat*, 9 June, DAF/COMP(2017).

OECD (2017) Executive Summary of the Roundtable on Algorithms and Collusion Annex to the Summary Record of the 127th meeting of the Competition Committee.

OECD (2000) *Report on hard core cartels*.

Sanchez-Graells, A. (2019). *Data-driven and digital procurement governance: Revisiting two well-known elephant tales*. Available at SSRN 3440552.

Vestager, M. (2017) *Algorithms and competition*, speech, Bundeskartellamt 18th Conference on Competition, Berlin, 16 March https://ec.europa.eu/commission/commissioners/2014-

Yavar Bathaee (2018) *The artificial intelligence black box and the failure of intent and causation*. Harvard Journal of Law & Technology Vol. 31(2).

KONKURRENSVERKET
Swedish Competition Authority

Adress 103 85 Stockholm
Telefon 08-700 16 00
konkurrensverket@kkv.se

www.konkurrensverket.se