Algorithmic and Human Collusion

Tobias Werner*

August 2022

Abstract

I study self-learning pricing algorithms and show that they are collusive in market simulations. To derive a counterfactual that resembles traditional tacit collusion, I conduct market experiments with humans in the same environment. Across different treatments, I vary the market size and the number of firms that use a pricing algorithm. I demonstrate that oligopoly markets become more collusive if algorithms make pricing decisions instead of humans. In two-firm markets, prices are weakly increasing in the number of algorithms in the market. In three-firm markets, algorithms weaken competition if most firms use an algorithm and human sellers are inexperienced.

JEL Classification: C90, D83, L13, L41

Keywords: Artificial Intelligence, Collusion, Experiment, Human–Machine Interaction

*Düsseldorf Institute for Competition Economics (DICE), University of Duesseldorf, Universitätsstr. 1, 40225 Duesseldorf, Germany, werner@dice.hhu.de

I thank Emilio Calvano, Joe Harrington, Adrian Hillenbrand, Matthias Hunold, Timo Klein, Nils Köbis, Ulrich Laitenberger, Alexander MacKay, Jeanine Miklós-Thal, Hans-Theo Normann, Vasilisa Petrishcheva, Catherine Roux, Christopher Snyder and Yossi Spiegel for helpful comments and suggestions. Additionally, I thank Robin Bitter, Leon Heidelbach and Marlene Merker for excellent research assistance. The pre-registration can be found here https://osf.io/yd32b and here https://osf.io/uxdcp. Ethical approval was granted by the German Association for Experimental Economic Research e. V. (No. vzRbKXHq).

1 Introduction

The use of autonomous pricing algorithms is on the rise in various industries.[1] When firms use those tools, the pricing decision for a given product is outsourced from the human decision-maker to a computer algorithm. While in the past most pricing algorithms have been rule-based with rules defined by the seller, there is a recent evolution towards self-learning algorithms (Ezrachi and Stucke 2017). These self-learning algorithms develop the strategies to achieve a specific goal, for instance, maximizing the firms' profits, without explicit instructions.

There are concerns among competition authorities (e.g., Bundeskartellamt and Autorité de la concurrence 2019, Competition & Markets Authority 2021) and academic scholars (e.g., Ezrachi and Stucke 2016, 2017, Mehra 2016) that pricing algorithms could not necessarily learn to price products more efficiently but also that there exists a possibility that they learn to collude tacitly.[2] In other words, algorithms could learn by themselves that tacit collusion benefits the firm.

While recent papers by Calvano et al. (2020a), Klein (2021) show that algorithms can learn to be collusive, it is unclear whether pricing algorithms are more collusive than humans and therefore harm competition. Tacit collusion in traditional markets amongst human decision-makers is a well-documented phenomenon in both empirical and experimental economics.[3] To assess the (anti-)competitive effects of algorithms, it is, therefore, necessary to establish a suitable baseline.

This paper provides a counterfactual for algorithmic collusion for a wide range of possible market compositions and highlights the impact of algorithms on competition. To examine whether commonly used self-learning algorithms make markets more collusive relative to the status quo of human collusion, I apply a two-step approach. In the first step, I consider self-learning pricing algorithms in an extensive simulation study to test whether algorithms learn to set supracompetitive prices and suitable strategies to support those prices as a collusive outcome. Here, I closely follow the approach from Calvano et al. (2020a) but consider a different

[1] The European Commission (2017) finds that two-thirds of sellers in digital markets use pricing tools. Prominent examples are Amazon (Chen et al. 2016b, Musolff 2021) or the gasoline market (Assad et al. 2020).

[2] For recent discussions about the possible policy and legal implications of algorithmic pricing see Kühn and Tadelis (2017), Schwalbe (2018), Calvano et al. (2019, 2020b), Assad et al. (2021), Harrington (2018).

[3] See e.g., Byrne and De Roos (2019), Miller and Weinberg (2017), Borenstein and Shepard (1996), Davies et al. (2011) for empirical and Engel (2015), Horstmann et al. (2018) for expperimental evidence.

market environment that is more tractable. In the second step, I conduct market experiments in which humans compete either against each other or self-learned pricing algorithms.[4] In the experiments, I closely mimic the market environment from the simulations. Across different treatments, I vary the market composition between algorithms and humans and the number of firms in the market. The experimental approach allows me to consider tacit collusion in a controlled setup and study the underlying mechanics. My design enables me to observe humans and algorithms in the same environment and, thus, to analyze whether algorithms promote collusion.

I find evidence that algorithms foster tacit collusion in duopolies. Two-firm markets with algorithms are always more collusive than markets with humans. In "mixed" markets, in which humans and algorithms compete with each other, self-learned pricing algorithms are as good as humans when colluding with the other market participant. Hence, pricing algorithms never promote competition but foster collusion if all firms use one. In triopolies, there exists a non-linear relationship between the number of firms with a pricing algorithm and the level of tacit collusion. Markets in which a single firm uses a pricing algorithm are more competitive than markets with only humans. Yet, as more firms use pricing algorithms, market prices can increase and may even exceed prices in human markets, especially if humans are inexperienced. Similar to Calvano et al. (2020a), Klein (2021), algorithms learn to punish price deviations. As I consider a stylized market environment, I can interpret the strategies of the algorithms. The most successful algorithms learn a win-stay lose-shift strategy that is common for the iterated prisoner's dilemma. The outcomes in mixed markets have a large variance as humans choose heterogeneous strategies when playing against the algorithms.

While there exists reoccurring support for the hypothesis that algorithms can learn to set non-competitive prices and develop reward-punishment strategies (Klein 2021, Calvano et al. 2020a, 2021, Abada and Lambin 2020, Johnson et al. 2020, Asker et al. 2021), it is unclear how algorithmic collusion compares to human collusion.[5] Market environments in previous studies on algorithmic collusion deviate substantially from the setting used in experimental market

[4] Many modern markets do not consist of just algorithms or only humans, but both can interact with each other in the same market environment. For example, according to Chen et al. (2016b), only one-third of the vendors selling the most popular products on amazon.com use some form of pricing algorithm, which gives rise to a mixture in market composition. Also, in the German gasoline industry Assad et al. (2020) identify local markets in which algorithms compete against firms in which arguably human managers make pricing decisions.

[5] Besides self-learned collusion, algorithms could also influence competition by offering better demand predictions (Miklós-Thal and Tucker 2019, O'Connor and Wilson 2021) or by serving as commitment devices (Brown and MacKay 2021, Leisten 2021). Furthermore, Harrington (2021) argues that outsourcing the devel-

games. My design allows me to compare the outcomes of pricing algorithms to human pricing directly as I can observe both in the identical market environment. A recent paper by Assad et al. (2020) identifies the adoption of algorithms in the German gasoline market. Within duopoly markets, price margins rise as both firms in the market begin to utilize an algorithm. The effect is comparable to my findings for two-firm markets. For the market studied by Assad et al. (2020) the exact algorithms are unobservable as they are usually proprietary. The combination of simulations and laboratory experiments enables me to examine human and algorithmic strategies to study the underlying mechanics that may drive those effects.

This research also allows studying cooperation between humans and algorithms, which is a topic in computer science and experimental economics. In computer science, the design of cooperative algorithms in repeated games is an active research area (e.g., Crandall et al. 2018, Lerer and Peysakhovich 2017). Here, cooperative algorithms are often the explicit objective. Similar to Calvano et al. (2020a), I consider a popular self-learning algorithm, which can be attractive as a pricing tool. While cooperation might be an outcome, it is not the initial design objective.

In experimental economics, on the other hand, the focus is often on deterministic algorithms, which do not learn themselves (see March (2021) for a recent literature review). Moreover, collusion in mixed markets with humans and algorithms is rarely studied. A notable exception is a recent paper by Normann and Sternberg (2021). They consider a tit-for-tat algorithm and find that three-firm markets with an algorithm are more collusive than classical human markets. There are no differences in market prices in four-firm markets. The authors vary whether participants know if they play against a computer or a person and find no differences in this domain. My approach differs as I study self-learning algorithms and the limit strategies which they develop by themselves. Furthermore, my design allows analyzing the entire array of market composition as I can observe algorithmic and human markets as well as mixed markets. Hence, I can directly compare algorithmic and human collusion and investigate the effect of pricing algorithms on a wide range of scenarios.

The remainder of the paper is structured as follows. In Section 2, I discuss the main concepts of the algorithm, which I consider in this study. Then, in Section 3, I explain the market environment that I use in all simulations and experiments. After discussing the

opment of algorithms to a third-party developer can affect market outcomes. For a general discussion of the possible effects, algorithms can have on the economy in general see Agrawal et al. (2019).

experimental design in Section 4 and the hypotheses in Section 5, I present my results in Section 6. Section 7 discusses the implications of my findings and concludes.

2 Pricing algorithms

Following the approach by Calvano et al. (2020a) and Klein (2021), I utilize Q-learning algorithms to study the collusive effects of self-learning pricing algorithms.[6] Q-learning is a reinforcement learning algorithm designed to solve Markov decision processes with an ex-ante unknown environment (Watkins 1989). In other words, Q-learning algorithms have to learn everything about the environment by themselves and are not instructed to follow a particular strategy.

Many of the most successful state-of-the-art reinforcement learning algorithms build on the main ideas of Q-learning (e.g., Mnih et al. 2015, Silver et al. 2016, Arulkumaran et al. 2017). Therefore, it appears reasonable to assume that self-learning pricing tools also use some form of Q-learning. As Q-learning is still interpretable in contrast to more modern approaches, it makes it a natural choice when studying algorithmic collusion.[7] In the following, I discuss some of the general concepts in Q-learning.[8]

2.1 Basic Q-learning

Optimization problem In each period t, a Q-learning algorithm, often referred to as an agent, observes the current state $s_t \in \mathbf{S}$ of its environment and chooses some action $a_t \in \mathbf{A}$. Here, $\mathbf{A}$ is the set of feasible actions and $\mathbf{S}$ the set of possible states. Picking the action results in a reward signal $\pi_t \in \mathbb{R}$ and the next state $s_{t+1} \in \mathbf{S}$. The objective of the agent is to maximize the sum of discounted future expected rewards given the current state s_t over $\mathbf{A}$.

[6] Earlier work by Waltman and Kaymak (2008) shows that Q-learning algorithms can converge to non-competitive quantities in a Cournot framework. However, they do not obtain collusion as algorithms also learn this behavior if they are memoryless. Hence, punishment strategies, which are essential for collusion to be sustainable in the long run, could never arise.

[7] While most of the literature on algorithmic collusion focuses on Q-learning algorithms, there are some exceptions. For instance, Hansen et al. (2020) study algorithmic pricing as a multiarmed bandit problem, in which each firm uses an Upper Confidence Bound Algorithm. Supracompetitive prices can arise in this setup as firms tend to run correlated experiments. Recent studies by Hettich (2021), Jeschonneck (2021) consider reinforcement learning algorithms that use function approximation methods.

[8] For a more detailed discussion of Q-learning see Sutton and Barto (2018).

This maximization problem is commonly expressed by the Bellman equation

$$V(s_t) = \max_{a_t}\{\mathbb{E}[\pi_t|s_t, a_t] + \delta\mathbb{E}[V(s_{t+1})|s_t, a_t]\} \tag{1}$$

with $\delta \in [0, 1)$ being the discount rate. The Bellman equation described by Equation 1 is recursive. The value of being in state s_t is given by the current reward signal π_t plus the discounted value of the continuation state s_{t+1}.

Conditional on having perfect knowledge over a stationary environment the agent is interacting in, Equation 1 could be solved using dynamic programming techniques. Yet, in Q-learning, the environment is typically unknown to the agent. Before learning, the agent does not know which actions result in which states or which state-action combinations lead to which rewards. Furthermore, the environment might be non-stationary in the sense that the same state-action combinations in period t may lead to a different reward and another continuation state in a different period t'. Thus, classical dynamic programming techniques, like recursively solving the Bellman equation, do not work given the unknown and possibly non-stationary environment.

In Q-learning, the Bellman equation is rewritten as the Q-function

$$Q(s_t, a_t) = \mathbb{E}[\pi_t|s_t, a_t] + \delta\mathbb{E}[\max_a Q(s_{t+1}, a)|s_t, a_t] \tag{2}$$

In this paper, $\mathbf{A}$ and $\mathbf{S}$ are finite, and hence the Q-function is given by a $|\mathbf{S}| \times |\mathbf{A}|$ matrix, where $Q(s_t, a_t)$ represents the expected net present value of picking action a_t in state s_t.[9] The goal of the Q-learning agent is to repeatedly interact with the environment to iteratively update the cells of its Q-matrix to obtain an approximation for the state-action value $Q(s_t, a_t)$ for each state-action combination.

In all simulations, the Q-matrix is initialized with random numbers drawn from a uniform distribution with support on the unit interval.[10] For each subsequent iteration t, the agent picks some action a_t conditional on the current state s_t, which yields π_t and s_{t+1}. Then, the Q-matrix gets updated as the weighted average of the past estimate of $Q(s_t, a_t)$ and the newly

[9] Note that the $V(s_t) \equiv \max_{a_t} Q(s_t, a_t)$.

[10] Klein (2021), Abada and Lambin (2020) initialize the Q-matrix with zeros. Calvano et al. (2020a), Johnson et al. (2020) use an initialization that corresponds to the discounted profit if all firms randomize their prices. Calvano et al. (2020a) show that outcomes are similar for different initialization of the Q-matrix.

learned value

$$Q_{t+1}(s_t, a_t) = (1-\alpha)Q_t(s_t, a_t) + \alpha(\pi_t + \delta \max_a Q_t(s_{t+1}, a)) \tag{3}$$

where $\alpha \in (0, 1)$ is referred to as the *learning rate.* For the subsequent states the same procedure is applied in each iteration until some convergence rule is met.

Exploration versus exploitation When selecting the action in s_t, the agent faces a trade-off. On the one hand, picking the action $a_t^* = \arg\max_a Q(s_t, a)$ yields the highest expected payoff given the current approximation of the Q-matrix. On the other hand, the agent can explore the environment only by picking new actions. By exploring the action space for each state of the Q-matrix the agent gradually learns the value of each state-action combination. As a result, the agent may find actions, which were underestimated beforehand. To balance exploration and exploitation, I use the ε-greedy algorithm. When using the ε-greedy algorithm, the current optimal action a_t^* is picked with probability $(1-\varepsilon_t)$ with $\varepsilon_t \in [0, 1]$. With the probability ε_t the algorithm selects a random action from $\mathbf{A}$. I follow the approach by Calvano et al. (2020a), Johnson et al. (2020) and define $\varepsilon_t = e^{-\beta t}$ for some small $\beta > 0$. Note that ε_t decays over time. At the beginning of the learning process, when the Q-matrix is rather uninformative about the value of picking an action in a specific state, the agent picks actions at random with a high probability and explores the action space. Overtime ε_t decreases and the agent chooses the action, which offers the highest expected long-run reward, more often. Eventually, the agent does not explore anymore but always picks the action with the highest expected value given that $\lim_{t\to\infty} \varepsilon_t = 0$.

2.2 Simulation setup

Learning and convergence In each simulation, the agents play against independent copies of itself and learn by interacting simultaneously with each other. For a stationary environment, Q-learning agents converge to the optimal solution under mild conditions.[11] For my design, this is, however, not the case as multiple agents learn in the same environment at the same

[11] For a stationary Markov decision process, Q-learning converges to the optimal policy when using the ε-algorithm if the rewards signals are bounded. Furthermore, the learning rate must decrease over time, the sum of learning rates must diverge, while the squared sum of learning rates must converge. For a proof see Watkins and Dyan (1992).

time. The environment of each agent is influenced by the decisions of other agents taking actions and learning at the same time by picking action simultaneously. Furthermore, the learning procedure when using the ε-greedy algorithm is stochastic for finite t. Given this constant change in strategies for the competitors, the environment is non-stationary, and thus convergence is not guaranteed. Therefore, I rely on simulations to derive results on algorithmic collusion in this non-stationary setup.

To determine the state of convergence, in each training iteration, I follow the approach by Calvano et al. (2020a) and look at the cells of the Q-matrix. If the best action for each state does not change for 100,000 subsequent periods, I assume that the agents in the market have converged and found a stable strategy.

Uniqueness For a given environment, three hyperparameters define the learning process of the agent: α, β and δ. The learning rate α controls how much the agent values new information relative to the current approximation of the Q-matrix. For large values of α, the agent does not put much weight on past interactions with the environment. If α is small, the agent does not learn much from the newly arriving information in each period. The value of β captures the extent of exploration of the agent. The larger β, the faster ε_t converges to zero. Lastly, the discount rate δ captures the importance of future rewards. All three hyperparameters are not learned by the agents but are set exogenously. While there are some rules of thumb, their choice remains essentially unclear from a theoretical perspective. Given that an agent's outcome usually depends on those hyperparameters, the state of convergence is not unique for a given environment. This problem gets amplified further as the learning process of all agents is stochastic, which hinders uniqueness.

In the entire study, I keep the discount rate fixed at $\delta = 0.95$. It corresponds to the continuation probability in the market experiments with human participants described in Section 4.2. For α and β, I consider a parameter grid with $\alpha \in [0.025, 0.25]$ and $\beta \in [1 \times 10^{-8}, 2 \times 10^{-5}]$ with 100 points in each dimension evenly spaced from one another. For each grid-point, I simulate 1,000 distinct markets which differ in the underlying stochastic process. Hence, I run in total 10,000,000 simulations for each market size. The grid and the simulation setup, again, follows Calvano et al. (2020a) and Johnson et al. (2020). I evaluate the results from this grid simulation using different performance measures, which I describe below. Furthermore, I use the grid to find a specific algorithm that competes with humans in the experiments.

2.3 Performance evaluation

Profitability and prices Given the definition of the Bellman equation, $V(s)$ provides an unbiased estimate of the expected sum of future discounted rewards for a given state s.[12] Thus, $V(s)$ is a natural measure for the profitability for agent i in a given state s. I utilize this direct interpretability and use $V(s)$ as a profitability measure. Additionally, I consider the (average) market price upon convergence as a way to measure tacit collusion. Market prices are of particular interest when comparing outcomes in algorithmic to experimental markets with humans as the value function is unobserved here.

Optimality High profitability alone does not necessarily imply that the agent has learned an optimal strategy against its competitors. To derive a measure of how well the agent does in comparison to how well it could do, I derive the optimality of the agent for state s as

$$\Gamma_i(s) = \mathbb{1}\{\arg\max_a Q_i(s,a) = \arg\max_a Q_i^*(s,a)\} \tag{4}$$

where $Q_i^*(s,a)$ is the optimal Q-matrix assuming that the opponents play their limit strategy. Here, $\mathbb{1}$ denotes the indicator function.

I can obtain this optimal Q-matrix in the following way: After convergence, the strategies of the other agents, which learned in the same environment, are held constant. Thereby, the environment becomes stationary as the state-transition probabilities do not change anymore. Next, I initialize a new agent, which competes against the limit strategy of the opponents. I utilize dynamic programming to repeatedly iterate over each state-action combination of its Q-matrix until convergence. Within this stationary environment, convergence is guaranteed. The Q-matrix of the new agent corresponds to the optimal Q-matrix $Q_i^*(s,a)$ that the actual agent could have learned against the limit strategy of the other agents.

[12] I evaluate the performance of the agent only upon convergence. Therefore, for ease of notation, I omit the period subscript t.

Note that $\Gamma_i(s) = 1$ implies that the agent has learned to play a Nash equilibrium for state s. The agent has learned a subgame perfect Nash equilibrium if and only if $\overline{\Gamma} = \frac{1}{|\mathbf{S}|}\sum_s \Gamma_i(s) = 1$. For $\overline{\Gamma} < 1$ there are states for which the agent did not learn to play a Nash equilibrium.

Selecting a specific algorithm I study experimental treatments in which humans compete against algorithms in the same market. To conduct laboratory experiments in those "mixed" markets, I have to decide on a specific parameterization of the agent with a specific stochastic process. To select an algorithm for this purpose, I take the perspective of a firm that would want to deploy a pricing algorithm to a market. It is reasonable to assume that this firm would want the pricing algorithm to be (i) profitable and (ii) optimal in the sense that it is not easily exploitable by other market participants. Considering both objectives in isolation is not sufficient when selecting an agent for the deployment to the market. Importantly, high profitability does not necessarily imply high optimality or vice versa. It is vital to rule out that the high profitability is driven by a lack of sophistication of the algorithms.[13] For instance, the agents in the environment could jointly converge to a seemingly collusive outcome in which they price at the monopoly price but fail to learn a strategy that accounts for certain deviations. Such a myopic strategy is unlikely to perform well against new competitors, which would result in a lower profitability than in the simulation. When both performance measures are maximized jointly, the agent has learned a profitable strategy that also accounts for the strategic element of the environment. Accordingly, it increases the likelihood that the algorithms perform well against new (possibly human) competitors. Therefore, I propose the following criterion for agent i that combines both performance measures

$$\Psi_i = \frac{\overline{\Gamma}_i}{|\mathbf{S}|}\sum_s V_i(s). \tag{5}$$

Since $\overline{\Gamma}_i \in [0, 1]$, the selection criterion is the average profitability over the entire state space shrunk towards zero by the degree of suboptimality. Hence, $\overline{\Gamma}_i$ can be interpreted as a shrinkage penalty in this context. The intuition is that high profitability is only valuable if other players cannot exploit the agent easily with a possibly more sophisticated strategy. In the simulation, algorithms usually converge to a specific state. Within the experiments with humans, different states are potentially relevant as human and algorithmic strategies can differ. For that reason,

[13] The situation can arise, for example, if the exploration of the agent is limited.

I consider the entire state space and not only the state of convergence when defining Ψ_i. For treatments in which algorithms interact with humans, I select the algorithm from the simulation in which $\overline{\Psi} = \frac{1}{N}\sum_i^N \Psi_i$ is maximized over the parameter grid for α and β. I will refer to it as the selected algorithm.

In mixed markets, the algorithms learn "offline" in advance. Put differently, they develop their strategies in a simulated market environment against other algorithms. Once they interact with humans in the actual experimental market, they do not learn any more from new market information but use the strategy obtained during the training in the simulation.[14] While this approach might appear restrictive, it is arguably realistic. Q-learning is a slow learning algorithm as it updates only one cell of the Q-matrix in each training step. Furthermore, each cell has to be visited by the agent multiple times to obtain an accurate estimate of the value of this state-action combination. As a result, Q-learning algorithms usually learn too slow to be trained "online", meaning in the actual market environment. Furthermore, this learning strategy has been used in other groundbreaking Q-learning applications, and it is a common standard for successful reinforcement learning applications.[15]

Also, from an industry perspective, offline training makes intuitive sense. A firm using a pricing algorithm would likely want to evaluate its performance before its deployed to the market to avoid any possible loss given a potential suboptimal pricing strategy. The risk that the agent learns a suboptimal strategy can be partially mitigated by offline training as an ex-ante evaluation is feasible.[16]

3 Market environment

I consider a stylized Bertrand market environment, which is commonly used in the experimental economics literature on collusion (see, for instance, Fonseca and Normann 2012, Horstmann et al. 2018).

[14] Thus, I do not consider how humans and algorithms may interact during the learning process. Yet, it can be an interesting avenue for future research.

[15] For instance, AlphaGo, which is a reinforcement learning algorithm that outperforms humans in the board game Go, uses offline learning (Silver et al. 2016). For a discussion of offline learning also see Calvano et al. (2020a).

[16] When using offline training for pricing tools that build on reinforcement learning algorithms, the market environment has to be known to the developing firm. However, with modern tools in demand estimation and supervised machine learning, it is reasonable to assume that firms can derive this environment.

There are $N \in \{2, 3\}$ firms in the market, which face a perfectly inelastic demand function and have zero marginal costs. Each firm produces the same homogeneous good. The market consists of $m = 60$ computerized consumers, which are all willing to purchase exactly one unit of this good in each round and have a maximum willingness to pay of $\overline{p} = 4$. The price of firm i in period t is denoted by $p_t^i \in \mathscr{P} := \{0, 1, 2, ..., 5\}$. Consumers buy the good at the lowest offered price. If multiple firms offer the lowest price in a given round, the market is shared equally. Firms are always either represented by a human or by a Q-learning algorithm. This market environment is the same for the simulation study and all experimental treatments. It allows me to directly compare the simulation and outcomes and derive a counterfactual for algorithmic collusion. In the simulation treatments, firms compete in an infinitely repeated game with a discount rate of $\delta = 0.95$. To mimic the features of an infinitely repeated game in the experimental treatments, I use a repeated game with random stopping, where the continuation probability for playing another round is given by 95% (Roth and Murnighan 1978). While this environment is less complex than many actual markets, it yields a suitable setting for my design as it distills the main components of price competition when studying collusion.

There exists a stage game Nash equilibrium at $p^{NE} = 1$. The monopoly price of $p^M = \overline{p} = 4$ maximizes joint profits.[17] When all firms charge the same price, the profit is given by $\pi_t^i = p\, m/N$. The profit for a single deviating firm is $\pi_t^i = p\, m$. Collusion is sustainable at the monopoly price for the given discount factor, for instance by grim-trigger strategies. Crucially, the environment is stylized and tractable, which allows for an analysis of the strategies algorithms learn in the game. Furthermore, it is arguably easy to understand for experimental subjects due to its simple mechanics. Also, the environment offers a different extension to algorithmic price competition to markets with a perfectly inelastic demand function, which has not been studied before.

Q-learning and the market environment If a firm uses a Q-learning algorithm, the algorithm takes over the pricing decision in each period. Hence, the action a_t of the agent corresponds to a price and the set of possible actions corresponds to the price set.

[17] There are two prices ($p = 5$ and $p = 0$) which are (weakly) dominated. I include both in the set of possible prices $\mathscr{P}$ to rule out that convergence to the boundaries of the price set is equivalent to collusion or competition at the stage game Nash equilibrium.

Similar to Calvano et al. (2020a), Johnson et al. (2020), I define the state of the environment for each agent as the set of past prices from the previous period $s_t = \{p_1^{t-1}, ..., p_N^{t-1}\}$. Notably, this state representation corresponds to memory-one strategies, which are predominantly used by humans in the prisoner's dilemma and market games (Dal Bó and Fréchette 2019, Romero and Rosokha 2018, Wright 2013). Calvano et al. (2020a) also consider state representations that allow for a two-period memory. This larger memory does not improve the collusive abilities of the algorithms. Importantly, it is straightforward to construct memory-one strategies that make collusion incentive-compatible in two and three-firm markets.[18]

The economic profit obtained in each respective period is the reward signal for the Q-learning algorithms.

4 Experimental design

4.1 Treatments

I consider five experimental treatments and two treatments based on simulations. Across treatments, I vary the market composition between algorithms (A) and humans (H) and the number of firms in the market (see Table 1). I label the treatments with the number of human firms followed by the number of firms that use an algorithm. For example, treatment 2H1A stands for two human players and one algorithmic player operating in one market. Thus, I consider treatments without any algorithms (2H0A and 3H0A) and without any humans (0H2A and 0H3A).[19] Comparisons between those treatments reveal whether algorithms are

Table 1: Treatment composition

Number of Human Players	Number of Algorithms			
	0	1	2	3
3	3H0A	-	-	-
2	2H0A	2H1A	-	-
1	-	1H1A	1H2A	-
0	-	-	0H2A	0H3A

[18] For instance, consider a memory-one strategy that mimics a grim-trigger strategy in the sense that the agent always plays the monopoly price in the state $s = (p^M, p^M, p^M)$ but chooses the stage game Nash equilibrium in any other state. This strategy is a possible outcome for the simulations from an ex-ante perspective and makes collusion sustainable for the given discount factor of $\delta = 0.95$.

[19] Note that the latter are simulation studies.

more collusive than humans. Additionally, I consider treatments in which humans compete against algorithms (1H1A, 1H2A and 2H1A). I utilize those treatments to examine the way humans and pricing algorithms interact with each other. Furthermore, they show if an increase in the share of algorithms in the market fosters tacit collusion for different market sizes.

4.2 Procedure

Each experimental treatment is repeated for three supergames to observe learning effects. Within each supergame, there is a fixed group composition. Across supergames, I use a perfect stranger matching scheme. This matching scheme is common knowledge. Hence, participants know that they interact with each person only within one supergame throughout the entire experiment. It rules out any reputation effects that could be present elsewise.

In my experiment, each round has a continuation probability of 95% in each supergame. Hence, with a 5% chance, a given supergame ends after each respective round. The instructions mention the continuation probability to the subjects at the beginning of each supergame. It corresponds to the discount rate of $\delta = 0.95$ that is used for the algorithms in the simulation treatments.[20] To allow for different experimental sessions with the same supergame lengths, the round numbers are pre-drawn with a random number generator.[21] At the end of each round, participants receive information about all prices in the market. Furthermore, they see their own profit in the given round.

Each participant has complete information on which firms use a pricing algorithm. While this is arguably not the case in actual markets, Normann and Sternberg (2021) show that participants are insensitive to the knowledge of playing against an algorithm or a human player.

Following Normann and Sternberg (2021), all profits obtained by a firm that uses a pricing algorithm are given to a passive human player, who does not take any active decision. It rules out any differences in social or distributional preferences that might arise elsewise across treatments.

The framing regarding the algorithmic decisions in the experiment is neutral. In all treatments, participants do not know the objective of the algorithm, they do not know that it is

[20] For a risk-neutral player the continuation probability is theoretically equivalent to the discount rate (see for instance Roth and Murnighan 1978, Dal Bó 2005).

[21] The exact round numbers are 25 (supergame 1), 17 (supergame 2) and 11 (supergame 3).

self-learned or how it learned its strategy.[22] Subjects receive the instructions at the start of the session, but they are also available during the experiment at any point. After the participants read the instructions, I ask them a set of control questions.[23] If a participant gives three times a wrong answer, I show an additional explanation for the respective question. One person dropped out of the experiment due to technical problems. I exclude the entire matching group of this subject from the analysis.

As described in Section 3, the algorithms always condition their current pricing decision on a state which is the set of prices from the previous period. In the first round of each supergame, the algorithm has no state to condition upon as the state $s_{t=0}$ is undefined. To circumvent this initial condition problem, I define $s_{t=0}$ as the state of convergence from the learning process. Thus, the algorithm always begins each supergame with the same action it played last in the simulated environment.

The experiments were conducted online in May and June 2021. I recruited the participants using ORSEE (Greiner 2015) from the subject pool of the DICE Lab, University of Düsseldorf. A web-conference call, in which participants could ask clarifying questions and receive technical assistance if required, accompanied each session.[24] In total, 313 participants were recruited with between 60 to 64 subjects in each experimental treatment (see Table 2). For each treatment without any humans, I use 1,000 independent simulation runs for each parameterization of the algorithms as the respective comparison unit.

Each session lasted for approximately 30 minutes, and subjects earned on average 11.3 Euro, including a show-up fee of 4 Euro. Within the experiment, I used an experimental currency unit (ECU) where one Euro corresponded to 130 ECU. The experiment employed a between-subject design and thus, each subject participated only in one treatment. I programmed the experiment in oTree (Chen et al. 2016a).

[22] It mimics the information structure in actual markets, in which firms do not know much about the algorithms of other market participants. Yet, varying the information participant get about the objective and the learning process of the algorithm can be an interesting path for future research.

[23] See Appendix A for the full set of instructions, the control questions, and screenshots of the relevant decision screens.

[24] The procedure is similar to Zhao et al. (2020), Danz et al. (2021). Li et al. (2021) find that this procedure offers comparable results to lab experiments in different economic games.

Table 2: Number of observations by treatment

Treatment	Number of participants	Number of independent observations
3H0A	63	7
2H0A	60	10
1H1A	64	32
1H2A	63	21
2H1A	63	7
0H2A	-	1,000
0H3A	-	1,000

* The number of independent observations for the experimental treatments refers to the last supergame. It is determined by the size of the matching group. A matching group consists of six (nine) firms for markets with two (three) firms. Since the algorithms do not learn anymore during the experiment and the respective participants do not take any active decisions, there are no reputation effects across super games. Hence, I exclude the algorithmic firms from the matching scheme.

5 Hypotheses

The experimental design allows for the comparison of human and algorithmic collusion for different market compositions. Furthermore, I can investigate the interaction between humans and pricing algorithms if both populate the same market. As a measure of the degree of tacit collusion I use the respective market price. Within the experimental treatments, an independent observation is the average market price for a given matching group. For the simulations, I average the market prices for each independent simulation over 1,000 rounds after convergence. All hypotheses and the corresponding statistical tests have been pre-registered.[25]

From a theoretical perspective, punishment strategies are vital for collusion to be sustainable in the long run (Friedman 1971, Abreu 1988). While humans often fail to employ punishment strategies that appear desirable from the theoretical perspective[26], Calvano et al. (2020a), Klein (2021) find that self-learned pricing algorithms learn harsh punishment strategies that make collusion incentive compatible. I expect the pricing algorithms in my design to

[25] The pre-registration uses the template from *AsPredicted.org* and can be found here https://osf.io/yd32b and here https://osf.io/uxdcp. Ethical approval was granted by the German Association for Experimental Economic Research e.V. (No. vzRbKXHq).

[26] For a discussion of the strategies that humans use in experimental market games see for instance Wright (2013).

learn comparable punishment strategies, which would theoretically foster collusion compared to lenient strategies often used by humans.

Algorithms may also reduce the strategic uncertainty within the game. After convergence, algorithms follow the same strategy for all of the subsequent rounds. In the mixed market treatments, the algorithm plays according to their limit strategy against humans. Hence, they play the strategy that they learned in the simulations. Crucially, after the learning process of the algorithms, their strategy is deterministic and does not change anymore. Normann and Sternberg (2021) argue that playing against a deterministic algorithm reduces strategic uncertainty compared to playing against a human, who might change the strategy during the game and are less committed to a particular behavior. They demonstrate theoretically that the postulated reduction in strategic uncertainty fosters collusion. In line with this prediction, they show that in experimental market games where humans compete against a tit-for-tat algorithm, markets become more collusive compared to markets with only humans. As the algorithms in my design are also deterministic after convergence, I expect higher degrees of tacit collusion in mixed markets relative to human markets. Since humans first have to learn about the algorithm's strategy, it appears natural that this effect especially materializes in later supergames. Harsher punishment strategies and the reduction in strategic uncertainty by algorithms should both foster collusion. Thus, I hypothesize that market prices increase as more firms use a pricing algorithm for a given market size.

Hypothesis 1. *The level of tacit collusion increases in the share of firms using self-learned pricing algorithms.*

It is a well-documented finding in the literature on experimental market games that tacit collusion becomes less likely as the number of firms in the market increases (Engel 2007, Huck et al. 2004, Harrington et al. 2016). Within my design, a larger market size implies higher deviation profits. That, in turn, increases the incentive to deviate from a collusive price level. Furthermore, the strategic complexity of the game grows as the number of firms increases. With more firms in the market, market participants have to condition their behavior on additional factors such as the previously chosen prices from the extra competitor. This increase in strategic complexity may further hinder collusion.[27] Similar to the findings in

[27] For a discussion on the influence of strategic complexity on cooperation see Jones (2014), Gale and Sabourian (2005).

experimental market games, Calvano et al. (2020a), Johnson et al. (2020) find decreasing prices in algorithmic markets in their simulations as the number of firms increases. I expect comparable results in my experimental design, which leads to the following hypothesis.

Hypothesis 2. *The level of tacit collusion decreases in the number of firms in the market for human and algorithmic markets alike.*

Note that it is unclear how those number effects differ between algorithmic and human markets. While the decline in market prices in the previous studies on algorithmic collusion appear smaller than in human markets, the market setup deviates substantially from the environments usually used in experimental market games. Hence, it is an open question whether algorithms are better at colluding as the market size increases. I investigate this question in the following sections.

6 Results

In this section, I discuss the results and examine whether algorithms foster tacit collusion for different market compositions. I begin by considering the performance measures for the algorithms in Section 6.1. Then, I discuss the exact strategies that the selected algorithms learn upon convergence in Section 6.2. In Section 6.3, I investigate how algorithmic collusion compares to human collusion. Lastly, I discuss the results on mixed market compositions in Section 6.4 to shed light on the interaction between pricing algorithms and human decision-makers if both populate the same market at the same time.

6.1 Performance of the algorithms

In the following, I discuss the performance of algorithm in the simulation treatments without any humans for duopolies (0H2A) and triopolies (0H3A).

Profitability Figure 1 shows the profitability of the algorithms in the state of convergence[28] for a parameter grid over the learning rate α and the exploration decay β for the simulations 0H3A and 0H2A.[29] As a benchmark, I provide the value function under collusion V^C at the

[28] While it is not guaranteed, the algorithms always convergence using the respective convergence criterion defined in Section 2.2.

[29] Computational support and infrastructure was provided by the "Centre for Information and Media Technology" (ZIM) at the University of Düsseldorf (Germany).

monopoly price and under competitive pricing V^{NE}. Those can be derived by considering the individual fixed profit π in those states and then rewriting the Bellman equation as the arithmetic series $V = \frac{1}{1-\delta}\pi$. Lighter colors in Figure 1 show a profitability that is close to collusion at the monopoly price, while darker colors indicate that the algorithms are not profitable.

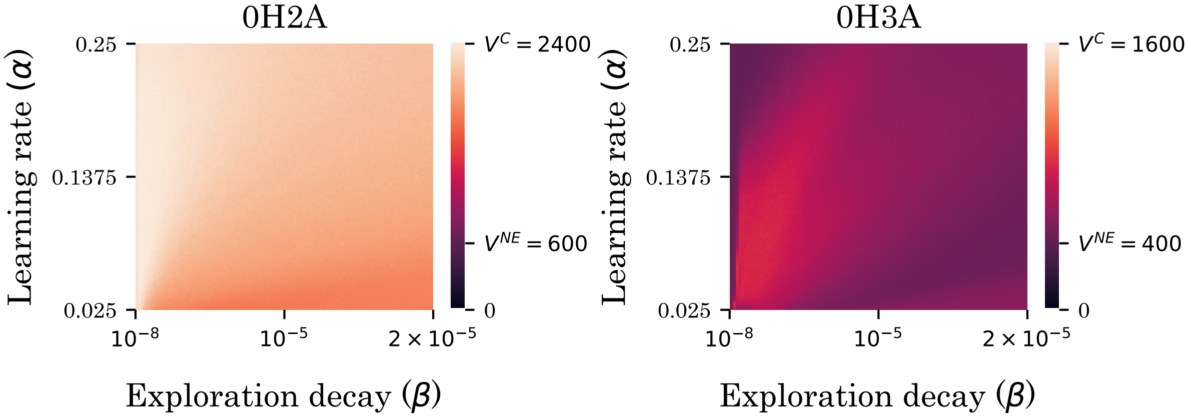


Figure 1: Profitability of the Q-learning agents in the state of convergence averaged over 1,000 simulation runs for different grid points.

In both treatments, the algorithms have a high profitability after convergence. For most grid points, the algorithms learn a strategy that is more profitable than with competitive pricing. This becomes evident by comparing the value function for the grid points in Figure 1 to V^{NE}. Notably, the profitability is usually greater for 0H2A compared to 0H3A. This pattern is also confirmed when considering the average prices the algorithms play upon convergence shown in Figure 2.

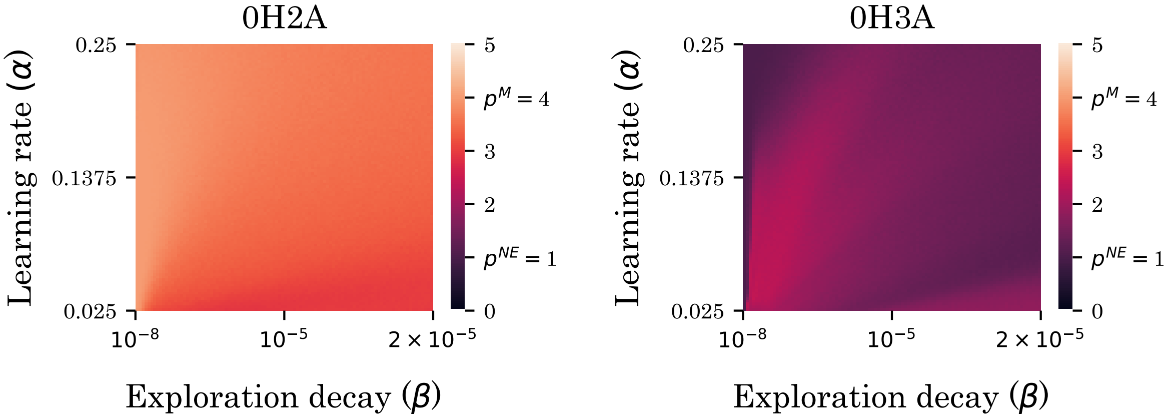


Figure 2: Average market price of the Q-learning agents after convergence averaged over 1,000 simulation runs for different grid points. This market price average is obtained by considering 1,000 periods after convergence.

On average, the algorithms learn to set non-competitive prices for a wide range of parameterizations. For each grid point, the average market price is above the stage game Nash equilibrium in 0H2A. It is also the case for 99.2% of all grid points in 0H3A. Hence, in both treatments market prices are above the competitive benchmark. Notably, the market prices in 0H2A are on average higher compared to 0H3A. While in 0H2A the average market price is above $p = 3$ for more than 93.4% of all grid points, it never exceeds $p = 2.4$ in 0H3A. Indeed, for each grid point, the average price in 0H2A is statistically significantly higher than in 0H3A (Two-sided Mann–Whitney U test, p<0.01 for each grid point separately). Accordingly, the level of tacit collusion is higher in two-firm algorithmic markets. This result is in line with Hypothesis 2 and previous findings on algorithmic collusion.

Result 1. *Algorithms learn to set non-competitive prices in two and three-firm markets. In these markets, tacit collusion is significantly higher with two than with three firms.*

Optimality Figure 3 shows the share of all simulation runs in which both algorithms converge mutually to a Nash equilibrium.

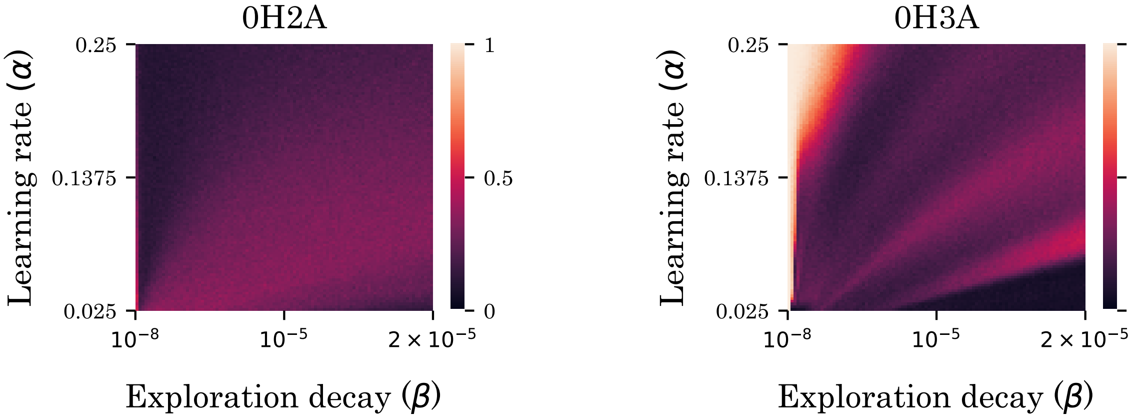


Figure 3: Share of simulations that converge to a Nash equilibrium.

On average, learning to play a Nash equilibrium appears difficult for the algorithms in 0H2A and 0H3A. While for certain parameterizations the algorithms manage to play a mutual best response, the optimality measure is below one for most grid points. Also, in comparison to previous findings by Klein (2021), Calvano et al. (2020a) the share of outcomes converging to a Nash equilibrium seems smaller. The market environment with a perfectly inelastic demand function proves to be challenging for the algorithms. A possible reason for this is that small changes in prices lead to drastic shifts in profits, which may hinder a smooth convergence to an equilibrium strategy.

6.2 Strategies of the algorithms

Next, I examine the limit strategies that the algorithms learn once they converge. I focus on the parameterizations of the algorithm that perform best, given the selection criterion discussed in Section 2.3.[30] Thus, I concentrate on the algorithm that is likely to have high profitability while being harder to exploit by other market participants. It appears reasonable to assume that a firm would select such an algorithm when deploying a pricing tool to an actual market environment.

Punishment behaviour I consider the average punishment behavior of the algorithms upon a deviation by another market participant. High price levels alone are not proof of algorithmic collusion as learning to play those prices could be merely myopic. In other words,

[30] For 0H2A those parameters are $\alpha \approx 0.027$ and $\beta \approx 1 \times 10^{-8}$. The values for 0H3A are $\alpha \approx 0.029$ and $\beta \approx 6.16 \times 10^{-7}$.

the algorithm could have learned to play certain prices without developing an underlying understanding of the strategic components of the market environment. From a theoretical perspective, punishment strategies are vital for collusion to be sustainable in the long run. Thus, to confirm that the market price of the algorithms are indeed collusive, it is essential to consider their behavior after deviations by other market participants from a potentially collusive outcome.

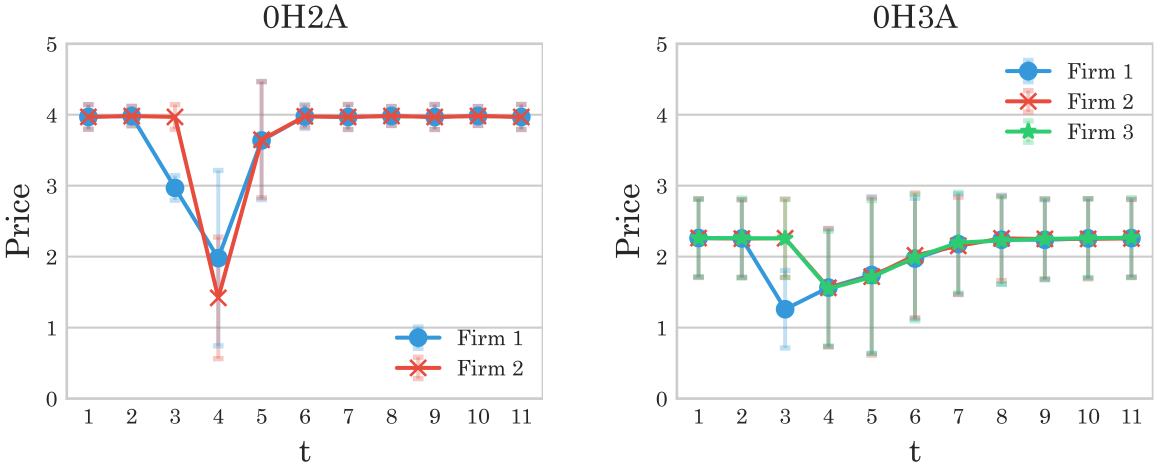


Figure 4: Punishment behavior of the algorithms after convergence. Starting from the state of convergence, the algorithms play according to their limit strategy. I induce an exogenous deviation from Firm 1 in $t = 3$ to observe the reaction of the other firms. I use 1,000 independent simulation runs. The error bars represent the standard deviation.

The left-hand side of Figure 4 shows the behavior of the algorithms in 0H2A after a deviation by one of them. In the first two periods, the algorithms play according to their limit strategy. Then, in period $t = 3$, I force Firm 1 to deviate by undercutting the price of the competitor. The deviation price is always just below the price it would have played according to its limit strategy. Thus, Firm 1 always chooses the most profitable one-period deviation.[31] Afterwards, I allow both algorithms again to play according to their limit strategy to observe their response to the deviation.

In the initial two periods, algorithms choose prices that are close to the monopoly level. After the exogenously induced deviation of Firm 1 in the third period, Firm 2 lowers its price below the price of Firm 1. Then, after this phase of lower prices, both firms revert to the initial price level within the next couple of periods. The behavior of the algorithms is consistent with

[31] In more than 99% of simulation runs the algorithms learn to play a price above the stage game Nash equilibrium.

a punishment scheme. By undercutting the price of Firm 1 after the deviation, Firm 2 makes the deviation from the initial price level unprofitable. Indeed, for 81.4% of all simulation runs, algorithms learn a limit strategy that makes deviations as shown in Figure 4 unprofitable. Thus, algorithms do not only learn to play high prices but also strategies that make collusion incentive compatible.

The right-hand side of Figure 4 shows the results for the same exercise for 0H3A. As discussed in Section 6.1, prices are significantly lower in three-firm markets compared to two-firm markets. Similarly to 0H2A, the other firms in the market decrease their price after the deviation by Firm 1. After a punishment phase of multiple periods, the algorithms return to the initial price level that they played before the deviation.[32] The punishment behavior of the algorithms is incentive compatible in 62.7% of all simulation runs.

Result 2. *Algorithms in 0H2A and 0H3A learn punishment strategies that can make collusion incentive compatible.*

Limit strategy of the selected algorithm In the previous sections, I considered the average behavior of the algorithms with a fixed parameterization over multiple simulation runs with different underlying stochastic processes. Next, I present the exact limit strategy of the algorithm which maximizes the selection criterion Ψ as discussed in Section 2.3 in 0H2A and 0H3A. It is also the strategy the algorithm will use within the experimental treatments with humans.

Equation 6 describes the core idea of the strategy. It is nearly identical for the algorithm in two and three-firm markets.[33]

[32] While the average punishment strategy appears like a smooth transition between periods of punishment and cooperation, it is usually not the case for each simulation in isolation. Large and sudden price jumps after deviations are common. The transition only appears smooth when averaged over all simulation runs.

[33] There are minor differences between the strategies in 0H2A and 0H3A. Namely, in 0H3A, there is a small number of states that trigger a different response by the algorithm after deviations from the monopoly price. For instance the state $s_t = (4, 3, 0)$ leads to $a_t = 4$ or $s_t = (4, 4, 2)$ yields $a_t = 3$. However, those states are never reached after the algorithms converged. Furthermore, in mixed market experiments, those states only account for approximately 1% of all rounds. Additional details are provided in Appendix B.2.

$$p_i^t(s^t) = \begin{cases} p^M & \text{if } s^t = \{p_i^{t-1} = p^M | \forall i\} \\ p^M & \text{if } s^t = \{p_i^{t-1} = p^{NE} | \forall i\} \\ p^{NE} & \text{otherwise} \end{cases} \tag{6}$$

Upon cooperation at the monopoly price p^M in the previous period, the algorithm always chooses the monopoly price again. Any deviation from the cooperative outcome is punished by playing the stage-game Nash equilibrium p^{NE}. If and only if all firms played p^{NE} in the previous period, the algorithm reverts to playing p^M. In every other relevant state, the algorithm plays p^{NE}.[34] While algorithms in 0H2A learn this strategy frequently, it only arises occasionally in 0H3A, which is also indicated by the overall lower price level in this treatment.

Interestingly, this strategy is similar to the win-stay, lose-shift strategy (WSLS) discussed by Nowak and Sigmund (1993) in the context of the iterated prisoner's dilemma. Whenever an agent uses WSLS in the iterated prisoner's dilemma, she conditionally cooperates. Upon any deviation, the agent defects and reverts back to cooperation if and only if both players defected in the previous period. WSLS has several desirable properties from an (evolutionary) game-theoretical perspective.[35] If actions are noisy, WSLS can correct for unintended deviations when playing with another agent that uses WSLS. That is not the case for other popular strategies like tit-for-tat. Furthermore, WSLS can detect and exploit unconditional cooperators after unintended deviations, which may arise if the action implementation is noisy. However, depending on the exact payoff structure, agents that always defect can exploit WSLS. Nowak and Sigmund (1993) show that WSLS arises naturally as the most widespread strategy in an evolutionary simulation in a noisy iterated prisoner's dilemma.[36]

The strategy of the algorithm is as an application of WSLS to the market environment. Similar to WSLS in the iterated prisoner's dilemma, the selected Q-learning algorithm restricts its attention to two actions: Cooperation at p^M and defection at p^{NE}. Just as the classical

[34] This refers to all possible states that are reachable given the limit strategy of the algorithm. Thus, I do not consider states that would require the algorithm to play prices that it never plays itself when following its limit strategy.

[35] For a discussion of win-stay, lose-shift also see Imhof et al. (2007), Posch (1999).

[36] For Q-learning algorithms, actions are also implemented with noise during the learning process as the exploration of the environment is stochastic. Hence, there might exist a possible relation between the evolutionary processes that lead to WSLS in simulation studies and the learning of Q-learning agents.

WSLS, the algorithm only cooperates if all firms in the market jointly played p^M or p^{NE} in the previous round and defects elsewise.[37]

Importantly, the WSLS strategy does not arise by construction but as a result of the learning procedure of the algorithms. Even with a state representation that is restricted to the prices of the previous period, it would be straightforward to construct strategies that punish deviations for more than one period.[38] However, the algorithms do not coordinate on strategies that outperform WSLS.

Result 3. *The algorithm that maximizes the selection criterion learns a win-stay lose-shift strategy.*

While this strategy can correct unintended deviations and punish intended deviations by other firms, it is also possible to construct strategies that exploit the algorithm. As an example, consider a three-firm market where two firms use the strategy of the algorithm described by Equation 6 and firm k uses the following strategy

$$p_k^t(s^t) = \begin{cases} p^D = 3 & \text{if } s^t = \{p_i^{t-1} = p^{NE} | \forall i\} \\ p^{NE} & \text{otherwise} \end{cases} \tag{7}$$

Given that the algorithm always plays p^M after all firms played p^{NE} in the previous round, the strategy by firm k triggers the algorithms to cooperate every other round only to exploit their cooperative phase by choosing the most profitable deviation p^D. It is straightforward to show that in an infinitely repeated game with $\delta = 0.95$ the strategy of firm k strictly dominates cooperation at the monopoly price in three-firm markets. Note that this strategy is however dominated by always cooperate for two-firm markets.[39] While Q-learning algorithms never learn the strategy described by equation 7 during their simultaneous and dynamic learning process, I will analyze if humans manage to exploit the limit strategy of the algorithm in mixed markets in Section 6.4.

[37] Calvano et al. (2019) find that Q-learning algorithms often learn similar one-period-punishment strategies in the iterated prisoners dilemma.

[38] A simple example is a strategy that mimics the behavior of a grim-trigger strategy. Yet, also other strategies are feasible. For instance, consider strategies that do not revert to the monopoly price immediately after playing the stage game Nash equilibrium but to intermediate values of the price set.

[39] For details see Appendix B.

6.3 Comparing algorithmic and human collusion

While algorithms converge to non-competitive prices and learn punishment strategies, it is unclear whether algorithms are more collusive than humans. Therefore, in this section, I compare the market outcomes from the experiments with humans to the algorithmic markets. Figure 5 shows the average market prices by supergames (SG) for the treatments with only humans (2H0A and 3H0A) and outcomes for the selected algorithmic markets (0H2A and 0H3A).

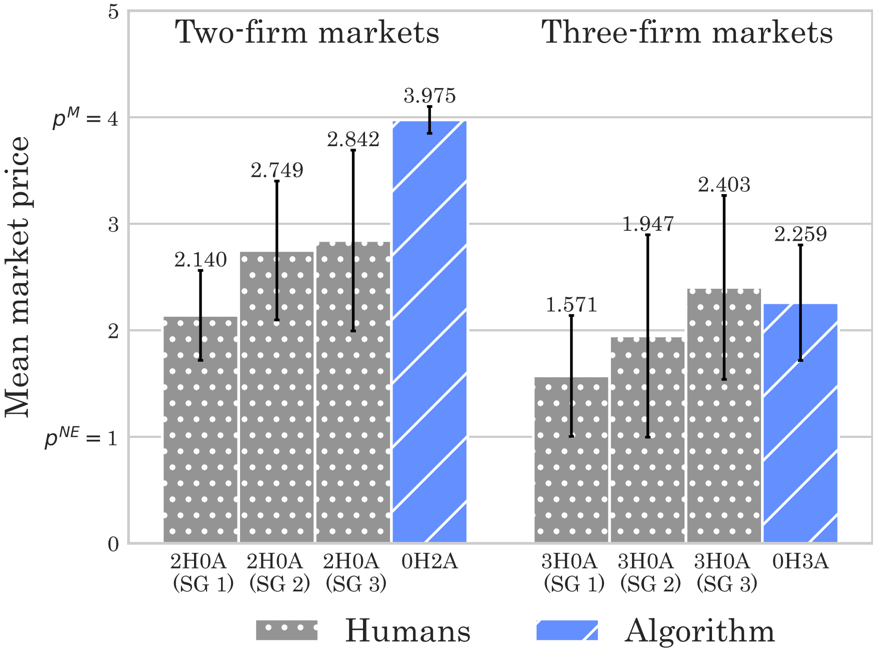


Figure 5: Market prices for the algorithmic and human markets for each supergame (SG). I derive the prices for the algorithmic markets upon convergence as an average over 1,000 subsequent periods for 1,000 independent simulations. The error bars represent the standard deviation.

Similar to previous findings in the literature (e.g., Huck et al. 2004), collusion becomes more difficult for humans as the market size increases. Average market prices are higher for each supergame in 2H0A compared to 3H0A. Those differences are (weakly) statistically significant for the first and second supergame but insignificant for the last supergame (SG1 $p = 0.045$; SG2 $p = 0.055$, SG3 $p = 0.283$; two-sided Mann–Whitney U tests). Thus, while the disparity in prices becomes smaller after learning, prices are always higher in two-firm markets compared to three-firm markets, which is in line with Hypothesis 2. While both algorithms and humans see a drop in price due to the expanded market size, the decline is greater for algorithmic markets. It suggests that the market size within the discussed environment might be more harmful to algorithmic than for human collusion.[40]

Result 4. *Similar to algorithmic markets, the level of tacit collusion declines for humans as the market size increases. Price drops due to the increase in market size are higher for algorithmic compared to human markets.*

In two-firm markets, algorithms outperform humans at colluding. Average market prices in 0H2A are statistically significantly higher than in 2H0A for each supergame when using the selected algorithm as a comparison unit ($p < 0.01$ for all supergames; two-sided Mann–Whitney U tests). Also, when considering the average market price of the entire parameter grid discussed in Section 6.1, prices in 2H0A are smaller ($p < 0.05$ for all supergames; one-sample two-sided t-tests against the average grid price of 3.51).

Furthermore, in three-firm markets, the selected algorithms are more collusive than humans in the first two supergames. However, this advantage entirely fades after the first two supergames as there are no differences between algorithms and humans in the third and last supergame (SG1 $p < 0.01$; SG2 $p < 0.01$, SG3 $p = 0.980$; two-sided Mann–Whitney U tests). Hence, after humans had the chance to learn about the game, they are as good as self-learned algorithms at colluding in three-firm markets. If I compare prices in human markets to the average algorithmic outcome in the parameter grid, there exist no statistically significant price differences for the first and second supergame. Moreover, in the last supergame, the average grid price within three-firm human markets even exceeds the average price of the parameter grid (SG1 $p = 0.991$; SG2 $p = 0.340$, SG3 $p = 0.044$; one-sample two-sided t-tests against the average grid price of 1.57). Hence, trained algorithms can outperform inexperienced humans at colluding in markets with three firms. Yet, humans are as good as algorithms at colluding after they gain experience. If a firm fails to pick an optimal algorithm, humans can even surpass algorithmic performance in this environment.

Result 5. *Algorithms are more collusive than humans in two-firm markets. In three-firm markets, algorithms outperform inexperienced humans at colluding but there are no price differences if humans are experienced.*

[40] For future research, it can be interesting to see the development of those number effects for even larger markets.

Result 5 is in line with Hypothesis 1 for two-firm markets. It is also the case for three-firm markets, if humans are inexperienced. There is no evidence that algorithms hurt competition in three-firm markets after humans adapt and learn themselves.

6.4 Collusion between humans and algorithm

In this section, I consider the outcomes for mixed markets in which humans compete against algorithms. The algorithms always play according to the limit strategy of the selected algorithm. Hence, similarly to Normann and Sternberg (2021) who consider a tit-for-tat algorithm, the humans compete against a fixed strategy in the experiments. In contrast to Normann and Sternberg (2021) the strategy is a result of the learning procedure of the algorithms instead of being chosen by a researcher. Furthermore, the WSLS algorithm maximizes the selection criterion Ψ_i. Thus, it would arguably be used by firms.

Figure 6 shows the average market price pooled across all supergames for each treatment with human involvement.

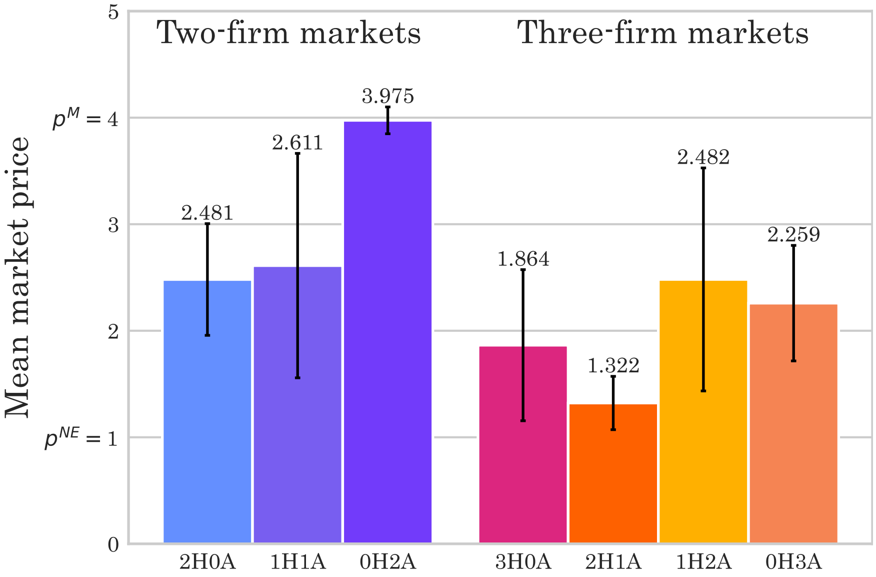


Figure 6: Average market prices for all treatments. For treatment with humans, I pool market prices across all super games. For algorithmic markets, I use the parameterization of the selected algorithm as a comparison unit. The error bars represent the standard deviation.

Within two-firm markets, there are no statistically significant differences in market prices between two humans (2H0A) and one human competing with one algorithm (1H1A) ($p = 0.84$, two-sided Mann–Whitney U test). Thus, contrary to Hypothesis 1 the pricing algorithm does not foster collusion. Nevertheless, on average, a single algorithm is as good at colluding with a human as another human player. Furthermore, prices in 1H1A are significantly lower than in the fully algorithmic market 0H2A ($p < 0.01$, two-sided Mann–Whitney U test). Hence, while algorithms never foster competition in a duopoly, they can make markets more collusive if all firms utilize them.

In three-firm mixed markets, I observe a non-linear relationship between the level of tacit collusion and the number of algorithms in the market. Market prices in 2H1A are *lower* than in 3H0A ($p = 0.07$, two-sided Mann–Whitney U test).[41] Adding another algorithm to the market (1H2A) increases prices again compared to 2H1A ($p < 0.01$, two-sided Mann–Whitney U test). There are no statically significant differences between 1H2A and 0H3A using algorithms with the parameterization of the selected algorithms as a comparison unit ($p = 0.76$, two-sided Mann–Whitney U test). However, average prices in 0H3A are higher than market prices in 3H0A if the outcomes are pooled across supergames ($p < 0.01$, two-sided Mann–Whitney U test).

Result 6. *Humans manage to cooperate with pricing algorithms. In duopolies, algorithms (weakly) foster tacit collusion. In triopolies, there exists a non-linear relationship between the level of tacit collusion and the number of algorithms in the market. If most firms use pricing algorithms, markets can become less competitive.*

Within my framework, firms have a clear incentive to use pricing algorithms in a duopoly. If only a single firm adopts, prices do not change. Yet, if both firms outsource their pricing decisions to an algorithm, markets become more collusive, which in turn increases firms' profits.[42] This effect resembles recent findings on algorithm pricing in the German gasoline market. Assad et al. (2020) show that market-level margins do not increase if only one gas station in a local market adopts a pricing algorithm. Yet, if both gas stations in the duopoly adopt the price algorithm margins increase by 28%. For triopolies, I find vastly different outcomes depending on the exact market composition in mixed markets, but also with three firms in the market, algorithms can hurt competition. It is especially the case if the majority

[41] Note that the results differs from Normann and Sternberg (2021) who find that a single tit-for-tat algorithm fosters collusion with three firms in a simpler market environment. The strategies of the human sellers drive the results in my setup and I analyze them in the subsequent paragraph.

[42] Köbis et al. (2021) argue that the decision to delegate the pricing to an algorithm can be particularly relevant as it allows the firms' manager to morally distance herself from the unethical behavior of collusion. In my experiment, firms cannot decide whether they want to adopt a pricing algorithm as it is determined exogenously. However, it can be a path for future research to examine the adoption decision of participants.

of firms decide to use pricing algorithms and humans lack experience. However, adoption incentives are less obvious compared to a duopoly, as firms' profits can decrease if only a single firm utilizes them.

Heterogeneous strategies in mixed market In Figure 7, I plot the average market price by round and supergame for each experimental treatment. While 1H2A and 1H1A have a similar trend as 2H0A in the first supergame, 3H0A and 2H1A have noticeable lower market prices. In fact, after some initial rounds, average prices in 2H1A are at the stage game Nash equilibrium. In the later supergames, after the participants learn about the game, some interesting patterns emerge. Prices in 2H1A are still close to the stage game Nash equilibrium. While average market prices in 1H1A and 1H2A are similar to 2H0A, there are sharp spikes in every other round.

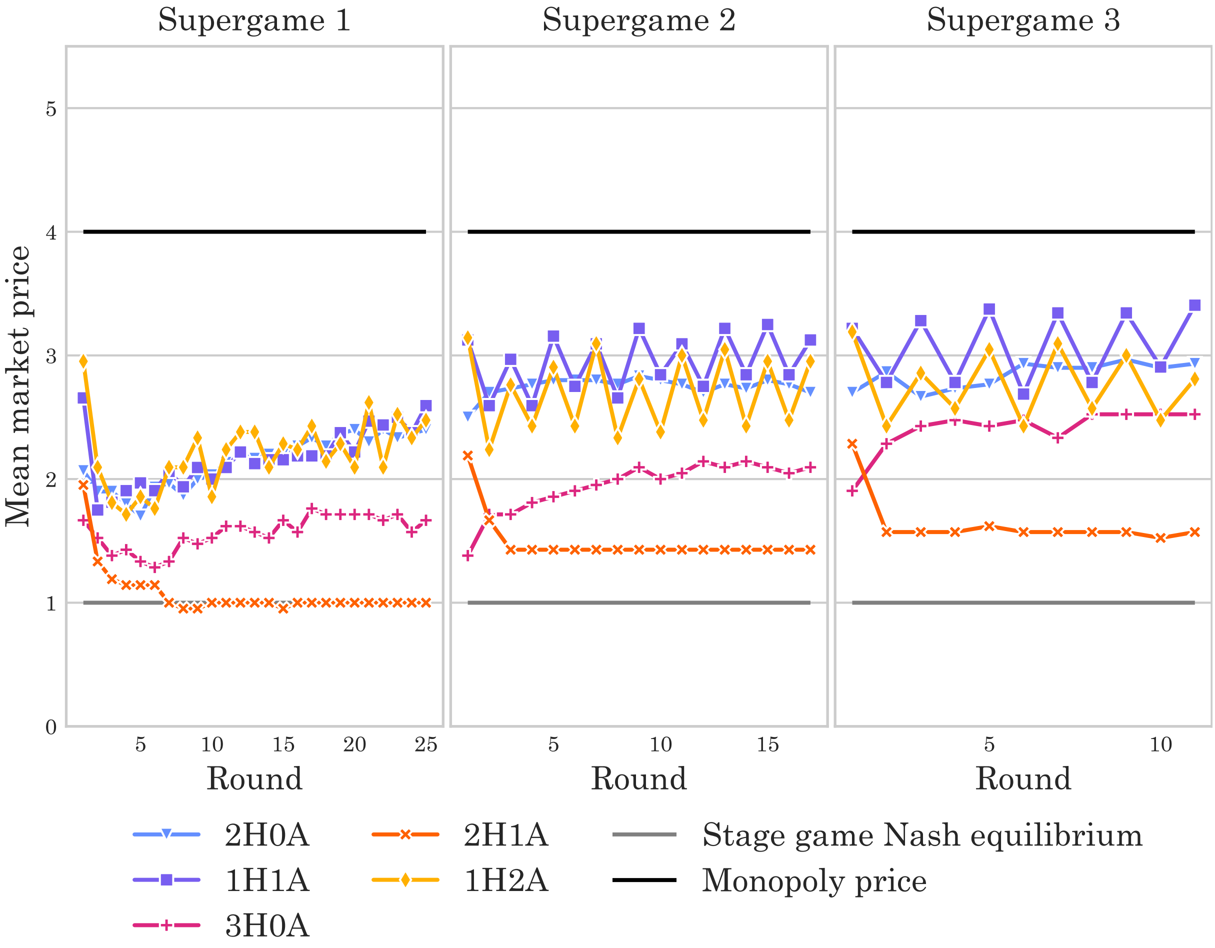


Figure 7: Average market prices by supergame and round for all experimental treatments.

To understand those price patterns, it is important to remember that participants during the experiment play against the limit strategy of the selected algorithm, which is described by Equation 6. In other words, participants play against a variation of a win-stay lose-shift (WSLS) strategy. Normann and Sternberg (2021) demonstrate that the algorithm's strategy is a significant determinant of outcomes in human-machine interactions. The expectations that participants have about the algorithm's behavior are mostly irrelevant. Hence, it is essential to understand how participants respond to the strategy of the algorithm in the presented setup.

While participants do not know the strategy of the algorithm initially, they have the opportunity to learn about it during the first supergame. Once a participant understands how the algorithm works, there are different ways to adapt her strategies as a response. First, she can ALWAYS COOPERATE with the algorithm at the monopoly price. Second, she can try to EXPLOIT the algorithm by playing the price cycle strategy described in Equation 7. This strategy dominates ALWAYS COOPERATE in 1H2A and is dominated by ALWAYS COOPERATE in 1H1A.[43] Other strategies are possible. Namely, participants can ALWAYS DEFECT at the stage game Nash equilibrium. Furthermore, they can play an imperfect exploitation strategy by playing a price of $p = 2$ in the cooperative phase of the algorithm and $p = 1$ elsewise. I denote this strategy by EXPLOIT2. The strategies ALWAYS DEFECT and EXPLOIT2 are dominated by ALWAYS COOPERATE and the EXPLOIT respectively.

To investigate which strategies are used by the participants in 1H1A and 1H2A against the algorithm, I estimate a mixture model using the Strategy Frequency Estimation Method (SFEM) proposed by Dal Bó and Fréchette (2011). The method is highly influential for estimating strategy choices in infinitely repeated games, especially the Prisoner's Dilemma (e.g., Fudenberg et al. 2012, Romero and Rosokha 2018, Dal Bó and Fréchette 2019). Starting from a predefined set of strategies, SFEM assumes that subject i, chooses strategy s^k with probability ϕ^k and follows this strategy for all rounds of the game. In each period, participant i selects her price according to strategy s^k with probability $\sigma \in (1/2, 1)$ but makes an error with probability $1-\sigma$. The individual likelihood that participant i plays according to strategy k is given by $P_i(s^k) = \prod_t \sigma^{I_{t,i}}(1-\sigma)^{1-I_{t,i}}$. The identifier variable $I_{t,i}$ is equal to 1 if the price of participant i in period t corresponds to the price she would have played if she followed strategy s^k. Otherwise, $I_{t,i}$ is equal to zero. The log-likelihood function is given by $\mathscr{L} =$

[43] For details see Appendix B.

$\sum_i ln(\sum_k \phi^k P_i(s^k))$. The estimate of ϕ^k represents the share of participants in the population that uses strategy k. The value of σ can be interpreted as a goodness of fit parameter. If σ is close to its lower bound of 0.5, the model is noisy. The model describes the data well for values of σ that are close to 1. For the estimation procedure, I focus on the strategies that are reasonable when competing against the algorithm (ALWAYS COOPERATE, ALWAYS DEFECT, EXPLOIT and EXPLOIT2). Moreover, I restrict the analysis to the last supergame. Table 3 shows the results of the estimation procedure.

Table 3: Estimated proportion for each strategy

Strategy	Treatment	
	1H1A	1H2A
ALWAYS COOPERATE	0.61	0.48
	(0.09)	*(0.11)*
ALWAYS DEFECT	0.10	0.22
	(0.05)	*(0.10)*
EXPLOIT	0.29	0.29
	(0.08)	*(0.10)*
EXPLOIT2	0.00	0.02
	(0.00)	*(0.05)*
σ	0.92	0.84

[*] The mixture model is estimated by maximum likelihood estimation. I restrict the data to the last supergame. The bootstrapped standard errors are in parentheses.

The most frequent strategy that participants play against the algorithm is ALWAYS COOPERATE in both treatments. The estimated proportion is, however, smaller in 1H2A compared to 1H1A. Also, EXPLOIT is prevalent in the population, but the estimates do not differ between 1H1A and 1H2A. Notably, the share of ALWAYS DEFECT is higher in 1H2A compared to 1H1A. The imperfect exploitative strategy EXPLOIT2 is never played in 1H1A, and it only accounts for a share of 0.02 of the data in 1H2A.

In line with the shift in incentives when increasing the market size, fewer participants play a cooperative strategy against the algorithm in 1H2A. Yet, participants often fail to learn the best response as EXPLOIT and ALWAYS COOPERATE dominate ALWAYS DEFECT in 1H2A. A possible reason is that learning about the environment is more difficult in 1H2A due to higher strategic complexity. While both algorithms in 1H2A use a WSLS strategy, participants still have to consider additional information compared to 1H1A. That can impede learning for a subset of participants. Individual prices reveal that some participants circle between prices of 1, 2, and 3 without a clear pattern. It appears that those participants did not learn to follow a fixed strategy (see Appendix B.2 for the price patterns on an individual level). This argument is also supported by the smaller value of σ and higher standard errors in 1H2A, as it indicates a more noisy behavior of the participants. While average market prices in 1H1A (1H2A) and 2H0A (3H0A) are similar, it is usually not the case for individual markets. Depending on the particular strategies that humans learn, mixed markets can be more or less collusive than their entirely human counterparts.[44]

In 2H1A, it is also crucial to consider the possible strategies humans can use against the algorithm. While ALWAYS COOPERATE and EXPLOIT are both still viable options to play against the algorithm, they now require joint coordination by two humans. Indeed, low prices in 2H1A can be explained by a frequent failure to coordinate simultaneously against the algorithm. While some markets manage to collude at the monopoly price with the algorithm, most participants fail to coordinate on any other strategy than ALWAYS DEFECT against the algorithm.[45]

Result 7. *Most humans always cooperate with the algorithm or try to exploit it in 1H1A and 1H2A. In 2H1A, most humans always defect as they fail to coordinate on a joint strategy against the algorithm. Market outcomes differ substantially conditionally on the exact strategies that humans learn.*

7 Concluding Remarks

In this paper, I study the collusive potential of self-learning pricing algorithms and show that pricing algorithms can weaken competition. Similar to previous results by Calvano et al. (2020a), Klein (2021), I observe that algorithms learn to set prices above the competitive benchmark and develop reward-punishment strategies in simulations. As the market environment is stylized and therefore highly tractable, I can analyze the strategies of the algorithms. I find that the most successful algorithms learn a win-stay lose-shift strategy. To derive a counterfactual for algorithmic collusion and observe the interaction of humans and pricing

[44] Also Wieting and Sapi (2021) find heterogeneous market outcomes depending on the exact number of algorithms in the market using data from the Dutch online retailer *bol.com*.

[45] Figure 12 in Appendix B.2 highlights those price patterns.

algorithms, I conduct laboratory experiments with the same market environment as in the simulations. Across different treatments, I vary the market size and the number of firms that use a self-learned pricing algorithm. This approach allows me to pin down the anti-competitive effects algorithms can have across a wide range of market compositions.

In duopolies, algorithmic markets are always more collusive than human markets. Markets with one human and one algorithm have similar average market prices compared to entirely human markets. In three-firm markets, market prices decrease if a single firm uses a pricing algorithm. It is driven by the specific strategy the algorithms learn and the failure of humans to coordinate with the algorithm. As more firms utilize pricing algorithms, prices increase again in three-firm markets. If all firms in the market use an algorithm, market prices can be higher than in human markets. However, the effect fades after humans have the chance to learn about the market environment. Most participants cooperate with the algorithm, but the strategies are heterogeneous, and some participants try to exploit the algorithm.

My results highlight the potential anti-competitive effects of self-learning algorithms. While market outcomes vary depending on the exact parameterization and market composition, algorithms rarely foster but often weaken competition if they populate the market. The considered pricing algorithms are simple, and the experimental environment is stylized. Yet, it appears probable that more complex algorithms can achieve similar results and scale to more complex real-world markets.[46] Within the presented framework, the fear from competition authorities that algorithms can harm the competitive landscape is justified.

Current research in computer science focuses on explainable artificial intelligence (see Barredo Arrieta et al. 2020). The development objective for those algorithms is that humans can understand their results and the decision process. Also, for pricing algorithms, explainable artificial intelligence is desirable. It is critical to understand why algorithms learn to be collusive and how algorithms have to be designed to prevent collusive market outcomes. Asker et al. (2021) underline the significance of the algorithmic design on its collusive behavior, but more research is needed to determine a suitable procedure to regulate pricing algorithms. Furthermore, recent work by Calvano et al. (2020b) proposes to audit algorithms before firms can use them as a pricing tool. Competition authorities could examine the algorithm in a simulated market environment to evaluate its potential for tacit collusion and ban specific al-

[46] Hettich (2021) shows that deep reinforcement learning algorithms can be collusive in a different market environment with up to ten firms.

gorithms if necessary. Auditing is not feasible for tacit collusion among humans. My findings indicate that it is not only possible for algorithms, but it is also necessary to prevent harm to competition.

References

Abada, Ibrahim and Xavier Lambin, "Artificial Intelligence: Can Seemingly Collusive Outcomes Be Avoided?," 2020.

Abreu, Dilip, "On the Theory of Infinitely Repeated Games with Discounting," *Econometrica*, 1988, *56* (2), 383–396.

Agrawal, Ajay, Joshua Gans, and Avi Goldfarb, *The Economics of Artificial Intelligence: An Agenda*, University of Chicago Press, 2019.

Arrieta, Alejandro Barredo, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera, "Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, 2020, *58* (October 2019), 82–115.

Arulkumaran, Kai, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, 2017, *34* (6), 26–38.

Asker, John, Chaim Fershtman, and Ariel Pakes, "Artificial Intelligence and Pricing: The Impact of Algorithm Design," 2021.

Assad, Stephanie, Emilio Calvano, Giacomo Calzolari, Robert Clark, Vincenzo Denicolò, Daniel Ershov, Justin Johnson, Sergio Pastorello, Andrew Rhodes, Lei Xu, and Matthijs Wildenbeest, "Autonomous algorithmic collusion: Economic research and policy implications," *Oxford Review of Economic Policy*, 2021, *37* (3), 459–478.

____ , Robert Clark, Daniel Ershov, and Lei Xu, "Algorithmic Pricing and Competition : Empirical Evidence from the German Retail Gasoline Market," 2020.

Bó, Pedro Dal, "Cooperation under the Shadow of the Future : Experimental Evidence from Infinitely Repeated Games," *American Economic Review*, 2005, *95* (5), 1591–1604.

____ and Guillaume R. Fréchette, "The Evolution of Cooperation in Infinitely Repeated Games: Experimental Evidence," *The American Economic Review*, 2011, *101* (1), 411–429.

____ and ____ , "Strategy Choice in the Infinitely Repeated Prisoner's Dilemma," *American Economic Review*, 2019, *109* (11), 3929–3952.

Borenstein, Severin and Andrea Shepard, "Dynamic Pricing in Retail Gasoline Markets," *The RAND Journal of Economics*, 1996, *27* (3), 429–451.

Brown, Zach and Alexander MacKay, "Competition in Pricing Algorithms," *American Economic Journal: Microeconomics (forthcoming)*, 2021.

Bundeskartellamt and Autorité de la concurrence, "Algorithms and Competition," 2019.

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

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello, "Algorithmic Pricing What Implications for Competition Policy?," *Review of Industrial Organization*, 2019, *55* (1), 155–171.

____ , ____ , ____ , and ____ , "Artificial intelligence, algorithmic pricing, and collusion," *American Economic Review*, 2020, *110* (10), 3267–3297.

____ , ____ , ____ , and ____ , "Algorithmic collusion with imperfect monitoring," *International Journal of Industrial Organization (forthcoming)*, 2021.

____ , ____ , ____ , Joseph E. Harrington, and Sergio Pastorello, "Protecting consumers from collusive prices due to AI," *Nature*, 2020, *370* (6520), 1040–1042.

Chen, Daniel L., Martin Schonger, and Chris Wickens, "oTree-An open-source platform for laboratory, online, and field experiments," *Journal of Behavioral and Experimental Finance*, 2016, *9*, 88–97.

Chen, Le, Alan Mislove, and Christo Wilson, "An empirical analysis of algorithmic pricing on amazon marketplace," *Proceedings of the 25th international conference on World Wide Web*, 2016, pp. 1339–1349.

Competition & Markets Authority, "Algorithms: How they can reduce competition and harm consumers," 2021.

Crandall, Jacob W., Mayada Oudah, Tennom, Fatimah Ishowo-Oloko, Sherief Abdallah, Jean François Bonnefon, Manuel Cebrian, Azim Shariff, Michael A. Goodrich, and Iyad Rahwan, "Cooperating with machines," *Nature Communications*, 2018, *9* (1), 1–12.

Danz, David, Neeraja Gupta, Marissa Lepper, Lise Vesterlund, and K Pun Winichakul, "Going virtual : A step-by-step guide to taking the in-person experimental lab online," 2021.

Davies, Stephen, Matthew Olczak, and Heather Coles, "Tacit collusion, firm asymmetries and numbers: Evidence from EC merger cases," *International Journal of Industrial Organization*, 2011, *29* (2), 221–231.

Engel, Christoph, "How much collusion? A meta-analysis of oligopoly experiments," *Journal of Competition Law and Economics*, 2007, *3* (4), 491–549.

——, "Tacit Collusion: The Neglected Experimental Evidence," *Journal of Empirical Legal Studies*, 2015, *12* (3), 537–577.

European Commission, "Final report on the E-commerce Sector Inquiry," 2017.

Ezrachi, Ariel and Maurice E. Stucke, "Virtual Competition," *Journal of European Competition Law & Practice*, 2016, *7* (9), 585–586.

—— **and** ——, "Artificial intelligence & collusion: When computers inhibit competition," *University of Illinois Law Review*, 2017, *2017* (5), 1775–1810.

Fonseca, Miguel A. and Hans-Theo Normann, "Explicit vs. tacit collusion-The impact of communication in oligopoly experiments," *European Economic Review*, 2012, *56* (8), 1759–1772.

Friedman, James W., "A non-cooperative equilibrium for supergames," *Review of Economic Studies*, 1971, *38* (1), 1–12.

Fudenberg, Drew, David G. Rand, and Anna Dreber, "Slow to anger and fast to forgive: Cooperation in an uncertain world," *American Economic Review*, 2012, *102* (2), 720–749.

Gale, Douglas and Hamid Sabourian, "Complexity and competition," *Econometrica*, 2005, *73* (3), 739–769.

Greiner, Ben, "Subject pool recruitment procedures: organizing experiments with ORSEE," *Journal of the Economic Science Association*, 2015, *1* (1), 114–125.

Hansen, Karsten, Kanishka Misra, and Mallesh Pai, "Algorithmic Collusion: Supra-competitive Prices via Independent Algorithms," *Marketing Science*, 2020, *40* (1), 1–12.

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

——, "The Effect of Outsourcing Pricing Algorithms on Market Competition," *Management Science (forthcoming)*, 2021.

——**, Roberto Hernan Gonzalez, and Praveen Kujal**, "The relative efficacy of price announcements and express communication for collusion: Experimental findings," *Journal of Economic Behavior and Organization*, 2016, *128* (051), 251–264.

Hettich, Matthias, "Algorithmic Collusion: Insights from Deep Learning," 2021.

Horstmann, Niklas, Jan Krämer, and Daniel Schnurr, "Number Effects and Tacit Collusion in Experimental Oligopolies," *Journal of Industrial Economics*, 2018, *66* (3), 650–700.

Huck, Steffen, Hans-Theo Normann, and Jörg Oechssler, "Two are few and four are many: Number effects in experimental oligopolies," *Journal of Economic Behavior and Organization*, 2004, *53* (4), 435–446.

Imhof, Lorens A., Drew Fudenberg, and Martin A. Nowak, "Tit-for-tat or Win-stay, Lose-shift?," *Journal of Theoretical Biology*, 2007, *247* (3), 574–580.

Jeschonneck, Malte, "Collusion among Autonomous Pricing Algorithms Utilizing Function Approximation Methods," 2021.

Johnson, Justin, Andrew Rhodes, and Matthijs Wildenbeest, "Platform Design When Sellers Use Pricing Algorithms," 2020.

Jones, Matthew T., "Strategic complexity and cooperation: An experimental study," *Journal of Economic Behavior and Organization*, 2014, *106*, 352–366.

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

Köbis, Nils, Jean François Bonnefon, and Iyad Rahwan, "Bad machines corrupt good morals," *Nature Human Behaviour*, 2021, *5* (6), 679–685.

Kühn, Kai-Uwe and Steve Tadelis, "Algorithmic Collusion," *Prepared for CRESSE 2017*, 2017.

Leisten, Matthew, "Algorithmic Competition , with Humans," 2021.

Lerer, Adam and Alexander Peysakhovich, "Maintaining cooperation in complex social dilemmas using deep reinforcement learning," 2017.

Li, Jiawei, Stephen Leider, Damian R. Beil, and Izak Duenyas, "Running Online Experiments Using Web-Conferencing Software," *Journal of the Economic Science Association*, 2021, *7* (2), 167–183.

March, Christoph, "Strategic interactions between humans and artificial intelligence: Lessons from experiments with computer players," *Journal of Economic Psychology*, 2021, *87.*

Mehra, Salil K., "Antitrust and the robo-seller: Competition in the time of algorithms," *Minnesota Law Review*, 2016, *100* (4), 1323–1375.

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

Miller, Nathan H. and Matthew C. Weinberg, "Understanding the Price Effects of the Miller-Coors Joint Venture," *Econometrica*, 2017, *85* (6), 1763–1791.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, "Human-level control through deep reinforcement learning," *Nature*, 2015, *518* (7540), 529–533.

Musolff, Leon, "Algorithmic Pricing Facilitates Tacit Collusion: Evidence from E-Commerce," 2021.

Normann, Hans-Theo and Martin Sternberg, "Hybrid Collusion: Algorithmic Pricing in Human-Computer Laboratory Markets," 2021.

Nowak, Martin and Karl Sigmund, "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game," *Nature*, 1993, *364* (6432), 56–58.

O'Connor, Jason and Nathan E. Wilson, "Reduced demand uncertainty and the sustainability of collusion: How AI could affect competition," *Information Economics and Policy*, 2021, *54* (100882).

Posch, Martin, "Win-stay, lose-shift strategies for repeated games - Memory length, aspiration levels and noise," *Journal of Theoretical Biology*, 1999, *198* (2), 183–195.

Romero, Julian and Yaroslav Rosokha, "Constructing strategies in the indefinitely repeated prisoner's dilemma game," *European Economic Review*, 2018, *104*, 185–219.

Roth, Alvin E. and J. Keith Murnighan, "Equilibrium behavior and repeated play of the prisoner's dilemma," *Journal of Mathematical Psychology*, 1978, *17* (2), 189–198.

Schwalbe, Ulrich, "Algorithms, Machine Learning, and Collusion," *Journal of Competition Law & Economics*, 2018, *14* (4), 568–607.

Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, and Koray Kavukcuoglu, "Mastering the game of Go with deep neural networks and tree search," *Nature*, 2016, *529* (7585), 484–489.

Sutton, Richard S. and Andrew G. Barto, *Reinforcement learning: An introduction*, 2 ed., MIT Press, 2018.

Waltman, Ludo and Uzay Kaymak, "Q-learning agents in a Cournot oligopoly model," *Journal of Economic Dynamics and Control*, 2008, *32* (10), 3275–3293.

Watkins, Christopher John Cornish Hellaby, "Learning from Delayed Rewards." PhD dissertation, King's College, Cambridge 1989.

____ **and Peter Dyan**, "Q-Learning," *Machine Learning*, 1992, *8*, 279–292.

Wieting, Marcel and Geza Sapi, "Algorithms in the Marketplace: An Empirical Analysis of Automated Pricing in E-Commerce," 2021.

Wright, Julian, "Punishment strategies in repeated games: Evidence from experimental markets," *Games and Economic Behavior*, 2013, *82*, 91–102.

Zhao, Shuchen, Kristian López Vargas, Daniel Friedman, and Marco Antonio Gutierrez Chavez, "UCSC LEEPS Lab Protocol for Online Economics Experiments," 2020.

A Implementation details

The instructions were translated from German. Section A.1 provides a translation for the 2H1A treatment.

A.1 Instructions

Particularly important: If you have any questions, contact the administrator using the chat function in the Webex conference.

Once you took a decision on the respective page and read all the information, **please click on the "Next" button so that the experiment can continue**. If you **do not make an input for an extended time** or temporarily leave this website, we will remove you from the experiment.

In this case, you will **not receive any payment and will be banned from future online experiments.**

Instructions

In this experiment, you will repeatedly make price decisions. These allow you to earn real money. How much you earn depends on your decisions and those of the other participants. **Regardless, you will receive 4.00 euros for participating**. In the experiment, we use a fictional monetary unit called ECU. After the experiment, the ECUs are converted into euros and you will be paid accordingly. **Here, 130 ECUs correspond to one euro**.

In this experiment, you represent a firm in a virtual product market. In each market, two other firms sell the same product as you do. These firms are represented by two other experiment participants. All firms offer 60 units of the same product. There occur no costs of production to the firms. The game has multiple rounds, with the exact number being decided by a random mechanism. You play the game for three sessions. In each round of a session, you meet the same firms (i.e., experiment participants). However, the firms in your market change after each session.

The market has 60 identical customers. Each customer in each round of a session intends to buy **one unit** of the product as cheaply as possible. Each customer is willing to spend a maximum of **4 ECUs** for this unit of the product. All firms decide in each round again **at the same time** for how many ECUs they want to sell their product. You can sell your product for 0 ECU, 1 ECU, ... or 5 ECUs (only whole ECUs). Your profit is the price multiplied by the number of units sold. Formally expressed:

Profit = Price x Units sold

The firm with **the lowest price in the given round** sells its products as long as the price is not greater than 4 ECUs. Firms with a higher price do not sell their product. **The lowest price is the market price in the given round**. Firms with a **price higher than the market price do not sell their products in that round**. If two or all three firms want to sell their products for the same market price, the demand is divided equally between the two or three firms.

Examples:

Example 1: Firm A sets a price of 3 ECUs, firm B sets a price of 3 ECUs, firm C sets a price of 4 ECUs. Thus, firms A and B together set the lowest price. Firm A and B both sell the same number of products. Both firms have 30 customers each and thus get the same profit of 90 ECUs. Firm C sells nothing and has a profit of 0 ECUs.

	Firm A	**Firm B**	**Firm C**
Price	3	3	4
Profit	90	90	0

Example 2: Firm A sets a price of 2 ECUs, firm B sets a price of 2 ECUs, firm C sets a price of 2 ECUs. Thus, firms A, B, and C together set the lowest price. They all sell the same number of products (20 each) and thus get the same profit of 40 ECUs.

	Firm A	**Firm B**	**Firm C**
Price	2	2	2
Profit	40	40	40

Example 3: Firm A sets a price of 1 ECUs, firm B sets a price of 2 ECUs, firm C sets a price of 3 ECUs. Thus, firm A set the lowest price. Firm A is the only one that sells the

product for 1 ECU to all 60 customers and thus gets a profit of 60 ECUs. Firms B and C both sell nothing and have a profit of 0 ECUs.

	Firm A	**Firm B**	**Firm C**
Price	1	2	3
Profit	60	0	0

Example 4: Firm A sets a price of 5 ECUs, firm B sets a price of 5 ECUs, firm C sets a price of 5 ECUs. Thus, firms A, B, and C together set the lowest price. However, customers are only willing to buy the product for 4 ECUs. Therefore, no firm sells its products and all firms get a profit of 0 ECUs.

	Firm A	**Firm B**	**Firm C**
Price	5	5	5
Profit	0	0	0

Market decisions by algorithms:

In your markets, two participants decide at which price they want to sell their firm's product and are paid the profit their firms earn at the end of the experiment.

Firm C will be equipped with an algorithm in all rounds, which will make the necessary price decisions for the participant. In this case, the participant does not take any decisions, but **still receives the profit** that his or her firm earns.

The procedure of the experiment:

After each round, all firms are informed about the prices chosen by each firm and about their profit. In the next round, each firm again has the chance to re-select its price. You interact with the same participants in each round within one session.

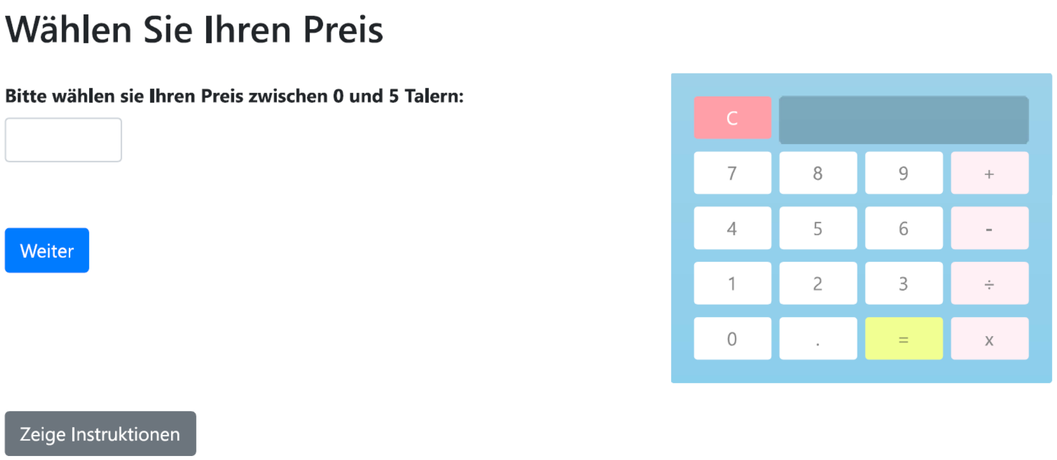
After each round, a random mechanism decides whether another round is played or the session ends. The probability that another round will be played is 95%. Thus, the session ends after each round with a probability of 5%. The session continues until the end is determined randomly.

Figuratively, the computer throws a virtual dice with 20 sides before each possible further round. The result decides whether another round is played or not. If the number is 20, the session is over; for all other numbers, another round is played.

Note:

You will play the game described for a total of three sessions. After each session, you will be paired with other participants to form a new market. This means that you interact with other participants in each of the three sessions. **After all, sessions are completed, it will be randomly decided which of the three sessions will be paid for**. You will receive this profit after the experiment. You will also receive additional 4.00 euros for participating in this experiment.

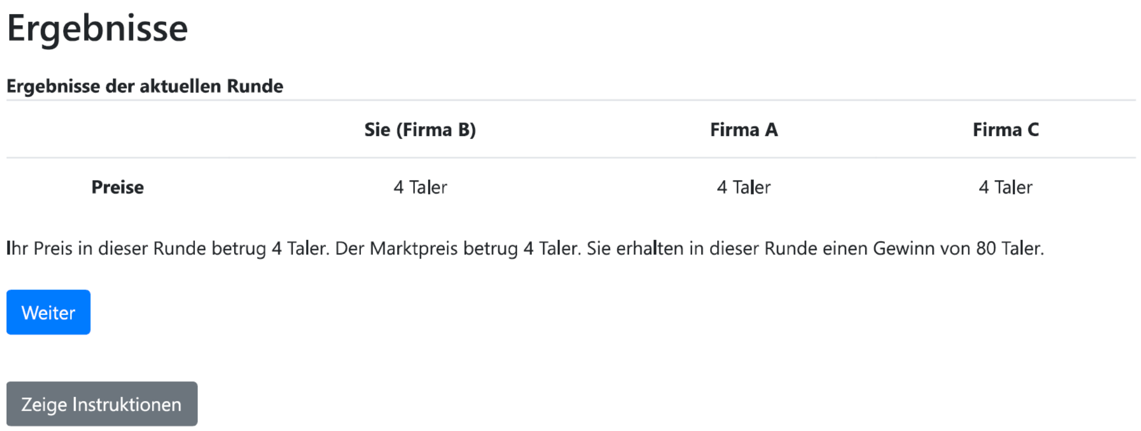
A.2 Screenshots of the Experiment



Wählen Sie Ihren Preis

Bitte wählen sie Ihren Preis zwischen 0 und 5 Talern:

Figure 8: Decision screen



Ergebnisse

Ergebnisse der aktuellen Runde

	Sie (Firma B)	Firma A	Firma C
Preise	4 Taler	4 Taler	4 Taler

Ihr Preis in dieser Runde betrug 4 Taler. Der Marktpreis betrug 4 Taler. Sie erhalten in dieser Runde einen Gewinn von 80 Taler.

Figure 9: Profit information screen

A.3 Survey & Control Questions

A.3.1 Control Questions

Question 1: How many consumers are in the market who want to buy the product?

- 35
- 30
- 40
- 60

Question 2: What is the probability of playing another period after completing one?

- 95%
- 20%
- 50%

Question 3: What is the maximum price consumers are willing to pay for the product?

Question 4: You are firm A and choose a price of 2, firm B chooses a price of 3, firm C chooses a price of 5. What is your profit in ECU in this round?

Question 5: You are firm A and choose a price of 3, firm B chooses a price of 3, firm C chooses a price of 3. What is your profit in ECU in this round?

B Strategy analysis

B.1 Incentive compatibility of the algorithm's strategy

I want to show that the algorithms' strategy can be exploited by the strategy described by Equation 7. I focus on the case in which a player k uses this strategy, and all other players use the limit strategy of the algorithm (Equation 6). When using the Exploit strategy, the firms enter a price cycle. After a deviation to $p_k = 3$, all firms play the stage game Nash equilibrium. In the following period, firms $-k$ play the monopoly price while firm k plays again $p_k = 3$ to restart the cycle. Hence, in every other period firm k receives the deviation profit of $\pi_D = 3m = 180$. In the other periods, all firms share the market and firm k receives $\pi_{NE}/N = m/N = 60/N$. Cooperation at the monopoly price yields $\pi_M/N = 4m/N = 240/N$. Thus, the exploitative strategy dominates cooperation in an infinitely repeated game with discount factor δ if

$$\begin{aligned} & V^{Cooperate}(N) \geq V^{Exploit}(N) \\ \Leftrightarrow & \sum_{t=0}^{\infty} \delta^t \frac{\pi_M}{N} \geq \pi_D + \delta \frac{\pi_{NE}}{N} + \delta^2 \pi_D + \delta^3 \frac{\pi_{NE}}{N} + ... \\ \Leftrightarrow & \frac{1}{1-\delta} \frac{\pi_M}{N} \geq \frac{1}{1-\delta^2} \pi_D + \frac{\delta}{1-\delta^2} \frac{\pi_{NE}}{N} \\ \Leftrightarrow & \frac{1}{1-\delta} \frac{240}{N} \geq \frac{1}{1-\delta^2} 180 + \frac{\delta}{1-\delta^2} \frac{60}{N} \end{aligned} \tag{8}$$

Within the experiment and the simulation, the discount is $\delta = 0.95$. Hence, for a two-player game we have $V^{Cooperate}(N = 2) = 2400$ and $V^{Exploit}(N = 2) \approx 2138.46$. It implies that cooperation with the algorithm at the monopoly price dominates the exploitative strategy. For a three-player game it is $V^{Cooperate}(N = 3) = 1600$ and $V^{Exploit}(N = 3) \approx 2041.03$. Thus, exploiting the algorithm dominates cooperation.

B.2 Individual prices in the last supergame

Figure 10 and 11 show the individual prices for the treatments 1H1A and 1H2A respectively. Furthermore, I plot the price of the algorithm for each round. Note that this prices is the same for both algorithms in 1H2A.

Both figures reveal the price patterns associated with the strategies described in Section 6.4. In very few rounds in 1H2A, the strategy of the algorithm diverges from the win-stay lose-shift strategy described by Equation 6. While this may delay efficient learning in the game, participants never play a strategy that exploits those negligible deviations.

Figure 12 shows all prices for each market in the last supergame in 2H1A. In few markets, both humans manage to collude with the algorithm. The other markets usually have a market price that is equal to the stage game Nash equilibrium.

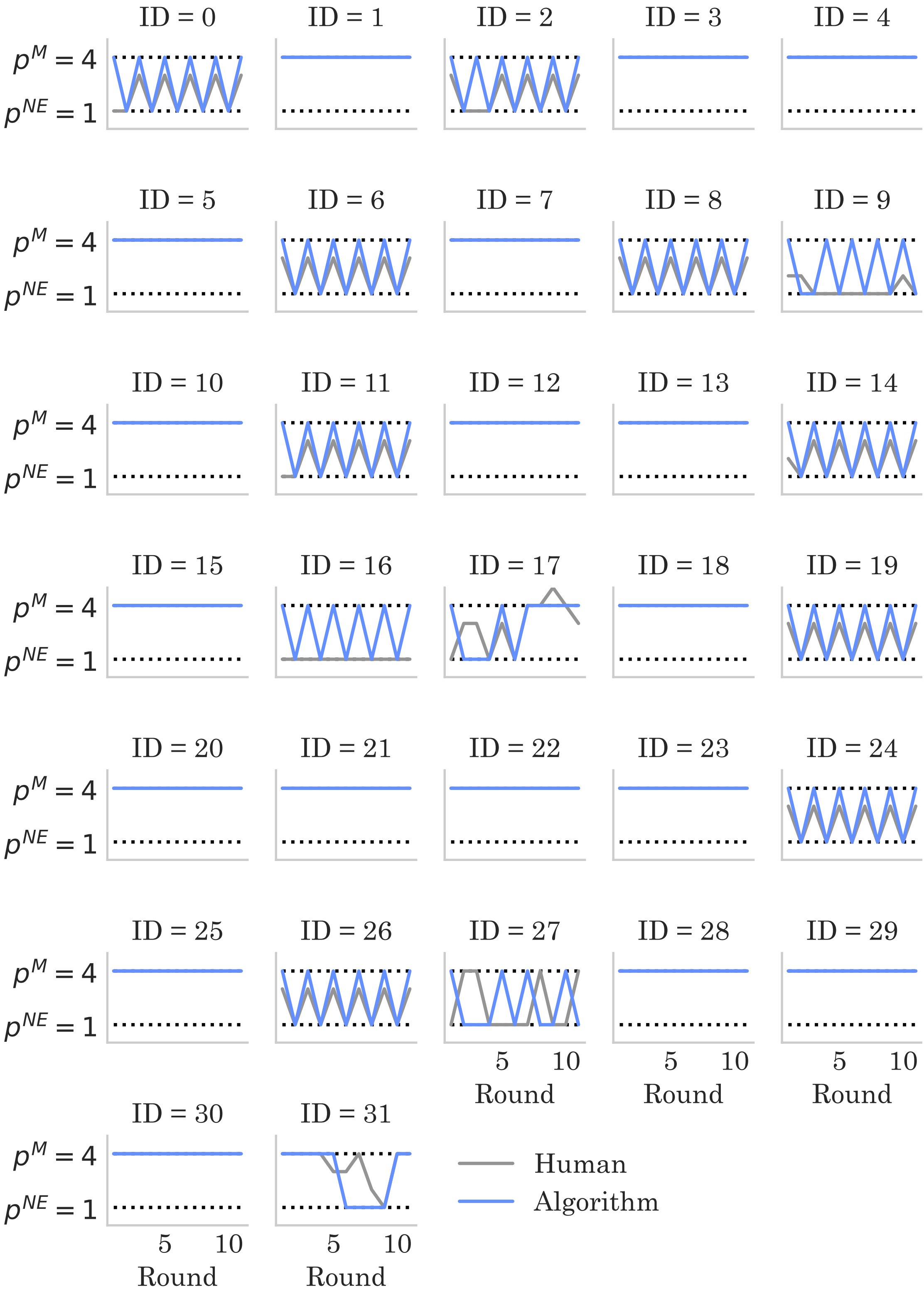


Figure 10: Prices for each human participant in the last supergame in 1H1A.

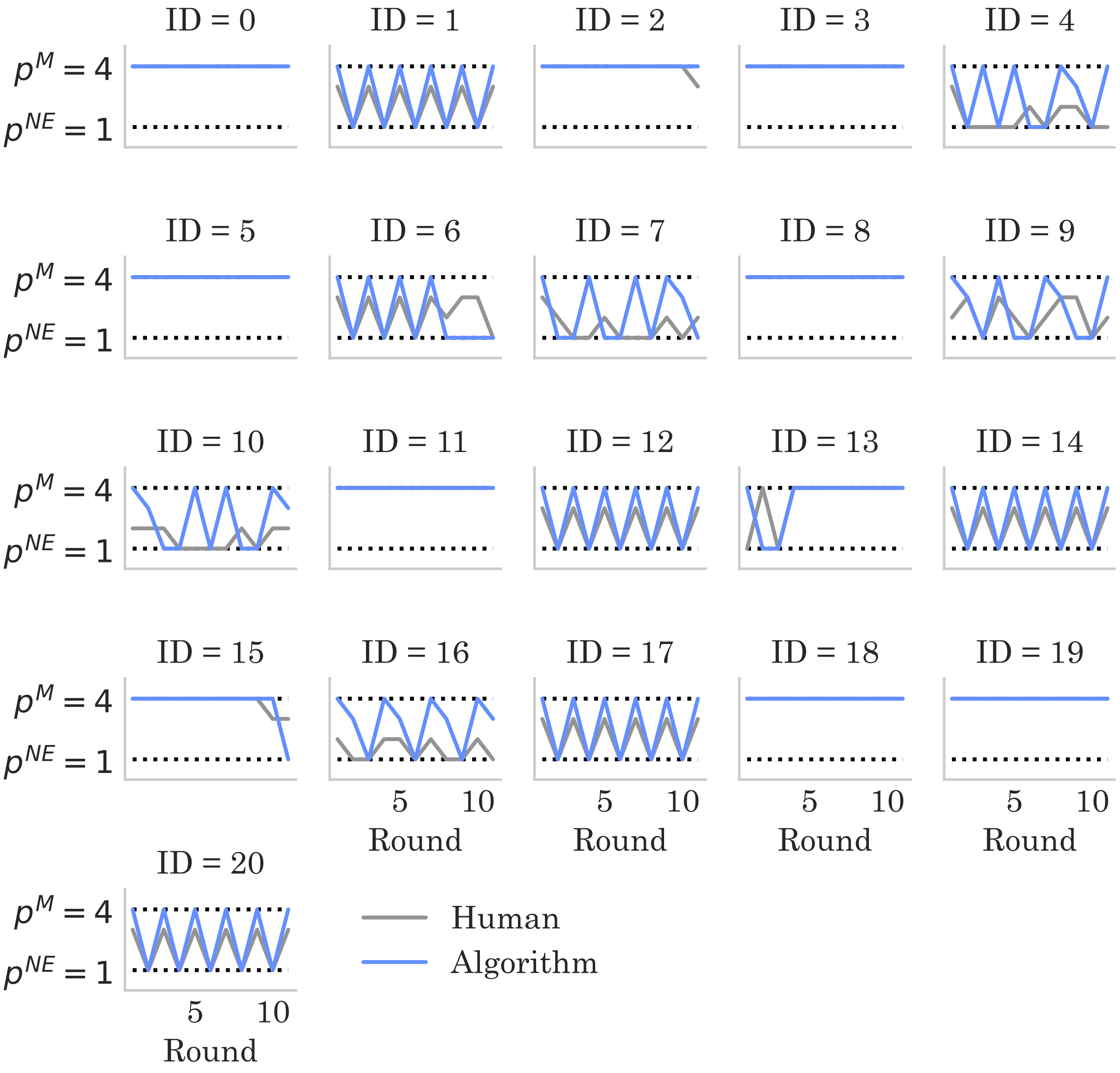


Figure 11: Prices for each human participant in the last supergame in 1H2A. Note that both algorithm always play the same price.

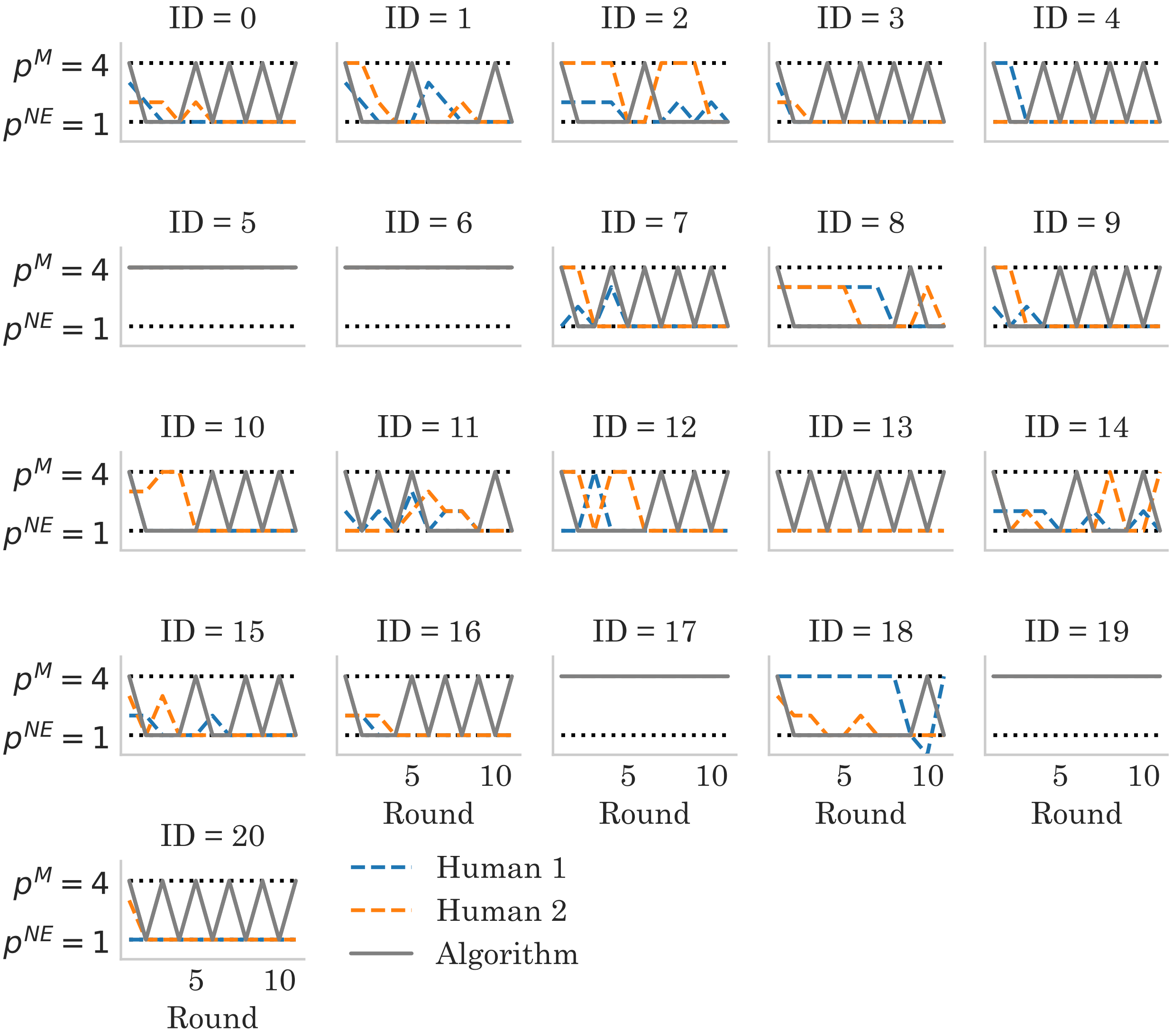


Figure 12: Prices for each market in the last supergame in 2H1A.