Algorithmic collusion: Genuine or spurious?

ALGORITHMIC COLLUSION: GENUINE OR SPURIOUS?[†]

EMILIO CALVANO[**‡§], GIACOMO CALZOLARI[&§],
VINCENZO DENICOLÒ[*§] AND SERGIO PASTORELLO[*]

APRIL 2023

Reinforcement-learning pricing algorithms sometimes converge to supra-competitive prices even in markets where collusion is impossible by design, or cannot be an equilibrium outcome. We analyze when such spurious collusion may arise, and when instead the algorithms learn genuinely collusive strategies, focusing on the role of the rate and mode of exploration.

Keywords: Artificial Intelligence, Reinforcement Learning, Collusion, Exploration.

J.E.L. codes: L41, L13, D43, D83.

1. INTRODUCTION

Economists generally agree that collusion is not synonymous with high prices but exists only when the high prices are sustained by a suitable reward-punishment scheme (Harrington, 2018). Since Calvano et al. (2020) and Klein (2021), a number of studies have demonstrated that reinforcement-learning algorithms can learn such collusive schemes in a completely autonomous way, without external supervision. At the same time, other studies (e.g., Asker et al., 2022a, 2022b; Banchio and Mantegazza, 2022) have shown that reinforcement-learning algorithms sometimes converge to high prices without colluding properly. In other words, these algorithms do not cut their prices not because they anticipate retaliatory responses by rivals but simply because they are missing opportunities to earn more. We refer to the former case as *genuine* collusion, and to the latter as *spurious*.

This note contributes to understanding when algorithmic collusion is genuine or spurious.[1] We focus specifically on the role of exploration. It is well known that in order to acquire

Corresponding author: Vincenzo Denicolò, vincenzo.denicolo@unibo.it. We thank without implicating Joe Harrington, Timo Klein and Piercarlo Zanchettin for useful comments and discussions. Emilio Calvano gratefully acknowledges research funding from the Italian Ministry of Education, University and Research, (PRIN project 2017TMFPSH).

[*]Bologna University; [**] University of Rome, Tor Vergata; [‡]Toulouse School of Economics; [&]European University Institute; [§]CEPR

[1]Empirical analyses that have found evidence that pricing algorithms may increase prices (e.g., Assad

more information, the algorithms must select, with a certain frequency, actions that may appear sub-optimal in the light of the knowledge they already possess. We show that the mode and intensity of this experimentation activity may affect what the algorithms eventually learn.

2. EXPLORATION AND LEARNING IN PRICING GAMES

As most of the previous literature, we focus on Q-learning algorithms. These algorithms estimate the value of making a certain action a in a certain state s, denoted by $Q(a, s)$, and base their behavior on such estimation. For a brief introduction to Q-learning, see Calvano, Calzolari, Denicolò and Pastorello (2020) (henceforth CCDP); for a more comprehensive treatment, Sutton and Barto (2018).

Using these algorithms, Asker, Fershtman and Pakes (2022a, 2022b) (henceforth AFP) document the possibility of spurious collusion in two settings: when the competing algorithms play a sequence of independent pricing games and cannot condition their current price on past prices, and when in each period the algorithms care only about their current profits. In both cases, the only equilibrium is to charge, period after period, the static Bertrand price (i.e., with homogeneous products, the unit cost). AFP find, instead, that even after settling to stable behavior, the algorithms charge positive and large price-cost margins.[2]

Clearly, AFP's algorithms do not come to realize that they could gain by undercutting the rivals. In what follows, we inquire into the possible sources of such failure.

2.1. *The mode of exploration*

Specifically, we contrast two modes of exploration. In the first one, which is used by AFP, in each period t the algorithm adopts the action that it currently perceives as optimal; that is, in each state s, the action that maximizes $Q_t(a, s)$, where $Q_t(a, s)$ is the algorithm's assessment of $Q(a, s)$ in period t. However, the initial values $Q_0(a, s)$ are all set at very high levels. This produces a pattern of non-random exploration, as follows. When the algorithm tries a certain action a in a certain state s, it learns that the associated payoff is lower than it thought. This brings its estimate of $Q(a, s)$ down, whereas the estimates

et al., 2022, for the German retail gasoline market, and Mussolf (2022) for Amazon.com marketplace) are typically unable to identify the mechanism underlying the high prices.

[2] AFP demonstrate this pattern for the case of asynchronous algorithms. Their synchronous algorithms are different, as discussed below.

of the Q-values of different actions a' do not change. This automatically induces the algorithm to try different actions in the future. The process continues until the estimated Q-values get in line with the actual payoffs. This mode of exploration is often referred to as *optimistic initialization*.[3]

The second mode of exploration, which is used by CCDP, is the so-called ε-greedy model. This type of exploration is purely random. The algorithm explores with probability $1-\varepsilon$, and when it does, it tries all actions currently perceived as sub-optimal with the same probability. Initially, ε is nil, so the algorithm explores with probability one, but as learning proceeds, ε increases and eventually converges to one. The Q-values are initialized in a neutral way, which does not entail any further systematic incentive or disincentive to explore.[4]

The two modes of exploration can be combined, but to better highlight the differences we focus on pure cases. We briefly consider mixed cases later.[5]

2.2. *Economic framework*

As for the economic framework, we use the same as AFP: a homogenous-good duopoly with rectangular demand and price competition. To take advantage of the codes for numerical simulation developed by CCDP, we generate this case by taking the limit of CCDP's logit model as the degree of production differentiation goes to zero and the outside option has zero value. In the resulting set-up, the unit cost is 1 and the reser-

[3]Optimistic initialization is popular in computer science, but Sutton and Barto warn against its use in non stationary environments (Sutton and Barto, 2018, § 2.6, p. 34). Repeated games such as those studied here are inevitably non stationary, because even if the stage game does not change, rivals may change their behavior from one period to the next owing to learning, experimentation, or both.

[4]This is achieved by setting $Q_0(a,s)$ equal to the average payoff that a player would obtain by choosing action a in state s if the rival responded in a purely random way, which is precisely what the algorithms do initially.

[5]Another popular form of exploration is the so-called Boltzman model. In this case, actions are chosen with probabilities

$$\Pr(a_t = a|\, s) = \frac{e^{Q_t(a,s)/T}}{\sum_{a'} e^{Q_t(a',s)/T}}$$

where the parameter T is often called the system's "temperature." This pattern of exploration is adopted for instance by Waltman and Kaymak (2008) in their pioneering study of algorithmic collusion. They specifically assume that the temperature T initially is 1 (so choices are purely random) but then declines at a certain rate κ, which is a hyper-parameter of the algorithm, and eventually vanishes (so the algorithm chooses the action with the highest Q-value with probability 1). Waltman and Kaymak (2008) find what we call spurious collusion. The problem, in this case, seems due to their choice of a value of κ so high as to undermine learning.

vation price is 2; this is also the monopoly price. (The codes are publicly available at `https://www.aeaweb.org/articles?id=10.1257/aer.20190623`.)

Since Q-learning posits a finite number of actions and states, we discretize the price space using a grid of 15 equally spaced values. An action a consists in the choice of one of these prices. The grid is set so that the two lowest prices are lower than 1 (the unit cost), and the highest price is greater than 2 (the monopoly price).

The algorithms maximize the discounted sum of the profit they earn in each period, with a discount factor $\delta \in [0, 1)$. They may have either no memory, in which case the Q-values are simply $Q(a)$, or else a one-period memory. In this latter case, the state s in period t is the pair of prices charged by the algorithms in the previous period, $t - 1$.

For the case of optimistic initialization, the initial values $Q_0(a, s)$ are iid draws from a uniform distribution whose support lies entirely above the true values $Q(a, s)$. For the case of random experimentation, on the other hand, we set the learning and exploration parameters α and β of CCDP at $\alpha = 0.3$ and $\beta = 4 \times 10^{-4}$, respectively. For each set of parameter values, we run 1,000 numerical simulations and average the results.

2.3. *Results*

To begin with, consider the case of zero memory. In this case, the algorithms cannot condition their current choices on the past history of the game. With continuous prices, the only equilibrium would be to price at cost, period after period. With discretized prices, the equilibrium involves pricing one step above the unit cost (this weakly dominates pricing at or below cost).

Figure **??** contrasts the evolution of prices under random exploration (left-hand side panel) and optimistic initialization (right-hand side panel). Under optimistic initialization, the algorithms converge to prices substantially higher than the cost, as in AFP. Under random exploration, on the other hand, the algorithms learn to play competitively. Prices almost always converge to the lowest feasible price that is higher than the unit cost (≈ 1.07) and never exceed it by more than one step.[6]

As noted, one can also mix optimistic initialization and random exploration. In this case, the greater the weight of random exploration, the closer the prices to the unit cost.[7]

[6]Obviously, the finer the price grid, the better the unit cost can be approximated. We have also run simulations with 100 price levels, in which case the algorithms converge to prices almost indistinguishable from the unit cost.

[7]This confirms the findings of AFP: when they add random exploration to optimistic initialization, their

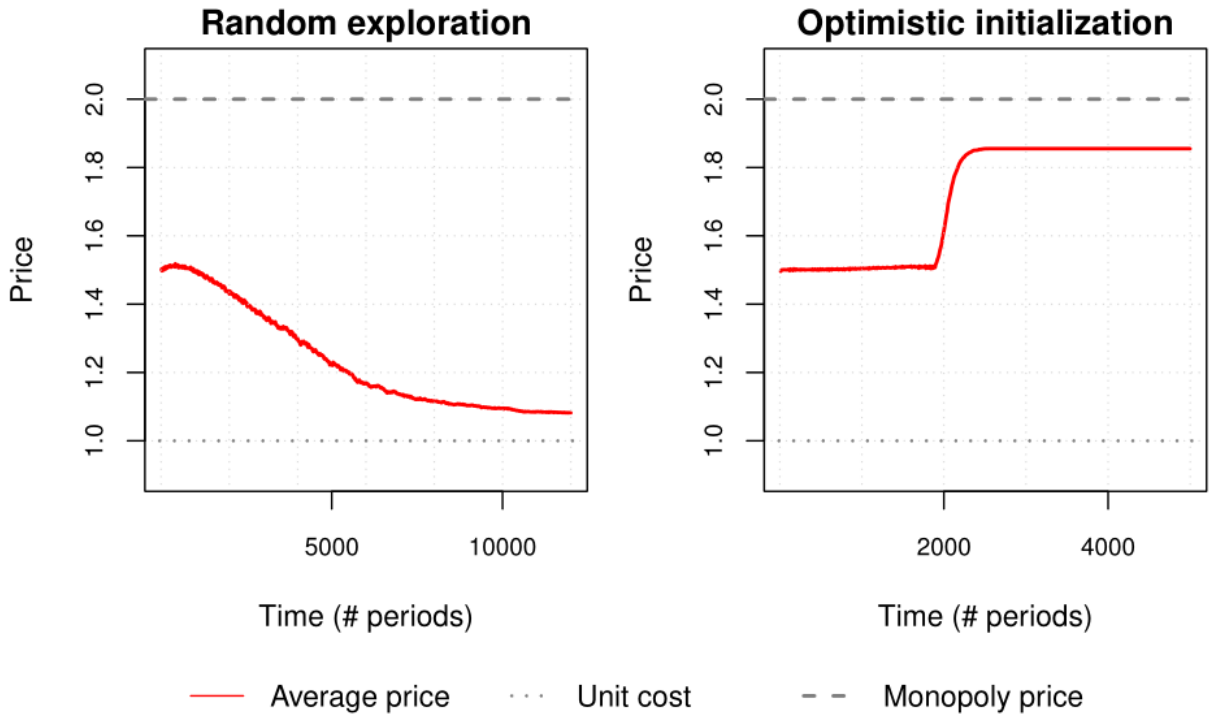

Figure 1: The price path with different exploration mechanisms. Memoryless algorithms repeatedly play a static Bertrand game. Market demand is 1 if price is less than 2, zero otherwise. Marginal cost is 1. There are 15 equally spaced feasible prices. The right panel represents the case of mechanical exploration, as in AFP (2022a,b); the left panel that of random exploration, as in CCDP (2020).

Next, consider the case in which the algorithms have a one-period memory and thus can condition their current price on the previous period's prices. This in principle allows for collusive strategies. However, such strategies can represent an equilibrium of the repeated game only if the discount factor δ is sufficiently high. For δ close to 0, the only equilibrium is again to price at cost period after period, as when the algorithms are memoryless.[8]

This notwithstanding, under optimistic initialization the algorithms converge to high prices even if δ is close to zero (Figure 2). In this case, the high prices cannot be sustained by a dynamic reward-punishment scheme, as the algorithms are so impatient that dynamic considerations are irrelevant. The algorithms' failure to optimize is therefore apparent, just as in the memoryless case.

algorithms indeed converge to lower prices. A similar pattern emerges, for a constant rate of random exploration, by lowering the initial values $Q_0(a)$.

[8]These results are consistent with AFP's finding that if one combines optimistic initialization with random exploration, the price-cost margin decreases.

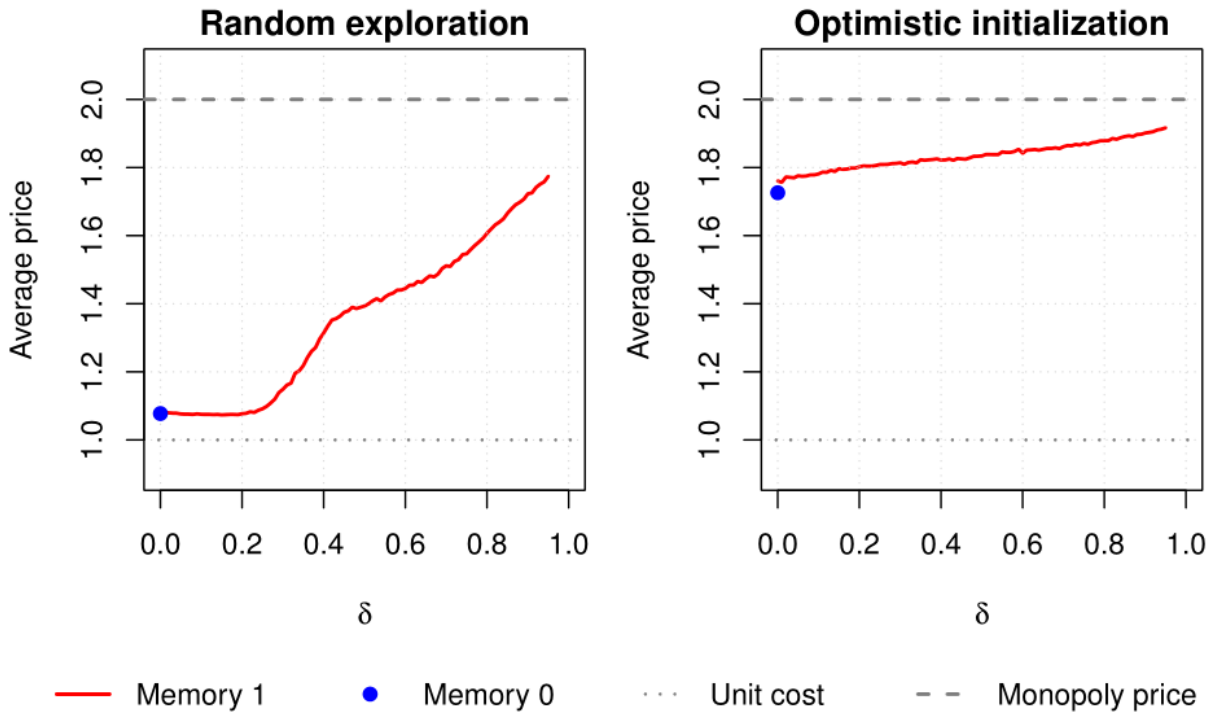

Figure 2: The long-run prices which the algorithms converge to under the two exploration mechanisms, for different values of the discount factor δ. The economic environment is as in Figure 1, but the algorithms now have a one-period memory and thus can condition their current price on the previous period prices.

Under random exploration, on the other hand, the algorithms price competitively when δ is low.[9] Prices increase only as δ gets larger, and when δ is large enough they are so high as to deliver more than 80% of the monopoly profits. Under random exploration, therefore, the high prices that the algorithms converge to, when they are patient enough can be genuinely collusive.

These observations are confirmed by the study of algorithms' responses to exogenous defections from the high prices. The "impulse-response" analysis developed in CCDP (2020) indeed shows that the high prices obtained under random exploration are sustained by punishments of defections, while those obtained under optimistic initialization, either with no memory or else when δ is close to 0, are not.

[9]Naturally, the algorithms may make sub-optimal choices even under random exploration. However, CCDP (2020) show that in this case the deviations from optimal behavior are small.

2.4. *Mechanism*

The foregoing results raise the problem of why Q-learning algorithms can sometimes get stuck in evidently sub-optimal choices.

Banchio and Mantegazza (2022) address this problem by focusing on a simpler stage game, i.e., a prisoner's dilemma. They allow for a mix of optimistic initialization and random exploration and consider memoryless algorithms, which as such should learn to defect. Using a combination of numerical simulations and analytical techniques, they show that the algorithms may instead alternate between cooperation and defection.

Specifically, this outcome occurs when the dynamics of the Q-values brings them in a region they call *sliding region*,[10] where the Q-values evolve along the boundary between cooperation and defection. For this to be possible, however, random exploration must be limited. When the algorithms explore more, or are initialized in a neutral way, they learn to defect.

Banchio and Mantegazza (2022) further show that the possibility of sliding along the boundary arises because the algorithms update only one Q-value at a time, and the Q-values associated with defection gets updated more quickly than those associated with cooperation.

2.5. *Smarter algorithms*

This last observation suggests that besides relying on sufficiently intense random experimentation, spurious collusion could be avoided by considering algorithms that update the estimates of all Q-values simultaneously, rather than one at a time. AFP refer to these algorithms as *synchronous*. They are indeed less prone to generating spuriously collusive outcomes, as shown by AFP numerically and by Banchio and Mantegazza (2022) analytically.

However, synchronous Q-learning algorithms need to be provided with information about the underlying economic environment, which allows them to compute counterfactuals. These algorithms, therefore, are not completely unsupervised, and their behavior depends on what structural information they are fed with. AFP, for instance, instruct the algo-

[10]These regions are defined in the space of the estimated Q-values of the two actions that players can choose in a prisoner's dilemma, cooperate and defect. Banchio and Mantegazza focus on the case where the algorithms are symmetric and thus always share the same estimates of the Q-values in the deterministic approximation of the stochastic dynamical system.

rithms to anticipate competitive behavior by rivals. This makes it more difficult for the algorithms to learn to collude, so it is even more noteworthy that genuine collusion still emerges (when it is feasible).

Another route to avoiding spurious collusion may be the use of reinforcement-learning algorithms that do not estimate the Q-values – a precursor of the value function – but the value function itself, the associated policy function, or both.[11] For example, Frick (2023) uses actor-critic algorithms, which estimate simultaneously the value function and the policy function, in order to address the issue of the time scale of algorithmic collusion. He shows that these algorithms can learn to genuinely collude much faster than Q-learning, and still in a completely unsupervised fashion.

3. CONCLUSION

This paper confirms that with random experimentation, even asynchronous Q-learning algorithms learn genuinely collusive strategies when they are forward looking and can condition their current prices on past prices. When the algorithms are myopic or memoryless, on the other hand, they learn to price competitively.

The possibility that memoryless or myopic algorithms may converge to supra-competitive prices seems to be due to a specific mode of exploration, optimistic initialization. But even in this case, smarter algorithms may avoid spurious collusion by updating many Q-values at a time, or by estimating directly the value and policy functions rather than the Q-values.

The problems discussed in this paper ultimately revolve around the way AI-based algorithms learn. We believe that understanding the mechanisms of algorithmic learning is a fascinating and important topic for research. Such research may benefit more from the study of learning "anomalies," such as spurious collusion, than of cases where the algorithms learn well.

In this spirit, it may be instructive to consider other anomalies. For example, Epivent and Lambin (2022) point out that the algorithms of CCDP learn to punish not only deviations to prices lower than the collusive prices, but also higher. Naturally, this is still genuine collusion, not spurious: even grim trigger strategies, for instance, entail punishments of higher prices. However, the algorithms do not have an analytical comprehension of the strategic environment but learn purely by trial and error. That by doing so they may learn

[11] These algorithms allow for continuous rather than discrete actions, and with continuous actions there are no boundaries to slide along.

to punish also higher prices seems therefore puzzling. This is another topic that deserves further research.

REFERENCES

Asker, J., Fershtman, C. and Pakes, A. (2022a). Artificial Intelligence, Algorithm Design and Pricing. *AEA Papers and Proceedings*.

—, — and — (2022b). Artificial Intelligence and Pricing: The Impact of Algorithm Design.

Assad, S., Clark, R., Ershov, D. and Xu, L. (2020). Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market.

Banchio, M. and Mantegazza, G. (2022). Adaptive algorithms and collusion via coupling.

Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. *The American economic review*, **110** (10), 3267–3297.

Decarolis, F., Rovigatti, G., Rovigatti, M. and Shakhgildyan, K. (2022). Artificial Intelligence, Algorithmic Bidding and Collusion in Online Advertising. *mimeo*.

Epivent, A. and Lambin, X. (2022). On Algorithmic Collusion and Reward-Punishment Schemes.

Frick, K. (2022). Autonomous Pricing using Policy-Gradient Reinforcement Learning.

Hansen, K. T., Misra, K. and Pai, M. M. (2021). Frontiers: Algorithmic Collusion: Supra-competitive Prices via Independent Algorithms. *Marketing Science*, **40** (1), 1–12.

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

Johnson, J., Rhodes, A. and Wildenbeest, M. R. (2023). Platform Design When Sellers Use Pricing Algorithms. *Econometrica (forthcoming)*.

Klein, T. (2021). Autonomous algorithmic collusion: Qlearning under sequential pricing. *The Rand journal of economics*, **52** (3), 538–558.

Musolff, L. (2022). Algorithmic Pricing Facilitates Tacit Collusion: Evidence from E-Commerce. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, New York, NY, USA: Association for Computing Machinery, pp. 32–33.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Waltman, L. and Kaymak, U. (2008). Q-learning agents in a Cournot oligopoly model. *Journal of economic dynamics & control*, **32** (10), 3275–3293.