DISCUSSION PAPER SERIES

DP16393

Algorithmic collusion, genuine and spurious

Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò and Sergio Pastorello

INDUSTRIAL ORGANIZATION

ISSN 0265-8003

Algorithmic collusion, genuine and spurious

Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò and Sergio Pastorello

Discussion Paper DP16393
Published 25 July 2021
Submitted 20 July 2021

Centre for Economic Policy Research
33 Great Sutton Street, London EC1V 0DX, UK
Tel: +44 (0)20 7183 8801
www.cepr.org

Algorithmic collusion, genuine and spurious

Abstract

We clarify the difference between the asynchronous pricing algorithms analyzed by Asker, Fershtman and Pakes (2021) and those considered in the previous literature. The difference lies in the way the algorithms explore: randomly or mechanically. We reaffirm that with random exploration, asynchronous pricing algorithms learn genuinely collusive strategies.

JEL Classification: L41, L13, D43, D83

Keywords: artificial intelligence, Reinforcement Learning, Collusion, exploration

Emilio Calvano - emilio.calvano@unibo.it
University of Bologna and CEPR

Giacomo Calzolari - giacocalzolari@gmail.com
European University Institute and CEPR

Vincenzo Denicolò - vincenzo.denicolo@unibo.it
University of Bologna and CEPR

Sergio Pastorello - sergio.pastorello@unibo.it
University of Bologna

Acknowledgements
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).

ALGORITHMIC COLLUSION, GENUINE AND SPURIOUS†

Emilio Calvano*‡§, Giacomo Calzolari&§,
Vincenzo Denicolò*§ and Sergio Pastorello*

July 2021

We clarify the difference between the asynchronous pricing algorithms analyzed by Asker, Fershtman and Pakes (2021) and those considered in the previous literature. The difference lies in the way the algorithms explore: randomly or mechanically. We reaffirm that with random exploration, asynchronous pricing algorithms learn genuinely collusive strategies.

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

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

1. INTRODUCTION

In an important contribution to the literature on pricing algorithms and collusion, Asker, Fershtman and Pakes (2021) (henceforth AFP) have introduced the distinction between synchronous and asynchronous learning. Asynchronous learning is when the algorithm, after taking a certain action in a certain state and obtaining the associated payoff, updates its assessment of the value of that action in that state only, leaving its assessment of all other states and actions unchanged. Synchronous learning, on the other hand, is when the algorithm updates its assessment of a number of actions or states at a time, using some prior knowledge of the economic environment in which it operates. Synchronous algorithms have therefore the potential to learn faster and, if fed with accurate prior information, more effectively than the asynchronous ones.

To highlight the difference between synchronous and asynchronous algorithms, AFP (2021) present a figure, re-produced here as Figure 1, which depicts the evolution of prices when

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; ‡Toulouse School of Economics; &European University Institute; §CEPR

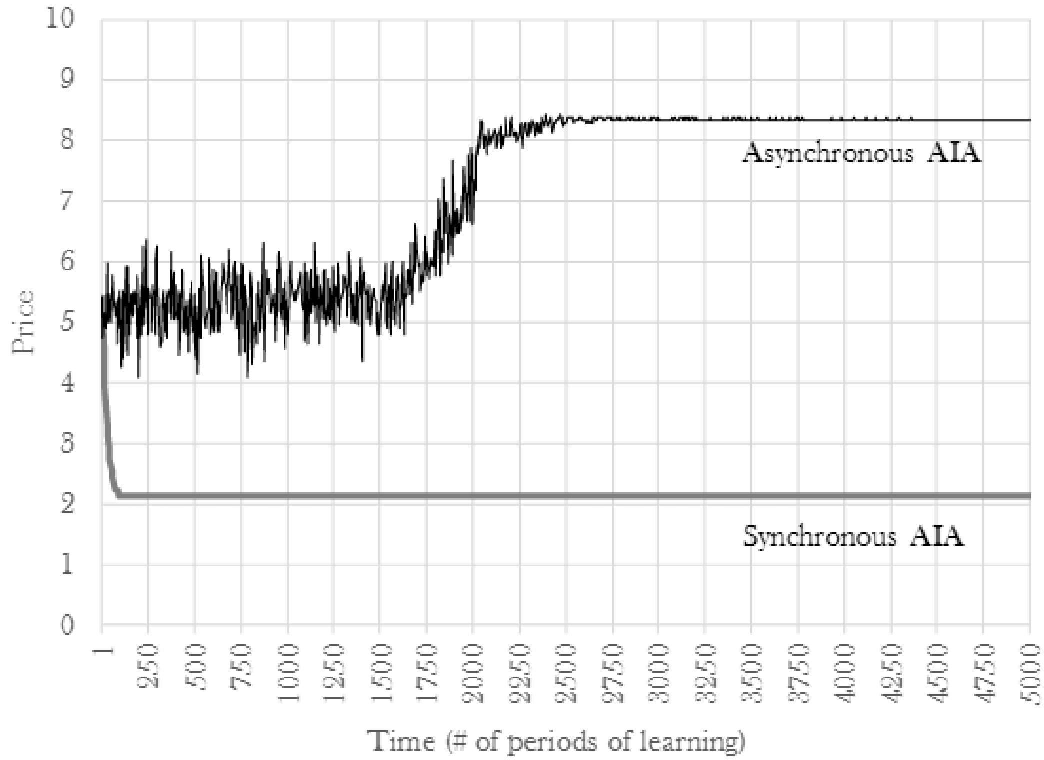

Figure 1: Price paths with synchronous and asynchronous algorithms (reproduced from Asker, Fershtman and Pakes (2021)). Firms repeatedly play a static Bertrand game and cannot condition current price on past prices. Market demand is 1 if price is less than 10, zero otherwise. Marginal cost is 2. There are 100, equally spaced feasible prices.

two competing algorithms play a sequence of independent games, where they cannot condition current on past prices. In this setting, the only equilibrium is to charge, period after period, the static Bertrand price (i.e., with homogeneous products, the unit cost). AFP's synchronous algorithms learn to play this equilibrium quite rapidly, whereas their asynchronous algorithms converge to prices substantially higher than the cost.

Economists generally agree that collusion is not synonymous with high prices but exists only when the high prices are sustained by a reward-punishment scheme that makes them an equilibrium outcome. Collusion is therefore impossible by design if players cannot condition their choices on the past history of the game. Thus, the supracompetitive prices which AFP's asynchronous algorithms converge to cannot reflect genuine collusion. Since most of the existing literature on algorithmic collusion, including our own previous work (Calvano, Calzolari, Denicolò and Pastorello, 2020; henceforth CCDP),[1] focuses on asyn-

[1]Other notable contributions include Klein (forthcoming), Johnson, Rhodes and Wildenbeest (2020) and Abada Lambin (2020).

chronous algorithms, one may wonder that the collusion documented in this literature is spurious. This note explains why it is not.

2. RANDOM V. AUTOMATIC EXPLORATION

A key difference between AFP's asynchronous algorithms and those of CCDP (2020) and others pertains to the mode of exploration. Generally speaking, reinforcement-learning algorithms must gather new information by experimenting. Following a common approach in computer science, CCDP (2020) focus on *random* experimentation: in order to acquire more information, with a certain probability the algorithm selects actions that may appear sub-optimal in the light of the knowledge it already possesses. By design, the probability of making such random choices is initially high but decreases as the learning proceeds.[2]

AFP (2021), on the other hand, adopt a different mode of experimentation. At each iteration, their asynchronous algorithms adopt the action that is currently perceived as optimal. However, they initialize the algorithm in such a way that it attributes very high values to all actions in all states. As the algorithm does take a certain action in a certain state and learns that the associated payoff is lower than it thought, it is automatically led to try different actions in the future. The process continues until the estimated values get in line with the payoffs actually experienced. This pattern of exploration, which is also popular in computer science, is often referred to as *optimistic initialization*.

Both modes of exploration can produce optimal choices in decision-theoretic settings, but less is known on how they perform in games of strategy. Given the paucity of theoretical results, only practice can tell which mode of exploration works better in specific multi-agent settings. In the case of competing pricing algorithms, it appears that optimistic initialization may, in some cases, mislead the algorithms into evidently suboptimal behaviors. This seems to be true to a much lesser extent for random exploration.

This point is illustrated in Figure 2. The economic environment is similar to that of Figure 1: that is, two pricing algorithms repeatedly interact in a market for a homogeneous product without conditioning their current choices on the past history of the game.[3] Under optimistic initialization, the asynchronous algorithms' behavior is, qualitatively, the same

[2] Random experimentation comes in two main variants, ε-greedy and Boltzman. In the ε-greedy model, when the algorithm is in exploration mode, all actions that are currently perceived as sub-optimal are tried with the same probability. In the Boltzman model, on the other hand, the probability with which an action is tried is a decreasing function of its perceived quality. Here we focus on the ε-greedy model. However, CCDP (2020) show that results are similar under Boltzman exploration.

[3] The figure depicts the average prices across 1,000 numerical simulations.

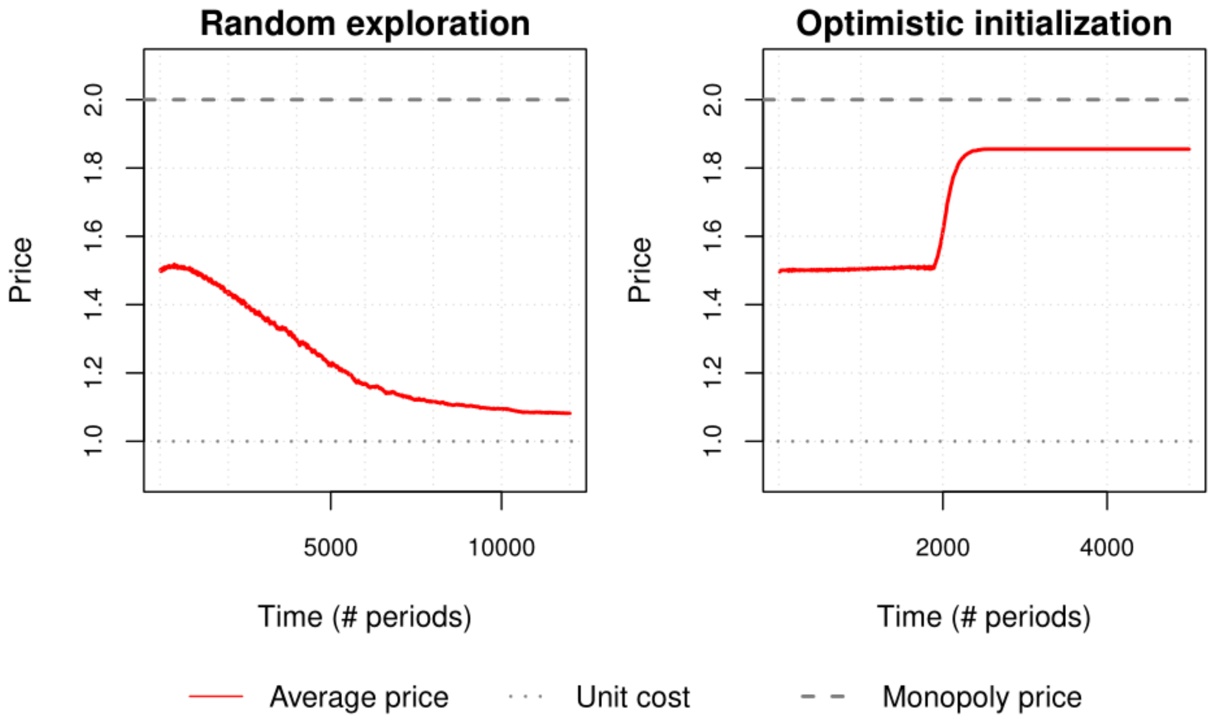

Figure 2: Average price path with different exploration mechanisms. As in Figure 1, algorithms repeatedly play a static Bertrand game and cannot condition their current price on past prices. Market demand is 1 if price is less than 2, zero otherwise. Marginal cost is 1. There are 15, equally spaced feasible prices. The right panel represents the case of mechanical exploration, as in AFP (2021); the left panel that of random exploration, as in CCDP (2020).

as in AFP, in spite of small differences with their set-up.[4] Under random exploration, on the other hand, the asynchronous algorithms learn to play competitively. Prices converge, in almost all of the cases, to the lowest feasible price that is higher than the unit cost (≈ 1.07), and never exceed that price by more than one step.[5]

The role of random exploration in facilitating learning is confirmed by AFP's own analysis.

[4]These differences arise because we use the same analytical setting (and the same codes for the numerical simulations) as in CCDP (2020). Therefore, we generate the case of homogenous products with rectangular demand by taking the limit of our 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 reservation price is 2; this is also the monopoly price. For the case of random experimentation, we set the learning and exploration parameters α and β of CCDP (2020) at $\alpha = 0.3$ and $\beta = 4 \times 10^{-4}$, respectively. For the case of optimistic initialization, the Q-matrix is initialized at very high values generated by a uniform distribution whose support lies above the equilibrium values. The codes used for the numerical simulations are publicly available at `https://www.aeaweb.org/articles?id=10.1257/aer.20190623`.

[5]As in CCDP (2020), we use a grid of 15 prices, of which the two lowest ones are lower than the unit cost. With discrete price levels, pricing one step above the unit cost weakly dominates pricing at or below cost. With a finer price grid (i.e., 100 prices rather than 15) the unit cost can be better approximated, and the algorithms indeed converge to prices closer to the unit cost.

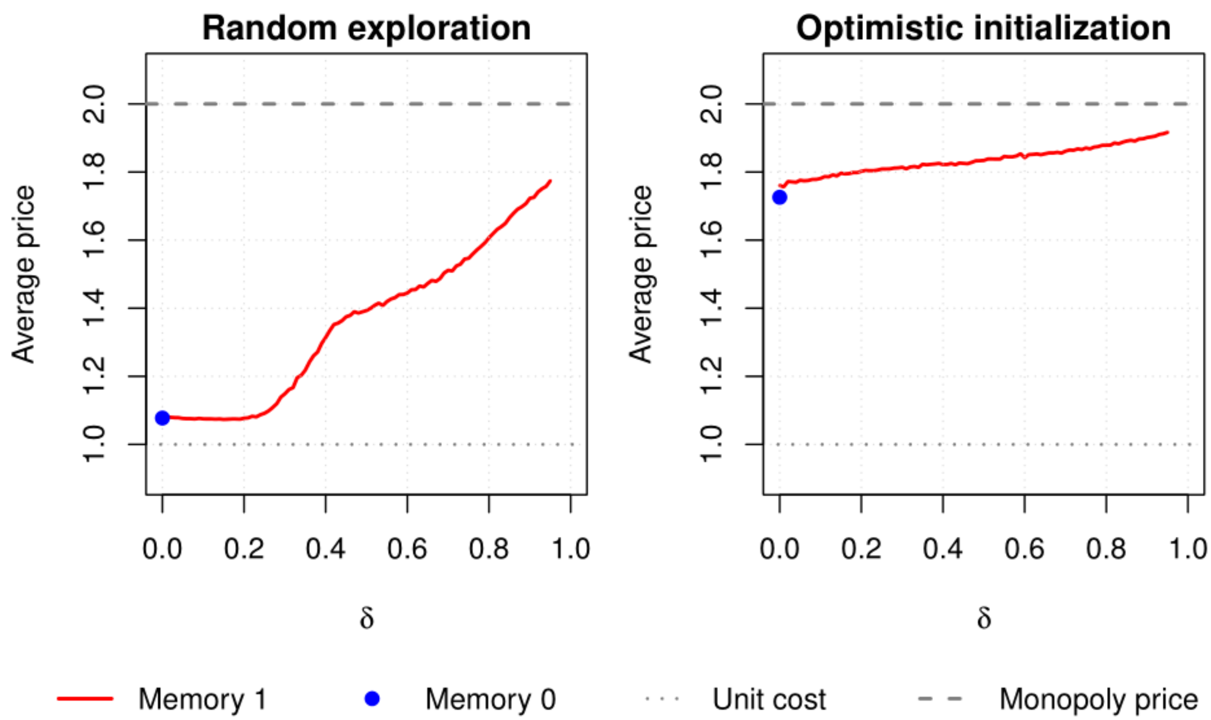

Figure 3: Average long-run prices with different exploration mechanisms for different values of the discount factor δ. The economic environment is as in Figure 2, but we now consider the possibility that algorithms may condition current price on past period's prices (we refer to this case as memory one, and to the case where no such conditioning is possible as memory zero).

In an extension, they combine optimistic initialization with random exploration and show that this brings the asynchronous algorithms closer to equilibrium play – the more so, the more extensive the random exploration (see Table 2 in AFP, 2021). However, the optimistic initialization they continue to use still affects the outcomes. Our simulations with random exploration use a neutral initialization that does not entail any systematic exploration.

Figure 3 compares the two modes of exploration when the algorithms can condition their current price on the previous period's prices. This in principle allows for collusive strategies. As is well known, however, such strategies can represent an equilibrium of the repeated game only if the discount factor δ is sufficiently high. Under random exploration, the asynchronous algorithms continue to price competitively when δ is low, and start converging to higher prices only when δ gets larger. Under optimistic initialization, on the other hand, prices are high even if δ is close to zero. In this case, the algorithms are so impatient that the high prices cannot be sustained in equilibrium by any dynamic reward-punishment scheme. The algorithms' failure to optimize is therefore apparent.[6]

Figures 2 and 3 show that the "anomalous" behavior of AFP's asynchronous algorithms is not due to asynchronicity in itself, but to the way they explore. Asynchronous algorithms can produce genuinely collusive outcomes with random experimentation.

3. CONCLUSION

As AFP (2021) emphasize, collusion is not synonymous with supracompetitive prices but arises when the high prices are sustained by a dynamic reward-punishment scheme. This note has reaffirmed that with random experimentation, asynchronous algorithms learn genuinely collusive strategies when they are forward looking. When the algorithms are myopic, on the other hand, they learn to price competitively. AFP's finding that even myopic asynchronous algorithms converge to supracompetitive prices is due to the specific mode of exploration they adopt.

The fact that asynchronous algorithms can collude genuinely does not mean that synchronous algorithms are not worth studying. To the extent that programmers possess reliable information on the structure of the market, they will try to use that information to improve the performance of the algorithms. Understanding how this impacts on the algorithms' behavior is clearly an important task for research.

REFERENCES

Abada, I. and Lambin, X. (2020). Artificial Intelligence: Can seemingly collusive otcomes be avoided? https://papers.ssrn.com/abstract=3559308.

Asker, J., Fershtman, C., and Pakes, A. (2021). Artificial Intelligence and Pricing: The Impact of Algorithm Design, CEPR discussion paper #DP 15880.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. American Economic Review, 110(10), 3267-3297.

Johnson, J., Rhodes, A., and Wildenbeest, M. R. (2020). Platform Design When Sellers Use Pricing Algorithms. Available at SSRN. https://doi.org/10.2139/ssrn.3691621

Klein, T. (forhcoming). Assessing Autonomous Algorithmic Collusion: Q-Learning Under Short-Run Price Commitments. RAND Journal of Economics.

[6]CCDP (2020) show that algorithms may make sub-optimal choices even under random exploration, but deviations from optimality are limited.