AEA Papers and Proceedings 2022, 112: 452–456
https://doi.org/10.1257/pandp.20221059

ALGORITHMIC PRICING[‡]

Artificial Intelligence, Algorithm Design, and Pricing[†]

By John Asker, Chaim Fershtman, and Ariel Pakes*

Firms increasingly delegate their pricing to algorithms that exploit detailed data on customers' preferences and that, in some instances, use a learning process to develop strategies to play an oligopolistic pricing game. Policymakers and researchers have raised the concern that this may diminish the competitiveness of markets and even lead to collusive behavior.[1]

This paper investigates how the design of artificial intelligences' (AIs) learning protocols can lead to competitive or supracompetitive price outcomes when competing in a simple Bertrand pricing game. We begin by describing the pricing game (Section I) and then the details of the AIs' learning protocols that will ultimately shape play and pricing outcomes (Section II). We then discuss the results (Section III) and conclude with some broader observations (Section IV).

I. The Bertrand Pricing Game

Consider two firms $i \in \{1,2\}$ with equal marginal costs, $c_1 = c_2 = 2$. Firms sell homogenous goods and compete by setting prices. We discretize the set of feasible prices to be 100 equally spaced numbers between 0.01 and 10, inclusive. Denote this set of feasible prices to be $\mathcal{P} = \{p^1, \ldots, p^M\}$. The price of firm i is denoted $p_i \in \mathcal{P}$. We assume that consumers buy from the firm with the lowest price. In case of a tie, firms split demand equally. We parameterize the model such that demand faced by firm i is

$$d_i(p_i,p_j) = \begin{cases} 1 & \text{if } p_i < p_j \text{ and } p_i \leq 10 \\ \frac{1}{2} & \text{if } p_j = p_i \text{ and } p_i \leq 10 \\ 0 & \text{otherwise} \end{cases}.$$

There are two Nash equilibria for this (static) Bertrand game. The first one is $p_i = p_j = 2.0282$. The second one is $p_i = p_j = 2.1291$.[2]

We allow this stage game to be repeated many times. This repetition allows the AIs, which set firms' prices, to learn to play. Ultimately, we are interested in the pricing outcomes at which the AIs converge.

II. Design Features of a Pricing AI

The artificial intelligence algorithms (AIAs) playing this Bertrand pricing game use a reinforcement learning algorithm.[3] The AIAs in this paper do not value the future (that is, they only value profit in the current period). This rules out "collusive"-style equilibria in which play is shaped by perceptions of the future returns flowing from a present action.[4]

‡*Discussants:* Emilio Calvano, University of Bologna; Alexander MacKay, Harvard University; Ignacia Mercadal, Columbia University; Gaurab Aryal, University of Virginia.

*Asker: UCLA and NBER (email: johnasker@econ.ucla.edu); Fershtman: Eitan Berglas School of Economics, Tel Aviv University and CEPR (email: fersht@post.tau.ac.il); Pakes: Harvard University and NBER (email: apakes@fas.harvard.edu). We thank participants in various seminars and conferences for their comments. Excellent research assistance was provided by Lorenzo Cattivelli, Masha Melkonyan, and Kylie Toh.

†Go to https://doi.org/10.1257/pandp.20221059 to visit the article page for additional materials and author disclosure statement(s).

[1] See, for instance, the discussions in Competition and Markets Authority (2018); Brown and MacKay (forthcoming); Assad et al. (2021); and Calvano et al. (2020).

[2] This equilibrium structure is a consequence of the set of feasible prices being discrete. If prices were continuous, this would reduce to the familiar undergraduate example in which $p = c$ is the equilibrium. We assume that firms do not have capacity constraints.

[3] We use variants of Q-learning, as discussed by Watkins and Dayan (1992). For a survey of reinforcement learning algorithms, see Sutton and Barto (2018).

[4] Similarly, the state space is a singleton, ruling out history-dependent strategies. Asker, Fershtman, and Pakes

In this simple setting, a reinforcement learning algorithm has the following ingredients:

(i) A set of values for every algorithm (equivalently, firm) that can be interpreted as the algorithm's perception of the expected profit conditional on setting each feasible price. That is, for every firm i in each period k,

$$\mathcal{W}_i^k = \left\{W_i^k(p)\right\}_{p \in \mathcal{P}}.$$

(ii) A method for choosing the price p_i^k in every period, conditional on $\mathcal{W}_i^k$. Here, the AIA chooses $p_i^k = \operatorname{argmax}_{p \in \mathcal{P}}\left\{W_i^k(p)\right\}$.

(iii) An updating rule. The updating rule uses the information the firm observed during that period to update $\mathcal{W}_i^k$ to generate $\mathcal{W}_i^{k+1}$.

The updating rule determines how the algorithm learns the returns to an action. In period 1, initial values of $\mathcal{W}_i^k$ need to be imposed. Here, $W_i^1(p) \sim_{i.i.d} U(10, 20)$. This sets all starting values higher than monopoly profits and so encourages exploration during the AIAs' learning process.[5] In subsequent periods, updating can occur in a number of ways. We consider three approaches, which vary in the amount of information required and in economic sophistication:

(i) *Asynchronous Updating.*—Only $W_i^k\left(p_i^k\right)$ is updated, where p_i^k is the price chosen by i in period k. Given a speed-of-learning parameter, α,

$$W_i^{k+1}\left(p_i^k\right) = \alpha\pi\left(p_i^k, p_j^k\right) + (1-\alpha)\, W_i^k\left(p_i^k\right)$$

and $W_i^{k+1}(p) = W_i^k(p)$ for all $p \neq p_i^k$.

(ii) *Synchronous Updating.*—$W_i^k(p)$ is updated for all prices with an estimate, denoted $\pi^e\left(p, p_j^k\right)$, of what profits would have been had the firm chosen each price $p \in \mathcal{P}$; i.e.,

$$W_i^{k+1}(p) = \alpha\,\pi^e\left(p, p_j^k\right) + (1-\alpha)\, W_i^k(p).$$

Algorithms using asynchronous or synchronous updating require access to different amounts of information. Asynchronous updating only requires knowledge of the profits received from the price actually played. The amount of information that synchronous updating requires depends on how profits from counterfactual prices, i.e., $\pi^e\left(p, p_j^k\right)$ for $p \neq p_i^k$, are calculated.

One extreme is when the algorithm sees the competitor's price and knows the demand and cost functions. Then it can calculate what actual profits would have been had the algorithm played a different price. A much weaker assumption is that all the algorithm knows is that demand slopes downward. Then, if $p > p_i^k$ and $W^k(p) > (p-c)q\left(p_i^k, p_j^k\right)$, we use the updating equation in part (ii) above with $\pi^e\left(p, p_j^k\right) = (p-c)q\left(p_i^k, p_j^k\right)$, as downward-sloping demand implies that profits could not be as high as $W^k(p)$.

Similarly, if $p < p_i^k$ and $(p-c)q\left(p_i^k, p_j^k\right) > W^k(p)$, then $\pi^e\left(p, p_j^k\right) = (p-c)q\left(p_i^k, p_j^k\right)$. If neither of these conditions are met, $W^{k+1}(p) = W^k(p)$.

We call the first case, in which the algorithm can calculate counterfactual profits exactly, *perfect synchronous updating*. Its observational and computational requirements typically grow with increases in the complexity of the underlying model (more firms, differentiated products, …). We call the second case *synchronous updating using downward demand*. Note that all it ever requires in addition to the current profits needed for the asynchronous case is knowledge of the quantity actually sold during the period.

III. Results

Figure 1 summarizes pricing outcomes when both firms use asynchronous updating, perfect synchronous updating, and downward-sloping demand (panels A, B, and C, respectively). Each panel shows outcomes

(2021) contain a more general discussion in which AIs may put a positive weight on future profits, and they obtain results that are qualitatively similar to those presented here.

[5] In our example, exploration is induced by high initial conditions, but one could also explore through periodic choice of policies that do not maximize perceived values; see Asker, Fershtman, and Pakes (2021) and the literature cited there.

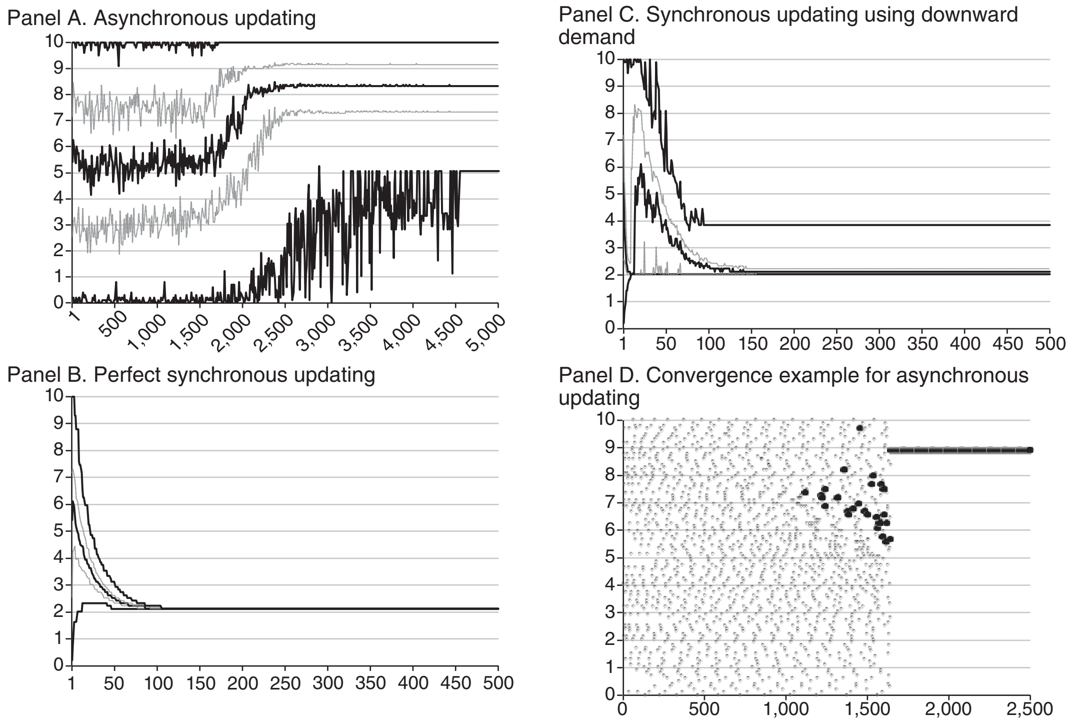

FIGURE 1. PRICE OUTCOMES WITH DIFFERENT ALGORITHM DESIGNS

Notes: Panels A, B, and C: Prices (vertical axis) by period (horizontal axis) from 100 simulations are shown. The lines, from bottom to top, represent the minimum (thin, black solid line), twenty-fifth percentile (thin, grey dashed line), median (thick, black solid line), seventy-fifth percentile (thin, grey dashed line), and maximum (thin, black solid line) of the distribution of prices in each period. Results are for a static Bertrand market with two firms selling homogeneous goods. Results are shown for firm 1. The model is parametrized as follows. Demand, $Q = 1$ if $P \leq 10$, 0 otherwise. Marginal cost = 2. Feasible prices exist on a grid with 100 elements equally spaced between 0.1 and 10, inclusive. AIs put a zero weight on future profits (the future is discounted to zero). The weight on current returns in updating is given by $\alpha = 0.1$. Initial conditions are i.i.d. draws from $U[10,20]$, for each $W(p)$ for each firm. In panel A, only every tenth period is shown. In panel A, the minimum, twenty-fifth percentile, median, seventy-fifth percentile, and maximum values after 5000 periods are 5.06, 7.33, 8.34, 9.14, and 10, respectively. In panel B, the minimum, twenty-fifth percentile, median, seventy-fifth percentile, and maximum values after 500 periods are all equal to 2.1291. In panel C, the minimum, twenty-fifth percentile, median, seventy-fifth percentile, and maximum values after 500 periods are 2.0282, 2.129, 2.1291, 2.23, and 3.84, respectively. In panel D, prices (vertical axis) by period (horizontal axis) are shown for firm 1 during a single simulation chosen from the 100 runs to generate panel B. Hollow light-grey circles indicate (chosen) prices for which the $W(p)$ is updated downward. Solid heavy-black circles indicate (chosen) prices for which the $W(p)$ is updated upward.

from 100 simulations, which differ only in the initialization of $\mathcal{W}_i^k$. There are five lines in each of the panels. From bottom to top, these are, by period, the minimum, twenty-fifth percentile, median, seventy-fifth percentile, and maximum price across the 100 simulations that were run. Panel D provides an example of the convergence process for the asynchronous updating case.

Panel A shows the price distribution by period when both firms use asynchronous updating. Convergence is relatively slow, but after 4,600 periods, none of the paths generated in any of the 100 simulations change, and all of the percentiles stay constant. Most notably, all simulations reach rest points that are significantly higher than the Nash equilibrium prices. The median price is 8.34, and the minimum is 5.06. Note also that all quantiles tend to increase over the course of the learning process; i.e., they move *away* from the Nash equilibria.

By contrast, when the AIAs learn via perfect synchronous updating (panel B), prices converge quickly to Nash pricing levels. In this instance, prices across all 100 simulations settled on 2.13

after approximately 105 iterations. Thus, learning was much faster, prices converged to a static Nash equilibrium,[6] and initial conditions had no influence on the limiting prices.

Panel C implements a synchronous protocol that exploits the economic premise that demand slopes downward. This adds simple economic reasoning to the asynchronous algorithm without imposing the observational and computational requirements needed for the counterfactuals calculations used in perfect synchronous updating. In panel C, the minimum rest point is 2.03 and the maximum is 3.84. Recall that the desire to explore alternative policies induced a choice of high initial values. The reason that the outcomes in panel C differ so markedly from those in panel A is that imposing downwardly sloping demand insures that the W's attached to a higher price than the price actually played can only be updated downward.

The outcomes in panel A, in which both firms use asynchronous updating, contrast with the outcomes generated in panels B and C. This is true for both the speed of convergence and the ultimate prices that are realized. In particular, panel C suggests that leveraging minimal assumptions about the economic environment can have a significant mitigating impact on any tendency for an asynchronous algorithm to generate supracompetitive prices and also dramatically decreases the iterations needed until convergence.

Panel D provides more detail on the convergence process occurring when firms use asynchronous updating. For a single simulation run, the prices chosen by firm 1's AI are shown with circles. The hollow, small light-grey circles indicate chosen prices for which the $W_1^k\left(p_1^k\right)$ is updated downward; solid heavy-black circles indicate chosen prices for which the $W_1^k\left(p_1^k\right)$ is updated upward. In panel D, prices converge to 8.89 in period 1,645. The fact that the initial condition draws are higher than monopoly profits implies that in early periods, the evaluations fall. That is, when a price p_1^k is chosen, it generates a lower profit than the $W_1^k\left(p_1^k\right)$ associated with p_1^k, and so $W^{k+1}\left(p_1^k\right)$ is lower than $W_1^k\left(p_1^k\right)$. Eventually, the initial valuation of some other price is higher than the profit earned from the given price, so some other price will be chosen.

The first time the chosen price, p_i^k, generates a profit that is higher than $W_1^k\left(p_1^k\right)$ is in iteration $k = 1{,}119$. This leads to an upward adjustment and a solid black circle in panel D. As long as the price of the rival is not reduced to match or undercut the price, firm i does not change its price. Since at $k = 1{,}119$, the rival's price is greater than p_1^{1119}, it obtains no profits and $W_2^{1119}\left(p_2^{1119}\right)$ is updated downward. Eventually, it chooses a different price that, if equal to or lower than p_1^{1119}, generates positive profits. If the rival's new price is lower, firm 1's profits go to zero and its evaluation of p_1^{1119} falls. However, if the rival were to choose p_1^{1119}, both firms earn positive profits. If, in addition, at that k, $\pi\left(p^k, p^k\right) \geq \max\left[W_i^k(p), W_j^k(p)\right]$, the algorithm will have converged to p^k. In panel D, the two firms match on price 22 times before converging in period 1,645.[7]

The comparison of outcomes in panels A, B, and C illustrates the importance of the learning (or updating) process in determining the outcomes that may emerge when AIs play against each other. At least for this particular setting, asynchronous updating leads to supracompetitive ("high") prices. Importantly, this happens in a setting in which firms do not care about the future and are unable to play history-dependent strategies. Thus, though standard collusive equilibria of the sort familiar from the repeated game literature are not feasible, the asynchronous algorithm always generates rest point prices that are significantly above the Nash equilibrium values. This is a consequence of the convergence process that occurs when asynchronous algorithms play against each other in an environment with exploration. By contrast, synchronous updating leads to pricing that is in line with what economists would think of as a competitive outcome (the static Nash equilibrium in prices). When asynchronous algorithms are imbued with some limited economic sophistication (here, by allowing them to understand that demand slopes downward), the supracompetitive prices that asynchronous algorithms generate are substantially mitigated,

[6] In the static case, one can prove convergence to Nash more generally, without assuming a particular structure for demand. See Asker, Fershtman, and Pakes (2021).

[7] By contrast, convergence for AIs using a synchronous algorithm follows, after a small amount of initial randomness, a best response dynamic.

and the speed of convergence is greatly accelerated. This suggests that an understanding of the competitive impacts of algorithmic pricing games requires knowledge of how the algorithms learns.

IV. Concluding Remarks

These results are presented in a setting in which the AIAs do not care about future payoffs and history-dependent strategies are not possible (the state space is a singleton). If these restrictions are relaxed, "collusive"-type strategies become possible (see Calvano et al. 2020). By ruling out the potential for this "collusive"-type play, this paper illustrates an additional channel through which market interaction between AIAs can lead to price elevation and shows that the design of the learning protocol at the heart of the AIA can be a key determinant of the extent to which pricing outcomes are competitive. In related work, we verify that even when AIAs care about the future and state spaces allow for history-dependent strategies, the nature of the learning protocol is likely to still have an economically significant impact on the degree to which prices are supracompetitive (see Asker, Fershtman, and Pakes 2021).

REFERENCES

Asker, John, Chaim Fershtman, and Ariel Pakes. 2021. "Artificial Intelligence and Pricing: The Impact of Algorithm Design." NBER Working Paper 28535.

Assad, Stephanie, Robert Clark, Daniel Ershov, and Lei Xu. 2021. "Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market." Unpublished.

Brown, Zach Y., and Alexander MacKay. Forthcoming. "Competition in Pricing Algorithms." *American Economic Journal: Microeconomics*.

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

Competition and Markets Authority. 2018. "Pricing Algorithms: Economic Working Paper on the Use of Algorithms to Facilitate Collusion and Personalised Pricing." CMA Working Paper CMA94.

Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press.

Watkins, Chris, and Peter Dayan. 1992. "Q-learning." *Machine Learning* 8: 279–92.