

July 2021

“Platform Design when Sellers Use Pricing Algorithms”

Justin Pappas Johnson, Andrew Rhodes and Matthijs Wildenbeest

PLATFORM DESIGN WHEN SELLERS USE PRICING ALGORITHMS

JUSTIN P. JOHNSON[†] ANDREW RHODES[‡] MATTHIJS WILDENBEEST[§]

July 9, 2021

ABSTRACT. Using both economic theory and Artificial Intelligence (AI) pricing algorithms, we investigate the ability of a platform to design its marketplace to promote competition, improve consumer surplus, and even raise its own profits. We allow sellers to use Q-learning algorithms (commonly used in computer science) to devise pricing strategies in a setting with repeated interactions, and consider the effect of platform rules that reward firms that cut prices with additional exposure to consumers. Overall, the evidence from our experiments suggests that platform design decisions can meaningfully benefit consumers, but that achieving these gains may require demand-steering policies that condition on past behavior and treat sellers in a non-neutral fashion. These more sophisticated policies disrupt the ability of algorithms to rotate demand and split industry profits, leading to very low prices. This is consistent with our theoretical results, which show that a platform can undermine collusion even when firms become infinitely patient.

Many products and services are sold through online platforms such as Amazon, Booking, and eBay. Sellers on these platforms are increasingly using algorithms to decide what prices they should charge.¹ This trend towards algorithmic pricing has generated significant debate. On the one hand algorithms might be procompetitive, enhancing efficiency by enabling sellers to react more quickly to changes in demand and supply conditions. On the other hand, policymakers and legal scholars have raised the concern that algorithms might be used to foster and sustain collusion.² Indeed, several legal cases involving algorithms facilitating collusion have already been tried in court.³ Furthermore, recent experimental work by economists

We thank Emilio Calvano, Han Huynh, Bruno Jullien, Timo Klein, Scott Duke Kominers, Alexander MacKay, Patrick Rey, Tat-How Teh, Johannes Walter, and participants at the Bates-White Antitrust Conference, the University of California (Berkeley), the University of California (Irvine), Compass Lexecon, the Conference on Algorithmic Collusion (University of Amsterdam), the Conference on Digital Platforms, the DERN Webinar (Monash University), the 2020 Econometric Society Meetings (San Diego), the Toulouse School of Economics Online Economics of Platforms Seminar, the Workshop on Pricing Technologies (European University Institute), and the ZEW Conference on the Economics of Information and Communication Technologies.

[†]Cornell University, justin.johnson@cornell.edu.

[‡]Toulouse School of Economics and CEPR, andrew.rhodes@tse-fr.eu. Rhodes acknowledges funding from the European Research Council (grant No 670494) and Agence Nationale de la Recherche (grant ANR-17-EURE-0010).

[§]Indiana University and CEPR, mwildenb@indiana.edu.

¹According to the OECD (2017a) “the use of pricing algorithms by professional sellers is becoming increasingly common, if not ubiquitous.” Meanwhile Chen, Mislove, and Wilson (2016) estimated that, even in 2015, algorithms were involved in the pricing of around one-third of the roughly 1600 best-selling products on the Amazon marketplace. Since then a whole industry has emerged specializing in automated pricing software for use on Amazon.

²Regulatory concerns are evident in, for example, OECD (2017b), CMA (2018), and DOJ (2018).

³See the *Topkins* case, pursued by the U.S. Department of Justice, and the *Eturas* case in the European Union.

has shown that Artificial Intelligence (AI) algorithms may learn to play collusive strategies, while empirical work suggests that adopting AI algorithms may lead to higher prices.⁴

At the same time, policymakers are increasingly highlighting the importance of platform design—namely the rules that govern interactions between users on a platform.⁵ One leading example is the rules that govern how different products are ranked or displayed to consumers. Platforms can use these rules to actively shape the nature of competition on their marketplaces. This has prompted some scholars to argue that dominant platforms should have a responsibility to design rules which promote competition on their marketplaces.⁶

In this article we use both economic theory and extensive experiments on artificial intelligence pricing algorithms to demonstrate that platforms may indeed be able to design rules that increase competition on their marketplaces. Relatively simple rules may suffice when sellers (or their algorithms) behave competitively, but more subtle ones—which condition on past behavior and treat sellers asymmetrically—may be required when there is a risk of collusion. Although we do not claim that platforms should be compelled by policymakers to actively promote competition, our results suggest that platforms may have the tools do so. Even if platforms were not compelled to do so, platforms might wish to implement such policies: sometimes the rules we suggest simultaneously benefit consumers and raise platform profits.

In more detail, we consider a setting with multiple firms whose products are differentiated according to the standard logit model. These firms compete over an infinite horizon on a monopoly retail platform. We consider two platform design rules, both of which involve steering consumer demand towards a subset of the sellers. The simpler of these policies, *price-directed prominence* (PDP), involves the marketplace simply steering demand towards sellers that are charging lower prices within a given category. Relatively high-priced sellers are shown to fewer consumers, reducing their profits. The more subtle of our two policies, *dynamic price-directed prominence* (Dynamic PDP), also conditions on past prices. These two policies capture a platform’s ability to rank and display products, and thereby restrict a consumer’s attention to a subset of the available products.⁷

Section 1 of our article examines the theoretical impact of the two design policies. Theory predicts that prices fall, and so PDP introduces a tradeoff for consumers between lower prices and less product variety. Whether consumers on balance benefit hinges on whether sellers

⁴See Calvano, Calzolari, Denicolò, and Pastorello (2020) and Klein (2021), discussed further below, for the experimental evidence, and Assad, Clark, Ershov, and Xu (2020) for the empirical evidence. Brown and MacKay (2019) show that adopting (non-AI) algorithms allows firms to achieve a form of price commitment that raises overall prices.

⁵See ec.europa.eu/digital-single-market/en/digital-services-act-package and Furman et al. (2019).

⁶For example Crémer, de Montjoye, and Schweitzer (2019) write that “Dominant platforms have ‘regulatory power’ and have a responsibility to use that power in a pro-competitive manner.”

⁷A related policy is the Amazon “buy box,” which shows consumers one seller among those offering a given homogeneous product (in contrast our analysis considers differentiated products). Current price substantially determines which firm “wins” the buy box, but past performance also matters. See Gómez-Losada and Duch-Brown (2019).

are cartelized or not. When sellers do not collude, PDP raises consumer surplus so long as no more than about two-thirds of products are obscured, but when sellers collude PDP lowers consumer surplus irrespective of the number of products that are obscured.

Because PDP is not predicted to work well when sellers are colluding, our second technique, Dynamic PDP, attacks the foundations of collusion more directly. It does so by steering additional demand to a firm not just in the period in which it cuts prices but also in some later periods, subject to the firm not raising prices and not being undercut by more than a “cushion” set by the platform. The net effect is that it is more difficult to punish a firm that has cut prices, and therefore incentives to deviate are amplified, reducing cartel stability.

Dynamic PDP has excellent theoretical properties: collusion becomes unsustainable even as firms become nearly infinitely patient, and consumers often benefit. This is important because in principle algorithms might operate in real time, which is theoretically equivalent (in terms of cartel stability) to being very patient.

Starting in Section 2, we use experiments to assess the impact of the two design rules when prices are set by simple AI algorithms. Our focus on AI algorithms is motivated by the growing use of AI in a range of business applications. It seems likely that such techniques will soon be widely employed to set prices, and indeed many providers of repricing software already market their products as using artificial intelligence.⁸ We use a particular type of AI algorithm called Q-learning, which is popular among computer scientists. As we describe more fully in Section 2, Q-learning is a reinforcement-learning algorithm which attempts to find an optimal solution to a potentially complex problem based on its own historical experience with the environment.

In our algorithmic experiments we find, as a preliminary result, that absent PDP the algorithms typically set prices that exceed the predicted static Bertrand-Nash prices. This confirms a key contribution of Calvano, Calzolari, Denicolò, and Pastorello (2020), who study interactions between algorithms in a setting closely related to ours.

Our experiments with these AI algorithms reveal that PDP lowers prices, but that the price declines need not be large enough to compensate consumers for the loss of variety. In this sense, sometimes algorithms behave more like competitive firms (with PDP helping consumers) but other times behave more like a cartel (with PDP harming consumers). PDP is less likely to benefit consumers when the algorithms value future profits highly, that is when they operate with high discount factors, which is intuitive given that economic theory predicts cartels are easier to support when this is so. But even when firms are patient, consumers sometimes benefit even though theory suggests they should not; this is more likely when product differentiation (as measured by the scale parameter in the standard

⁸See goaura.com, sunpricer.com, feedvisor.com/amazon-repricer, kalibrate.com, and a2isystems.com.

logit model) is lower. When consumers do benefit, the consumer-surplus gains can be large, sometimes exceeding 35%.

Our experiments with our second rule, Dynamic PDP, reveal that the algorithms respond very strongly to it, even when the discount factor is high. Prices drop substantially both compared to (regular) PDP and compared to no PDP. And there are also large increases in consumer surplus—we sometimes observe gains higher than 75%. Thus, our results suggest that Dynamic PDP may be a more effective tool than PDP to fight algorithmic collusion.

Beyond the evidence on price levels discussed above, we also examine the pricing patterns adopted by the algorithms. We find that, absent platform intervention, the algorithms often set constant prices that result in industry profits being split nearly equally. The algorithms typically respond to PDP by adopting price cycles rather than constant prices. Surprisingly, by doing this the algorithms manage to rotate demand and again split industry profits almost equally over the long run, even though by its nature PDP makes profits highly asymmetric in any given period. When Dynamic PDP is implemented, however, the algorithms rotate demand much less often, especially when the algorithms are able to track who has the pricing “cushion.” This results in very low prices and highly asymmetric profits.

Overall, the evidence from our experiments suggests that platform design decisions can meaningfully benefit consumers even when algorithmic collusion might otherwise emerge. However, achieving these gains may require more subtle steering policies such as Dynamic PDP. We also find that rules that raise consumer surplus can raise the profits of the platform, depending on the platform’s revenue model.

Our work is part of a broader literature on collusion and algorithms that spans multiple disciplines, including computer science, economics, and law. Mehra (2015) and Ezrachi and Stucke (2017) raise the concern that algorithms may facilitate collusion. They identify several ways this might occur, including AI algorithms learning to collude without explicit human direction. Harrington (2018) discusses policy issues, such as whether the proper definition of collusion ought to require an explicit agreement among conspirators or whether collusion is better defined as elevated prices supported by a reward-and-punishment scheme.

The prospect of AI algorithms learning to cooperate in prisoner’s dilemma games has been studied by Sandholm and Crites (1995), Tesauro and Kephart (2002), and Waltman and Kaymak (2008). Recent work goes further. Calvano, Calzolari, Denicolò, and Pastorello (2020) study collusion by AI algorithms in a logit model of differentiated products. In addition to finding prices that are elevated compared to theoretical predictions of non-collusive behavior, they identify that algorithms may learn to support collusive outcomes using reward-and-punishment schemes. Klein (2021) also studies the strategies algorithms utilize in a setting where firms selling homogeneous products take turns changing prices, and finds Edgeworth price cycles supporting supranormal profits. These articles use the

same type of reinforcement-learning algorithms that we do.⁹ They are part of a broader investigation into cooperation in games dating back to Axelrod and Hamilton (1981).¹⁰

In terms of structural market changes to fight collusion, Kovacic, Marshall, Marx, and Raiff (2006) detail methods that may apply in procurement auctions. The study of buyer groups by Dana (2012) is closer in spirit to our steering proposals, although he does not consider a collusive framework or the role of algorithmic pricing. Like Dana and us, Dinerstein, Einav, Levin, and Sundaresan (2018) recognize the tradeoff between prices and variety in their empirical study of platform design. Teh (2020) examines how trade-offs associated with platform design are affected by the platform’s fee structure. Also related are studies of intermediaries that bias recommendations or otherwise steer demand to raise their own static monopoly profits (Hagiu and Jullien, 2011; Inderst and Ottaviani, 2012; De Corniere and Taylor, 2019; Teh and Wright, 2020). Our approach differs in a number of ways but particularly in that we allow for (algorithmic) collusion and focus on how an intermediary can raise consumer surplus.

1. MODEL

Here we lay out a theoretical model of how platform design affects prices and other outcomes. There are n firms, each of which sells one differentiated product on a monopoly retail platform. The firms have the same constant marginal cost $c > 0$, which is inclusive of any fees paid to the platform. Firms interact repeatedly over time. In each period $t = 0, 1, \dots$, each firm i observes all past prices and other outcomes and simultaneously sets a price $p_i^t \geq 0$ (negative prices are not allowed). The firms have a common discount factor $\delta \in (0, 1)$.

In each period t there is also a unit mass of consumers, each of whom wishes to buy at most one product. Consumers spend one period in the market and then exit and are replaced by a new cohort. A representative consumer who buys product i in period t obtains utility

$$u_i^t = a - p_i^t + \zeta_i.$$

If a consumer buys no product then she obtains the outside option utility

$$u_0^t = a_0 + \zeta_0.$$

We assume that ζ_0 and each ζ_i are independent random variables with a type I extreme value distribution that has common scale parameter $\mu > 0$.

⁹Asker, Fershtman, and Pakes (2021) use a different type of algorithmic learning which conducts counterfactuals to learn about the returns that would have been earned if an alternative action had been taken. Other papers use economic theory but not actual algorithms. Miklós-Thal and Tucker (2019) and O’Connor and Wilson (2019) assume that algorithms improve the quality of information available and ask how that changes the structure of collusion. Salcedo (2015) explores the effect on prices when sellers commit to particular pricing algorithms.

¹⁰There is also a large literature specifically on whether and how humans learn to collude; Dal Bó and Fréchet (2018) provide an extensive summary and review of this literature. Deck and Wilson (2003) examine the use of pricing algorithms in an experimental setting in which humans choose which types of (non-AI) algorithms to use.

In each period t the platform displays a subset \mathcal{N}_t of the products to consumers. Consumers can only buy a displayed product. Therefore given our assumptions, each firm not in \mathcal{N}_t receives zero demand, whereas a firm $i \in \mathcal{N}_t$ receives (standard logit) demand

$$D_i(p^t) = \frac{\exp\left(\frac{a-p_i^t}{\mu}\right)}{\sum_{j \in \mathcal{N}_t} \exp\left(\frac{a-p_j^t}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)}, \quad (1)$$

where p^t is the vector of prices of the n firms at time t . Consumer surplus in period t is

$$U^t(p^t) = \mu \log \left[\sum_{j \in \mathcal{N}_t} \exp\left(\frac{a-p_j^t}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right) \right], \quad (2)$$

while total industry output is $\sum_{j \in \mathcal{N}_t} D_j(p^t)$ and total industry revenue is $\sum_{j \in \mathcal{N}_t} p_j^t D_j(p^t)$.

Below we consider two ways in which the set of displayed firms \mathcal{N}_t could be generated.

1.1. Price-Directed Prominence. The first platform intervention that we consider is price-directed prominence (PDP), which is formally defined as follows.

Definition 1 (Price-Directed Prominence). *In any given period t each consumer only observes k firms with the lowest prices, for some fixed integer $k \in \{1, 2, \dots, n-1\}$.*

In case two or more firms are tied for the k th lowest price, then a subset of the firms with the k th lowest price is randomly chosen so as to ensure that exactly k firms are displayed.

We now evaluate the theoretical effectiveness of price-directed prominence. Its predicted performance depends on several factors, including whether firms behave like a cartel or not. First we assess predicted outcomes when the market is competitive.

1.1.1. PDP in a Competitive Market. Suppose that in each period the n firms choose prices as if this is the only period in which they compete, that is, firms play a one-shot Bertrand-Nash pricing game. If PDP is not in effect—and so all firms are displayed—there is a unique equilibrium where each firm charges $p_{BN}^* > c$. In contrast, under PDP prices are driven down to marginal cost as firms compete for the right to be displayed to consumers.

Lemma 1. *When PDP is in effect, in a competitive market the k lowest prices equal marginal cost in equilibrium.*

PDP therefore presents a tradeoff for consumers between lower prices but less variety.

Proposition 1. *In a competitive market, compared to when all n firms are displayed:*

- (1) *If $k/n > e^{-1} \approx 0.368$ then PDP increases consumer surplus and total output. It increases revenue if, in addition, $a - c < a_0$ and μ is sufficiently small.*
- (2) *If $k/n < e^{-\frac{n}{n-1}}$ then PDP decreases consumer surplus, total output, and revenue.*

The first part of Proposition 1 says that PDP raises consumer surplus and total output whenever the proportion k/n of firms displayed is large enough. Indeed, even if as many as 63% of products are not shown, the resulting gains from intensified price competition are enough to benefit consumers.¹¹ (Of course, it is better to show consumers more products conditional on them being priced at marginal cost.) Unfortunately it is hard to provide general conditions under which PDP increases revenue. However, revenue does increase when, for example, the outside option a_0 is high and the level of differentiation μ is low.

On the other hand, the second part of Proposition 1 says that PDP reduces consumer surplus and total output if too few firms are displayed, because the loss of variety then dominates.¹² Since PDP reduces prices, whenever it reduces output it also reduces revenue.

Overall, our results show that when markets are competitive PDP may benefit consumers and increase both the volume and value of transactions on the platform. However some markets may not be well characterized by our assumption of Bertrand-Nash competition. We next consider the possibility that the market is instead cartelized.

1.1.2. PDP in a Cartelized Market. We focus on the theoretical performance of PDP under *full collusion*—whereby in each period the n firms set prices to maximize their joint profits. Fully collusive prices therefore maximize the profit of an n -product monopolist. The following simple lemma characterizes monopoly prices.

Lemma 2. *An n -product monopolist would charge the same price $p^m(k)$ on k of the products, and charge weakly more than $p^m(k)$ on the remaining $n - k$ products.*

We first look at how PDP affects the sustainability of full collusion as an equilibrium outcome. Full collusion is sustainable if and only if, in each period, each firm's net present value of colluding exceeds its discounted payoff from deviating. As is typical in models of collusion, for a given k full collusion is easier to maintain when δ is larger. It turns out that full collusion is sustainable for the broadest range of discount factors when (i) all firms charge $p^m(k)$ in each period until there is a deviation, and (ii) after a deviation firms play the one-shot Bertrand-Nash equilibrium in Lemma 1 forevermore. Part (i) implies that in each period the platform randomly selects which firms to display, and so in each period in expectation every firm gets an equal share of the monopoly profit. (Alternative schemes, where firms alter their prices over time so as to control how demand is rotated between cartel members, can be shown to sustain collusion for a narrower range of δ .) Meanwhile part (ii) ensures that a firm earns zero in each period after it deviates, despite intrinsic differentiation.

¹¹This 63% threshold is certainly met if only one firm is not shown, that is if $k = n - 1$.

¹²The condition $k/n < e^{-\frac{n}{n-1}}$ is satisfied if, for example, $k = 1$ and $n \geq 4$. We also note that as $n \rightarrow \infty$ the two thresholds in the proposition coincide, and so PDP increases consumer surplus and output if and only if $k/n > e^{-1}$.

Denote by $\widehat{\delta}_k$ the critical discount factor above which full collusion satisfying parts (i) and (ii) in the previous paragraph can be sustained when k firms are shown. A natural question is how $\widehat{\delta}_k$ varies with k . There are two effects. First, as k decreases the value of being in the cartel also decreases because the monopoly profit is lower. Second, as k decreases the value of deviating from full collusion by undercutting $p^m(k)$ increases. Roughly speaking, this is because the deviating firm is shown alongside fewer (that is, $k - 1$) other firms and so gets higher demand. Both of these effects decrease the stability of a cartel.

Proposition 2. *Under PDP, showing fewer products to consumers makes it harder to fully collude, in the sense that the critical discount factor $\widehat{\delta}_k$ increases as k become smaller:*

$$\widehat{\delta}_1 \equiv 1 - 1/n > \widehat{\delta}_2 > \dots > \widehat{\delta}_{n-1}.$$

This proposition says that—presuming some PDP is in effect—showing fewer firms makes full collusion less stable. Numerical simulations also suggest that $\widehat{\delta}_1 > \widehat{\delta}_n$, which means that PDP also makes collusion harder compared to the case where all n firms are displayed. Consequently for moderate δ our results support the idea that PDP can destabilize full collusion, thereby forcing the cartel to lower its prices to the potential benefit of consumers.¹³

On the other hand, if δ is sufficiently large then full collusion is sustainable whether PDP is used or not. For example, Proposition 2 implies that under PDP full collusion is sustainable whenever $\delta > 1 - 1/n$. This may not be a very stringent condition because a high degree of patience is similar to being able to rapidly observe and adjust prices—which in principle algorithms might excel at doing.

We therefore now look at the effect of PDP when full collusion remains sustainable.

Proposition 3. *Suppose δ is large enough that full collusion is sustainable (with or without PDP). Fully collusive prices are lower when fewer firms are displayed to consumers:*

$$p^m(1) < p^m(2) < \dots < p^m(n).$$

However, consumer surplus and total output are also lower: as fewer firms are displayed, the decline in prices is too small to offset the loss of variety. Revenue is lower as well.

Fully collusive prices fall as PDP is implemented more aggressively, that is, as fewer firms are shown to consumers. Intuitively, recall that fully collusive prices are the same as those optimally chosen by a multi-product monopolist. And a monopolist with fewer products optimally sets lower prices because it is less concerned about cannibalizing existing sales.

Consumers are harmed by any implementation of PDP, according to Proposition 3, because under full collusion the decline in prices is too small to offset the decrease in variety. Output

¹³For example, if $k = 1$, it is straightforward to prove that when $\delta < \widehat{\delta}_1$ the firm that is displayed to consumers charges marginal cost in any (pure-strategy) subgame-perfect Nash equilibrium.

falls for the same reason, and since prices are also lower so is revenue. Therefore, depending on what fees it charges the platform may also be harmed by PDP. This differs from what happens in competitive markets, where a modest loss of variety intensifies price competition so much that consumer surplus, output and revenue may all increase (Proposition 1).

In light of Proposition 3 we now turn to another platform intervention which is capable of destabilizing collusion and benefiting consumers even for very high discount factors.

1.2. Dynamic PDP: A Stronger Tool for Breaking Collusion. The second platform intervention that we consider is *Dynamic Price-Directed Prominence* (DPDP or Dynamic PDP). The idea is to reward firms that set low prices not only with additional demand today but also with an enhanced opportunity to gain future demand.

Definition 2 (Dynamic PDP). *In period $t = 0$ one firm with the lowest price is the only firm shown to consumers, its “display clock” is set to 1 at the end of the period, and it is given an “advantage” in period 1.*

In any period $t > 0$ in which firm i has the advantage, firm i is the only firm shown to consumers, it has 1 added to its display clock at the end of the period, and it receives the advantage in period $t + 1$, so long as:

- (1) *Firm i has not raised its price compared to the previous period, and*
- (2) *Firm i is not undercut in period t by strictly more than a fixed value $ADV > 0$, and*
- (3) *Firm i ’s display clock at the start of period t is weakly less than $\tau \geq 1$, where τ is a fixed value (possibly equal to infinity).*

If any of these three conditions is violated, then in period t a firm with the lowest price is given all demand in that period, its display clock is (re)set to 1 at the end of the period, and it receives the advantage in period $t + 1$.

Under Dynamic PDP the firm that is shown today gets a “pricing advantage”, making it easier for that firm to be shown to consumers tomorrow, so long as it does not raise its price and has not held the advantage for more than τ periods (as measured by its “display clock”).

In competitive markets Dynamic PDP works the same as regular PDP (for $k = 1$). In particular, at least one firm charges marginal cost in each period, and so for example Dynamic PDP benefits consumers for low n but otherwise makes them worse off (Proposition 1).

However in cartelized markets Dynamic PDP is much more effective than regular PDP in reducing prices. To interpret the next result, recall that $\widehat{\delta}_1$ is the critical discount factor for full collusion under PDP when $k = 1$.

Proposition 4. *Consider Dynamic PDP with an advantage $0 < ADV \leq p^m(1)$.*

- (1) *There exists a $\widehat{\delta} \geq \widehat{\delta}_1$ such that if $\delta < \widehat{\delta}$, then in any pure-strategy subgame-perfect Nash equilibrium the equilibrium transaction price equals marginal cost in all periods.*
- (2) *Moreover, for any δ , there exist ADV and τ sufficiently large such that the equilibrium transaction price is marginal cost in all periods.*

Proposition 4 says that for $0 < ADV \leq p^m(1)$ collusion is harder to sustain when the platform implements dynamic rather than regular PDP. Indeed, when ADV and τ are sufficiently large, firms are unable to collude at any price above marginal cost even as they become arbitrarily patient, that is, as $\delta \rightarrow 1$.

We now sketch an intuition for why collusion can be made infeasible, focusing for simplicity on the case where $\tau = \infty$. Let $\tilde{\pi}(p)$ denote per-period profit of a firm that charges p and is the only firm shown to consumers. Suppose that firms try to fully collude. At time zero a firm could deviate by charging $p^m(1) - \psi$ for $\psi > 0$ small, be the only firm shown to consumers, and win the advantage. This firm could then charge $\min\{p^m(1) - \psi, ADV\}$ in each future period and keep the advantage forever, because no firm could undercut it by strictly more than ADV . Setting ADV high enough such that $\tilde{\pi}(ADV) \geq \tilde{\pi}(p^m(1))/n$, and for $\psi > 0$ small, the firm's payoff from this deviation would strictly exceed $\tilde{\pi}(p^m(1))/n(1 - \delta)$.¹⁴ However any of the n firms could so deviate. Therefore a necessary condition for full collusion to be an equilibrium is that the joint discounted profit from collusion strictly exceeds $\tilde{\pi}(p^m(1))/(1 - \delta)$ —which is impossible. A similar argument can be used to rule out any collusive scheme which tries to sustain prices above marginal cost.

Finally, Dynamic PDP may again introduce a trade-off between lower prices but less variety.

Proposition 5. *Suppose δ is sufficiently high that, absent platform intervention, firms would fully collude. Suppose also that ADV and τ are sufficiently large that the transaction price equals marginal cost. Compared to the case where all n firms are shown to consumers, Dynamic PDP increases consumer surplus and total output if and only if*

$$n < \tilde{n} = \exp\left(1 + \exp\left(\frac{a - c - a_0}{\mu}\right)\right).$$

Dynamic PDP can benefit consumers and increase output in a wide variety of circumstances when firms are patient and markets are cartelized. (Note that $n < \tilde{n}$ holds for many parameterizations—including those used in our later experiments—even for very large n .) Although it is again hard to provide general conditions under which revenue increases, this happens for example when $n = 2$, $a - c < a_0$, and μ is small. This contrasts with Proposition 3 which showed that regular PDP performs badly in cartelized markets.

¹⁴In particular, the firm's profit strictly exceeds $\pi^m(1)/n$ at time zero and weakly exceeds it in all later periods.

2. A MULTI-AGENT REINFORCEMENT LEARNING APPROACH

In the remainder of this article we investigate how reinforcement-learning algorithms respond to PDP and Dynamic PDP. Reinforcement learning represents a class of techniques from computer science whereby algorithms learn about their environment based on their own past experiences with it. When multiple reinforcement-learning agents interact this is referred to as Multi-Agent Reinforcement Learning (MARL) (see Buşoniu, Babuška, and De Schutter (2010) and Bloembergen, Tuyls, Hennes, and Kaisers (2015) for surveys).

We use a common technique called (tabular) Q-learning which itself is part of a subclass of reinforcement learning called temporal-difference learning (Watkins, 1989; Sutton and Barto, 2018). Q-learning is motivated by the theory of dynamic programming applied to Markov Decision Environments. A powerful feature of Q-learning is that it requires little knowledge of the underlying environment. In particular, it assumes that the algorithm can recognize which state an underlying system is in and knows which actions are available in each state, but does not assume any prior knowledge about the payoff functions or the state transition probabilities (or even of the prior distribution of such probabilities). Thus, Q-learning is a robust technique that has been applied to diverse applications.

Before turning to the details of our MARL approach, we first review the basics of Q-learning in stationary single-player environments. We follow the standard modern reference on this topic (Sutton and Barto, 2018) for Q-learning in which the state space is fairly small. Cutting-edge techniques for larger state spaces involve approximating the state space. Much recent progress on “deep reinforcement learning” has been made. See Mnih et al. (2015) and Silver et al. (2016), which develop and apply both novel and existing techniques such as “experience replay” (introduced by Lin (1992) but see also the discussion in de Bruin, Kober, Tuyls, and Babuška (2015)) to confront the various convergence issues that such state-space approximation induces. Li (2017) provides a recent overview of the topic. Deep reinforcement learning can be powerful but it involves many customization decisions by the designer of the algorithm; our approach grants less discretion to the researcher.

2.1. Q-learning with a Single Agent. To understand how Q-learning works, consider a single algorithm facing an unknown stationary Markov Decision Environment with a finite set of states indexed by $s \in \mathcal{S}$, a finite set of actions indexed by $x \in \mathcal{X}$, and with transition probabilities between states that depend on the current action and state. Given action x in state s the agent receives a payoff $\pi(s, x)$ that could be random.

Let $x^*(s)$ represent an optimal policy. That is, denoting by s_t the state at time t and the initial state by s_0 , $x^*(s)$ maximizes the future expected discounted profits

$$\mathbb{E} \sum_{t=0}^{\infty} \delta^t \pi(s_t, x^*(s_t)).$$

Q-learning is motivated by the theory of dynamic programming. Let $V(s)$ denote the value of being in state s . Rather than working with $V(s)$ directly, Q-learning involves iteratively estimating the “action-value function” $Q^*(s, x)$ where $Q^*(s, x)$ gives the expected discounted payoffs of taking action x at state s today and then using the optimal policy function $x^*(s)$ in all future periods. Thus

$$Q^*(s, x) = \mathbb{E}\pi(s, x) + \delta \mathbb{E}V(s'|s, x).$$

If $Q^*(s, x)$ were known then the optimal policy $x^*(s)$ would also be known,

$$x^*(s) = \arg \max_{x \in \mathcal{X}} Q^*(s, x).$$

Because the state and action spaces are finite, $Q^*(s, x)$ is simply a matrix and so $x^*(s)$ is determined by looking at the row (say) corresponding to state s and then choosing the column with the largest element in that row, which corresponds to the optimal action $x^*(s)$.

Because $Q^*(s, x)$ is not known it must be estimated as follows. Beginning from a given matrix Q , at time t in state s the algorithm decides which action to take. With probability $1 - \epsilon_t$ it chooses the action that is optimal according to the current Q-matrix. However, with probability ϵ_t it experiments by uniformly randomizing over all available actions. Such experimentation ensures that Q-learning sufficiently explores all states and actions.

After choosing the action x , the realized payoff $\pi(s, x)$ is observed, as is the new state s' . The one element of the Q-matrix corresponding to (s, x) is then updated as follows.

$$Q(s, x) \leftarrow (1 - \alpha)Q(s, x) + \alpha \left[\pi(s, x) + \delta \max_{\tilde{x} \in X} Q(s', \tilde{x}) \right],$$

for some $\alpha \in (0, 1)$. This notation is from the computer science literature: the quantity to the left of the arrow refers to the new update of $Q(s, x)$ and all quantities to the right of the arrow refer to the previous “un-updated” Q-matrix.

Updating is characterized by the learning rate α by which old information is replaced with new information and by the probability of experimentation ϵ_t . We parameterize ϵ_t as follows,

$$\epsilon_t = e^{-\beta t}$$

for some experimentation parameter $\beta > 0$. A higher β means that experimentation tapers off more quickly. An algorithm’s learning is therefore characterized by the couple (α, β) .

In stationary single-player environments such as the one considered above, Q-learning is guaranteed to converge to the true action-value function $Q^*(s, x)$ under fairly weak conditions and hence to uncover the true optimal policy $x^*(s)$ (see Watkins and Dayan (1992), Jaakkola, Jordan, and Singh (1994), and Tsitsiklis (1994)). However, we will consider interactions among multiple algorithms. As is well known, in MARL settings there is no theoretical guarantee of convergence. The reason is simply that each agent is changing the strategy that it uses over time as it updates its own Q-matrix, and so from the standpoint of other agents the environment is no longer stationary. Nevertheless, as in some other studies (Waltman and Kaymak, 2008; Calvano, Calzolari, Denicolò, and Pastorello, 2020), we nearly always obtain convergence (defined precisely below).

2.2. Our MARL Approach. Our specification substantially follows that in Calvano, Calzolari, Denicolò, and Pastorello (2020) (although those authors do not examine PDP). We simulate interactions among multiple agents (algorithms), with our baseline specification having $n = 2$ and $\delta = 0.95$. Each agent has marginal cost $c = 1$. Demand is given by equation (1) with each firm having the same quality component $a = 2$ and the outside good having $a_0 = 0$. We consider two values for product differentiation, $\mu = 1/4$ and $\mu = 1/20$.

To implement MARL we make the following choices. In each period the action space is the set of prices that agents can set. We discretize this set of prices to contain fifteen elements in the set $[1, 2.1]$. The lower bound of this set is marginal cost and the upper bound is slightly above the fully collusive prices in the absence of PDP (and also above the corresponding prices with PDP). The state space is the set of possible prices charged by agents in the previous period, with 15^n elements. Thus, each agent conditions its prices at time t on the prices set by all agents at time $t - 1$.

We generalize the theoretical model of PDP from Section 1 by introducing a parameter γ . This parameter measures the extent to which the platform implements PDP, and we vary it in increments of 0.01 over the region $[0, 1]$. Specifically, in each period a representative proportion $1 - \gamma$ of consumers is shown all n goods whereas a proportion γ is shown only one good ($k = 1$). To handle ties we slightly smooth the process that determines which firm is shown to the mass γ of consumers. We introduce a smoothing parameter $\sigma > 0$ such that, given prices $\{p_i\}$ in a period, the probability that i is shown to the γ consumers is

$$\frac{\exp\left(\frac{-p_i}{\sigma}\right)}{\sum_{j=1}^n \exp\left(\frac{-p_j}{\sigma}\right)}, \quad (3)$$

where $\sigma = 0.01$. For Dynamic PDP there are additional details which we defer to Section 4.

Each algorithm maintains its own Q-matrix and updates it over time in the manner described in Section 2.1. Implementing Q-learning requires an initial “time zero” Q-matrix, which we build as follows. Fixing an agent and state, for each action available to that agent we derive

the within-period payoff that would be expected if all other agents uniformly randomized their actions. We then divide this value by $1 - \delta$ so that the Q-matrix indeed contains an initial estimate of the total future payoffs of taking different actions today.

Unless stated otherwise, our default specification is $\alpha = 0.15$ and $\beta = 10^{-5}$. We will vary the learning parameters (α, β) across a range to assess the robustness of our results.

We run these algorithms until the induced strategy of each agent does not change for 100,000 periods.¹⁵ In other words, for each agent in each period we take that agent’s Q-matrix and determine, for each possible state, which action is associated with the highest payoff in terms of the Q-matrix. This procedure induces a policy function for each agent in each period. If, for any 100,000 period horizon, this policy function is stable for each agent then we say that the algorithms have converged. We then compute payoffs and other relevant metrics by averaging over this 100,000 period horizon.

For each set of parameters we consider, we repeat this procedure 1000 times, in each instance restarting the algorithms from their initial time-zero Q-matrices, resetting experimentation levels to those at time zero, and running them until they again converge. Finally, we average across these 1000 iterations for all values that we report.

3. PRICE-DIRECTED PROMINENCE: EXPERIMENTAL RESULTS

Here we present the results of our MARL experiments on the effects of PDP when $n = 2$ and $\delta = 0.95$ (Sections 4, 6 and 8 consider Dynamic PDP, lower values of δ , and $n = 3$, respectively). We first consider higher product differentiation ($\mu = 1/4$), then lower product differentiation ($\mu = 1/20$), using our default learning parameters $\alpha = 0.15$, and $\beta = 10^{-5}$. Then we explore the robustness of our results to changes in these learning parameters.

3.1. Higher Product Differentiation. To set a baseline for $\mu = 1/4$, Table 1 reports results for when price-directed prominence is not in effect ($\gamma = 0$). The Bertrand-Nash price is what theory predicts both firms would charge in the absence of collusion, and is given by 1.473. The collusive price maximizes the joint profits of the firms, with both firms charging the same price 1.925 (Lemma 2).¹⁶ Table 1 also reports the (share-weighted) AI prices that the algorithms actually charge, given by 1.682. Thus, our algorithms typically converge to a price exceeding the Bertrand-Nash price, in line with what we would expect from Calvano, Calzolari, Denicolò, and Pastorello (2020).

¹⁵We stop the algorithms if they do not converge after 1 billion periods, as in Calvano, Calzolari, Denicolò, and Pastorello (2020).

¹⁶We compute Bertrand-Nash and collusive outcomes for continuous prices. The optimal collusive prices when $\gamma \in (0, 1)$ may be asymmetric, which our later computations accommodate. For this reason, and also because some firms may have very small sales when PDP is in effect, we always report share-weighted prices.

Bertrand-Nash Price	Collusive Price	AI Price
1.473	1.925	1.682

TABLE 1. Benchmark assessment of AI pricing relative to Bertrand-Nash and collusive pricing, for $\mu = 1/4$, when there is no price-directed prominence.

Figure 1 lays out the effects of price-directed prominence. The left panel shows how both the algorithmic “PDP price” and the collusive price vary with the mass γ shown only one product, while the right panel shows the corresponding consumer surplus changes. The PDP price slightly increases for low values of γ but thereafter mostly decreases. Over the entire range of γ , prices decline by 7% under AI pricing. This is quite similar to the decline in the fully collusive price over this range, which is 6.4%. The fact that prices decline is consistent with our cartel analysis in Proposition 3, which seems relevant given that $\delta = 0.95$.

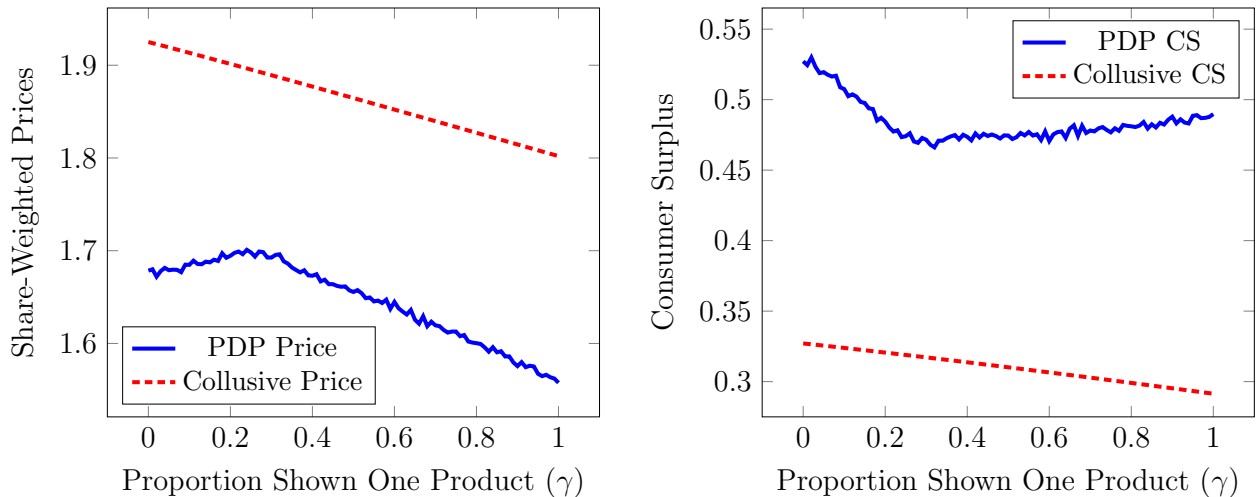


FIGURE 1. The effect of price-directed prominence on prices (left panel) and consumer surplus (right panel), with differentiation $\mu = 1/4$.

However, the lower prices come at a cost to consumers in the form of lower variety, where the loss of variety becomes larger as γ increases. Taking both price decreases and variety loss into account, it is clear that PDP harms consumers. Indeed, any positive value of γ (other than $\gamma = 0.02$) leads to lower consumer surplus than $\gamma = 0$, and moving from $\gamma = 0$ to $\gamma = 1$ lowers consumer surplus by about 7% under AI pricing. This decline in consumer surplus is also in line with our theoretical predictions from Proposition 3.

Although the algorithms do not achieve the fully collusive prices even in the absence of PDP, nonetheless the effect of PDP is consistent with what theory predicts for a fully collusive cartel: prices fall but not enough to benefit consumers (Proposition 3).

3.2. Lower Product Differentiation. We now report the same results as above except with $\mu = 1/20$, corresponding to the case with lower product differentiation. Table 2 summarizes for the situation with no PDP ($\gamma = 0$). Bertrand-Nash competition exhibits prices of 1.100. Full collusion generates prices of about 1.893, and AI prices are 1.737.

Bertrand-Nash Price	Collusive Price	AI Price
1.100	1.893	1.737

TABLE 2. Benchmark assessment of AI pricing relative to Bertrand-Nash and collusive pricing, for $\mu = 1/20$, when there is no price-directed prominence.

Figure 2 presents the effects of PDP on prices (left panel) and consumer surplus (right panel). AI prices initially increase with γ but then at $\gamma = 0.33$ tend to decrease for the remaining range of γ . Compared to AI prices at $\gamma = 0$, prices are lower at $\gamma = 1$. Specifically, there is an 8% decline in prices across this range. This PDP price decrease is substantial compared to the 2% decline in the fully collusive price over this range.

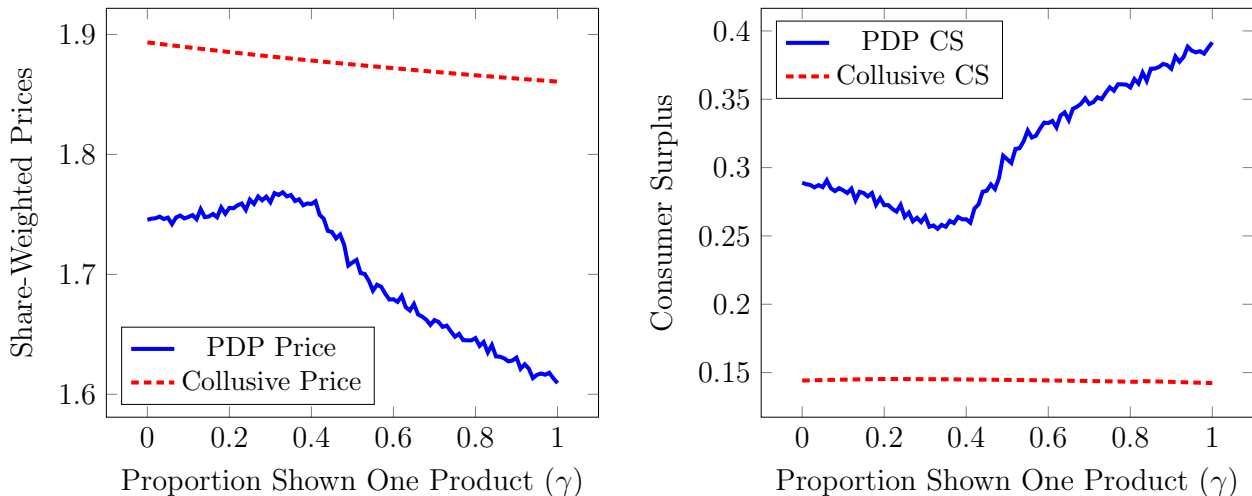


FIGURE 2. The effect of price-directed prominence on prices (left panel) and consumer surplus (right panel), with differentiation $\mu = 1/20$.

The effect of PDP on consumer surplus is large and positive. Moving from $\gamma = 0$ to $\gamma = 1$ increases consumer surplus by 35%. Hence, the AI price decrease more than compensates the variety loss, benefiting consumers. Even at moderate values of γ , such as $\gamma = 0.6$, there is a 15% increase in consumer surplus.

As when $\mu = 1/4$, the algorithms do not achieve the fully collusive prices even in the absence of PDP. In contrast, here the effect of PDP is inconsistent with what theory predicts for a fully collusive cartel: prices fall enough that consumers benefit.

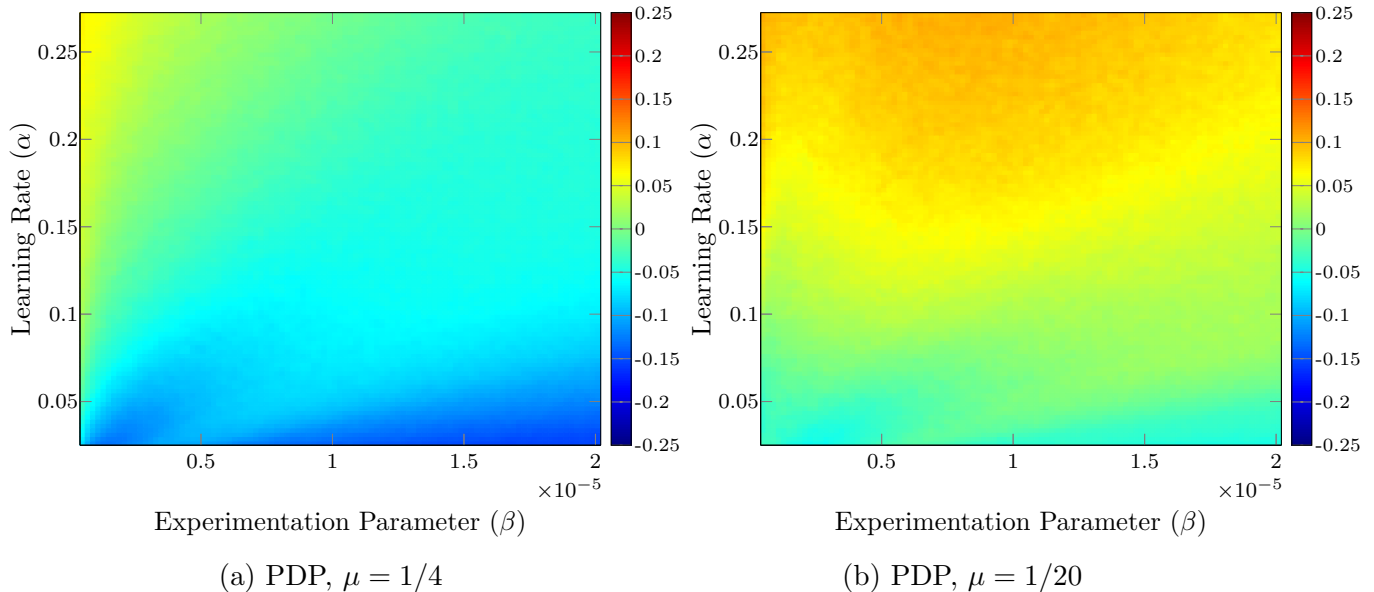


FIGURE 3. Heatmaps of the effects of PDP on consumer surplus. Consumer surplus increases in 13.43% of the cases in (a) and 82.52% of the cases in (b).

3.3. Learning-Parameter Robustness. The results reported above suggest that PDP may either increase or decrease consumer surplus, depending on product differentiation. To explore the robustness of this conclusion under our MARL specification, here we vary the learning parameters α and β .

As explained in Section 2.1, the parameter $\alpha \in (0, 1)$ measures the extent to which new information is incorporated, whereas $\beta > 0$ measures how quickly experimentation tapers off over time. In applications α is often set to values such as 0.1 and 0.15 and indeed, even in single-agent settings, theoretically guaranteeing convergence requires that α eventually becomes small (although convergence often occurs despite a lack of theoretical guarantee even if α is fixed as in our analysis). In light of this, and also in line with Calvano, Calzolari, Denicolò, and Pastorello (2020), we consider α in $[0.025, 0.2725]$.

In selecting a range for β , it is important to allow the algorithms a sufficient opportunity to explore the state space. We restrict attention to β in the range $[4 \times 10^{-7}, 2.02 \times 10^{-5}]$, which is similar to that chosen by Calvano, Calzolari, Denicolò, and Pastorello (2020). We note that our default specification of $(\alpha, \beta) = (0.15, 10^{-5})$ is the midpoint of this region.

The overall grid encompasses $(\alpha, \beta) \in [0.025, 0.2725] \times [4 \times 10^{-7}, 2.02 \times 10^{-5}]$. To implement, we discretize this grid into 10,000 separate (α, β) pairs and run our algorithms 1000 times for each pair. To focus on whether consumer surplus increases following the implementation of PDP, for each pair (α, β) we compare consumer surplus with $\gamma = 0$ to consumer surplus with $\gamma = 0.7$. We choose $\gamma = 0.7$ as the point of comparison because our initial exploratory

simulations as reported above for the case of $\mu = 1/20$ suggest that this is a value where the algorithms are responding strongly to PDP in a way that benefits consumers. Setting $\gamma = 0.7$ typically leads to conservative predictions compared to larger γ .

We present the resulting assessment in two heatmaps contained in Figure 3, with $\mu = 1/4$ in the left panel and $\mu = 1/20$ in the right panel. More red colors indicate a more positive effect of PDP on consumer surplus and more blue colors indicate a more negative effect.

From these heatmaps two facts are apparent. First, changes in the learning parameters can have significant effects on the outcome even for fixed μ . Second, over a broad range of learning parameters, PDP appears more effective at raising consumer surplus when differentiation is low. Precisely, for $\mu = 1/20$ consumer surplus increases in 82.52% of the considered learning-parameter pairs, versus 13.43% of the pairs when $\mu = 1/4$. Similarly, when PDP has a positive effect, the magnitude of the effect appears larger when μ is smaller.

4. DYNAMIC PDP: EXPERIMENTAL RESULTS

Although our experiments with price-directed prominence show some success in lowering prices and benefiting consumers, algorithmic prices remain high and consumers are often harmed when $\mu = 1/4$. Therefore, we now implement our policy of Dynamic PDP experimentally, recalling that Propositions 4 and 5 predict that this policy can lower prices and improve consumer welfare especially when firms are very patient and the market is cartelized. We let a firm with the pricing advantage keep it in all future periods ($\tau = \infty$) so long as it is not undercut by more than ADV and does not raise its price.

As we did for PDP, we suppose that Dynamic PDP involves showing a $1 - \gamma$ proportion of consumers all n products and the remaining γ proportion a single product. In the initial period, or in any period in which the firm with the advantage has raised its price, equation (3) determines which firm is shown to consumers and receives the advantage. If the firm with the advantage has not raised its price then again equation (3) applies, except that the firm with the advantage has ADV subtracted from its price. For now we set $ADV = 0.3$, but in Section 4.2 we assess the performance of a range of ADV values.

Figure 4 displays the effect of Dynamic PDP when $\mu = 1/4$. The left panel shows the effect on prices and the right panel shows the effect on consumer surplus. Each panel also displays the effect of (regular) PDP (these are the same numbers as presented earlier in Figure 1).

Figure 4 shows that Dynamic PDP has a large effect on prices and consumer surplus. The consumer surplus effects are strongest at $\gamma = 0.96$. Comparing that level to $\gamma = 0$, prices are about 18% lower and consumer surplus is 25% higher. The decline in prices is approximately three times the decline under regular PDP. The consequent increase in consumer surplus is particularly notable given that regular PDP lowers consumer surplus for these parameters.

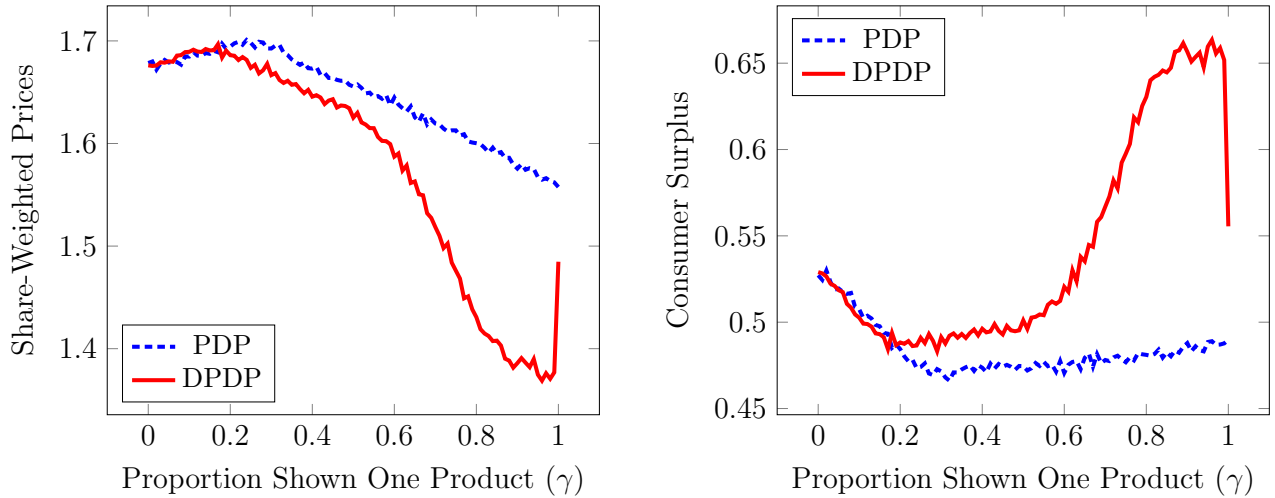


FIGURE 4. The effect of Dynamic PDP on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/4$.

However, there is a sharp decrease in consumer surplus and increase in prices at $\gamma = 1$. It appears that this value presents a learning challenge for the algorithms. To see why this may be, observe that when $\gamma = 1$ one firm will have zero demand. The additional presence of a pricing advantage means that it may be that this firm's demand and profits are completely insensitive to prices across a significant range. In contrast, when $\gamma < 1$ each firm has positive demand and each firm's profit is always affected by marginal price changes.

Figure 5 presents results for the case of $\mu = 1/20$ (comparison prices and consumer surplus for regular PDP are the same as in Figure 2). The strongest effects occur at $\gamma = 0.99$, leading to about a 15% price decrease and a 75% increase in consumer surplus compared to not using PDP at all. We again see that for $\gamma = 1$ there is a sharp decrease in consumer surplus and increase in prices, which we attribute to the learning challenge mentioned above.

We also performed an analysis of how the learning parameters affect outcomes. Extending our work from Section 3.3, we assessed consumer surplus both without any price-directed prominence and also with DPDP, for $\gamma = 0.7$. Figure 6 presents the heatmap of the change in consumer surplus from adopting DPDP across a range of hyperparameters.

Overall, Dynamic PDP appears to perform well in terms of lowering prices and benefiting consumers, although consumers do not always benefit from it. When $\mu = 1/4$, consumer surplus increases in 67.75% of the learning-parameter pairs, versus 80.43% of the pairs when $\mu = 1/20$. For $\mu = 1/4$ this is a significant difference compared to the 13.43% of cases where consumer surplus increased with regular PDP. For $\mu = 1/20$ the percentage of cases where consumer surplus increases is about the same as for regular PDP. However, the level of consumer surplus increases is much higher, so that DPDP still has a larger effect on consumer surplus compared to PDP when $\mu = 1/20$.

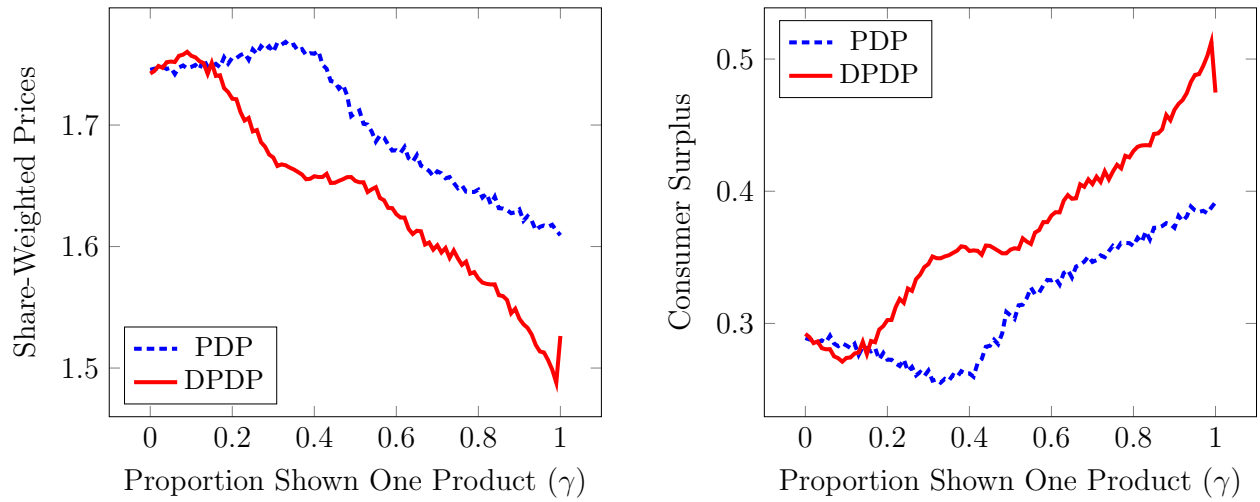


FIGURE 5. The effect of Dynamic PDP on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/20$.

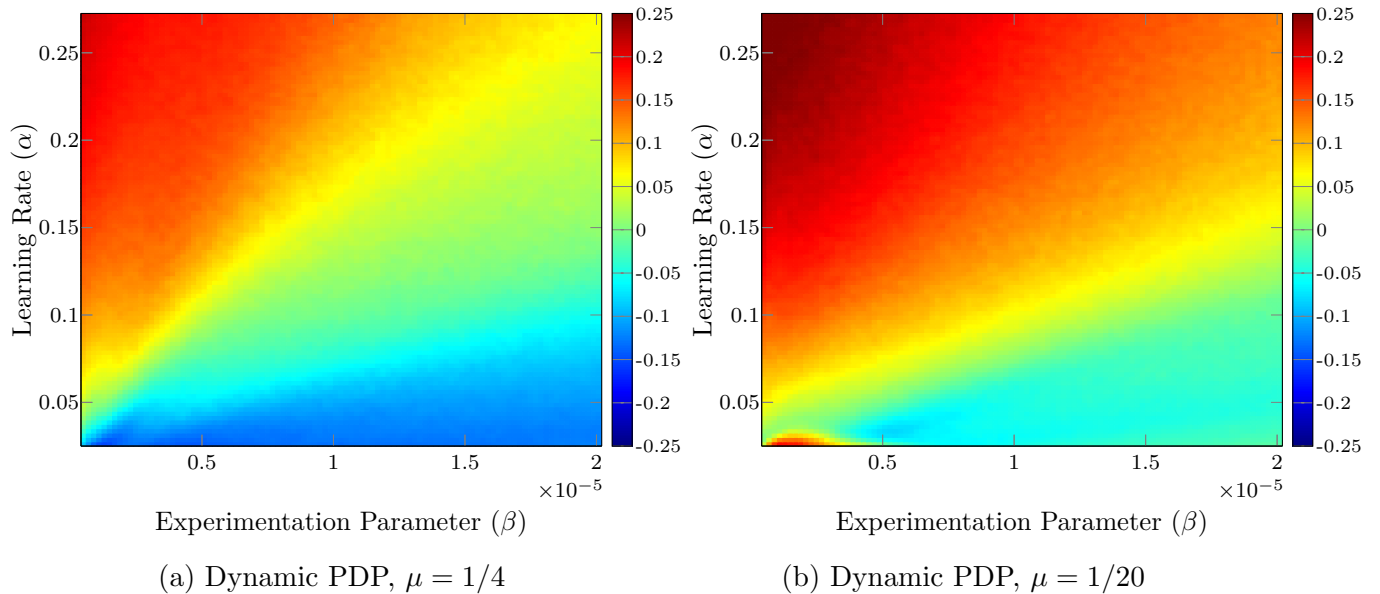


FIGURE 6. Heatmaps of the effects of Dynamic PDP on consumer surplus, for $\gamma = 0.7$ and $ADV = 0.3$. Consumer surplus increases in 67.75% of the cases in (a) and 80.43% of the cases in (b).

4.1. Smarter AI: Lower Prices and Higher Consumer Surplus. We note that a firm's optimal price in period t may depend on whether it has the advantage or not, but that information is not included in the state space observed by the algorithms (only the previous

period’s prices are).¹⁷ It is plausible that real-world algorithms would be designed to incorporate such additional information. To investigate, we allow our algorithms to track the identity of the firm with the pricing advantage, in addition to prices from the previous period.

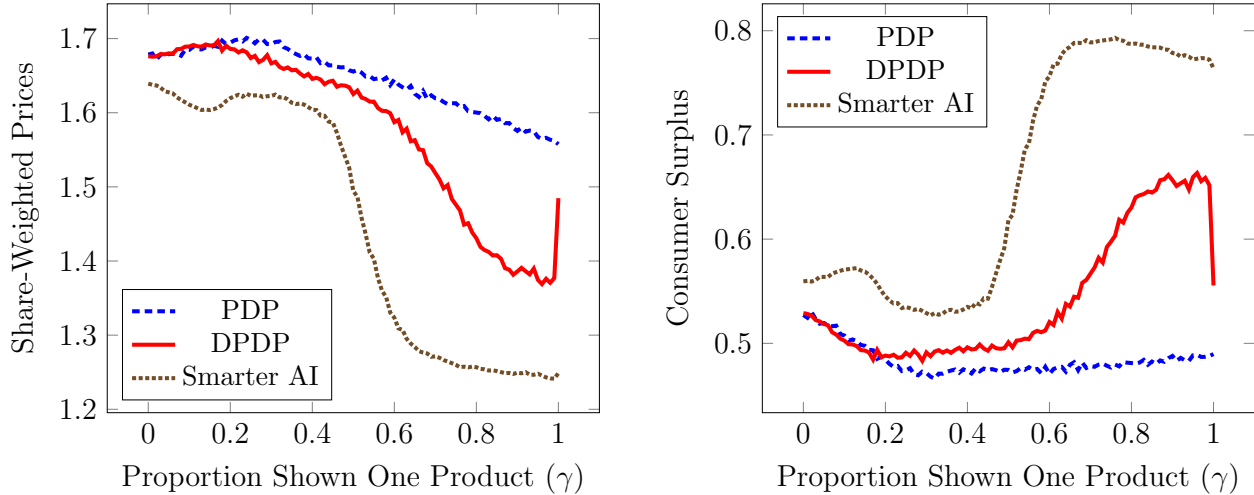


FIGURE 7. The effect of a smarter AI that includes in its state space the identity of the firm with the pricing advantage, on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/4$.

Allowing the AI to be “smarter” in this sense has a substantial effect on prices and consumer surplus. The case where $\mu = 1/4$ is shown in Figure 7 (similar results emerge for $\mu = 1/20$). Recall that in this case (for our baseline learning parameters), regular PDP harms consumers. With smarter AI, consumer surplus is highest at $\gamma = 0.76$, representing a 50% increase in consumer surplus compared to no PDP, and a 20% increase in consumer surplus compared to the best performance of Dynamic PDP without the extended algorithm (at $\gamma = 0.96$).¹⁸ We also note that the smarter AI does not exhibit the sharp decrease in consumer surplus near $\gamma = 1$ that we observe for DPDP.

4.2. Different Levels of the “Pricing Advantage”. Here we explore how the level of the pricing advantage ADV affects outcomes. We consider values of ADV in the interval $[0, 1]$, in increments of 0.01. Given that we have restricted our algorithms’ prices to lie in $[1, 2.1]$ and that $c = 1$, even moderate values in this range are large enough that our theory predicts large gains for consumers. Indeed, at $ADV = 0.3$, theory predicts marginal cost pricing for $\mu = 1/4$; for $\mu = 1/20$ a value of slightly more than $ADV = 0.4$ is sufficient.

¹⁷Thus, as firms update their Q-matrices at time t , they incorporate their realized profits from period $t-1$, but—even fixing prices—these payoffs may vary substantially based on which firm in fact had the pricing advantage.

¹⁸Note that when $\gamma = 0$ the smarter AI delivers different prices and consumer surplus than PDP or DPDP do, because adding an additional (payoff-irrelevant) element to the state space influences how the algorithms learn.

We will assess changes to ADV when $\gamma = 0.7$. Thus, in all of our results here we will compare the outcome of Dynamic PDP with $\gamma = 0.7$ to the case of no PDP at all. Note that when $ADV = 0$, Dynamic PDP corresponds to regular PDP in our experimental setting (and so there may be effects on consumer surplus even when $ADV = 0$).

Figure 8 shows how consumer surplus is affected for both values of differentiation and also for the “smarter AI” from Section 4.1. Three main outcomes are apparent. First, consumer surplus generally increases initially as ADV rises from zero. Second, consumer surplus typically begins declining as ADV grows larger. Third, the benefits then level off. In all cases prices follow a similar (but inverted) pattern, decreasing but then rising again. A fourth observation that applies only to the smarter AI is that consumer surplus is roughly a step function for higher ADV values. A closer inspection of the data reveals that these steps correspond to the underlying pricing grid that we used.

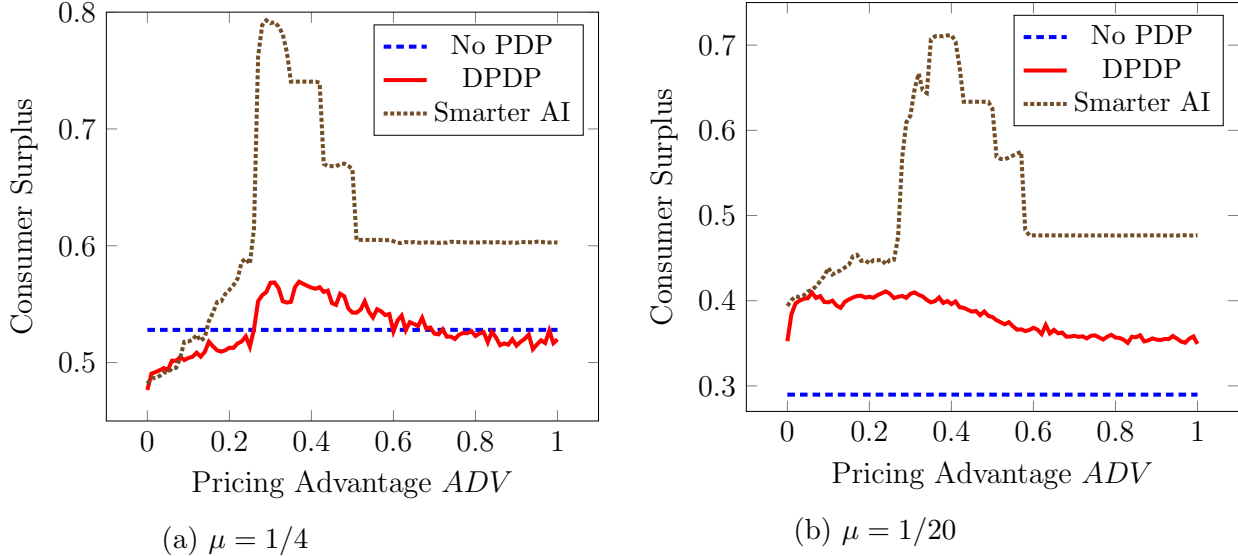


FIGURE 8. The effect of ADV on Dynamic PDP performance (consumer surplus), with $\gamma = 0.7$, compared to no PDP.

5. PRICING AND PROFIT PATTERNS

In this section we look in detail at the pricing and profit behavior displayed by the algorithms under the various platform-design policies considered above. We consider four distinct policies: (i) no platform intervention, (ii) PDP, (iii) DPDP, and (iv) Smarter AI, where $\gamma = 0.7$ for the three “active” interventions.

In looking at the converged behavior of the algorithms for each of these four possibilities, sometimes the algorithms each set a constant price over time (“1-price cycle”), but other times the algorithms converge to a price cycle of length two periods or more.

	No Intervention	PDP	DPDP	Smarter AI
Average Cycle Length	1.321	3.131	4.581	1.151
% Cycles of Length 1	72.5%	7.1%	14.1%	88.6%
% Cycles of Length 2	24.7%	44.6%	9.0%	8.6%
1-price Cycles				
% Symmetric	83.9%	100%	0%	0.2%
Industry Profit	0.594	0.548	0.247	0.242
High-Profit Firm's Share	50.8%	50.8%	83.7%	83.6%
Other Cycles				
Industry Profit	0.586	0.488	0.453	0.335
High-Profit Firm's Share	51.5%	52.7%	54.1%	84.2%

TABLE 3. Frequency of pricing cycles of various lengths and corresponding profit information. Cycles of length one (1-price cycles) are degenerate cycles in which each firm's price is constant over time. The high-profit firm's share represents the share of industry profits obtained by whichever firm earns higher profits over the long run.

Table 3 provides an overview. For each policy, the table reports the average cycle length observed, as well as the frequency of 1-price and 2-price cycles. Additionally, when there is a 1-price cycle, we report the frequency with which the firms charge symmetric (that is, identical) prices. Finally, we report the average industry profit and the share of this profit obtained by whichever of the two firms earned the highest long-run profits. We break these results out depending on whether there is a 1-cycle or a higher-order price cycle. (We do this because price and profit metrics are similar for all cycles of length two or longer.)

We highlight two results before delving into the details. First, the algorithms often reach outcomes where the two firms earn approximately equal profits. This can be seen in looking at the “high-profit firm's share” entries in the table, which indicate the share of industry profits received by the higher-profit firm over a long horizon. The only cases in which there is substantial profit asymmetry correspond to Smarter AI and the 1-price cycles under DPDP. Given that both PDP and DPDP by design push towards highly asymmetric profits within any given period, it follows that the algorithms frequently learn to “rotate demand” over time, so that sometimes one firm receives the guaranteed 70% ($\gamma = 0.7$) of consumers and sometimes the other firm does. Second, the most substantial decreases in industry profits (relative to no intervention) occur with Smarter AI and the 1-price cycles under DPDP.

We now turn to the details. Begin by considering the outcomes when there is no intervention. Of these simulations, 72.5% of outcomes exhibit a 1-price cycle (that is, constant prices) and

24.7% exhibit a 2-price cycle. 83.9% of the cycles of length one exhibit symmetric prices. Finally, as noted above, the two firms are able to split the industry profits nearly evenly between themselves, with the highest-profit firm receiving 50.8% of industry profits.

Moving from no intervention to PDP, the first notable change is a substantial decline in the frequency of 1-price cycles (7.1% rather than 72.5%). However, we now see that 100% of these 1-cycles exhibit symmetric prices. This contributes to the symmetry of profits, given that even small price asymmetries would result in highly asymmetric profits under PDP.¹⁹ The decreased frequency of 1-price cycles necessarily implies an increase in the frequency of cycles of length two or greater. Interestingly, among all such cycles the firms' rotation of demand allows them to keep splitting industry profits nearly evenly, although not quite as evenly as when there is a 1-price cycle.

Moving from PDP to DPDP, one notable change is a large increase in the average cycle length (4.581 rather than 3.131). Although the frequency of 1-price cycles doubles (from 7.1% to 14.1%), there is a large shift from 2-price cycles to higher-order cycles. Unlike outcomes discussed earlier, we now observe extreme profit differences between the 1-cycle outcomes and the outcomes with proper price cycles. In particular, when there is a 1-price cycle, profits are highly asymmetric with the high-profit firm receiving 83.7% of long-run industry profits (versus 50.8%). One firm holds the advantage in every period and prices at such a level that, given the value of ADV , it never loses the advantage. The other firm only makes profits from the 30% of the market that is not subjected to the platform's policy intervention. But when there is a 2-price or higher-order price cycle, firms change their prices over time so as to rotate who wins the guaranteed 70% of demand, with there being very little difference between the long-run profits of the two firms. Industry profits are much higher when firms rotate demand and split profits in this way (0.453 versus 0.247).

Finally, moving to Smarter AI, the most notable change is that now 88.6% of outcomes exhibit a 1-price cycle, and average cycle length decreases sharply (1.51 versus 4.581). As was true for DPDP, when there is a 1-price cycle industry profits are quite low and very asymmetric; one firm prices to keep the advantage over time. Unlike for DPDP, even when there are longer price cycles, profits remain highly asymmetric; firms are not rotating demand.

6. LESS PATIENT FIRMS

So far in our experiments we have considered firms that are fairly patient, with a discount factor of $\delta = 0.95$. But we know from a theoretical perspective that the performance of PDP depends on whether the market is cartelized or not, with PDP often increasing consumer surplus when firms behave competitively but decreasing consumer surplus when firms collude.

¹⁹In more detail, within any given period there is necessarily substantial asymmetry in profits because $\gamma = 0.7$, but over a long time horizon these asymmetries average out if firms set symmetric prices.

To explore this issue we allow δ to vary in increments of 0.01 over the interval $[0, 0.99]$, and for each value of δ record consumer surplus as γ varies in increments of 0.01 on $[0, 0.99]$. We report on the case with $\mu = 1/4$ but obtain similar qualitative results when $\mu = 1/20$.

We present the results of these experiments involving PDP in two different ways. Figure 9a shows raw consumer surplus. The main message from this figure is that, for any level of γ , consumer surplus substantially increases as δ becomes smaller. This is an intuitive result that also suggests our algorithms work in a sensible manner and are not able to maintain higher prices when δ is lower.

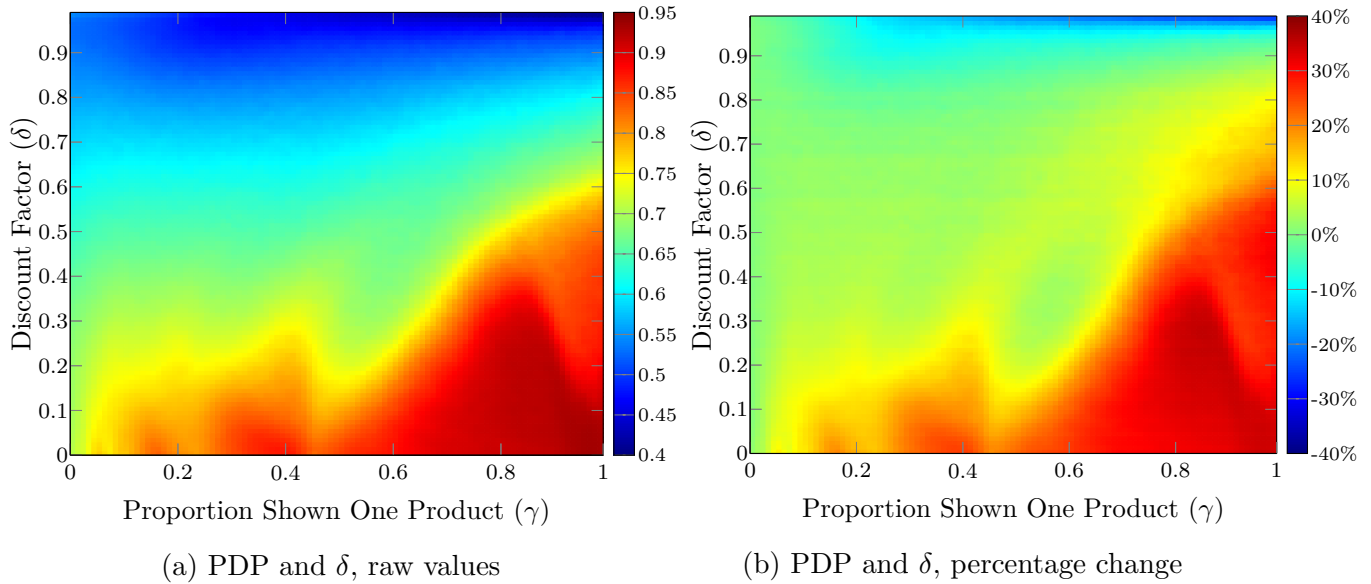


FIGURE 9. PDP heatmap of raw (9a) and percentage change in (9b) consumer surplus for different values of δ and γ , at $\mu = 1/4$. Percentage change is calculated relative to $\gamma = 0$ for each given δ value.

Figure 9b uses the same data but presents it in way that tracks how effective PDP is in percentage terms, normalizing consumer surplus to its level when $\gamma = 0$ for each δ value. More precisely, for each given (γ, δ) pair, the heatmap displays the difference in consumer surplus between (γ, δ) and $(0, \delta)$, divided by the level of consumer surplus at $(0, \delta)$. Thus, for any δ , reading from left to right shows how consumer surplus changes with PDP.

Figure 9b shows that PDP raises consumer surplus across a very broad region of δ . In fact, only for very high levels of δ does PDP lower consumer surplus for all γ values. For example, even at $\delta = 0.9$ consumer surplus goes up (slightly) for several high values of γ such as $\gamma = 0.98$, while at $\delta = 0.85$ consumer surplus is up for all $\gamma \geq 0.73$. Thus our finding in Section 3 that PDP lowers consumer surplus when $\mu = 1/4$ requires that δ be fairly high. For smaller δ , our experiments yield results closer to our predictions from competitive markets (Proposition 1).

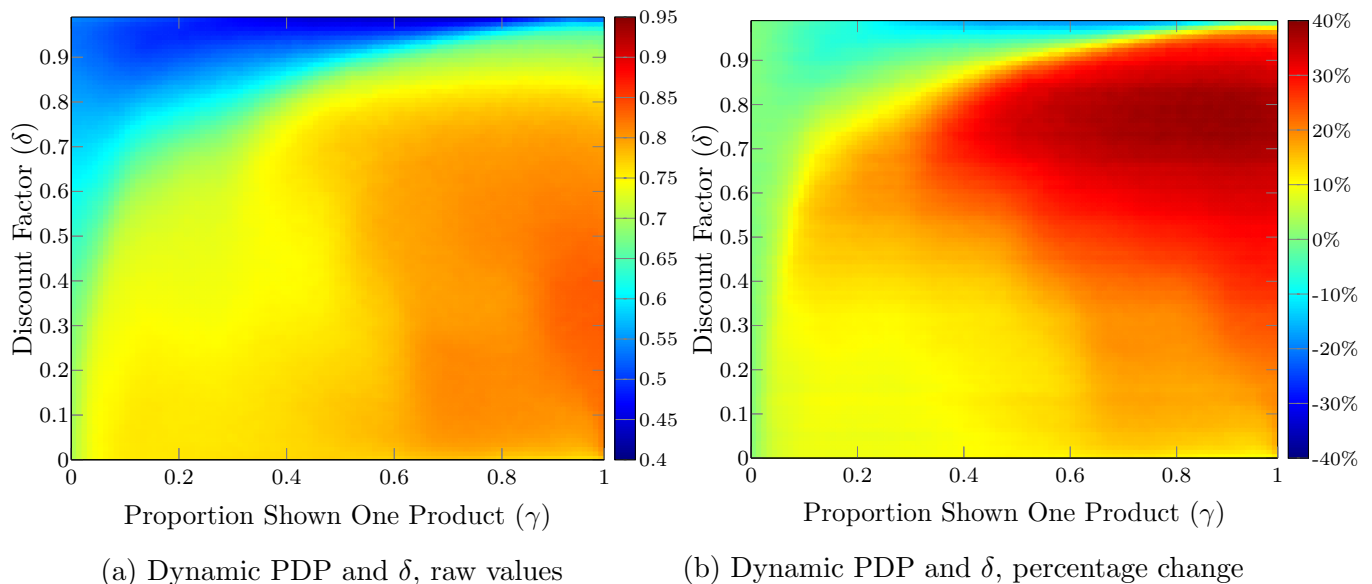


FIGURE 10. Dynamic PDP heatmap of raw (10a) and percentage change in (10b) consumer surplus for different values of δ and γ , at $\mu = 1/4$ and $ADV = 0.3$. Percentage change is calculated relative to $\gamma = 0$ for each given δ value.

We also examine the performance of Dynamic PDP as δ changes. Figure 10 presents results as was done for PDP, with the left panel showing raw consumer surplus and the right panel showing percentage changes after normalizing to the level of consumer surplus at $\gamma = 0$ for each δ . DPDP continues to work very well across a broad range of δ . In fact, for all δ values considered except for the highest ($\delta = 0.99$), there exist values of γ such that DPDP raises consumer surplus at that δ . We note that the percentage increases are not necessarily higher when δ is lower. This is because at $\gamma = 0$ for lower values of δ prices are already much lower than at higher δ values; there is less room for prices to drop. Another reason is that for many values of δ the prices at high values of γ are similar (and low, around 1.18).

7. PLATFORM PROFITS

So far we have focused on whether platform interventions can raise consumer surplus. Here we assess how such policies might influence the profits of the platform.

Many online marketplaces impose fees on sellers. The most common fees are fixed shares of revenue and per-unit charges. Some platforms utilize both types of fees. For example, on the Amazon Marketplace sellers pay a proportion of the revenues they earn. But some categories have additional per-unit fees (or minimum per-sale fees from revenue sharing, which is equivalent to a per-unit fee when binding), and some sellers adopt a pricing plan

offered by Amazon that also involves a per-unit fee. Additionally, some platforms such as Amazon offer fulfillment services to sellers, which (for Amazon) involve per-units fees.²⁰

Therefore, a platform might benefit from either increasing the total revenue or the total output sold. One way to do this is by simply increasing the revenue or units sold per shopper. Another way to do this is by increasing the market size (that is, the number of shoppers on the platform), for example by implementing policies that raise consumer surplus.

Under our chosen specifications it is difficult—for a fixed market size—for revenue to increase following one of our policy interventions. For example, from a theoretical perspective, Proposition 3 indicates that PDP always lowers revenue when the market is cartelized. Practically speaking, for our chosen specifications at $\gamma = 0$ both the Bertrand-Nash prices and the prices set by the algorithms are beneath the revenue-maximizing prices in the absence of any platform involvement. Thus, because both PDP and DPDP lower prices and decrease variety, neither can raise revenue for these parameterizations. In specifications outside those considered in this article, however, both in theory (for example, Proposition 1) and in the associated experiments, we have confirmed that revenue can go up.

Specification	Critical Growth Elasticity ($\gamma = 0.7$)
PDP ($\mu = 0.05, \delta = 0.95$)	0.297
DPDP ($\mu = 0.05, \delta = 0.95$)	0.293
DPDP ($\mu = 0.25, \delta = 0.95$)	2.002
PDP ($\mu = 0.05, \delta = 0.2$)	1.044
PDP ($\mu = 0.25, \delta = 0.2$)	0.864

Effects of Policies on Revenue

TABLE 4. Critical Growth Elasticities for revenue for different policies and μ and δ values, using $\gamma = 0.7$. If a one percentage point increase in consumer surplus increases market size by at least this amount, then overall revenues increase.

But the policies we have considered might raise total revenues even if they lower per-customer revenue, if offering consumers higher utility grows the market size. Especially when platforms sell many products, there may be an externality: individual sellers ignore how lowering their own prices might bring more overall shoppers to the platform, thereby benefiting sellers in other categories when consumers one-stop shop.

To assess this possibility, we ask what elasticity of market size with respect to consumer surplus would be required to raise overall revenues, according to our experiments. Specifically,

²⁰Amazon Marketplace fees are described at sell.amazon.com/pricing.html, which also notes that fulfillment “includes picking and packing your orders, shipping and handling, customer service, and product returns.”

we look at various settings in which consumer surplus increases but per-customer revenue decreases, for $\gamma = 0.7$, including cases with high ($\delta = 0.95$) but also low ($\delta = 0.2$) patience. We suppose the initial market size is one. For each setting, we compute the percentage increase in market size which would keep overall revenues constant, and then divide that by the computed percentage increase in consumer surplus of moving from $\gamma = 0$ to $\gamma = 0.7$. We call the final number the “Critical Growth Elasticity” for revenue and present results in Table 4. For example, under PDP with $\mu = 0.05$ and $\delta = 0.95$, the platform would break even on revenue if a one percentage point increase in consumer surplus led to a 0.297% increase in the number of customers visiting the platform.

Turning to per-unit fees, theory predicts that PDP and Dynamic PDP can increase total output (Propositions 1 and 5) and thereby increase the profits of a platform using such fees.

We present results on output in Table 5. The second column gives the percentage of γ values such that consumer surplus is strictly higher at that value compared to $\gamma = 0$. Similarly, the third column gives the percentage of γ values such that total output is strictly higher at that value compared to $\gamma = 0$. For example, when $\mu = 0.05, \delta = 0.95$, and PDP is used, consumer surplus is higher for 54% of the values of γ in (our grid on) $[0.01, 1]$, and output is higher for 22% of those values. The fourth and final column reports the critical growth elasticity as defined above except looking at output rather than revenue, again focusing on $\gamma = 0.7$ (values of 0.000 indicate that output is already higher at $\gamma = 0.7$ than at $\gamma = 0$). We see that these tend to be smaller than the required values for revenue. For example, under PDP with $\mu = 0.05$ and $\delta = 0.95$, the critical growth elasticity for output is now only 0.032; this follows because PDP with $\gamma = 0.7$ increases consumer surplus by 19.93% and because the needed market growth to break even on total output is 0.64%.

Specification	% of γ values for which:		Critical Growth Elasticity ($\gamma = 0.7$)
	CS Up	Output Up	
PDP ($\mu = 0.05, \delta = 0.95$)	54%	22%	0.032
DPDP ($\mu = 0.05, \delta = 0.95$)	83%	3%	0.045
DPDP ($\mu = 0.25, \delta = 0.95$)	38%	24%	0.268
PDP ($\mu = 0.05, \delta = 0.2$)	77%	37%	0.000
PDP ($\mu = 0.25, \delta = 0.2$)	100%	92%	0.000

Effects of Policies on Output

TABLE 5. The second and third columns give the percentage of γ values such that consumer surplus or output increases, compared to $\gamma = 0$. The final column gives the Critical Growth Elasticities for output for different policies and μ and δ values, using $\gamma = 0.7$. If a one percentage point increase in consumer surplus increases market size by at least this amount, then overall output increases.

8. THREE FIRMS

Here we provide results for the case with $n = 3$, again focusing on $\mu = 1/4$.²¹ All other parameters are as in Sections 3 and 4. As with $n = 2$, we find that DPDP often increases consumer surplus. Figure 11 shows the performance for $\mu = 1/4$ at the baseline learning parameters. In this case DPDP does best at $\gamma = 0.73$, leading to a 13% consumer surplus gain and substantial price drops. However, we once again see that at $\gamma = 1$ the algorithms appear less able to learn to charge lower prices (as discussed in Section 4). We also see that regular PDP has only a modest effect on prices and no clear consumer surplus gains.

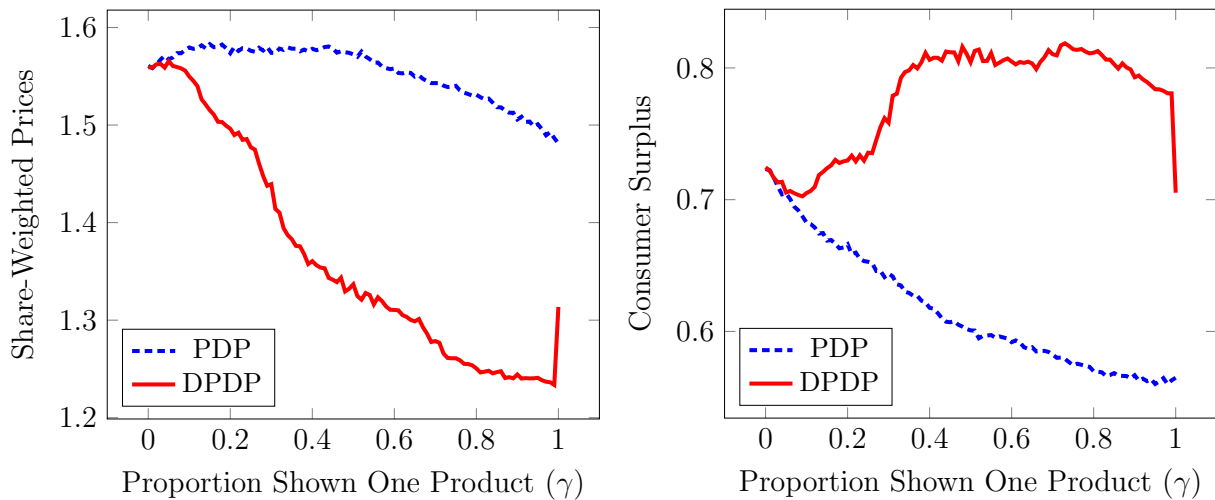


FIGURE 11. For $n = 3$, the effect of Dynamic PDP on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/4$.

Varying the learning parameters (over the same range considered earlier) tells a similar story. Although we do not display the data here, PDP lowers consumer surplus in 100% of the learning-parameter pairs considered, similar to the $n = 2$ case in which PDP only raised consumer surplus in 13.43% of the cases (Figure 3a). We also find that PDP almost always lowers consumer surplus even if $\mu = 1/20$, unlike the case with $n = 2$. Thus, at least when δ is high, our predictions from Proposition 3 hold true when $n = 3$: prices fall when PDP is imposed, but not enough to benefit consumers.²²

Although we do not display the data here, Dynamic PDP performs extremely well, increasing consumer surplus in 97.8% of learning-parameter pairs considered. Varying ADV leads to the same qualitative effects as with two firms, as discussed in Section 4.2.

²¹The main difference when $\mu = 1/20$ is that consumer surplus gains are larger for DPDP than when $\mu = 1/4$.

²²We also explored the effect of PDP across a broader range of δ values and found results consistent with the case of $n = 2$: PDP reliably raises consumer surplus for smaller values of δ and, for any fixed γ , lowering δ tends to significantly increase consumer surplus.

9. CONCLUSION

In this article we examined the intersection of several important trends: the growing use of pricing algorithms, increased concerns about the potential anticompetitive role of algorithms, and heightened interest among policymakers about the rules that govern interactions between platform users. Using both theory and experiments, we showed that platform rules that steer demand can increase competition between sellers, and simultaneously raise a platform's profits. Which rules are most effective depends on whether sellers (or their algorithms) behave competitively or collusively. When collusion is a concern, for example, rules that condition on past prices and treat sellers in a non-neutral fashion may be required.

Given our stylized modeling choices and the many practical considerations faced by a platform, we do not suggest that our exact policies are the ideal ones. For instance, we have focused solely on price as the strategic variable. But in the real world, if steering were based entirely on price, then sellers might offer inefficiently low-cost and low-quality products. And even if price is the primary consideration, there are many potential rules that we have not considered that may be more effective in promoting competition.

In deciding which rules to adopt, the platform must be cognizant of differences between how algorithms behave and what theory or simple intuition might suggest. One reason that reinforcement-learning algorithms need not obtain outcomes predicted by economic theory is that these algorithms must learn the rules of the game by playing the game. This means that they are simultaneously implicitly learning the rules while also trying to find the optimal strategy. Especially when theoretical results lean on backwards induction, which assumes that all players understand all the rules of the game beginning with the first time period, it is not surprising that algorithms do not behave entirely according to theoretical predictions. For example, in theory Dynamic PDP leads to marginal cost pricing but in our experiments it leads to very low but above-cost prices. Our experiments have also identified some learning challenges of algorithms. Most speculatively, it may be possible for a platform to exploit these and other idiosyncrasies of how algorithms learn. At the very least, a platform should account for such issues as it sets its rules.

Despite such challenges and limitations, our work is proof of concept that demand-steering policies may be procompetitive, both in competitive and cartelized markets, even when sellers use pricing algorithms. And, although we are not advocating that platforms should have a regulatory duty to promote competition, our work provides some evidence that it may be feasible for platforms to do so.

APPENDIX: OMITTED PROOFS

Proof of Lemma 1: We restrict attention to pure strategies. (It is lengthy but straightforward to prove the result when firms can use mixed strategies.) Consider a given period and relabel the firms such that $p_1 \leq p_2 \leq \dots \leq p_n$ (for simplicity we drop time superscripts). It is straightforward to verify that any prices satisfying $p_1 = p_2 = \dots = p_m = c$ for $m \geq k + 1$ constitute a Nash equilibrium. We now prove there is no Nash equilibrium in which $p_{k+1} > c$.

On the way to a contradiction, suppose that there is an equilibrium in which $p_{k+1} > c$. Note that firm $k + 1$ cannot be shown with probability one, and if it is shown with positive probability then $p_{k+1} = p_k$. If $p_k > c$ then firm $k + 1$ could lower its price to just slightly less than p_k and thereby ensure it is shown with probability one and so increase its profits. Hence $p_k = c < p_{k+1}$, but this means firm k could increase its profits by instead setting $p_k \in (c, p_{k+1})$. We have arrived at a contradiction and so conclude $p_{k+1} = c$. ■

We use the following lemma in some subsequent proofs.

Lemma 3. *Total industry output is monotonically increasing in consumer surplus, because equation (2) can be rewritten as*

$$\sum_{j \in \mathcal{N}_i} D_j(p^t) = 1 - \frac{\exp(a_0/\mu)}{\exp(U^t(p^t)/\mu)}.$$

Proof of Proposition 1: First we determine consumer surplus with and without PDP. Suppose only $k < n$ firms are shown to consumers. From Lemma 1 all firms (that are shown to consumers) charge c , and so using equation (2) consumer surplus is

$$\mu \log \left\{ k \exp\left(\frac{a-c}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right) \right\}. \quad (4)$$

Suppose instead that all n firms are shown to consumers. There is a unique and symmetric equilibrium. Denote by p_{BN}^* the equilibrium price. Using equation (2) consumer surplus is

$$\mu \log \left\{ n \exp\left(\frac{a-p_{BN}^*}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right) \right\}. \quad (5)$$

We now derive p_{BN}^* . If firm i deviates by charging p_i its profit is

$$(p_i - c) \frac{\exp\left(\frac{a-p_i}{\mu}\right)}{\exp\left(\frac{a-p_i}{\mu}\right) + (n-1) \exp\left(\frac{a-p_{BN}^*}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)}.$$

Taking a first-order condition and imposing symmetry, we find that

$$\frac{p_{BN}^* - c}{\mu} - \frac{n \exp\left(\frac{a-p_{BN}^*}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)}{(n-1) \exp\left(\frac{a-p_{BN}^*}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)} = 0. \quad (6)$$

We note for future reference that $\mu < p_{BN}^* - c < \mu n / (n-1)$. This follows from the fact that the lefthand side of equation (6) is strictly increasing in p_{BN}^* , is strictly negative at

$p_{BN}^* = c + \mu$, and is strictly positive at $p_{BN}^* = c + \mu n / (n - 1)$. We also note for later that when $a - c < a_0$, $\lim_{\mu \rightarrow 0} (p_{BN}^* - c) / \mu = 1$ and $\lim_{\mu \rightarrow 0} p_{BN}^* = c$. To see this, rewrite (6) as

$$\frac{p_{BN}^* - c}{\mu} = \frac{n \exp\left(\frac{a-c-a_0}{\mu}\right) \exp\left(-\frac{p_{BN}^*-c}{\mu}\right) + 1}{(n-1) \exp\left(\frac{a-c-a_0}{\mu}\right) \exp\left(-\frac{p_{BN}^*-c}{\mu}\right) + 1}. \quad (7)$$

We have already proved that $(p_{BN}^* - c) / \mu$ is bounded. Note also that $a - c < a_0$ implies $\lim_{\mu \rightarrow 0} e^{(a-c-a_0)/\mu} = 0$. Hence as $\mu \rightarrow 0$ the limit of the righthand side of (7) is 1. This establishes that $\lim_{\mu \rightarrow 0} (p_{BN}^* - c) / \mu = 1$. The claim that $\lim_{\mu \rightarrow 0} p_{BN}^* = c$ follows immediately.

PDP strictly increases consumer surplus if and only if (4) strictly exceeds (5), or

$$\frac{k}{n} > \exp\left(-\frac{p_{BN}^* - c}{\mu}\right). \quad (8)$$

By Lemma 3 PDP also strictly increases total output if and only if (8) holds.

Consider part (1) of the proposition. We proved earlier that $p_{BN}^* - c > \mu$, and so from (8) $k/n > e^{-1}$ is a sufficient condition for PDP to increase consumer surplus and output. Next, assume that $a - c < a_0$, and note that PDP increases revenue if and only if $\Delta r > 0$ where

$$\Delta r = c \frac{k \exp\left(\frac{a-c-a_0}{\mu}\right)}{1 + k \exp\left(\frac{a-c-a_0}{\mu}\right)} - p_{BN}^* \frac{n \exp\left(\frac{a-p_{BN}^*-a_0}{\mu}\right)}{1 + n \exp\left(\frac{a-p_{BN}^*-a_0}{\mu}\right)}.$$

It is convenient to rewrite Δr as

$$\Delta r = \exp\left(\frac{a-c-a_0}{\mu}\right) \left[\frac{ck}{1 + k \exp\left(\frac{a-c-a_0}{\mu}\right)} - \frac{p_{BN}^* n \exp\left(-\frac{p_{BN}^*-c}{\mu}\right)}{1 + n \exp\left(\frac{a-c-a_0}{\mu}\right) \exp\left(-\frac{p_{BN}^*-c}{\mu}\right)} \right]. \quad (9)$$

We proved earlier that $\lim_{\mu \rightarrow 0} (p_{BN}^* - c) / \mu = 1$ and $\lim_{\mu \rightarrow 0} p_{BN}^* = c$. Hence the limit of the square-bracketed term in (9) as $\mu \rightarrow 0$ is $c(k - ne^{-1})$, which is strictly positive because by assumption $c > 0$ and $k/n > e^{-1}$. By continuity the square-bracketed term is also strictly positive for μ in a neighborhood of $\mu = 0$. Moreover $e^{(a-c-a_0)/\mu}$ for all $\mu > 0$. Hence $\Delta r > 0$ for μ in a neighborhood of $\mu = 0$.

Finally, consider part (2) of the proposition. We proved earlier that $p_{BN}^* - c < \mu n / (n - 1)$, and so from (8) $k/n < e^{-\frac{n}{n-1}}$ is a sufficient condition for PDP to decrease consumer surplus and output. Since the prices of displayed products also fall, revenue decreases as well. ■

Proof of Lemma 2: Dropping time superscripts and labeling products such that $p_1 \leq p_2 \leq \dots \leq p_n$, a monopolist's profit in any given period is

$$\sum_{i=1}^k (p_i - c) D_i(p) = \sum_{i=1}^k (p_i - c) \frac{\exp\left(\frac{a-p_i}{\mu}\right)}{\sum_{j=1}^k \exp\left(\frac{a-p_j}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)},$$

and its derivative with respect to the price of product $i = 1, \dots, k$ is

$$D_i(p) \left[1 - \frac{p_i - c}{\mu} + \frac{\sum_{j=1}^k (p_j - c) D_j(p)}{\mu} \right]. \quad (10)$$

At the optimum (10) should equal zero for each $i = 1, \dots, k$ —which is only possible if $p_1 = p_2 = \dots = p_k$.

Substituting $p_1 = \dots = p_k = p^m(k)$ in equation (10) and setting it to zero, $p^m(k)$ satisfies

$$k \exp\left(\frac{a - p^m(k)}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right) - \left(\frac{p^m(k) - c}{\mu}\right) \exp\left(\frac{a_0}{\mu}\right) = 0. \quad (11)$$

The lefthand side of (11) is strictly decreasing in $p^m(k)$ and so there is a unique optimum. We also note for future reference that $p^m(k)$ is strictly increasing in k . This follows from the implicit function theorem because the lefthand side of (11) is strictly decreasing in $p^m(k)$ and strictly increasing in k . ■

Proof of Proposition 2: Let $\pi^m(k)$ denote the per-period fully collusive profit, that is, per-period industry profit when all firms charge $p^m(k)$ and k of them are shown. Let $\tilde{D}(p, k)$ be the demand faced by a single-product firm that charges p and is shown alongside $k - 1$ other single-product firms that charge $p^m(k)$.

We first argue that $\arg \max_p (p - c) \tilde{D}(p, k)$ equals $p^m(k)$ for $k = 1$, and is strictly less than $p^m(k)$ for $k > 1$. The proof for $k = 1$ is immediate. To prove the claim for $k > 1$, note that the derivative of $(p - c) \tilde{D}(p, k)$ with respect to p is proportional to

$$\frac{\exp\left(\frac{a-p}{\mu}\right) + (k-1) \exp\left(\frac{a-p^m(k)}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)}{(k-1) \exp\left(\frac{a-p^m(k)}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)} - \frac{p-c}{\mu},$$

which is strictly decreasing in p and strictly negative when evaluated at $p = p^m(k)$.

We now argue that full collusion is sustainable if and only if

$$\frac{\pi^m(k)}{1-\delta} \geq n \max_p (p - c) \tilde{D}(p, k). \quad (12)$$

For the “if” part, consider a collusive scheme where each firm charges $p^m(k)$ until there is a deviation, and then charges c forevermore. The platform randomly chooses each period which firms to display, so a firm’s expected payoff in each period from colluding is $\pi^m(k)/n$. The payoff from deviating is no more than $\max_p (p - c) \tilde{D}(p, k)$. Hence (12) is sufficient for the collusive scheme to be an equilibrium. For the “only if” part, note that in a fully collusive scheme per-period industry profit is $\pi^m(k)$, and that each firm could deviate by charging less than $p^m(k)$ and get $\max_p (p - c) \tilde{D}(p, k)$ (or arbitrarily close to it for $k = 1$). Hence if (12) does not hold at least one firm prefers to deviate from full collusion.

Next, let $\hat{\delta}_k$ be the unique δ such that (12) holds with equality.

To prove that $\widehat{\delta}_1 = 1 - 1/n$, recall that $\arg \max_p (p - c) \widetilde{D}(p, 1) = p^m(1)$, which in turn implies that $\max_p (p - c) \widetilde{D}(p, 1) = \pi^m(1)$.

To prove that $\widehat{\delta}_1 > \widehat{\delta}_2 > \dots > \widehat{\delta}_{n-1}$, first note that $\pi^m(k)$ is strictly increasing in k . Hence it is sufficient to show that $\max_p (p - c) \widetilde{D}(p, k)$ is decreasing in k . To do this, first rewrite the multiproduct firm's first-order condition in equation (11) as

$$(k - 1) \exp\left(\frac{a - p^m(k)}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right) = \left(\frac{p^m(k) - c}{\mu}\right) \exp\left(\frac{a_0}{\mu}\right) - \exp\left(\frac{a - p^m(k)}{\mu}\right). \quad (13)$$

We know from the proof of Lemma 2 that $p^m(k)$ strictly increases in k , and so the righthand side of (13) strictly increases in k . It then follows from (13) that $(k - 1) \exp\left(\frac{a - p^m(k)}{\mu}\right)$ also strictly increases in k . This further implies that $\widetilde{D}(p, k)$ strictly decreases in k because

$$\widetilde{D}(p, k) = \frac{\exp\left(\frac{a-p}{\mu}\right)}{\exp\left(\frac{a-p}{\mu}\right) + (k-1) \exp\left(\frac{a-p^m(k)}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)}.$$

It is then straightforward to argue that $\max_p (p - c) \widetilde{D}(p, k)$ decreases in k . ■

Proof of Proposition 3: The proof of Lemma 2 has derived an implicit expression for $p^m(k)$ and showed that $p^m(k)$ is strictly increasing in k , hence $p^m(1) < p^m(2) < \dots < p^m(n)$.

Consumer surplus from PDP when prices are determined by a monopolist can, using Equations (2) and (11), be written as

$$\mu \log \left\{ \frac{p^m(k) - c}{\mu} \right\} + a_0. \quad (14)$$

Consumer surplus therefore increases in k because $p^m(k)$ increases in k . Using Lemma 3 output also increases in k . Since output and prices increase in k so does revenue. ■

Proof of Proposition 4: We prove the result for the case $\tau < \infty$; similar techniques can be used to prove the result when $\tau = \infty$.

Consider a pure-strategy subgame perfect Nash equilibrium (SPNE), and let \widehat{p}_t denote the transaction price in period t along the realized equilibrium path. Let \mathcal{U} denote period 0 as well as all periods $t > 0$ in which $\widehat{p}_t > \widehat{p}_{t-1}$, and let \mathcal{D} denote all periods $t > 0$ in which $\widehat{p}_t \leq \widehat{p}_{t-1}$. Let $\widehat{\delta}$ be the unique solution to

$$\frac{\pi^m(1)}{n} = (1 - \delta)\pi^m(1) + \delta(1 - \delta^\tau)\tilde{\pi}(\max\{c, ADV\}), \quad (15)$$

where $\tilde{\pi}(\cdot)$ is a firm's per-period profit when it is the only firm shown to consumers. Note for future reference that $\widehat{\delta} = \widehat{\delta}_1 = 1 - 1/n$ when $ADV \leq c$, and $\widehat{\delta} \in \left(\widehat{\delta}_1, \widehat{\delta}_1^{1/(\tau+1)}\right]$ when $ADV > c$. To see the latter, note that the righthand side of (15) is strictly larger than the lefthand side when $\delta = \widehat{\delta}_1$, and because $\tilde{\pi}(\max\{c, ADV\}) \leq \pi^m(1)$, the righthand side of

(15) is also both strictly decreasing in δ and weakly smaller than the lefthand side when $\delta = \widehat{\delta}_1^{1/(\tau+1)}$.

We now prove part (1) of the proposition. For this part of the proof, assume $\delta < \widehat{\delta}$.

We first prove that $\widehat{p}_t \leq p^m(1)$ for all t . On the way to a contradiction, suppose there exists a $t \in \mathcal{U}$ such that $\widehat{p}_t > p^m(1)$; by the definition of Dynamic PDP any firm can win the advantage in such a period t by undercutting \widehat{p}_t . Hence any firm could charge $p^m(1)$ in period t and win the advantage. If $ADV > c$ that firm could charge $ADV \leq p^m(1)$ in the following τ periods and (by the definition of Dynamic PDP) thereby keep the advantage, and then charge c from period $t + \tau + 1$ onwards. If instead $ADV \leq c$ that firm could simply charge c from period $t + 1$ onwards. Because this is true for all firms, in period t the n firms' combined discounted payoff is at least

$$\begin{aligned} n \left[\pi^m(1) + \sum_{r=1}^{\tau} \delta^r \tilde{\pi}(\max\{c, ADV\}) \right] &= \frac{n [(1 - \delta)\pi^m(1) + \delta(1 - \delta^\tau)\tilde{\pi}(\max\{c, ADV\})]}{1 - \delta} \\ &> \frac{n [(1 - \widehat{\delta})\pi^m(1) + \widehat{\delta}(1 - \widehat{\delta}^\tau)\tilde{\pi}(\max\{c, ADV\})]}{1 - \delta} \\ &= \frac{\pi^m(1)}{1 - \delta}, \end{aligned}$$

where the strict inequality uses $\delta < \widehat{\delta}$ and the fact that the square-bracketed term on the righthand side of the first line is strictly decreasing in δ , and the final equality uses equation (15). But this is a contradiction, because the joint profit in each period cannot exceed $\pi^m(1)$. Therefore $\widehat{p}_t \leq p^m(1)$ for all $t \in \mathcal{U}$. This further implies that $\widehat{p}_t \leq p^m(1)$ in any period $t \in \mathcal{D}$. Hence $\widehat{p}_t \leq p^m(1)$ for all t .

Thus the supremum $\bar{p} \leq p^m(1)$ over transaction prices exists. We now prove that $\bar{p} = c$. On the way to a contradiction suppose that $\bar{p} \in (c, p^m(1)]$. Let \bar{t} be the first period t in which $\widehat{p}_t > \bar{p} - \Delta$ for any $\Delta \in (0, \bar{p} - c)$, and note that $\bar{t} \in \mathcal{U}$. By the definition of Dynamic PDP, any firm could charge $\bar{p} - \Delta$ in period \bar{t} and win the advantage. If $ADV > c$ that firm could charge $\min\{\bar{p} - \Delta, ADV\}$ in the following τ periods and thereby keep the advantage, and then charge c from period $\bar{t} + \tau + 1$ onwards. If instead $ADV \leq c$ that firm could simply charge c from period $\bar{t} + 1$ onwards. Because this is true for all firms, in period t the n firms' combined discounted payoff is at least

$$n \left[\tilde{\pi}(\bar{p} - \Delta) + \sum_{r=1}^{\tau} \delta^r \tilde{\pi}(\min\{\bar{p} - \Delta, \max\{c, ADV\}\}) \right],$$

which can be rewritten as

$$\frac{n}{1 - \delta} [(1 - \delta)\tilde{\pi}(\bar{p} - \Delta) + \delta(1 - \delta^\tau)\tilde{\pi}(\min\{\bar{p} - \Delta, \max\{c, ADV\}\})]. \quad (16)$$

Notice that the square-bracketed term in (16) is continuous in Δ , and continuous and strictly decreasing in δ . Hence, for any $\delta < \widehat{\delta}$, there exists Δ sufficiently small such that (16) strictly

exceeds

$$\frac{n}{1-\delta} \left[(1-\widehat{\delta})\tilde{\pi}(\bar{p}) + \widehat{\delta}(1-\widehat{\delta}^\tau)\tilde{\pi}(\min\{\bar{p}, \max\{c, ADV\}\}) \right]. \quad (17)$$

We now prove that (17) weakly exceeds $\tilde{\pi}(\bar{p})/(1-\delta)$. There are three cases to consider. First, if $ADV \leq c$, then recall from earlier that $\widehat{\delta} = 1 - 1/n$, and hence (17) equals $\tilde{\pi}(\bar{p})/(1-\delta)$. Second, if $c < \bar{p} \leq ADV$, then (17) equals $n\tilde{\pi}(\bar{p}) \left(1 - \widehat{\delta}^{\tau+1}\right) / (1-\delta)$, which weakly exceeds $\tilde{\pi}(\bar{p})/(1-\delta)$ because (as shown earlier) $\widehat{\delta} \leq (1 - 1/n)^{1/(\tau+1)}$. Third, if $c < ADV < \bar{p}$, then we can use (15) to rewrite (17) as

$$\frac{n}{1-\delta} \left[(1-\widehat{\delta})\tilde{\pi}(\bar{p}) + \pi^m(1) \left[\frac{1}{n} - (1-\widehat{\delta}) \right] \right] \geq \frac{\tilde{\pi}(\bar{p})}{1-\delta},$$

where the inequality uses $\widehat{\delta} > 1 - 1/n$ and $\pi^m(1) \geq \tilde{\pi}(\bar{p})$. We have thus proved that (17) weakly exceeds $\tilde{\pi}(\bar{p})/(1-\delta)$. This implies that (16) strictly exceeds $\tilde{\pi}(\bar{p})/(1-\delta)$. But this is impossible, because the fact that \bar{p} is the supremum over prices implies that per-period profit cannot exceed $\tilde{\pi}(\bar{p})$.

We have therefore established that $\bar{p} \leq c$. But since $\bar{p} < c$ is impossible it must be that $\bar{p} = c$. The final step is then to construct a SPNE where firms charge c in each period. Consider the following strategy:

- (1) Consider period $t = 0$, or any period $t > 0$ where at the start of the period the advantaged firm's "clock" strictly exceeds τ . Then each firm charges c .
- (2) Consider a period $t > 0$ where at the start of the period the advantaged firm's "clock" is weakly less than τ .
 - (a) If the firm that was displayed last period charged weakly less than c , then in the current period each firm charges c .
 - (b) If the firm that was displayed last period charged $p > c$, then in the current period it charges $\min\{p, \min\{c + ADV, p^m(1)\}\}$ and all other firms charge c .

Using the one-shot deviation principle it is straightforward to check that this strategy forms a SPNE. Moreover along the equilibrium path the transaction price is c in every period.

Finally, we need to prove part (2) of the proposition. Clearly if $\delta < 1 - 1/n$ then $\delta < \widehat{\delta}$ and we are done. Therefore suppose $\delta \geq 1 - 1/n$, and consider $ADV > c$. Since the righthand side of (15) is strictly decreasing in δ , $\delta < \widehat{\delta}$ if and only if

$$1 - \delta^\tau > \frac{\pi^m(1) \left[\frac{1}{n} - (1-\delta) \right]}{\delta \tilde{\pi}(ADV)}. \quad (18)$$

Notice that (18) is easier to satisfy when τ and ADV are larger. Notice also that when ADV is sufficiently large such that $\tilde{\pi}(ADV) > \pi^m(1)/n$, then the righthand side of (18) is strictly less than 1, and hence the inequality is certainly satisfied when τ is sufficiently large. ■

Proof of Proposition 5: Following similar arguments as in the proof of Proposition 1, Dynamic PDP (with $k = 1$) raises consumer surplus and output if and only if

$$\exp\left(\frac{p^m(n) - c}{\mu}\right) > n \iff p^m(n) > \mu \log(n) + c. \quad (19)$$

Recall from the proof of Lemma 2 that $p^m(n)$ solves

$$n \exp\left(\frac{a - p^m(n)}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right) - \left(\frac{p^m(n) - c}{\mu}\right) \exp\left(\frac{a_0}{\mu}\right) = 0. \quad (20)$$

The lefthand side is strictly decreasing in $p^m(n)$, so condition (19) is satisfied if and only if the lefthand side of (20) is strictly positive when evaluated at $p^m(n) = \mu \log(n) + c$ i.e.,

$$\exp\left(\frac{a - c}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right) - \exp\left(\frac{a_0}{\mu}\right) \log(n) > 0 \iff n < \tilde{n}. \quad \blacksquare$$

REFERENCES

- ASKER, J., C. FERSHTMAN, AND A. PAKES (2021): “Artificial Intelligence and Pricing: The Impact of Algorithm Design,” *NBER Working Paper 28535*.
- ASSAD, S., R. CLARK, D. ERSHOV, AND L. XU (2020): “Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market,” *Working Paper*.
- AXELROD, R., AND W. D. HAMILTON (1981): “The evolution of cooperation,” *Science*, 211(4489), 1390–1396.
- BLOEMBERGEN, D., K. TUYLS, D. HENNES, AND M. KAISERS (2015): “Evolutionary dynamics of multi-agent learning: A survey,” *Journal of Artificial Intelligence Research*, 53, 659–697.
- BROWN, Z. Y., AND A. MACKAY (2019): “Competition in Pricing Algorithms,” *Working Paper*.
- BUŞONIU, L., R. BABUŞKA, AND B. DE SCHUTTER (2010): “Multi-agent reinforcement learning: An overview,” in *Innovations in multi-agent systems and applications-1*, pp. 183–221. Springer.
- CALVANO, E., G. CALZOLARI, V. DENICOLÒ, AND S. PASTORELLO (2020): “Artificial Intelligence, Algorithmic Pricing and Collusion,” *American Economic Review*, 110(10), 3267–97.
- CHEN, L., A. MISLOVE, AND C. WILSON (2016): “An empirical analysis of algorithmic pricing on amazon marketplace,” in *Proceedings of the 25th International Conference on World Wide Web*, pp. 1339–1349.
- CMA (2018): “Pricing Algorithms: Economic working paper on the use of algorithms to facilitate collusion and personalised pricing,” *Competition and Markets Authority*.
- CRÉMER, J., Y.-A. DE MONTJOYE, AND H. SCHWEITZER (2019): “Competition policy for the digital era,” *European Commission*.

- DAL BÓ, P., AND G. R. FRÉCHETTE (2018): “On the determinants of cooperation in infinitely repeated games: A survey,” *Journal of Economic Literature*, 56(1), 60–114.
- DANA, J. D. (2012): “Buyer groups as strategic commitments,” *Games and Economic Behavior*, 74(2), 470–485.
- DE BRUIN, T., J. KOBER, K. TUYLS, AND R. BABUŠKA (2015): “The importance of experience replay database composition in deep reinforcement learning. Deep Reinforcement Learning Workshop,” *Advances in Neural Information Processing Systems (NIPS-DRLWS)*.
- DE CORNIERE, A., AND G. TAYLOR (2019): “A model of biased intermediation,” *The RAND Journal of Economics*, 50(4), 854–882.
- DECK, C. A., AND B. J. WILSON (2003): “Automated pricing rules in electronic posted offer markets,” *Economic Inquiry*, 41(2), 208–223.
- DINERSTEIN, M., L. EINAV, J. LEVIN, AND N. SUNDARESAN (2018): “Consumer price search and platform design in internet commerce,” *American Economic Review*, 108(7), 1820–59.
- DOJ (2018): “Life in the Fast Lane,” *Prepared remarks by Assistant Attorney General Makan Delrahim, US Department of Justice*.
- EZRACHI, A., AND M. E. STUCKE (2017): “Artificial intelligence & collusion: When computers inhibit competition,” *University of Illinois Law Review*, pp. 1775–1810.
- FURMAN, J., D. COYLE, A. FLETCHER, D. MCAULEY, AND P. MARSDEN (2019): “Unlocking digital competition: Report of the Digital Competition Expert Panel,” *HM Treasury*.
- GÓMEZ-LOSADA, Á., AND N. DUCH-BROWN (2019): “Competing for Amazon’s Buy Box: A Machine-Learning Approach,” in *International Conference on Business Information Systems*, pp. 445–456. Springer.
- HAGIU, A., AND B. JULLIEN (2011): “Why do intermediaries divert search?,” *The RAND Journal of Economics*, 42(2), 337–362.
- HARRINGTON, J. E. (2018): “Developing competition law for collusion by autonomous artificial agents,” *Journal of Competition Law & Economics*, 14(3), 331–363.
- INDERST, R., AND M. OTTAVIANI (2012): “Competition through commissions and kickbacks,” *American Economic Review*, 102(2), 780–809.
- JAAKKOLA, T., M. I. JORDAN, AND S. P. SINGH (1994): “Convergence of stochastic iterative dynamic programming algorithms,” in *Advances in neural information processing systems*, pp. 703–710.
- KLEIN, T. (2021): “Autonomous Algorithmic Collusion: Q-Learning Under Sequential Pricing,” *RAND Journal of Economics (forthcoming)*.
- KOVACIC, W. E., R. C. MARSHALL, L. M. MARX, AND M. E. RAIFF (2006): “Bidding rings and the design of anti-collusion measures for auctions and procurements,” *Handbook of procurement*, 15.
- LI, Y. (2017): “Deep reinforcement learning: An overview,” *arXiv preprint arXiv:1701.07274*.

- LIN, L.-J. (1992): “Self-improving reactive agents based on reinforcement learning, planning and teaching,” *Machine learning*, 8(3-4), 293–321.
- MEHRA, S. K. (2015): “Antitrust and the robo-seller: Competition in the time of algorithm,” *Minnesota Law Review*, pp. 1323–1375.
- MIKLÓS-THAL, J., AND C. TUCKER (2019): “Collusion by algorithm: Does better demand prediction facilitate coordination between sellers?,” *Management Science*, 65(4), 1552–1561.
- MNIH, V., K. KAVUKCUOGLU, D. SILVER, A. A. RUSU, J. VENESS, M. G. BELLEMARE, A. GRAVES, M. RIEDMILLER, A. K. FIDJELAND, G. OSTROVSKI, ET AL. (2015): “Human-level control through deep reinforcement learning,” *Nature*, 518(7540), 529–533.
- O’CONNOR, J., AND N. WILSON (2019): “Reduced Demand Uncertainty and the Sustainability of Collusion: How AI Could Affect Competition,” *Working Paper*.
- OECD (2017a): “Algorithms and Collusion - Note from Italy,” *OECD*.
- (2017b): “Algorithms and Collusion: Competition Policy in the Digital Age,” *OECD*.
- SALCEDO, B. (2015): “Pricing Algorithms and Tacit Collusion,” *Working Paper*.
- SANDHOLM, T. W., AND R. H. CRITES (1995): “On multiagent Q-learning in a semi-competitive domain,” in *International Joint Conference on Artificial Intelligence*, pp. 191–205. Springer.
- SILVER, D., A. HUANG, C. J. MADDISON, A. GUEZ, L. SIFRE, G. VAN DEN DRIESSCHE, J. SCHRITTWIESER, I. ANTONOGLU, V. PANNEERSHELVAM, M. LANCTOT, ET AL. (2016): “Mastering the game of Go with deep neural networks and tree search,” *Nature*, 529(7587), 484.
- SUTTON, R. S., AND A. G. BARTO (2018): *Reinforcement learning: An introduction*. MIT press.
- TEH, T.-H. (2020): “Platform governance,” *Working Paper*.
- TEH, T.-H., AND J. WRIGHT (2020): “Intermediation and steering: Competition in prices and commissions,” *American Economic Journal: Microeconomics*.
- TESAURO, G., AND J. O. KEPHART (2002): “Pricing in agent economies using multi-agent Q-learning,” *Autonomous Agents and Multi-Agent Systems*, 5(3), 289–304.
- TSITSIKLIS, J. N. (1994): “Asynchronous stochastic approximation and Q-learning,” *Machine learning*, 16(3), 185–202.
- WALTMAN, L., AND U. KAYMAK (2008): “Q-learning agents in a Cournot oligopoly model,” *Journal of Economic Dynamics and Control*, 32(10), 3275–3293.
- WATKINS, C. J. (1989): “Learning from delayed rewards,” *Thesis Chapter, King’s College, Cambridge*.
- WATKINS, C. J., AND P. DAYAN (1992): “Q-learning,” *Machine learning*, 8(3-4), 279–292.