Econometrica, Vol. 91, No. 5 (September, 2023), 1841–1879

PLATFORM DESIGN WHEN SELLERS USE PRICING ALGORITHMS

JUSTIN P. JOHNSON
Graduate School of Management, Cornell University

ANDREW RHODES
Toulouse School of Economics, University of Toulouse Capitole and CEPR

MATTHIJS WILDENBEEST
Department of Economics, University of Arizona and CEPR

We investigate the ability of a platform to design its marketplace to promote competition, improve consumer surplus, and increase its own payoff. We consider demand-steering rules that reward firms that cut prices with additional exposure to consumers. We examine the impact of these rules both in theory and by using simulations with artificial intelligence pricing algorithms (specifically Q-learning algorithms, which are commonly used in computer science). Our theoretical results indicate that these policies (which require little information to implement) can have strongly beneficial effects, even when sellers are infinitely patient and seek to collude. Similarly, our simulations suggest that platform design can benefit consumers and the platform, but that achieving these gains may require policies that condition on past behavior and treat sellers in a nonneutral fashion. These more sophisticated policies disrupt the ability of algorithms to rotate demand and split industry profits, leading to low prices.

KEYWORDS: Algorithms, collusion, platform design, prominence, Q-learning.

MANY PRODUCTS AND SERVICES are sold through online platforms such as Amazon, Booking, and eBay. These platforms design rules that govern interactions between buyers and sellers. A leading example is the rules for how different products are ranked or displayed to consumers. It is increasingly recognized that this "regulatory power" of platforms allows them to shape the nature of competition on their marketplaces.[1] At the same time, competition on these marketplaces is changing due to the increased use of pricing

Justin P. Johnson: jpj25@cornell.edu
Andrew Rhodes: andrew.rhodes@tse-fr.eu
Matthijs Wildenbeest: wildenbeest@arizona.edu
We thank Emilio Calvano, Han Huynh, Bruno Jullien, Timo Klein, Scott Duke Kominers, Alexander MacKay, Catarina Marvao, Volker Nocke, Patrick Rey, Antonio Russo, Tat-How Teh, Johannes Walter, four referees, and participants at the 2021 American Law and Economics Conference, the 2021 Bates–White Antitrust Conference, the University of California (Berkeley), the University of California (Irvine), Charles River Associates, Compass Lexecon, the 2020 Conference on Algorithmic Collusion (University of Amsterdam), the 2022 CRESSE Conference, the DERN Webinar (Monash University), DICE, the 2022 Duke-UNC Triangle Microeconomics Conference, the 2022 EARIE Conference (Vienna), the 2020 Econometric Society Meetings (San Diego), the 2022 ENSAI Economics Days, Hitotsubashi University, the 2022 University of Humboldt–University of Toronto Theory Conference, 2021 IIOC, Luohan Academy, MIT, the 2021 Northwestern University Conference on Antitrust Economics and Competition, the Paris Digital Economics Seminar, the Toulouse School of Economics Online Economics of Platforms Seminar, the 2022 Tsinghua BEAT, the 2021 Workshop on Pricing Technologies (European University Institute), UIBE (Beijing), the University of Virginia, Yale, and the ZEW Conference on the Economics of Information and Communication Technologies. Rhodes acknowledges funding from the European Research Council (Grant 670494) and Agence Nationale de la Recherche (Grant ANR-17-EURE-0010).

[1]See, for example, Crémer, de Montjoye, and Schweitzer (2019).

 https://doi.org/10.3982/ECTA19978

algorithms by sellers.[2,3] Yet little is known about how price-setting algorithms respond to platform-design rules.

In this article, we investigate the power of platforms to design rules that shape competition on their marketplaces, especially when algorithms set prices. The demand-steering rules that we develop are motivated by economic theory, but we use extensive simulations to assess their performance when algorithms set prices. We show that a platform can design rules that both increase its own payoff and that of consumers. Although relatively simple rules may suffice when sellers behave competitively, more subtle ones—which condition on past behavior and treat sellers asymmetrically—may be required when there is a risk of collusion. Even these more subtle policies impose only a light informational burden on the platform.

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 single retail platform. The platform earns commissions from sales on its marketplace, and may also put some weight on consumer utility—consistent with more dynamic considerations or unmodeled platform competition. We consider two platform design rules, both of which involve steering consumer demand toward a subset of the sellers. The simpler of these policies, *price-directed prominence* (PDP), involves the marketplace simply steering demand toward 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. A related policy is the Amazon "buy box," which shows consumers one seller among those offering a given homogeneous product.[4] As with our PDP policy, current price is believed to be a strong determinant of whether a firm is awarded the buy box (i.e., displayed by the platform to many consumers). As with our Dynamic PDP policy, past performance and behavior also appear to influence which firm is awarded the buy box.[5] How-

[2]According 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.

[3]This shift to algorithmic pricing has also raised regulatory concerns; see OECD (2017b), CMA (2018), and DOJ (2018). Simulation-based work by Calvano, Calzolari, Denicolò, and Pastorello (2020) and Klein (2021) shows that algorithms may learn to play collusive strategies, while empirical work by Clark, Assad, Ershov, and Xu (forthcoming) and Musolff (2022) finds that adopting AI algorithms may lead to higher prices. Brown and MacKay (2023) show that adopting (non-AI) algorithms allows firms to achieve a form of price commitment that raises overall prices.

[4]The buy box is reportedly effective at steering demand, with 83% of sales going to its winner (see `repricer.com`).

[5]Some recent empirical work has investigated factors that influence buy box allocation. Current price is an important determinant. For example, Lee and Musolff (2021) report that the buy box is awarded to the lowest-priced seller over 50% of the time, leading to a buy box price elasticity of around −19. However, past performance also matters. For example, Gómez-Losada, Asencio-Cortés, and Duch-Brown (2022) find that current price in relation to previous periods helps predict changes in the buy box winner, while Chen, Mislove, and Wilson (2016) find that sellers with more (and better) reviews are more likely to occupy the buy box. That higher reviews make a seller more likely to win the buy box is echoed by industry experts (see, e.g., `repricerexpress.com/win-amazon-buy-box/`). Because the number and level of reviews is correlated with past performance, including past sales, and hence past prices, this is similar in spirit to our Dynamic PDP policy.

ever, the policies we consider are different from the buy box. First, we allow for horizontally differentiated products whereas the buy box applies to firms selling an identical physical product (although sellers may vary in terms of the quality of customer service they offer, for example). Second, we allow multiple firms to be shown to consumers.

We now discuss our results in more detail. Section 1 of our article examines the theoretical impact of the two design policies. Theory predicts that prices fall under PDP, introducing a tradeoff for consumers between lower prices and less product variety. Whether consumers on balance benefit hinges on whether sellers 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; moreover, we show that the policy performs well relative to arbitrarily sophisticated policies, despite its light informational burden. However, when sellers collude PDP performs poorly, lowering 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.

Whether the platform benefits from these steering policies depends on its business model. When per-unit fees are used and PDP or Dynamic PDP benefits consumers, we find that overall output also increases, raising total commissions. As the platform cares about a weighted sum of commissions and consumer utility, its payoff increases. (In extensions, we consider revenue-sharing and identify conditions under which the platform similarly gains.)

Starting in Section 2, we use simulations 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.[6] 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 simulations, we find, as a preliminary result, that absent platform design the algorithms typically set prices that exceed the predicted static Bertrand–Nash prices. This confirms a key contribution of Calvano et al. (2020), who study interactions between algorithms in a setting closely related to ours.

[6]See `goaura.com`, `sunpricer.com`, `feedvisor.com/amazon-repricer`, `kalibrate.com`, and `a2isystems.com`.

Our simulations show that PDP lowers prices, but that these declines need not be large enough to compensate consumers for the loss of variety, or boost output enough to benefit the platform. PDP is less likely to benefit consumers and the platform 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 and the platform sometimes benefit even though theory suggests they should not; this is more likely when product differentiation is lower.

Our simulations 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 and moderate increases in platform commissions.

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.

We also consider several extensions and robustness checks. These consider different platform business models, market environments, and alternative platform-design rules. Overall, the evidence from our simulations suggests that platform design decisions can meaningfully benefit consumers and the platform when algorithms set prices. However, achieving these gains may require more subtle steering policies such as Dynamic PDP.

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 et al. (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.[7] They are part of a broader investigation into cooperation in games dating back to Axelrod and Hamilton (1981).[8]

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 (2022) examines how tradeoffs 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 (2022)). Our approach differs in a number of ways but particularly in that we focus on the power of an intermediary to design rules that shape competition, both to its own advantage and that of consumers, including when there is a risk of collusion and sellers use pricing algorithms.

1. THEORETICAL MODEL

Here, we lay out a theoretical model of how platform design affects prices and other outcomes. There are $n \geq 2$ firms, each of which sells one differentiated product on a monopoly retail platform. The firms have the same constant variable production cost $\tilde{c} > 0$, and pay the platform a fee $f \geq 0$ each time they sell a unit of the product; let $c = \tilde{c} + f$ denote a firm's effective marginal cost. Firms interact repeatedly over time. In each period $t = 0, 1, \ldots$, 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

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

whereas if she buys no product she obtains the outside option utility $u_0^t = \zeta_0$. 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 $\mu > 0$.

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

[7] Asker, Fershtman, and Pakes (forthcoming) 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 (2021) 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.

[8] There is also a large literature specifically on whether and how humans learn to collude; Dal Bó and Fréchette (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 $\mathcal{N}^t$ receives zero demand. Meanwhile, a firm $i \in \mathcal{N}^t$ receives (standard logit) demand

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

where p^t is the vector of prices of the n firms at time t. Let $Q(p^t) = \sum_{j \in \mathcal{N}^t} D_j(p^t)$ denote total industry output. Consumer surplus in period t is

$$U(p^t) = \mu \log\left[\sum_{j \in \mathcal{N}^t} \exp\left(\frac{a - p_j^t}{\mu}\right) + 1\right]. \tag{2}$$

After some manipulations, $U(p^t) = -\mu \log(1 - Q(p^t))$; consumer surplus is increasing in total industry output. This follows from the aggregative structure of the logit model.

We assume that, for given $\omega \in [0, 1]$, the platform's payoff in period t is

$$\Omega(p^t) = \omega f Q(p^t) + (1 - \omega) U(p^t), \tag{3}$$

that is, a weighted average of per-unit fee revenues, $fQ(p^t)$, and consumer surplus, $U(p^t)$. Allowing the platform to put weight $1 - \omega$ on consumer surplus is a reduced form way to capture, for example, the benefits of dynamically growing the consumer base over time, or competition with other platforms. For simplicity, we take the per-unit fee f as fixed so as to focus on the role of platform design. (In reality, many retail platforms set commissions which cover a wide range of products, so it is reasonable to assume that f is not customized at the product level. Nonetheless, we show in the Online Appendix in the Supplementary Material (Johnson, Rhodes, and Wildenbeest (2023)) that our main results extend to the case where the platform endogenously chooses f to maximize its payoff. We also show there that our main insights extend to the case where commissions are based on revenue sharing rather than per-unit fees.)

We now consider two platform design rules that determine the set of displayed firms $\mathcal{N}^t$, and investigate how they affect prices and other outcomes.

1.1. *Price-Directed Prominence*

The first platform design rule 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 kth lowest price, then a subset of the firms with the kth 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 sellers behave like a cartel or not. First, we assess predicted outcomes when sellers behave competitively.

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—it is well known that there is a unique equilibrium in which each firm charges a strictly positive mark-up over effective marginal cost. In contrast, under PDP prices are driven down to effective marginal cost as firms compete for the right to be displayed to consumers.

LEMMA 1: *Under PDP, in a competitive market the k lowest equilibrium prices equal effective marginal cost $c = \tilde{c} + f$.*

PDP therefore presents a tradeoff, for both consumers and the platform, between lower prices but less variety: consumer surplus and total output (and hence fee revenues) increase due to lower prices, but decrease because consumers now choose from only $k < n$ products.

PROPOSITION 1: *In a competitive market, compared to when all n firms are displayed, PDP increases consumer surplus and the platform's payoff if and only if $k/n > \lambda$. The critical threshold λ is such that $e^{-\frac{n}{n-1}} < \lambda < e^{-1}$.*

Because $e^{-1} \approx 0.368$, Proposition 1 says that PDP benefits consumers and the platform even if as many as 63% of products are not shown; the gains from intensified price competition outweigh the variety loss. This threshold is certainly satisfied if $k = n - 1$, such that all but one product is displayed. Indeed, $k = n - 1$ is the best case of PDP for both consumers and the platform—it minimizes variety loss while keeping prices at effective marginal cost.

PDP is a simple policy with a low informational requirement; the platform can implement PDP even if it has no information about costs or demand.[9] Of course, if the platform has more information it might be able to use more complex rules. However, as we now show, PDP performs very well relative to even the most sophisticated rules. To see this, consider a hypothetical scenario in which all n sellers are displayed and yet they price at effective marginal cost $\tilde{c} + f$.[10] Conditional on each seller earning nonnegative profit in each period, even the most sophisticated platform rules could not give consumers higher surplus than they would achieve in this scenario; similarly, because this scenario maximizes total output (and hence commissions), it also provides an upper bound on what is achievable for the platform's payoff. Let Ω^{Max} and U^{Max} denote, respectively, these upper bounds on possible platform payoff and consumer surplus. Analogously, let Ω^{PDP} and U^{PDP} denote these values under PDP when $k = n - 1$, and Ω^{BN} and U^{BN} under Bertrand–Nash (with no platform design).

[9] Another low-information policy would be to randomly display k firms irrespective of their prices. This policy is inferior to PDP: it induces the same variety loss, but does not reduces prices. We return to this policy in our simulations in Section 3.4.

[10] If the platform perfectly knew the production cost and was able to tailor design rules at the product level, it could enforce this outcome by threatening to delist any seller that does not price at effective marginal cost $\tilde{c} + f$.

PROPOSITION 2: *The platform's payoff under PDP (when $k = n - 1$) satisfies*

$$\frac{\Omega^{\mathrm{PDP}}}{\Omega^{\mathrm{Max}}} \geq 1 - \frac{1}{n} \quad \textit{and} \quad \frac{\Omega^{\mathrm{PDP}} - \Omega^{\mathrm{BN}}}{\Omega^{\mathrm{Max}} - \Omega^{\mathrm{BN}}} \geq 1 - \frac{e}{n(e-1)}. \tag{4}$$

These bounds also apply if the platform's payoff, Ω, is replaced by consumer surplus, U.

This proposition provides two bounds on the performance of PDP, when $k = n - 1$ sellers are displayed.[11] The first is a bound on the absolute performance of PDP, that is, the share of maximum possible payoff (for either the platform or consumers) that PDP obtains. The second is a bound on the relative performance of PDP, that is, the share of the maximum possible increase in payoff (for either the platform or consumers) relative to the case of no platform design. For example, thinking about platform payoffs, $\Omega^{\mathrm{Max}} - \Omega^{\mathrm{BN}}$ is the maximum possible increase due to platform design, while $\Omega^{\mathrm{PDP}} - \Omega^{\mathrm{BN}}$ is the increase due to PDP. These bounds are clearly most impressive when n is large; for example, when $n = 10$, the bounds equal 0.9 and 0.842, respectively. However, depending on the demand and cost parameters, PDP can perform much better than suggested by these bounds, even when n is small. To illustrate, consider the default specification used in our later algorithmic simulations ($n = 2$, $a = 2$, $c = 1$, $\mu = 1/4$): one can show that PDP gives consumers and the platform (for all $f > 0$ and $\omega \in [0, 1]$) at least 85% of their maximum possible payoff, and at least 62% of the maximum possible increase in payoff relative to Bertrand–Nash.

Overall, our results show that when markets are competitive PDP performs well. 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: *Under PDP, 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

[11]Note that the platform payoff bounds in Proposition 2 require either that $f > 0$ or $\omega < 1$, or both, because if instead $f = 0$ and $\omega = 1$ then the platform's payoff equals zero under any platform rule.

part (ii) ensures that a firm earns zero in each period after it deviates, despite intrinsic differentiation.

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 (i.e., $k-1$) other firms and so gets higher demand. Both of these effects decrease the stability of a cartel.

PROPOSITION 3: *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*:

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

This proposition says that—presuming some PDP is in effect—showing fewer firms makes full collusion less stable. Numerical computations 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.[12]

On the other hand, if δ is sufficiently large then full collusion is sustainable whether PDP is used or not. For example, Proposition 3 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 4: *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*:

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

However, consumer surplus and the platform's payoff are also lower: *as fewer firms are displayed*, *the decline in prices is too small to offset the loss of variety.*

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 multiproduct 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 4, because under full collusion the decline in prices is too small to offset the decrease in variety. Output and, therefore, total platform fees, falls for the same reason. This differs from what happens in competitive markets, where a modest loss of variety intensifies

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

price competition so much that consumer surplus and the platform's payoff both increase (Proposition 1).

In light of Proposition 4, we now turn to another platform design, which destabilizes collusion and benefits consumers and the platform even for 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* (Dynamic PDP or DPDP). 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 $t = 0$, firms set prices and one firm with the lowest price is the only firm shown to consumers, and 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, and also receives the advantage in period $t + 1$, so long as:

(1) firm i has not raised its price in period t compared to its price in period $t - 1$, and

(2) no rival in period t undercuts firm i by strictly more than a fixed value $\text{ADV} > 0$.

If either of these two conditions is violated, then in period t a firm with the lowest price is the only firm shown to consumers, and that firm also receives the advantage in period $t + 1$.

Under Dynamic PDP, the firm that is shown today gets a "pricing advantage" ADV, making it easier for that firm to be shown to consumers tomorrow, so long as it does not raise price.

As we now show, Dynamic PDP leads to marginal cost pricing even when the market would otherwise be cartelized. To interpret the next result, recall that $\widehat{\delta}_1$ is the critical discount factor for full collusion under PDP when $k = 1$.

PROPOSITION 5: *Consider Dynamic PDP with an advantage* $0 < \text{ADV} \leq p^m(1)$.

(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 transaction price equals effective marginal cost in all periods*.

(2) *Moreover, if* ADV *is sufficiently large*, *then* $\widehat{\delta} = 1$ *so that the transaction price is effective marginal cost for all values of* δ.

Proposition 5 says that for $0 < \text{ADV} \leq p^m(1)$ collusion is harder to sustain when the platform implements dynamic rather than regular PDP. Indeed, when ADV is sufficiently large, firms are unable to collude at any price above effective marginal cost even as they become arbitrarily patient, that is, as $\delta \to 1$.

We now sketch an intuition for why collusion is infeasible when ADV is sufficiently large. To illustrate this as simply as possible, suppose that $\text{ADV} = p^m(1)$ and that the firms try to fully collude at price $p^m(1)$; let $\pi^m(1)$ denote the corresponding per-period monopoly profits. 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 $p^m(1) - \psi$ in each future period and keep the advantage forever, because no firm could undercut it by strictly more than ADV. For ψ sufficiently small, this means the payoff to the deviating firm is arbitrarily close to $\pi^m(1)/(1 - \delta)$. Since any firm could so deviate, in equilibrium firms must collectively earn arbitrarily close to $n\pi^m(1)/(1 - \delta)$—but this is impossible, because per-period industry profit cannot exceed the monopoly profit $\pi^m(1)$. A similar argument can be used to rule out any collusive scheme that tries to sustain prices above effective marginal cost, and this argument does not require ADV as large as $p^m(1)$.

PROPOSITION 6: *Suppose δ is sufficiently high that, absent platform design, firms would fully collude. Compared to the case where all n firms are shown to consumers, Dynamic PDP with sufficiently large* ADV *increases consumer surplus and platform payoffs if and only if*

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

Dynamic PDP can benefit consumers and the platform in a wide variety of circumstances when firms are patient and markets are cartelized. Note that when $n = 2$, which is the baseline case considered in our later simulations, this condition always holds.[13]

A limitation of DPDP as given in Definition 2 is that only one firm is ever shown. However, we show in the Online Appendix that, depending on parameters, DPDP can be extended to allow more firms ($1 < k < n$) to be shown while still enforcing effective marginal cost pricing. Performance bounds related to those given in Proposition 2 can also be established.

An implication of DPDP as given in Definition 2 is that a firm that wins the advantage in the first period can hold it forever, so long as it does not raise its price and is not substantially undercut by its rivals. However, we show in the Online Appendix that DPDP can be extended to allow for an exogenous probability that the advantage is taken away from a firm and reassigned to the firm with the lowest price in that period. As long as this probability is sufficiently low, effective marginal cost pricing again ensues.

We note that one potential downside of reducing prices to marginal cost is that sellers' profits are compressed. If this reduces seller entry, both consumers and the platform may be harmed. We do not address this issue but acknowledge it may be important in practice (Lee and Musolff (2021)). Of course, in the broader economy, this is also true of any policy that reduces seller profits.

The platform design tools that we have considered are conceptually related to prior literature on buyer groups and bundling in auctions. In particular, there is a close connection between PDP and buyer groups, which restrict their access to variety in order to induce sellers to set lower prices; Dana (2012) shows that this tradeoff works well in competitive markets, but unlike us, does not consider collusive markets.[14] As pointed out by Dana (2012), buyer groups are in turn related to bundling in auctions (see, e.g., Palfrey (1983)). Along these lines, Kovacic et al. (2006) assess practical strategies to weaken the power of (non-algorithmic) cartels in procurement auctions. They argue that holding auctions at irregular and long intervals (which is like bundling time periods) can help destabilize collusion in procurement auctions. An extreme example would be to bundle all time periods together: firms set prices once in the initial period and are bound to supply the product at that price for all time. In theory, this would deliver effective marginal cost

[13]Note that this condition holds for a weakly larger range of n than the corresponding one for regular PDP (and $k = 1$) in competitive markets, given in Proposition 1; this is simply because fully collusive prices exceed competitive prices. Moreover, as per Proposition 4, in cartelized markets regular PDP harms both consumers and the platform.

[14]See also O'Brien and Shaffer (1997) and Marx and Shaffer (2010), who explore a related tradeoff when retailers are required or choose to purchase from a single supplier.

pricing, but in practice might not work well.[15] Kovacic et al. (2006) further argue that the use of favored or maverick bidders can also destabilize collusion, although they do not identify the variety losses associated with these policies. Such policies are conceptually related to Dynamic PDP (including the variant discussed above, where the advantage is probabilistically taken away and reassigned), because one firm tends to hold the advantage for multiple periods and is treated favorably today based on earlier low prices.

1.3. *Connecting Theory to Algorithms*

An important objective of this article is to examine how platform-design policies perform when algorithms set prices, in addition to examining the purely theoretical performance of such policies. In the next sections, we describe in detail the algorithms that we will deploy, and in turn examine their performance.

Our modeling choices in the theory section above have been made with algorithms in mind. First, we have chosen relatively simple policies that give clear direction to the algorithms—charging lower prices leads to additional consumers being steered to a firm. (As discussed earlier, these policies also impose a low informational requirement on the platform.)

Second, our Dynamic PDP policy, although more complicated, is low-dimensional and only requires that the state space be expanded to account for which firm was displayed in the previous period. This makes sense given the tabular Q-learning algorithms (discussed in more detail below) that we deploy are best suited to smaller state spaces. Moreover, in our simulations we will distinguish between whether or not the algorithms can track which firm was most recently displayed to consumers, and assess how this affects their performance.

Third, by design Dynamic PDP admits the possibility that which firm is preferentially displayed changes over time. In sharp contrast, as discussed just above, an alternative policy would let prices in the initial period determine which firms are displayed in all future periods. But such an alternative policy would not provide scope for the algorithms to experiment, learn, and adapt their strategies over time—and these are essential elements of AI algorithms. (For other limitations on this alternative policy, see our discussion just above.)

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).

We use a common technique, (tabular) Q-learning, which is motivated by the theory of dynamic programming applied to Markov Decision Environments (Watkins (1989), Sutton and Barto (2018)). We choose these algorithms because real-world pricing problems are often complex, with sellers needing to learn what is optimal. A powerful feature of Q-learning is that it requires little knowledge of the underlying environment. The algorithm

[15]It may be unrealistic to think that a marketplace could enforce such a policy (e.g., because small sellers are effectively anonymous and can leave the marketplace without repercussion) or that it would want to (e.g., because future entry by new, more efficient sellers would in effect be foreclosed). Also see our discussion in Section 1.3 for reasons that this policy might not perform well when algorithms set prices.

can recognize which state the system is in and which actions are available, but has no prior knowledge of the payoff functions or the state transition probabilities. Another feature is that Q-learning is not designed as such to foster collusion. Thus, although we recognize the possibility that an algorithm's designer might wish to explicitly foster collusion, we take a more neutral perspective that the designer simply wishes to earn high profits in a complex environment.

Before turning to the details of our MARL approach, we first review the basics of Q-learning in stationary single-player environments where the state space is fairly small.[16]

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

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

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,

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

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.[17] Such experimentation ensures that Q-learning sufficiently explores all states and actions.

[16]The standard modern reference on this topic is Sutton and Barto (2018). Cutting-edge techniques for larger state spaces include "deep reinforcement learning," which involves approximating the state space; see Mnih et al. (2015) and Silver et al. (2016), as well as Lin (1992). 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.

[17]This approach to Q-learning is often called ε-greedy. One common alternative for modeling experimentation is the Boltzman approach, whereby the algorithm is more likely to play actions with higher associated Q-values.

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 to be

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

where $Q(s, x)$ is the previous "un-updated" element of the Q-matrix, and $\alpha \in (0, 1)$ is the learning rate parameter, which captures the extent to which old information is replaced by new information. The probability of experimentation ϵ_t is given by

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

where $\beta > 0$ is the experimentation parameter. A higher β means that experimentation tapers off more quickly. An algorithm's learning is thus 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 et al. (2020)), we nearly always obtain convergence (defined precisely below).

2.2. *Our MARL Approach*

Our specification substantially follows that in Calvano et al. (2020), allowing us to assess the performance of our policies relative to the benchmark that they have established. 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 scale parameter $\mu = 1/4$. For Dynamic PDP, our baseline has ADV $= 0.3$, so that a firm with the pricing advantage keeps it so long as it is not undercut by more than 0.3 and does not raise its price.

We implement MARL as follows. In each period, the action space is the set of prices that agents can set. We discretize this set of prices to contain fifteen equally spaced elements in the set $[1, 2.1]$.[18] The lower bound of this set is marginal cost and the upper bound is slightly above the fully collusive prices in the absence of platform design. 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 model from Section 1 by introducing a parameter γ, which measures the extent to which the platform implements (Dynamic) PDP.[19] We vary γ in increments

[18]Hence, the pricing grid contains the following elements (to three decimal places): {1.000, 1.079, 1.157, 1.236, 1.314, 1.393, 1.471, 1.550, 1.629, 1.707, 1.786, 1.864, 1.943, 2.021, 2.100}.

[19]An alternative interpretation is that γ consumers are behaviorally biased and follow platform recommendations, whereas the remaining $1 - \gamma$ consumers will exert additional search to discover all available products. Recalling our earlier discussion of seller entry, having $\gamma < 1$ is also a way of guaranteeing positive profits to each firm.

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.

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.[20] 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 associated with these converged Q-matrices by averaging over a 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: SIMULATION RESULTS

Here, we present the results of our simulations on the effects of PDP when $n = 2$, $\mu = 1/4$, $\delta = 0.95$, $\alpha = 0.15$, and $\beta = 10^{-5}$. We then explore the robustness of our results to changes in the learning parameters and product differentiation, and relative to a policy where the displayed firm is chosen randomly. Section 4 considers Dynamic PDP.

3.1. *Default Specification*

To set a baseline, Table I 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 1.471. The collusive price maximizes the joint profits of the firms, with both firms charging the same price 1.943.[21] Table I also reports the (share-weighted) AI prices that the algorithms actually charge, given by 1.685. Thus,

TABLE I

BENCHMARK ASSESSMENT OF AI PRICING RELATIVE TO BERTRAND–NASH AND COLLUSIVE PRICING WHEN THERE IS NO PRICE-DIRECTED PROMINENCE.

Bertrand–Nash Price	Collusive Price	AI Price
1.471	1.943	1.685

[20]We stop the algorithms if they do not converge after 1 billion periods, as in Calvano et al. (2020).

[21]We compute Bertrand–Nash and collusive outcomes using the discrete pricing grid employed by the algorithms.

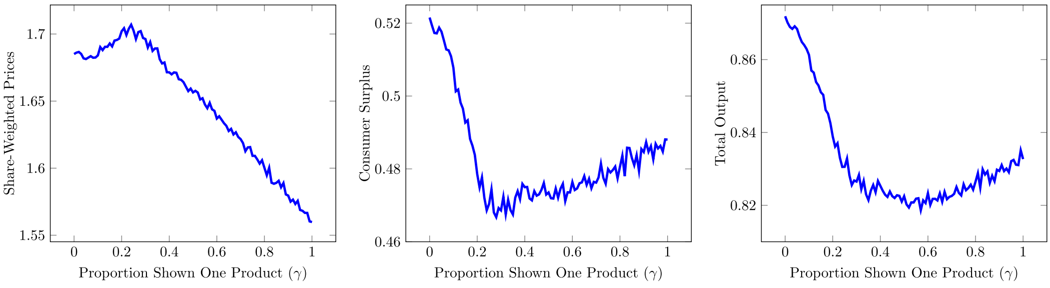


FIGURE 1.—The effect of PDP on prices (left panel), consumer surplus (middle panel), and total output (right panel).

our algorithms typically converge to a price exceeding the Bertrand–Nash price, in line with what we would expect from Calvano et al. (2020).

Figure 1 lays out the effects of price-directed prominence. The left panel shows how the algorithmic "PDP price" varies with the mass γ shown only one product, while the middle and right panels show the corresponding changes in consumer surplus and total output respectively. The PDP price slightly increases for low values of γ but thereafter mostly decreases. Over the entire range of γ, prices decline by about 7.4%. The fact that prices decline is consistent with our cartel analysis in Proposition 4, which seems relevant given that $\delta = 0.95$.

However, the lower prices come at a cost to consumers and the platform 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, PDP harms both consumers and the platform. Specifically, PDP reduces consumer surplus and total output (and hence total commissions) for all γ values. Indeed, moving from $\gamma = 0$ to $\gamma = 1$ lowers them by about 6.4% and 4.5%, respectively.

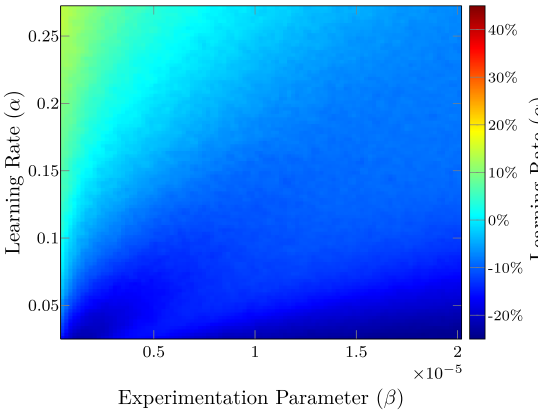
Although the algorithms do not achieve the fully collusive prices even in the absence of PDP, nonetheless the effect of PDP is consistent with what theory predicts for a fully collusive cartel: prices fall but not enough to benefit consumers and the platform (Proposition 4).

3.2. *Learning-Parameter Robustness*

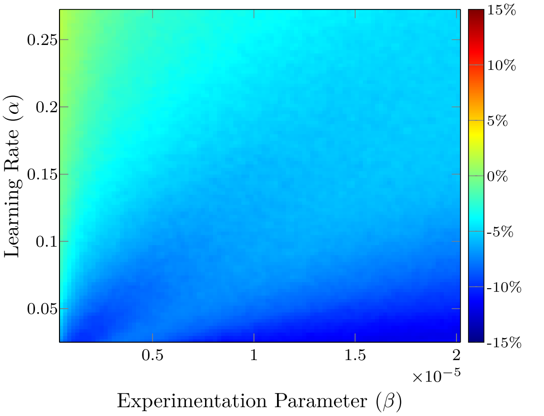
The results reported above suggest that PDP may decrease consumer surplus and platform payoffs, under our default specification. To explore the robustness of this conclusion, here we vary the learning parameters α and β.

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 et al. (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 et al. (2020). We note that our default specification of $(\alpha, \beta) = (0.15, 10^{-5})$ is the midpoint of this region.



(a) Consumer Surplus



(b) Total Output

FIGURE 2.—Heatmaps of the effects of PDP on consumer surplus and total output, for $\gamma = 0.7$. Consumer surplus increases in 13.8% of the cases, while total output increases in 1.7% of cases.

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. For each pair (α, β), we compute percentage changes in outcomes (consumer surplus and total output) when going from $\gamma = 0$ to $\gamma = 0.7$.

We present the resulting assessment in two heatmaps contained in Figure 2, with consumer surplus in the left panel and total output in the right panel.

From these heatmaps, three facts are apparent. First, although consumer surplus and total output continue to typically go down when PDP is introduced, for some learning parameters they increase. Specifically, consumer surplus increases for 13.8% of the (α, β) pairs, while total output increases for 1.7% of the pairs. Second, changes in the learning parameters can have significant effects on the magnitude of the outcome.

Third, we see from the heatmaps that PDP performs best in the region where α is large and β is small (top left of the heatmap), and worst when α is small and β is large (bottom right).[22] Although it is difficult to ascribe precise intuitions to the outcomes of AI learning processes, we now provide qualitative descriptions that help us better understand differences in how the algorithms are behaving in these two distinct regions. What we find is that prices before imposing PDP are higher in the top left region than in the bottom right, but after imposing PDP they are lower in the top left region than in the bottom right. Because the loss of variety is the same in either case (because α and β are not features of market demand), PDP is more effective in the top left region.[23]

[22]As one might expect, we also find that price decreases from the imposition of PDP tend to be larger in the region where consumer surplus gains are higher.

[23]PDP also affects other aspects of the algorithms' play differently in these two regions. In the top left of the heatmap, imposing the PDP policy induces the algorithms to switch away from charging constant prices to using price cycles, and also causes them to split profits less symmetrically. In contrast, in the bottom right of the heatmap, imposing the PDP policy causes little qualitative change either in how prices vary over time or how the algorithms split profits. We explore pricing and profit patterns in more detail in Section 5.

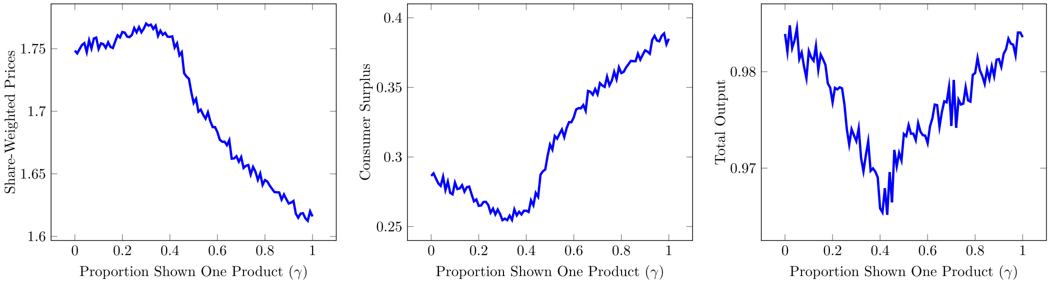


FIGURE 3.—The effect of PDP on prices (left panel), consumer surplus (middle panel), and total output (right panel), with lower differentiation ($\mu = 1/20$).

3.3. *Lower Product Differentiation*

So far, we have shown that PDP tends to reduce both consumer surplus and platform payoffs. We now show that when product differentiation is lower, PDP can have positive effects on these outcomes.

Here, we set $\mu = 1/20$, corresponding to lower product differentiation. Absent PDP ($\gamma = 0$), there are two Bertrand–Nash equilibria on the pricing grid (both firms charge either 1.079 or 1.157) while the fully collusive price is 1.864, and the AI share-weighted prices are 1.748.

Figure 3 presents the effects of PDP on prices (left panel), consumer surplus (middle panel), and total output (right panel). AI prices are highest at $\gamma = 0.3$, and then tend to decrease for the remaining range of γ. Moving from $\gamma = 0$ to $\gamma = 1$, prices decline by 7.6%.

The effect of PDP on consumer surplus is large and positive. Moving from $\gamma = 0$ to $\gamma = 1$ increases consumer surplus by 34.4%. Even at moderate values of γ, such as $\gamma = 0.7$, there is a 20.5% increase in consumer surplus. Intuitively, because $\mu = 1/20$ is lower than in our default specification, consumers place less weight on product variety. Hence, the AI price decrease more than compensates the variety loss, benefiting consumers.

In contrast, total output goes down for all but a few values of γ. This is not inconsistent with consumer surplus generally going up, however. To understand why, first note that there is randomness in the strategy profiles that the algorithms converge to, and that we are averaging over 1000 separate runs.[24] Second, consumer surplus is convex in output, and highly so when total quantity is close to one as is the case with $\mu = 1/20$.

Total output is slightly higher at $\gamma = 0.98$ and $\gamma = 0.99$ compared to no intervention. Moreover, for other high levels of γ, output is approximately equal to that with $\gamma = 0$. This means that when γ is high, PDP either slightly increases or else has little effect on the platform's total per-unit commissions. Since the platform's payoff depends on both its total commissions and consumer surplus (when $\omega < 1$), overall platform payoffs Ω can increase with PDP.

3.4. *Benchmark: Randomly Displaying Firms*

By design, PDP does two things. First, it limits how many products are displayed to consumers, and second, it displays products that have lower prices. To isolate the effect

[24]As is common in MARL settings, there is randomness in our outcomes both because the algorithms experiment, and because the presence of multiple players leads to a nonstationary environment.

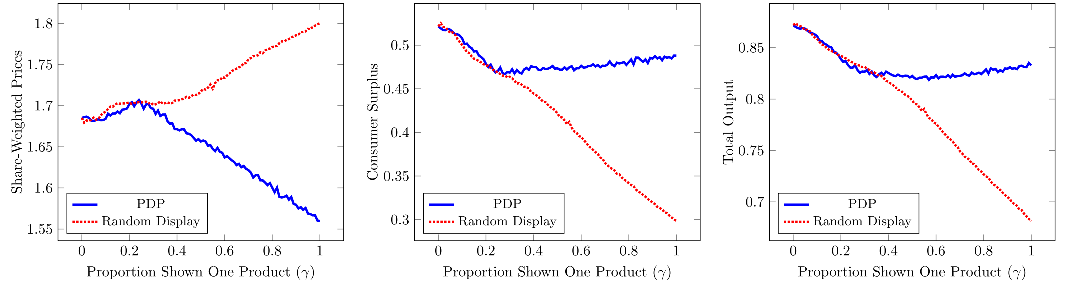


FIGURE 4.—A comparison of the effects of a benchmark of randomly displaying a firm on prices (left panel), consumer surplus (middle panel), and total output (right panel), compared to PDP.

of the first one, it is interesting to consider a benchmark where the number of displayed products is limited, but which products are displayed is entirely random. Intuitively, this benchmark should lead to higher prices, lower consumer surplus, and lower platform payoffs compared to PDP. This is because a firm's price no longer influences how many consumers are displayed its product, and so its demand is less price sensitive compared to PDP.

To investigate this, we consider a policy that displays one randomly chosen firm to γ of the consumers. Figure 4 shows that this random policy indeed leads to generally higher prices and lower consumer surplus and total output, compared to PDP. At high levels of γ, as one might expect, under the random policy the algorithms behave similarly to a single-product monopolist (which would charge 1.786, and generate consumer surplus of 0.303 and total output of 0.702). Moving from $\gamma = 0$ to $\gamma = 1$, under this random policy prices increase by 6.8% whereas they decrease by 7.4% under regular PDP. Similarly, consumer surplus and output decline by 42.8% and 21.8%, respectively, under the random policy, compared to 6.4% and 4.5% under regular PDP.

Overall, these results highlight the importance of using platform-design rules that condition on low prices as a way to increase competition, and hence mitigate the variety loss. Nevertheless, under our default specification PDP still reduces platform and consumer payoffs. We therefore now turn to our Dynamic PDP policy.

4. DYNAMIC PDP: SIMULATION RESULTS

Although our simulations with PDP show some success in lowering prices, AI prices remain high and consumers and the platform are harmed, at least in our default specification. Therefore, we now implement our policy of Dynamic PDP in our default specification, recalling that Propositions 5 and 6 predict that this policy can lower prices enough to benefit both consumers and the platform, even when firms are very patient and the market is cartelized.

We will show that the performance of Dynamic PDP depends on the details of the algorithms' state space. To begin with, we maintain the same state space as in our investigation of PDP—each algorithm tracks only the prices from the previous period. In Section 4.3 below, we expand this state space so that the algorithms can also track whichever firm was preferentially displayed in the previous period.

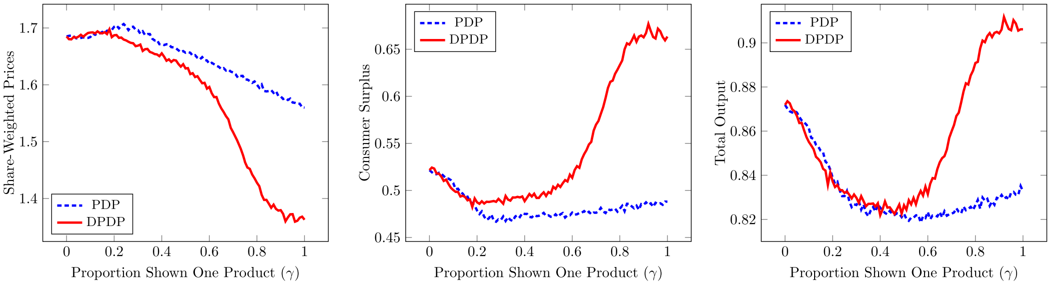


FIGURE 5.—The effect of Dynamic PDP on prices (left panel), consumer surplus (middle panel), and total output (right panel), with ADV = 0.3.

4.1. *Default Specification*

Our default specification uses the same demand and cost parameters as for regular PDP (as detailed in Section 3.1), sets ADV = 0.3, and assumes that the algorithms track only last-period prices. Figure 5 displays the effect of Dynamic PDP on prices (left panel), consumer surplus (middle panel), and total output (right panel). Each panel also displays the effect of regular PDP (as presented earlier in Figure 1).

Figure 5 shows that Dynamic PDP has a large effect on prices, consumer surplus, and total output. The effect is particularly strong at $\gamma = 1$. Comparing that level to $\gamma = 0$, prices are about 19.1% lower, consumer surplus is 27.2% higher, and total output is 3.9% higher. The decline in prices is approximately two and a half times the decline under regular PDP. The fact that consumer surplus and total output both increase is particularly notable given that they decrease under regular PDP for this specification.

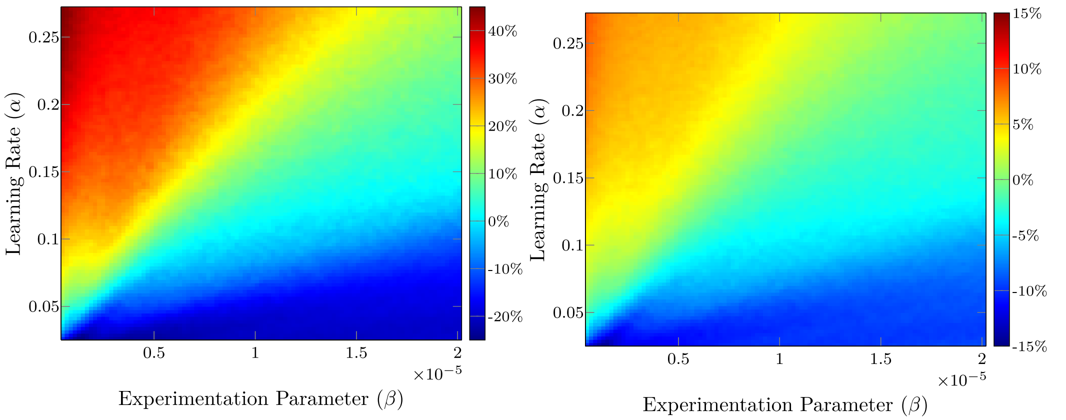
Because the platform's payoff is taken to be a weighted sum of consumer surplus and total per-unit commissions, it benefits from implementing Dynamic PDP for high values of γ.

4.2. *Learning-Parameter Robustness*

The results reported above suggest that Dynamic PDP may increase consumer surplus and platform payoffs, even when (regular) PDP would decrease them. To explore the robustness of this conclusion, here we vary the learning parameters. Extending our approach from Section 3.2, we compare consumer surplus and total output absent any platform intervention to the case of Dynamic PDP with $\gamma = 0.7$.

Figure 6 presents heatmaps of the percentage changes in consumer surplus (left panel) and total output (right panel) from adopting DPDP across a range of hyperparameters. Overall, Dynamic PDP appears to perform well, often benefiting consumers and the platform. Specifically, consumer surplus increases in 69.5% of the (α, β) pairs; this is a significant difference compared to the 13.8% of cases where consumer surplus increased with regular PDP. Similarly, total output increases in 39.8% of the (α, β) pairs; this is again a significant difference compared to the 1.7% of cases where total output increased with regular PDP. Moreover, when consumer surplus and total output do increase, the increases can be substantial.

As was the case for regular PDP, we see from the heatmaps that Dynamic PDP performs best in the region where α is large and β is small (top left of the heatmap), and worst when α is small and β is large (bottom right). We also find the same qualitative differences (across the regions) in how the algorithms adjust their behavior in response to Dynamic



(a) Consumer Surplus (b) Total Output

FIGURE 6.—Heatmaps of the effects of Dynamic PDP on consumer surplus and total output, for $\gamma = 0.7$ and ADV $= 0.3$. Consumer surplus increases in 69.5% of the cases, while total output increases in 39.8% of cases.

PDP that we found with regular PDP. That is, prices before imposing DPDP are higher in the top left region than in the bottom right, but after imposing DPDP they are lower in the top left region than in the bottom right.[25]

4.3. *Smarter AI: Augmented State Space*

In our theoretical analysis of DPDP, we assumed that sellers track the identity of the firm with the advantage. However, so far, we have not allowed the algorithms to do this, although such information is payoff-relevant.[26] Following our discussion in Section 1.3, we now allow our algorithms to track the identity of the firm with the pricing advantage ("Smarter AI").

Augmenting the state space in this manner has a substantial effect on outcomes, as shown in Figure 7. Consumer surplus and total output are both highest at $\gamma = 0.61$. For consumer surplus, this represents a 54.8% increase compared to no PDP, and a 19.3% increase compared to the best performance of Dynamic PDP without the extended algorithm (at $\gamma = 0.92$). Similarly, for total output it represents a 10% increase compared to no PDP, and a 5.2% increase compared to the best performance of Dynamic PDP without the extended algorithm (again at $\gamma = 0.92$).[27] (Recall that in our default specification regular PDP decreases both consumer surplus and total output.)

[25]We also find that, in each of these two regions, Dynamic PDP affects how the algorithms split profits and vary prices across periods in much the same way as regular PDP did (see footnote 23).

[26]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.

[27]Note that when $\gamma = 0$, Smarter AI delivers different prices (and consumer surplus and total output) than PDP or DPDP do, because adding an additional (payoff-irrelevant) element to the state space influences how the algorithms learn. For $\gamma > 0$, this additional state space element becomes payoff-relevant for DPDP but not for PDP. Consistent with this, as a robustness check we investigated the effect of augmenting PDP across the entire range of γ, and indeed find that its effect on outcomes is much less substantial than when DPDP is augmented.

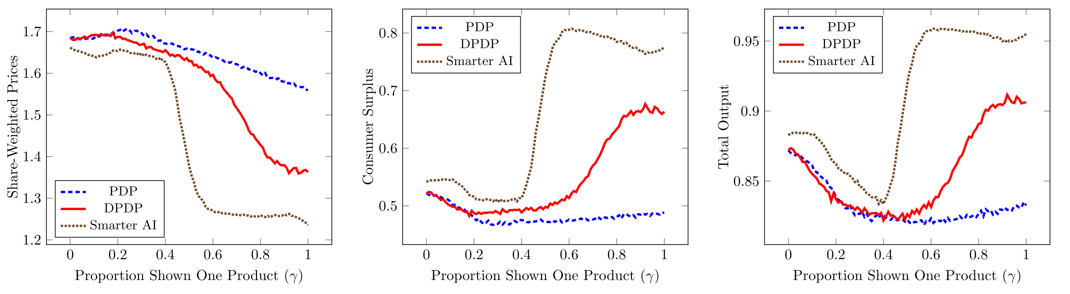


FIGURE 7.—The effect of a Smarter AI that includes in its state space the identity of the firm with the pricing advantage, on prices (left panel), consumer surplus (middle panel), and total output (right panel), with ADV = 0.3.

4.4. *Different Levels of the "Pricing Advantage"*

Here, we explore how the level of the pricing advantage ADV affects outcomes. We consider values of ADV > 0 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 and the platform. Indeed, theory predicts marginal cost pricing for our default specification with ADV = 0.3.

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 even when ADV is very small, Dynamic PDP is expected to change market outcomes given that 70% of consumers are shown only one product.

Figure 8 shows how consumer surplus is affected by changes in ADV. Two main outcomes are apparent. First, consumer surplus increases initially as ADV rises from zero, then typically declines as ADV grows larger, and finally levels off. In all cases, both total output and prices follow a nearly identical pattern (although inverted for prices). A second observation is that consumer surplus is a step function. Moreover, closer inspection of the data reveals that the jumps in consumer surplus occur when ADV is a multiple of the increments of the pricing grid. To see why this makes sense, note that generically a small change in ADV has no effect on which firm is displayed, and hence no effect on the algorithms' behavior. However, when ADV increases from below to above a multiple of

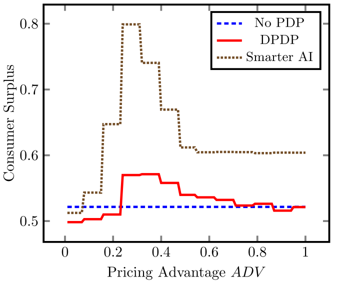


FIGURE 8.—The effect of ADV on Dynamic PDP performance (consumer surplus) with $\gamma = 0.7$, compared to no PDP.

the increments of the pricing grid, for given prices there can now be a change in which firm is displayed, and hence the algorithms' behavior is affected.[28]

Turning to platform payoffs, both consumers and the platform can benefit for a wide range of ADV values. For example, consider Smarter AI. Consumer surplus is up for all ADV values above 0.08; we also find that total output is up for all ADV values in excess of 0.16.

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.

Looking at the converged behavior of the algorithms for each of these four policies, sometimes the algorithms each set a constant price over time, and otherwise their prices follow a cycle.

Table II provides an overview. For each policy, the table reports the frequency of outcomes in which each algorithm sets a constant price. Additionally, when prices are constant, we report the frequency with which the firms charge symmetric (i.e., identical) prices. When instead there is a price cycle, we report the average length of the cycle. Finally, we also 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 prices are constant or follow a cycle.

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 by 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

TABLE II

PRICING AND PROFIT PATTERNS WITHOUT INTERVENTION, AND FOR PDP, DYNAMIC PDP, AND SMARTER AI (EACH AT $\gamma = 0.7$). NOTE: CYCLES OCCUR WHENEVER CONVERGED PRICES ARE NOT 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.

	No Intervention	PDP	DPDP	Smarter AI
% Outcomes with Constant Prices	74.2%	6.0%	14.7%	86.3%
Outcomes with Constant Prices				
% Symmetric	81.8%	100%	0%	0.1%
Industry Profit	0.596	0.552	0.243	0.244
High-Profit Firm's Share	50.9%	50.1%	83.6%	83.6%
Outcomes with a Price Cycle				
Average Cycle Length	2.089	3.265	4.975	2.095
Industry Profit	0.591	0.490	0.451	0.284
High-Profit Firm's Share	51.8%	52.6%	54.6%	84.4%

[28]To illustrate, suppose that the firm with the pricing advantage has not raised its price, and is charging one pricing increment (0.079) more than its rival. For ADV up to 0.07, the rival is always displayed, but if ADV increases by 0.01 (as in our simulations) to 0.08, then suddenly the rival is not displayed.

which there is substantial profit asymmetry correspond to Smarter AI, and to the outcomes under DPDP which feature constant prices. Given that our platform interventions by design push toward 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 constant-price outcomes under DPDP.

We now turn to the details. Begin by considering the outcomes when there is no platform design. Of these simulations, 74.2% of outcomes exhibit constant prices, 81.8% of which also 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 only slightly more than 50% of industry profits.

Moving from no platform design to PDP, the first notable change is a substantial decline in the frequency of outcomes with a constant price (6.0% rather than 74.2%). However, we now see that 100% of outcomes with a constant price are also symmetric. This contributes to the symmetry of profits, given that even small price asymmetries would result in highly asymmetric profits under PDP.[29] The decreased frequency of outcomes with constant prices necessarily implies an increase in the frequency of price cycles. 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 prices are constant over time.

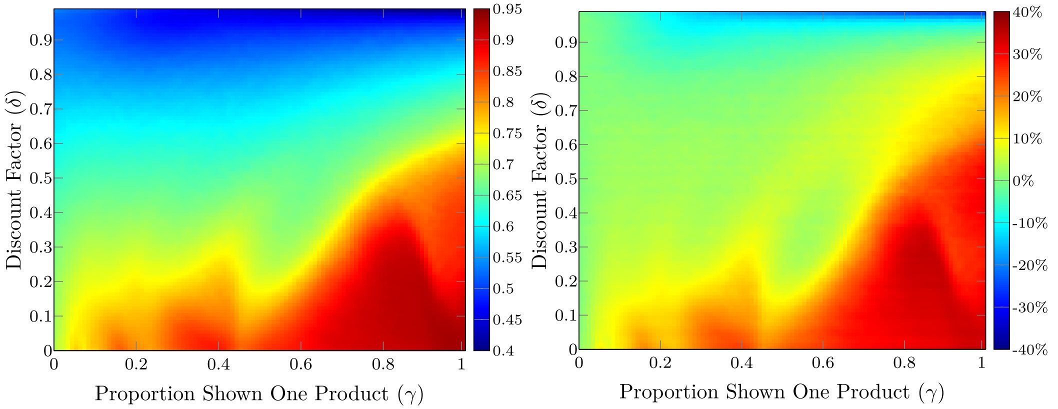
Moving from PDP to DPDP, the frequency of cycles decreases slightly (from 94% to 85.3%). One notable change is the increase in the average cycle length (4.975 rather than 3.265). Also notable is that under DPDP profit outcomes are qualitatively different depending on whether prices are constant or follow a cycle. In particular, when each firm sets a constant price, profits are highly asymmetric with the high-profit firm receiving 83.6% of long-run industry profits (versus 54.6% when there is a price cycle). 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 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 also much higher when firms rotate demand and split profits in this way (0.451 versus 0.243).

Finally, moving from DPDP to Smarter AI, the most notable change is a decrease in both the frequency of cycles (13.7% versus 85.3%) and their average length (2.095 versus 4.975). Moreover, unlike with DPDP, profits remain low and highly asymmetric irrespective of whether prices are constant or follow a cycle; firms do not rotate demand, and instead one of them prices so as to keep the advantage over time.

6. LESS PATIENT FIRMS

So far in our simulations, 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 discount factor can substantially influence market outcomes and the impact of our platform design policies.

[29] In 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.



(a) PDP and δ, Raw Values

(b) PDP and δ, Percentage Change

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

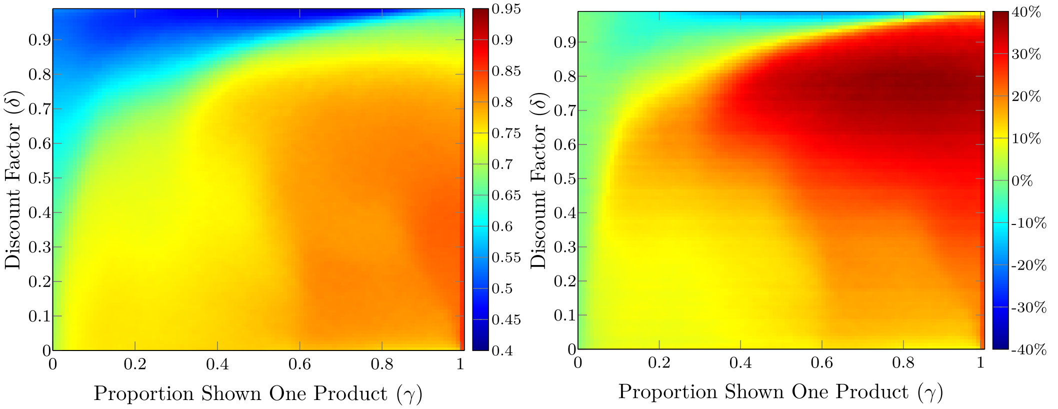
Here, we explore how varying the discount factor δ affects the performance of PDP and Dynamic PDP. To do this, we consider δ in increments of 0.01 over the interval $[0, 0.99]$, and for each value of δ record consumer surplus and total output as γ varies in increments of 0.01 on $[0, 0.99]$. We obtain qualitatively similar results for both outcomes; for brevity, this section therefore focuses on consumer surplus.

Beginning with PDP, we present the results of these simulations 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 decreases as δ becomes larger. This suggests that our algorithms are able to maintain higher prices when the discount factor δ is higher—and so behave in line with the usual intuition from models of collusion.

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 in percentage terms.

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 for all values of γ exceeding 0.9 except for $\gamma = 0.95$, while at $\delta = 0.85$ consumer surplus is up for all $\gamma \geq 0.67$. Thus, our finding in Section 3 that PDP lowers consumer surplus (when $\mu = 1/4$) requires that δ be fairly high. This is broadly consistent with our theoretical results from Section 1—with PDP benefiting consumers in competitive markets but harming them in cartelized markets.

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 each δ value considered, there exist values of γ such that DPDP raises consumer surplus. This tendency toward higher consumer surplus is again broadly consistent with the results from our earlier theoretical analysis showing that Dynamic PDP performs well even for very high discount factors. We note that the percentage increases are not nec-



(a) Dynamic PDP and δ, Raw Values

(b) Dynamic PDP and δ, Percentage Change

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

essarily 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.

7. EXTENSIONS

We now extend our simulation results to consider some alternative market environments, platform design choices, and business models.

7.1. *Stochastic Demand*

So far, we have considered a setting with deterministic (and constant) aggregate demand. However, when aggregate demand is stochastic, Calvano et al. (2020) show that the algorithms tend to achieve less of the potential collusive profit gains. This raises the question of whether our platform-design interventions become more or less effective as demand uncertainty increases.

Here, we explore this matter by allowing the demand parameter a to vary randomly over time. In particular, following Calvano et al. (2020), at the beginning of each period, a is uniformly randomly drawn from the set of values $\{2 - a_h, 2, 2 + a_h\}$. The algorithms do not observe this value before choosing their price.

We consider three values of a_h, namely $a_h = 0$ (no demand uncertainty) $a_h = 0.25$ (low demand uncertainty), and $a_h = 0.5$ (high demand uncertainty). For each value of a_h, we compute market outcomes (prices, consumer surplus, and output) for $\gamma = 0$ and $\gamma = 1$. We report in Table III the percentage changes going from the former to the latter.

We find that our policy interventions often perform better as uncertainty increases.[30] In particular, as uncertainty increases, the platform interventions become more effective in reducing prices and raising consumer surplus and output (although consumer surplus and output still decrease under PDP even at high uncertainty). For example, under Dynamic

[30]However, for Smarter AI the results are ambiguous: increasing demand uncertainty sometimes weakens its efficacy.

TABLE III

PERCENTAGE CHANGES IN MARKET OUTCOMES UNDER PDP AND DYNAMIC PDP, AT $\gamma = 1$ RELATIVE TO $\gamma = 0$ FOR DIFFERENT LEVELS OF DEMAND UNCERTAINTY.

	Percentage Change Relative to No Intervention		
	No Uncertainty	Low Uncertainty	High Uncertainty
Prices			
PDP	−7.4%	−7.8%	−9.1%
Dynamic PDP	−19.1%	−20.1%	−22.9%
Consumer Surplus			
PDP	−6.4%	−5.3%	−0.7%
Dynamic PDP	27.2%	28.9%	35.2%
Total Output			
PDP	−4.5%	−3.8%	−1.1%
Dynamic PDP	3.9%	6.0%	12.9%

PDP, going from $\gamma = 0$ to $\gamma = 1$ raises consumer surplus by respectively 27.2%, 28.9%, and 35.2% under no uncertainty, low uncertainty, and high uncertainty.

7.2. *Alternative Platform Intervention: A Richer Form of Dynamic PDP*

So far, we have considered a simple form of Dynamic PDP in which a firm that wins the advantage in some period can hold it all future periods, so long as it does not raise its price and is not substantially undercut by its rivals. Here, we explore a richer form of Dynamic PDP in which occasionally a firm exogenously loses the advantage.

In particular, we now suppose that in each period there is a probability $1 - \tau \in [0, 1]$ that the advantage is taken away from whichever firm has it, and then reassigned to whichever firm has the lowest price in this period; with complementary probability τ, the firm displayed in the last period continues to be displayed provided it does not raise its price and is not undercut by more than ADV. When $\tau = 0$, this policy corresponds to regular PDP, and when $\tau = 1$ it corresponds to our earlier version of Dynamic PDP.

Figure 11 illustrates the impact of τ for the case where the state space is not augmented to include the identity of the firm that was displayed last period. It does this for prices (left panel), consumer surplus (middle panel), and total output (right panel). A clear pattern emerges: as τ increases, and hence the firm with the advantage is more likely to keep

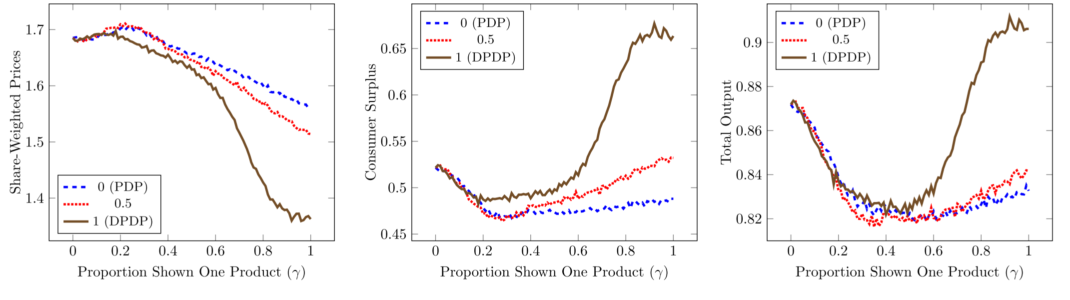
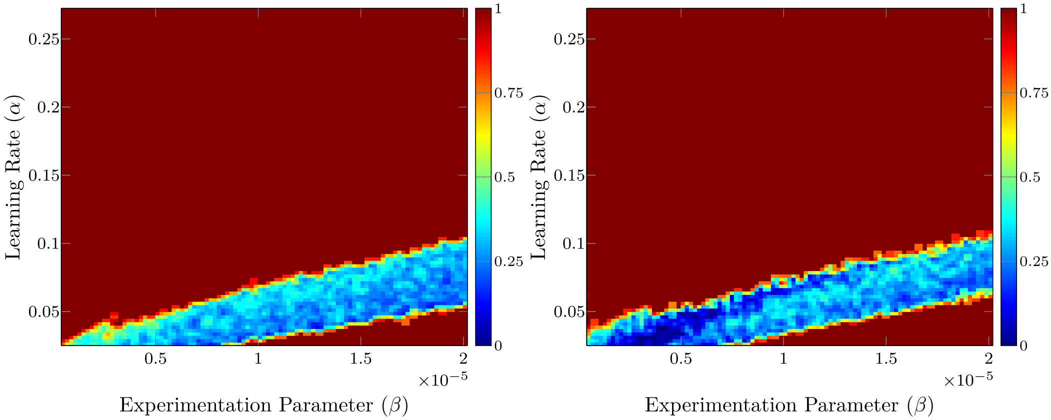


FIGURE 11.—The effect of Dynamic PDP on prices (left panel), consumer surplus (middle panel), and total output (right panel), with ADV = 0.3, and for different τ values. $\tau = 0$ corresponds to regular PDP, and $\tau = 1$ corresponds to the version of Dynamic PDP introduced earlier in the paper.



(a) Consumer Surplus (b) Total Output

FIGURE 12.—Heatmaps of the τ that maximizes consumer surplus (12a) and output (12b), under DPDP for $\gamma = 0.7$. The optimal τ value equals 1 in 86.2% of the cases for consumer surplus, and 85.9% of the cases for total output.

it (other things equal), prices tend to fall and both consumer surplus and total output tend to increase. Thus, to the extent that τ is a choice variable that the platform can optimize over, under our default specification the platform would like to make τ as large as possible.

To explore this further, we also examine how the τ that maximizes consumer surplus or output varies with the learning parameters α and β. Specifically, for each (α, β) pair, we allow τ to vary in increments of 0.1 on $[0, 1]$. We then select the values of τ that maximize either consumer surplus or output, and display them in the left and right panels, respectively, of Figure 12. We find that the optimal τ for both consumer surplus and total output (and hence platform commissions) equals one for around 86% of the learning parameters. However, there is a range of (α, β) pairs for which the optimal τ is at or close to zero; inspection of Figure 6 reveals that for these parameters DPDP (i.e., $\tau = 1$) performs poorly.

Finally, although we do not report formal results here, qualitatively similar patterns emerge when we perform the same exercise for Smarter AI. In particular, under our default specification, for a given γ value prices tend to be lower, and consumer surplus and total output higher, at larger values of τ. Moreover, we find that the optimal τ that maximizes consumer surplus or total output is equal to one for over 99% of the learning parameters.

7.3. *Revenue Sharing*

So far, we have focused on the case where the platform derives its commissions from per-unit fees. However, in practice platforms may also use revenue-sharing fees. 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

TABLE IV

THE PERFORMANCE OF PDP, DYNAMIC PDP, AND SMARTER AI UNDER REVENUE SHARING, FOR $\gamma = 0.7$ AND $\gamma = 1$. NOTE: THE FIRST TWO COLUMNS REPORT THE PERCENTAGE CHANGE IN REVENUE RELATIVE TO NO INTERVENTION. THE SECOND TWO COLUMNS REPORT THE CRITICAL WEIGHT ω SUCH THAT THE PLATFORM'S PAYOFF INCREASES SO LONG AS IT PUTS LESS THAN ω WEIGHT ON COMMISSIONS (OR, AT LEAST $1 - \omega$ WEIGHT ON CONSUMER SURPLUS), GIVEN IT TAKES A 20% SHARE OF REVENUES.

	Percentage Change in Revenue		Critical ω for Higher Platform Payoff	
	$\gamma = 0.7$	$\gamma = 1$	$\gamma = 0.7$	$\gamma = 1$
PDP	−10.3%	−12.6%	–	–
Dynamic PDP	−12.7%	−17.0%	0.564	0.740
Smarter AI	−17.7%	−19.5%	0.842	0.816

sellers, which (for Amazon) involve per-units fees.[31] Here, we consider simulation results, but we develop theoretical results for revenue sharing in the Online Appendix.

Total revenue falls under each of our platform design interventions, for each value of $\gamma > 0$. The reason is that even absent platform design, the prices set by the algorithms are already beneath the revenue-maximizing prices. Thus, because our platform interventions lower prices and decrease variety, they also reduce revenue.

The first two columns of Table IV report the percentage change in revenue (relative to no intervention) for $\gamma = 0.7$ and $\gamma = 1$. The effects are strongest at $\gamma = 1$, where total revenue falls by 12.6% under PDP, 17.0% under Dynamic PDP, and 19.5% under Smarter AI.

Nonetheless, platform payoffs can still increase. The reason is that our underlying assumption is that the platform's payoff is a weighted sum of its total commissions and consumer surplus. The last two columns of Table IV explore this. In particular, assuming the platform takes a 20% share of revenues, the table computes the critical weight ω such that the platform gains provided it places weight below ω on its commissions (or equivalently, more than weight $1 - \omega$ on consumer surplus). (Note that because PDP lowers both total revenues and consumer surplus, no such critical ω exists for this particular intervention.)

Because Dynamic PDP and Smarter AI raise consumer surplus, platform payoffs increase for many values of ω. For example, the final row of Table IV shows that for $\gamma = 0.7$, Smarter AI increases the platform's overall payoff provided $\omega \leq 0.842$, that is, if it puts weight of around 15% or more on consumer surplus.

7.4. *Three Firms*

Here, we provide results when $n = 3$. As with $n = 2$, our platform-design interventions tend to reduce prices but also reduce variety, presenting a tradeoff.

When a single firm is displayed to γ consumers, familiar results again emerge from our simulations. We see that regular PDP lowers prices but not enough to account for the decrease in variety; consumer surplus and output decrease for all γ values. Also, consistent with earlier results, we find that DPDP often increases both consumer surplus and output.

[31] Amazon Marketplace fees are described at `sell.amazon.com/pricing`, which also notes that fulfillment "includes picking and packing your orders, shipping and handling, customer service, and product returns.".

TABLE V

THE IMPACT ON PRICES, CONSUMER SURPLUS AND TOTAL OUTPUT, WHEN $n = 3$, $\gamma = 0.7$, AND THE NUMBER OF DISPLAYED FIRMS INCREASES FROM $k = 1$ TO $k = 2$.

	Percentage Difference From Displaying Two Firms Instead of One		
	Prices	Consumer Surplus	Total Output
PDP	3.6%	8.2%	4.8%
Dynamic PDP	23.5%	−20.2%	−5.7%
Smarter AI	0.8%	13.2%	1.4%

It does best for consumers at $\gamma = 0.63$, leading to a 15% gain relative to no intervention; the highest output gains occur at $\gamma = 0.71$, for a 2.3% increase. Similarly, Smarter AI performs very well, increasing consumer surplus by as much as 7.9% (at $\gamma = 0.28$) over the best performance of DPDP, and also outperforming in terms of output.

We also examined the effect of displaying two firms rather than one to γ consumers. This increases the overall variety available to those γ consumers, but intuitively may lead to higher prices because firms now need to compete less fiercely in order to be displayed. Indeed, this is what we find: for each of our platform-design interventions, displaying an additional firm raises prices for almost all γ values. Table V presents details for the case of $\gamma = 0.7$. For example, under PDP, displaying a second firm leads to a 3.6% increase in prices.

In terms of the overall effect of higher prices but more variety, we find mixed results. In particular, both PDP and Smarter AI perform better in terms of consumer surplus and output when two firms are displayed, whereas the opposite is true for DPDP. This is shown in the second and third columns of Table V. For example, under Smarter AI displaying a second firm increases consumer surplus by 13.2% and total output by 1.4%.[32]

8. CONCLUSION

In this article, we examined the intersection of several important trends: the growing "regulatory power" of platforms to manage interactions between buyers and sellers, the increased use of pricing algorithms by sellers, and concerns about the potential anticompetitive role of algorithms. Using both theory and simulations, we showed that platform rules that steer demand can increase competition between sellers, and simultaneously raise consumer surplus and the platform's payoff. Sometimes rules that condition on past prices and treat sellers in a nonneutral fashion may be required. These rules have a low informational burden, and can perform well relative to arbitrarily sophisticated policies.

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. Similarly, if firms are asymmetric, with one firm having slightly lower costs but dramatically lower quality, then preferentially displaying the firm with lower costs could harm consumers. When such concerns are important, the platform might wish to base

[32]Although Table V considers $\gamma = 0.7$, the same qualitative results are true even when the best-performing γ values are separately chosen for each policy intervention, and contingent on whether one or two firms are being displayed.

display decisions on some measure of quality-normalized prices. We have also not considered the entry decision of sellers; if seller competition is too intense on the platform, and the platform is a large source of revenue for the sellers, then entry may be restricted, possibly harming both consumers and the platform. When seller entry is indeed a concern for the platform, the platform may prefer less aggressive demand-steering. In our simulations, we have allowed for this prospect, by supposing that only some consumers may have their choice set restricted; this attenuates the competitive pressure on sellers. Finally, our analysis does not consider the case where the platform is itself a seller, and hence may have an incentive to self-preference.

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 simulations it leads to very low but above-cost prices. And of course we focused on a particular type of reinforcement-learning algorithm, which is not designed explicitly to foster collusion but instead to learn about the market environment. In future work, it would be interesting to investigate platform design when algorithms are explicitly designed to collude.

Despite such limitations, our work is proof of concept that a platform can design rules that shape competition on its marketplace, both benefiting consumers and raising its own payoff, even when sellers use pricing algorithms.

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 $p_1 \leq p_2 \leq \cdots \leq p_n$ (for simplicity, we drop time superscripts). Clearly, in a Nash equilibrium, $p_1 \geq c$. We now prove that in any Nash equilibrium $p_{k+1} = c$.

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 $p_{k+1} = c$. Finally, it is straightforward to verify that any prices satisfying $p_1 = p_2 = \cdots = p_m = c$ for $m \geq k+1$ constitute a Nash equilibrium. *Q.E.D.*

We use the following lemma in some subsequent proofs.

LEMMA 3: *Suppose all n firms are shown and they price competitively. The Bertrand–Nash equilibrium price* p^*_{BN} *is the unique solution to*

$$\frac{p^*_{\mathrm{BN}}-c}{\mu}-\frac{n\exp\left(\frac{a-p^*_{\mathrm{BN}}}{\mu}\right)+1}{(n-1)\exp\left(\frac{a-p^*_{\mathrm{BN}}}{\mu}\right)+1}=0, \tag{5}$$

and it satisfies $\mu < p^*_{\mathrm{BN}}-c<\mu n/(n-1)$.

PROOF OF LEMMA 3: Dropping time superscripts, if firm i charges a price p_i its profit is

$$(p_i-c)D_i(p)=(p_i-c)\frac{\exp\left(\frac{a-p_i}{\mu}\right)}{\sum_{j=1}^{n}\exp\left(\frac{a-p_j}{\mu}\right)+1}. \tag{6}$$

The derivative of firm i's profit with respect to p_i can be written as

$$D_i(p)\left\{1-\left(\frac{p_i-c}{\mu}\right)\left[1-D_i(p)\right]\right\}. \tag{7}$$

Bertrand–Nash equilibrium requires that equation (7) equal zero for each $i=1,\dots,n$, which is only possible if $p_1=p_2=\cdots=p_n=p^*_{\mathrm{BN}}$. Substituting these prices into equation (7) and setting it equal to zero, then gives equation (5). The left-hand side of equation (5) is strictly increasing in p^*_{BN}. This implies that a unique p^*_{BN} solves equation (5). The bounds on $p^*_{\mathrm{BN}}-c$ stated in the lemma then follow because the left-hand side of equation (5) is strictly negative at $p^*_{\mathrm{BN}}=c+\mu$ and strictly positive at $p^*_{\mathrm{BN}}=c+\mu n/(n-1)$. *Q.E.D.*

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

$$\mu\log\left\{k\exp\left(\frac{a-c}{\mu}\right)+1\right\}. \tag{8}$$

Suppose instead that all n firms are shown to consumers. Firms play the Bertrand–Nash price p^*_{BN} from Lemma 3. Using equation (2), consumer surplus is

$$\mu\log\left\{n\exp\left(\frac{a-p^*_{\mathrm{BN}}}{\mu}\right)+1\right\}. \tag{9}$$

PDP strictly increases consumer surplus if and only if (8) strictly exceeds (9), or

$$\frac{k}{n}>\exp\left(-\frac{p^*_{\mathrm{BN}}-c}{\mu}\right)=\lambda. \tag{10}$$

Since $U(p)=-\mu\log(1-Q(p))$, PDP strictly increases total output if and only if (10) holds. Hence, PDP strictly raises the platform's payoff if and only if (10) holds. Finally, note that Lemma 3 implies that $e^{-\frac{n}{n-1}}<\lambda<e^{-1}$. *Q.E.D.*

PROOF OF PROPOSITION 2: Let Q^{Max} denote total output when all n firms are displayed and price at c. Similarly, let Q^{PDP} and Q^{BN} denote total output under respectively PDP and Bertrand–Nash. Throughout this proof, let $X \equiv \exp(\frac{a-c}{\mu})$.

Start with the first inequality in the proposition. Using the expressions for Q^{PDP} and Q^{Max} and simplifying, we find that

$$Q^{\text{PDP}} - \left(1 - \frac{1}{n}\right)Q^{\text{Max}} = \frac{(n-1)X^2}{\big[(n-1)X+1\big][nX+1]} \geq 0. \tag{11}$$

Similarly, we find that

$$U^{\text{PDP}} - \left(1 - \frac{1}{n}\right)U^{\text{Max}} = \mu \log\big[(n-1)X+1\big] - \left(1 - \frac{1}{n}\right)\mu \log[nX+1] \geq 0, \tag{12}$$

where the inequality follows because the expression preceding it equals zero at $X = 0$ and is strictly increasing in X. Given that $U^{\text{Max}} > 0$, we then have that $U^{\text{PDP}}/U^{\text{Max}} > 1 - (1/n)$. Moreover, for $\omega < 1$ or $f > 0$ (or both), (11) and (12) imply that

$$\frac{\Omega^{\text{PDP}}}{\Omega^{\text{Max}}} = \frac{\omega f Q^{\text{PDP}} + (1-\omega)U^{\text{PDP}}}{\omega f Q^{\text{Max}} + (1-\omega)U^{\text{Max}}} \geq 1 - \frac{1}{n}.$$

Now consider the second inequality in the proposition. Note that

$$\big(Q^{\text{PDP}} - Q^{\text{BN}}\big) - \left[1 - \frac{e}{n(e-1)}\right]\big(Q^{\text{Max}} - Q^{\text{BN}}\big) \tag{13}$$

is decreasing in Q^{BN}. Also, recall from Lemma 3 that $p_{\text{BN}} > c + \mu$, which implies that

$$Q^{\text{BN}} \leq Q^{\text{BN}}_{\dagger} = \frac{nX\exp(-1)}{nX\exp(-1)+1}. \tag{14}$$

Hence, we can write

$$\begin{aligned} &\big(Q^{\text{PDP}} - Q^{\text{BN}}\big) - \left[1 - \frac{e}{n(e-1)}\right]\big(Q^{\text{Max}} - Q^{\text{BN}}\big) \\ &\quad \geq \big(Q^{\text{PDP}} - Q^{\text{BN}}_{\dagger}\big) - \left[1 - \frac{e}{n(e-1)}\right]\big(Q^{\text{Max}} - Q^{\text{BN}}_{\dagger}\big) \geq 0, \end{aligned} \tag{15}$$

where the final inequality is obtained by substituting in the expressions for Q^{PDP}, Q^{Max}, and $Q^{\text{BN}}_{\dagger}$ and then simplifying. Note also that

$$\big(U^{\text{PDP}} - U^{\text{BN}}\big) - \left[1 - \frac{e}{n(e-1)}\right]\big(U^{\text{Max}} - U^{\text{BN}}\big) \tag{16}$$

is decreasing in U^{BN}. Recalling again from Lemma 3 that $p^*_{\text{BN}} > c + \mu$, we have that

$$U^{\text{BN}} \leq \mu \log\big[nX\exp(-1)+1\big]. \tag{17}$$

Combining (16) and (17), substituting in the expressions for U^{PDP} and U^{Max}, we obtain

$$\begin{aligned}&\big(U^{\text{PDP}}-U^{\text{BN}}\big)-\left[1-\frac{e}{n(e-1)}\right]\big(U^{\text{Max}}-U^{\text{BN}}\big)\\&\quad\geq\mu\log\left[\frac{(n-1)X+1}{nX\exp(-1)+1}\right]-\left[1-\frac{e}{n(e-1)}\right]\mu\log\left[\frac{nX+1}{nX\exp(-1)+1}\right]\geq 0,\end{aligned}\tag{18}$$

where the final inequality follows because the expression preceding it equals zero at $X=0$ and is strictly increasing in X. Given that $U^{\text{Max}}-U^{\text{BN}}>0$, we have $(U^{\text{PDP}}-U^{\text{BN}})/(U^{\text{Max}}-U^{\text{BN}})\geq[1-\frac{e}{n(e-1)}]$. Moreover, for $\omega<1$ or $f>0$ (or both), (15) and (18) imply

$$\begin{aligned}\frac{\Omega^{\text{PDP}}-\Omega^{\text{BN}}}{\Omega^{\text{Max}}-\Omega^{\text{BN}}}&=\frac{\omega f\big(Q^{\text{PDP}}-Q^{\text{BN}}\big)+(1-\omega)\big(U^{\text{PDP}}-U^{\text{BN}}\big)}{\omega f\big(Q^{\text{Max}}-Q^{\text{BN}}\big)+(1-\omega)\big(U^{\text{Max}}-U^{\text{BN}}\big)}\\&\geq 1-\frac{e}{n(e-1)}.\end{aligned}$$

Q.E.D.

PROOF OF LEMMA 2: Dropping time superscripts and labeling products such that $p_1\leq p_2\leq\cdots\leq p_n$, a monopolist's profit in any given period is

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

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

$$D_i(p)\left[1-\frac{p_i-c}{\mu}+\frac{\displaystyle\sum_{j=1}^{k}(p_j-c)D_j(p)}{\mu}\right].\tag{19}$$

At the optimum, (19) should equal zero for each $i=1,\dots,k$, requiring $p_1=p_2=\cdots=p_k$. Substituting $p_1=\cdots=p_k=p^m(k)$ in equation (19) and setting it to zero, $p^m(k)$ satisfies

$$k\exp\left(\frac{a-p^m(k)}{\mu}\right)+1-\left(\frac{p^m(k)-c}{\mu}\right)=0.\tag{20}$$

The left-hand side of (20) is strictly decreasing in $p^m(k)$ and so there is a unique optimum. We note for future reference that, because the left-hand side of (20) is also strictly increasing in k, $p^m(k)$ is strictly increasing in k. *Q.E.D.*

PROOF OF PROPOSITION 3: 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 $k<n$ 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

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

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

$$\frac{\pi^m(k)}{1-\delta}\geq n\max_p (p-c)\tilde{D}(p,k). \tag{21}$$

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, (21) 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 (21) does not hold at least one firm prefers to deviate from full collusion.

Next, let $\widehat{\delta}_k$ be the unique δ such that (21) holds with equality. To prove that $\widehat{\delta}_1 = 1-1/n$, recall that $\arg\max_p (p-c)\tilde{D}(p,1)=p^m(1)$, which implies that $\max_p(p-c)\tilde{D}(p,1)=\pi^m(1)$.

To prove that $\widehat{\delta}_1>\widehat{\delta}_2>\cdots>\widehat{\delta}_{n-1}$, first note that $\pi^m(k)$ is strictly increasing in k. Hence, it is sufficient to show that $\max_p(p-c)\tilde{D}(p,k)$ is decreasing in k. To do this, first rewrite the multiproduct firm's first-order condition in equation (20) as

$$(k-1)\exp\left(\frac{a-p^m(k)}{\mu}\right)+1=\left(\frac{p^m(k)-c}{\mu}\right)-\exp\left(\frac{a-p^m(k)}{\mu}\right). \tag{22}$$

We know from the proof of Lemma 2 that $p^m(k)$ strictly increases in k, and so the right-hand side of (22) strictly increases in k. It then follows from (22) that $(k-1)\exp(\frac{a-p^m(k)}{\mu})$ also strictly increases in k. This further implies that $\tilde{D}(p,k)$ strictly decreases in k because

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

It is then straightforward to argue that $\max_p(p-c)\tilde{D}(p,k)$ decreases in k. *Q.E.D.*

PROOF OF PROPOSITION 4: The proof of Lemma 2 derived an implicit expression for $p^m(k)$, and showed that $p^m(k)$ is strictly increasing in k. Consumer surplus from PDP when prices are determined by a monopolist can, using equations (2) and (20), be written as

$$\mu \log\left\{\frac{p^m(k) - c}{\mu}\right\}. \tag{23}$$

Consumer surplus therefore increases in k because $p^m(k)$ increases in k. Total output also increases in k because $Q(p)$ satisfies $U(p) = -\mu \log(1 - Q(p))$. Hence, the platform's payoff also increases in k. *Q.E.D.*

PROOF OF PROPOSITION 5: Consider a pure-strategy subgame perfect Nash equilibrium (SPNE), and let $\widehat{p}_t$ denote the price of the product displayed to consumers in period t along the equilibrium path. Let $\tilde{\pi}(p)$ denote per-period profit of a firm that charges p and is the only firm shown to consumers. Note that $\tilde{\pi}(p)$ is strictly increasing in p up to $p^m(1)$. Let ADV* be the unique x which solves $\tilde{\pi}(x) = \pi^m(1)/n$. Also, define $\widehat{\delta}$ as

$$\widehat{\delta} = \begin{cases} \left(1 - \dfrac{1}{n}\right)\dfrac{\pi^m(1)}{\pi^m(1) - \tilde{\pi}\big(\max\{c, \text{ADV}\}\big)} & \text{if ADV} < \text{ADV}^*, \\ 1 & \text{if ADV} \geq \text{ADV}^*. \end{cases}$$

Proposition 3 shows $\widehat{\delta}_1 = 1 - 1/n$, and hence $\widehat{\delta} \geq \widehat{\delta}_1$. Also, note that by construction $\widehat{\delta} \leq 1$.

We start with part (1) of the proposition. Assume for this part of the proof that $\delta < \widehat{\delta}$.

We first prove that $\widehat{p}_t \leq p^m(1)$ for all t. Toward a contradiction, suppose this is not true, and let t' denote the first period in which $\widehat{p}_t > p^m(1)$. Note that either $t' = 0$, or $t' > 0$ and the firm that was displayed last period has raised its price; hence, by the definition of DPDP, the display slot is allocated in period t' to a firm with the lowest price. Note also that in period t', along the equilibrium path all firms charge strictly more than $p^m(1)$. Hence, any firm could charge $p^m(1)$ in period t' and win the display slot for sure, and then (i) if ADV $< c$, charge c in all future periods, and (ii) if ADV $\geq c$, charge ADV in all future periods and keep the slot. Thus, in period t' the n firms' combined discounted payoff is at least

$$n\left[\pi^m(1) + \frac{\delta \tilde{\pi}\big(\max\{c, \text{ADV}\}\big)}{1 - \delta}\right]. \tag{24}$$

Because $\delta < \widehat{\delta}$, this strictly exceeds $\pi^m(1)/(1 - \delta)$. To show this, there are three cases to consider. First, if ADV $\leq c$, then $\widehat{\delta} = 1 - (1/n)$ while (24) simplifies to $n\pi^m(1)$ and the claim is immediate from $\delta < \widehat{\delta}$. Second, if $c <$ ADV $<$ ADV* then, using the definition of $\widehat{\delta}$, $\delta < \widehat{\delta}$ implies that

$$n\tilde{\pi}(\text{ADV}) > \pi^m(1)\frac{1 - n(1 - \delta)}{\delta}, \tag{25}$$

which implies that (24) strictly exceeds $\pi^m(1)/(1 - \delta)$. Third, if ADV $\geq$ ADV* then, because by assumption ADV $\leq p^m(1)$, we have $n\tilde{\pi}(\text{ADV}) \geq \pi^m(1)$ and it follows that for any $\delta < \widehat{\delta} = 1$ (24) strictly exceeds $\pi^m(1)/(1 - \delta)$.

We have shown that the firms' combined discounted payoffs given in (24) strictly exceed $\pi^m(1)/(1 - \delta)$, which is a contradiction. Hence, $\widehat{p}_t \leq p^m(1)$ for all t.

It also follows that the supremum $\overline{p} \leq p^m(1)$ over equilibrium transaction prices exists. We now prove that $\overline{p} = c$.

On the way to a contradiction, suppose that $\overline{p} \in (c, p^m(1)]$. For any given $\Delta \in (0, \overline{p} - c)$, let t'' be the first period t in which $\widehat{p}_t \in (\overline{p} - \Delta, \overline{p}]$. Note that either $t'' = 0$, or $t'' > 0$ and the firm that was displayed in the previous period has raised its price. Hence, following earlier arguments, by the definition of Dynamic PDP any firm could charge $\overline{p} - \Delta$ in period t'' and win the advantage, and then (i) if ADV $> c$, charge $\min\{\overline{p} - \Delta, \text{ADV}\}$ in all future periods and keep the advantage, and (ii) if ADV $\leq c$, 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

$$n\left[\tilde{\pi}(\overline{p} - \Delta) + \frac{\delta \tilde{\pi}\big(\min\{\overline{p} - \Delta, \max\{c, \text{ADV}\}\}\big)}{1 - \delta}\right]. \tag{26}$$

Because $\delta < \widehat{\delta}$, at $\Delta = 0$ this is strictly more than $\tilde{\pi}(\overline{p})/(1 - \delta)$. To show this, there are three cases to consider. First, if ADV $\leq c$, then $\widehat{\delta} = 1 - (1/n)$ while (26) simplifies to $n\tilde{\pi}(\overline{p})$ and the claim follows from $\delta < \widehat{\delta}$. Second, if $c < \overline{p} \leq$ ADV, then (26) simplifies to $n\tilde{\pi}(\overline{p})/(1 - \delta)$, which strictly exceeds $\tilde{\pi}(\overline{p})/(1 - \delta)$ for all $\delta < 1$. Third, consider $c <$ ADV $< \overline{p}$. If ADV $\geq$ ADV* then, by the definition of ADV* and the assumption that ADV $\leq p^m(1)$, $n\tilde{\pi}(\text{ADV}) \geq \pi^m(1) \geq \tilde{\pi}(\bar{p})$, from which it follows that (26) strictly exceeds $\tilde{\pi}(\overline{p})/(1 - \delta)$ for all $\delta < \widehat{\delta} = 1$. If instead ADV $<$ ADV*, then $\delta < \widehat{\delta}$ implies (25), which combined with the fact that $\pi^m(1) \geq \tilde{\pi}(\overline{p})$, can be used to show that (26) strictly exceeds $\tilde{\pi}(\overline{p})/(1 - \delta)$.

Hence, we have shown that (26) evaluated at $\Delta = 0$ strictly exceeds $\tilde{\pi}(\overline{p})/(1 - \delta)$. By continuity, this implies that, for $\widehat{p} \leq \overline{p}$ sufficiently close to $\overline{p}$, firms' combined payoffs strictly exceed $\tilde{\pi}(\overline{p})/(1 - \delta)$, a contradiction. We have therefore established that $\overline{p} \leq c$. But since $\overline{p} < c$ is impossible, it must be that $\overline{p} = c$.

We now construct a SPNE where firms charge c in each period. Consider the strategy:

(1) In period 0, all firms charge c.
(2) Suppose that the firm that was displayed in period t charged weakly less than c in period t. Then in period $t + 1$ all firms charge c.
(3) Suppose that the firm that was displayed in period t charged strictly more than c in period t. Then in period $t + 1$ that firm charges the minimum of $\min\{c + \text{ADV}, p^m(1)\}$ and its price from period t, while 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 product that is displayed to consumers has price c in every period.

Finally, to prove part (2) of the proposition, note that when ADV $\geq$ ADV* then $\widehat{\delta} = 1$. *Q.E.D.*

PROOF OF PROPOSITION 6: Following similar arguments as in the proof of Proposition 1, Dynamic PDP (with $k = 1$) raises consumer surplus and output (and hence also platform payoff) if and only if

$$\exp\left(\frac{p^m(n) - c}{\mu}\right) > n \quad \Longleftrightarrow \quad p^m(n) > \mu \log(n) + c. \tag{27}$$

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

$$n \exp\left(\frac{a - p^m(n)}{\mu}\right) + 1 - \left(\frac{p^m(n) - c}{\mu}\right) = 0. \tag{28}$$

The left-hand side is strictly decreasing in $p^m(n)$, so condition (27) is satisfied if and only if the left-hand side of (28) is strictly positive when evaluated at $p^m(n) = \mu \log(n) + c$, that is,

$$\exp\left(\frac{a-c}{\mu}\right) + 1 - \log(n) > 0 \quad \iff \quad n < \tilde{n}. \qquad Q.E.D.$$

REFERENCES

ASKER, JOHN, CHAIM FERSHTMAN, AND ARIEL PAKES (forthcoming): "The Impact of Artificial Intelligence Design on Pricing," *Journal of Economics & Management Strategy*. [1845]

AXELROD, ROBERT, AND WILLIAM D. HAMILTON (1981): "The Evolution of Cooperation," *Science*, 211 (4489), 1390–1396. [1845]

BROWN, ZACH Y., AND ALEXANDER MACKAY (2023): "Competition in Pricing Algorithms," *American Economic Journal: Microeconomics*, 15 (2), 109–156. [1842]

CALVANO, EMILIO, GIACOMO CALZOLARI, VINCENZO DENICOLÒ, AND SERGIO PASTORELLO (2020): "Artificial Intelligence, Algorithmic Pricing and Collusion," *American Economic Review*, 110 (10), 3267–3297. [1842-1844,1854-1856,1866]

CHEN, LE, ALAN MISLOVE, AND CHRISTO WILSON (2016): "An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace," in *Proceedings of the 25th International Conference on World Wide Web*, 1339–1349. [1842]

CLARK, R., S. ASSAD, D. ERSHOV, AND L. XU (forthcoming): "Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market," *Journal of Political Economy*. [1842]

CMA (2018): "Pricing Algorithms: Economic Working Paper on the Use of Algorithms to Facilitate Collusion and Personalised Pricing," Competition and Markets Authority. [1842]

CRÉMER, JACQUES, YVES-ALEXANDRE DE MONTJOYE, AND HEIKE SCHWEITZER (2019): "Competition Policy for the Digital Era," European Commission. [1841]

DAL BÓ, PEDRO, AND GUILLAUME R. FRÉCHETTE (2018): "On the Determinants of Cooperation in Infinitely Repeated Games: A Survey," *Journal of Economic Literature*, 56 (1), 60–114. [1845]

DANA, JAMES D. (2012): "Buyer Groups as Strategic Commitments," *Games and Economic Behavior*, 74 (2), 470–485. [1845,1851]

DE CORNIERE, ALEXANDRE, AND GREG TAYLOR (2019): "A Model of Biased Intermediation," *The RAND Journal of Economics*, 50 (4), 854–882. [1845]

DECK, CARY A., AND BART J. WILSON (2003): "Automated Pricing Rules in Electronic Posted Offer Markets," *Economic Inquiry*, 41 (2), 208–223. [1845]

DINERSTEIN, MICHAEL, LIRAN EINAV, JONATHAN LEVIN, AND NEEL SUNDARESAN (2018): "Consumer Price Search and Platform Design in Internet Commerce," *American Economic Review*, 108 (7), 1820–1859. [1845]

DOJ (2018): ""Life in the Fast Lane," Prepared Remarks by Assistant Attorney General Makan Delrahim,", US Department of Justice. [1842]

EZRACHI, ARIEL, AND MAURICE E. STUCKE (2017): "Artificial Intelligence & Collusion: When Computers Inhibit Competition," 1775–1810, University of Illinois Law Review. [1844]

GÓMEZ-LOSADA, ÁLVARO, GUALBERTO ASENCIO-CORTÉS, AND NÉSTOR DUCH-BROWN (2022): "Automatic Eligibility of Sellers in an Online Marketplace: A Case Study of Amazon Algorithm," *Information*, 13 (2), 44. [1842]

HAGIU, ANDREI, AND BRUNO JULLIEN (2011): "Why Do Intermediaries Divert Search?" *The RAND Journal of Economics*, 42 (2), 337–362. [1845]

HARRINGTON, JOSEPH E. (2018): "Developing Competition Law for Collusion by Autonomous Artificial Agents," *Journal of Competition Law & Economics*, 14 (3), 331–363. [1844]

INDERST, ROMAN, AND MARCO OTTAVIANI (2012): "Competition Through Commissions and Kickbacks," *American Economic Review*, 102 (2), 780–809. [1845]

JAAKKOLA, TOMMI, MICHAEL I. JORDAN, AND SATINDER P. SINGH (1994): "Convergence of Stochastic Iterative Dynamic Programming Algorithms," in *Advances in Neural Information Processing Systems*, 703–710. [1854]

JOHNSON, JUSTIN P., ANDREW RHODES, AND MATTHIJS WILDENBEEST (2023): "Supplement to 'Platform Design When Sellers Use Pricing Algorithms'," *Econometrica Supplemental Material*, 91, https://doi.org/10.3982/ECTA19978. [1846]

KLEIN, TIMO (2021): "Autonomous Algorithmic Collusion: Q-Learning Under Sequential Pricing," *The RAND Journal of Economics*, 52 (3), 538–558. [1842,1844]

KOVACIC, WILLIAM E., ROBERT C. MARSHALL, LESLIE M. MARX, AND MATTHEW E. RAIFF (2006): "Bidding Rings and the Design of Anti-Collusion Measures for Auctions and Procurements," *Handbook of procurement*, 15. [1845,1851,1852]
LEE, KWOK-HAO, AND LEON MUSOLFF (2021): "Entry Into Two-Sided Markets Shaped by Platform-Guided Search," Working Paper. [1842,1851]
LIN, LONG-JI (1992): "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching," *Machine learning*, 8 (3–4), 293–321. [1853]
MARX, LESLIE M., AND GREG SHAFFER (2010): "Slotting Allowances and Scarce Shelf Space," *Journal of Economics & Management Strategy*, 19 (3), 575–603. [1851]
MEHRA, SALIL K. (2015): "Antitrust and the Robo-Seller: Competition in the Time of Algorithm," *Minnesota Law Review*, 100, 1323–1375. [1844]
MIKLÓS-THAL, JEANINE, AND CATHERINE TUCKER (2019): "Collusion by Algorithm: Does Better Demand Prediction Facilitate Coordination Between Sellers?" *Management Science*, 65 (4), 1552–1561. [1845]
MNIH, VOLODYMYR, KORAY KAVUKCUOGLU, DAVID SILVER, ANDREI A. RUSU, JOEL VENESS, MARC G. BELLEMARE, ALEX GRAVES, MARTIN RIEDMILLER, ANDREAS K. FIDJELAND, GEORG OSTROVSKI et al. (2015): "Human-Level Control Through Deep Reinforcement Learning," *Nature*, 518 (7540), 529–533.
MUSOLFF, LEON (2022): "Algorithmic Pricing Facilitates Tacit Collusion," Working Paper. [1842]
O'BRIEN, DANIEL P., AND GREG SHAFFER (1997): "Nonlinear Supply Contracts, Exclusive Dealing, and Equilibrium Market Foreclosure," *Journal of Economics & Management Strategy*, 6 (4), 755–785. [1851]
O'CONNOR, JASON, AND NATHAN WILSON (2021): "Reduced Demand Uncertainty and the Sustainability of Collusion: How AI Could Affect Competition," *Information Economics and Policy*, 54, 100882. [1845]
OECD (2017a): "Algorithms and Collusion—Note From Italy," OECD. [1842]
OECD (2017b): "Algorithms and Collusion: Competition Policy in the Digital Age," OECD. [1842]
PALFREY, THOMAS R. (1983): "Bundling Decisions by a Multiproduct Monopolist With Incomplete Information," *Econometrica*, 51 (2), 463–483. [1851]
SALCEDO, BRUNO (2015): "Pricing Algorithms and Tacit Collusion," Working Paper. [1845]
SANDHOLM, TUOMAS W., AND ROBERT H. CRITES (1995): "On Multiagent Q-Learning in a Semi-Competitive Domain," in *International Joint Conference on Artificial Intelligence*. Springer, 191–205. [1844]
SILVER, DAVID, AJA HUANG, CHRIS J. MADDISON, ARTHUR GUEZ, LAURENT SIFRE, GEORGE VAN DEN DRIESSCHE, JULIAN SCHRITTWIESER, IOANNIS ANTONOGLOU, VEDA PANNEERSHELVAM, MARC LANCTOT et al. (2016): "Mastering the Game of Go With Deep Neural Networks and Tree Search," *Nature*, 529 (7587), 484.
SUTTON, RICHARD S., AND ANDREW G. BARTO (2018): *Reinforcement Learning: An Introduction*. MIT Press. [1852,1853]
TEH, TAT-HOW (2022): "Platform Governance," *American Economic Journal: Microeconomics*, 14 (3), 213–254. [1845]
TEH, TAT-HOW, AND JULIAN WRIGHT (2022): "Intermediation and Steering: Competition in Prices and Commissions," *American Economic Journal: Microeconomics*, 14 (2), 281–321. [1845]
TESAURO, GERALD, AND JEFFREY O. KEPHART (2002): "Pricing in Agent Economies Using Multi-Agent Q-Learning," *Autonomous Agents and Multi-Agent Systems*, 5 (3), 289–304. [1844]
TSITSIKLIS, JOHN N. (1994): "Asynchronous Stochastic Approximation and Q-Learning," *Machine learning*, 16 (3), 185–202. [1854]
WALTMAN, LUDO, AND UZAY KAYMAK (2008): "Q-Learning Agents in a Cournot Oligopoly Model," *Journal of Economic Dynamics and Control*, 32 (10), 3275–3293. [1844,1854]
WATKINS, CHRISTOPHER J. C. H. (1989): *Learning From Delayed Rewards*. Thesis Chapter. Cambridge: King's College. [1852]
WATKINS, CHRISTOPHER J. C. H., AND PETER DAYAN (1992): "Q-Learning," *Machine learning*, 8 (3–4), 279–292. [1854]

Co-editor Alessandro Lizzeri handled this manuscript.

Manuscript received 16 July, 2021; final version accepted 16 February, 2023; available online 18 May, 2023.