Platform Design when Sellers Use Pricing Algorithms

Justin P. Johnson[†] Andrew Rhodes[‡] Matthijs Wildenbeest[§]

September 12, 2020

Abstract. Using both economic theory and Artificial Intelligence (AI) pricing algorithms, we investigate the ability of a platform to design its marketplace to promote competition, improve consumer surplus, and even raise its own profits. We allow sellers to use Q-learning algorithms (a common reinforcement-learning technique from the computer-science literature) to devise pricing strategies in a setting with repeated interactions, and consider the effect of steering policies that reward firms that cut prices with additional exposure to consumers. Overall, the evidence from our experiments suggests that platform design decisions can meaningfully benefit consumers even when algorithmic collusion might otherwise emerge but that achieving these gains may require more than the simplest steering policies when algorithms value the future highly. We also find that policies that raise consumer surplus can raise the profits of the platform, depending on the platform's revenue model. Finally, we document several learning challenges faced by the algorithms.

In this article we use both economic theory and extensive experiments on Artificial Intelligence (AI) pricing algorithms to assess the efficacy of two related policies that an online retail platform might implement to increase competition among the merchants who sell in its marketplace. Our analysis suggests that a platform can design itself to simultaneously promote competition, improve consumer surplus, and even raise its own profits.

We are motivated by a recent surge of interest by regulators and legal scholars about the growing use of pricing algorithms.[1] Although algorithms might behave in a competitive manner, a concern is that they might be used to foster and sustain collusion in online markets. Indeed, economists have recently shown that commonly used AI algorithms may learn to play collusive strategies even without being explicitly programmed to do so.[2] We show that the behavior of algorithms can be guided by the platform to benefit consumers and increase the platform's profits, irrespective of whether sellers act competitively or collusively.

We thank Emilio Calvano, Timo Klein, Scott Duke Kominers, and participants at the University of California (Berkeley) and the 2020 Econometric Society Meetings (San Diego).

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

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

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

[1]Regulatory concerns are evident in OECD (2017), CMA (2018), and DOJ (2018).

[2]See Calvano, Calzolari, Denicolò, and Pastorello (2020) and Klein (2019), also discussed further below.

Both policies that we propose, *price-directed prominence* (PDP) and its extension *dynamic price-directed prominence* (Dynamic PDP), involve steering consumer demand towards a subset of the sellers. The simpler of our two policies, PDP, involves the marketplace simply steering demand towards sellers that are charging lower prices within a given category. Relatively high-priced sellers are shown to fewer consumers, reducing their profits. The more subtle of our two policies, Dynamic PDP, also conditions on past prices. Most online marketplaces already help guide the consumer product-discovery process, for example by determining how different products are ranked or displayed.

The "buy box" on the Amazon Marketplace provides an example of a steering technique similar to those that we propose. The buy box shows consumers exactly one seller among the many who may be offering any given homogenous product. Current price substantially determines which firm "wins" the buy box, although past performance also has an influence.[3] The firm that wins the buy box may receive 80% or more of the demand for that product. PDP is related to the buy box, but is also different in some important ways. For example, PDP allows for any number of products to be shown to consumers, and for some consumers to see more products than others. We also allow for sellers within a category to be differentiated.

The economic theory that we develop suggests that PDP imposes a tradeoff for consumers between lower prices and less product variety, where product variety is described by the standard logit model of differentiated products. Theory predicts that prices fall but 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, but when sellers collude PDP lowers consumer surplus irrespective of the number of products that are obscured.

Because PDP is not predicted to work well when sellers are colluding, our second technique, Dynamic PDP, attacks the foundations of collusion more directly. It does so by steering additional demand to a firm not just in the period in which it cut prices but also in later periods, subject to the firm not raising prices and not being undercut by more than a small "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.

But how might pricing algorithms that interact repeatedly actually behave? To investigate, we deploy a common reinforcement-learning algorithm from computer science called

[3]Consumers can see additional firms by clicking on a button. See Chen, Mislove, and Wilson (2016) and Gómez-Losada and Duch-Brown (2019) for empirical analysis of how the buy box works.

Q-learning. Reinforcement-learning algorithms (discussed further in Section 2) are an approach to artificial intelligence in which algorithms attempt to find an optimal solution to a potentially complex problem based on their own historical experience with the environment.

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

Our experiments with these AI algorithms reveal that PDP lowers prices but that the price declines need not be large enough to compensate consumers for the loss of variety. In this sense, sometimes algorithms behave more like competitive firms (with PDP helping consumers) but other times behave more like a cartel (with PDP harming consumers). PDP is less likely to benefit consumers when the algorithms value future profits highly, that is when they operate with high discount factors, which is intuitive given that economic theory predicts cartels are easier to support when this is so. But even when firms are patient, consumers sometimes benefit, and we find this is more likely when product differentiation (as measured by the scale parameter in the standard logit model) is lower. When consumers do benefit, the consumer-surplus gains can be large, sometimes exceeding 35%.

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

Overall, the evidence from our experiments suggests that platform design decisions can meaningfully benefit consumers even when algorithmic collusion might otherwise emerge. However, achieving these gains may require more subtle steering policies such as Dynamic PDP. We also find that policies that raise consumer surplus can raise the profits of the platform, depending on the platform's revenue model. Finally, we document several learning challenges faced by the algorithms, with implications for the ideal design of the platform.

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 computers and algorithms may facilitate collusion. They identify several ways in which this might occur, including that humans might intentionally design algorithms to be the hub of a hub-and-spoke style conspiracy as well as artificial intelligence (AI) algorithms learning to collude without explicit human direction. Legal cases have already been brought although so far these cases have all involved humans as well as machines, as in the *Topkins* case, pursued by the U.S. Department of Justice, and the *Eturas* case in the European Union. Harrington (2018) discusses legal and policy issues

related to algorithmic collusion, including 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 simple prisoners' dilemma games has been studied at least as far back as Sandholm and Crites (1995) and Tesauro and Kephart (2002), and by Waltman and Kaymak (2008) in a setting with Cournot competition. Recent work in economics goes further. Calvano, Calzolari, Denicolò, and Pastorello (2020) study collusion by AI algorithms in a logit model of differentiated products. In addition to finding prices that are elevated compared to theoretical predictions of non-collusive behavior, they identify that algorithms may learn to support collusive outcomes using reward-and-punishment schemes. Klein (2019) 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 all use a class of algorithms called reinforcement-learning algorithms, commonly used in computer science.[4] As discussed in more detail later, such algorithms comprise an approach to artificial intelligence in which algorithms are not assumed to fully understand the environment in which they operate but instead must discover over time, based on experience and experimentation, which strategies work well. These algorithmic studies are part of a broader investigation into cooperation in games dating back to Axelrod and Hamilton (1981).[5]

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. Also related are studies of intermediaries that bias recommendations or otherwise steer demand to raise their own static monopoly profits (Hagiu and Jullien, 2011; Inderst and Ottaviani, 2012; De Corniere and Taylor, 2019; Teh and Wright, 2020). Our approach differs in a number of ways but particularly in that we allow for (algorithmic) collusion and focus on how an intermediary can raise consumer surplus.

[4]Other papers use economic theory but not actual algorithms. Miklós-Thal and Tucker (2019) and O'Connor and Wilson (2019) assume that algorithms improve the quality of information available and ask how that changes the structure of collusion. Salcedo (2015) and Brown and MacKay (2019) explore the effect on prices when sellers commit to particular pricing algorithms.

[5]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.

1. Model

Here we lay out a theoretical model to examine how a platform's design affects prices and other market outcomes. We test our results from Section 3 onwards using AI algorithms.

There are n firms, each of which sells its own differentiated product on a monopoly retail platform (alternatively, firms sell identical products but are differentiated in some other way). These firms have the same constant marginal cost $c > 0$, which is inclusive of any fees paid to the platform. Firms interact repeatedly over time. In each period $t = 0, 1, ...$, each firm i observes all past prices and other outcomes and simultaneously sets a price $p_i^t \geq 0$ (negative prices are not allowed). The firms have a common discount factor $\delta \in (0, 1)$.

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

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

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

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

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

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

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

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

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

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

Below we consider two ways in which the set of displayed firms $\mathcal{N}_t$ could be generated, and examine how these display methods influence prices, consumer surplus, output, and revenue.

1.1. **Price-Directed Prominence.**

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

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

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 firms behave like a cartel or not. First we assess predicted outcomes when the market is competitive.

1.1.1. *PDP in a Competitive Market.* Suppose that in each period t 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 with differentiated products. If PDP is not in effect—and so all firms are displayed—then there is a unique equilibrium where each firm charges $p^*_{BN} > c$.

In contrast, under PDP prices are driven down to marginal cost as firms compete for the right to be displayed to consumers.

Lemma 1. *In a competitive market in which PDP is in effect, at least $k+1$ firms price at marginal cost in equilibrium.*

Although PDP drives the prices of displayed products down to marginal cost, it also limits how much variety is available to consumers. PDP can therefore present a tradeoff.

Proposition 1. *Compared to the case where all n firms are shown to consumers, in a competitive market:*

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

The first part of Proposition 1 says that PDP raises consumer surplus and total output whenever the proportion k/n of firms displayed is large enough. Indeed, even if as many as 63% of products are intentionally obscured from consumers, the resulting gains from intensified price competition are enough to benefit consumers. (Note that this 63% threshold is certainly met if only one firm is obscured, that is if $k = n-1$.) Of course, it is always better to show more rather than fewer products to consumers, conditional on those products being priced at marginal cost. Unfortunately, it is hard to provide general analytical conditions under which PDP increases revenue (though it is easy to find numerical examples where this happens). However, as stated in the proposition, one stringent condition for revenue to increase is that the outside option a_0 is high and the level of differentiation μ is low.

On the other hand, the second part of Proposition 1 says that PDP reduces consumer surplus and total output if too few firms are displayed. The loss of variety dominates if the fraction of firms displayed is less than $e^{-\frac{n}{n-1}}$. (Note that this is satisfied if $n \geq 4$ and only one firm is displayed, that is if $k = 1$.[6]) Since PDP reduces prices and output it also reduces revenue.

Thinking in terms of an actual platform shifting demand across products and influencing product rankings, the proposition suggests that it is important to allow consumers to discover multiple products while also obscuring enough to substantially intensify price competition.

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

1.1.2. *PDP in a Cartelized Market.* Here we investigate the theoretical performance of PDP when the market is cartelized. We focus on *full collusion*—whereby in each period the n firms set prices to maximize their joint profits, that is, they charge the same prices as would a monopolist with n symmetric products. We begin with the following straightforward result.

Lemma 2. *Full collusion involves setting the same price in each period for each product. Denote this price by $p^m(k)$ for $k \in \{1, ..., n\}$.*

We first look at how PDP affects the sustainability of full collusion as an equilibrium outcome.

As is typical in models of collusion, for a given k full collusion is easier to maintain when δ is larger because the future is more valuable and so defecting from a collusive arrangement is less attractive. We assume that any defection leads firms to set the prices they would charge in a one-shot game. Lemma 1 therefore implies that firms price at marginal cost and so earn zero profits in all periods following a defection, despite intrinsic product differentiation.

Denote by $\widehat{\delta}_k$ the critical discount factor above which collusion 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 fully collusive profit is lower. Second, as k decreases the value to any given firm of deviating from full collusion increases. Roughly speaking, this is because when fewer firms are being shown consumers have fewer options to consider in the period in which a deviation occurs, thereby increasing the deviation profits. Both of these effects decrease the stability of a cartel.

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

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

[6]We also note that PDP increases consumer surplus and output if and only if k/n exceeds a threshold, and that as n gets large that threshold tends to e^{-1} (as does the condition in the second part of the proposition).

This proposition says that—presuming some PDP is in effect—showing fewer firms makes collusion less stable. We note that this does not ensure that PDP makes collusion harder compared to the case where all n firms are shown to consumers. However extensive numerical simulations suggest that this is the case, at least when only $k = 1$ firms are displayed.

Proposition 2 along with our numerical simulations supports the idea that PDP may help destabilize cartels. If full collusion becomes unsustainable, then the cartel must lower its prices and consumers may be better off. Indeed, if only one firm is shown to consumers and $\delta < \widehat{\delta}_1$, it is straightforward to prove that in any (pure strategy) subgame perfect Nash equilibrium the firm that is shown to consumers prices at marginal cost.

On the other hand, if δ is sufficiently large then full collusion is sustainable whether PDP is used or not. For example, one can prove that even when only one firm is shown to consumers, full collusion is sustainable if $\delta \geq \widehat{\delta}_1 = 1 - 1/n$. This may not be a very stringent condition however, because a high degree of patience is similar to being able to rapidly observe and adjust prices—which in principle algorithms might excel at doing.

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

Proposition 3. *Suppose δ is large enough that full collusion is sustainable. 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 total output are also lower: as fewer firms are displayed, the decline in prices is too small to offset the loss of variety. Revenue is lower as well.

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

What is most interesting about Proposition 3 is that, under full collusion, consumers are harmed by any implementation of PDP. Intuitively, as k decreases the decline in prices is too small to offset the decrease in variety of products shown to consumers. Output falls for the same reason, and since prices are also lower so is revenue. Therefore depending on what fees it charges the platform may also be harmed by PDP. This stands in sharp contrast to what happens in competitive markets, where a modest loss of variety intensifies price competition so much that consumer surplus, output and revenue may all increase (Proposition 1).

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

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

Definition 2 (**Dynamic PDP).** *In period $t = 0$ 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 compared to the previous period, and*
(2) *no rival in period $t+1$ undercuts firm i by strictly more than a fixed value $ADV > 0$.*

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

Dynamic PDP gives a "pricing advantage" of ADV to the firm that was shown to consumers yesterday, making it easier for that firm to be shown to consumers today, so long as it does not raise its price. In competitive markets Dynamic PDP works the same as regular PDP (for $k = 1$). At least two firms charge marginal cost in each period, and so for example Dynamic PDP benefits consumers for low n but otherwise makes them worse off (Proposition 1).

However in cartelized markets Dynamic PDP is much more effective than regular PDP in reducing prices. To illustrate this we require some additional notation. Let $\tilde{\pi}(p)$ denote per-period profit of a firm that charges p and is the only firm shown to consumers. Let $\pi^m(1) = \tilde{\pi}(p^m(1))$ denote that same firm's per-period profit at the fully collusive price.

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

(1) *There exists a $\widehat{\delta}$ such that if $\delta < \widehat{\delta}$, then in any pure-strategy subgame-perfect Nash equilibrium the equilibrium transaction price equals marginal cost in all periods.*
(2) *The critical $\widehat{\delta}$ is increasing in ADV and weakly exceeds $\widehat{\delta}_1$ from Proposition 2.*
(3) *The transaction price equals marginal cost for any discount factor δ (so $\widehat{\delta} = 1$) if*

$$\tilde{\pi}(ADV) \geq \frac{\pi^m(1)}{n}.$$

Proposition 4 says that collusion is harder to sustain when the platform implements dynamic rather than regular PDP. Collusion is also more difficult as ADV grows larger. Indeed, the cartel is completely destabilized when ADV is sufficiently large—firms are unable to collude at any price above marginal cost even as they become arbitrarily patient, that is, as $\delta \to 1$.

We now sketch an intuition for why collusion is infeasible for high ADV even as $\delta \to 1$. To simplify the exposition suppose that ADV satisfies $\pi^m(1)/n < \tilde{\pi}(ADV) < \pi^m(1)$ (because

$\tilde{\pi}(p)$ is increasing, this implies that $ADV < p^m(1))$. Consider time zero and imagine the cartel tries to implement fully collusive pricing. If successful, the firms could split the discounted value of $\pi^m(1)$ between them, and so (on average) each would earn the discounted value of $\pi^m(1)/n$. One deviation that a firm could undertake is to charge ADV in all periods. The firm would win the advantage at $t = 0$ because by assumption $ADV < p^m(1)$. The firm would also keep the advantage in all future periods, because no firm could undercut it by strictly more than ADV. Hence the deviator would receive a payoff of $\tilde{\pi}(ADV) > \pi^m(1)/n$ in each period. But this means that each of the n firms can assure itself of strictly more than $\pi^m(1)/n$ in each period, which is impossible because by definition per-period industry profits cannot exceed $\pi^m(1)$. Hence each firm should deviate from full collusion at time zero. Similar arguments can then be used to show that firms have an incentive to deviate from any collusive scheme which tries to sustain prices above marginal cost.

One might wonder whether Dynamic PDP could be simplified further. For example, consider a rule where one firm with the lowest price at $t = 0$ gets all future demand provided it never raises its price. In theory, this rule is at least as effective as Dynamic PDP in pushing down prices—it can be shown that for any $\delta < 1$ the equilibrium transaction price is c in each period. In practice, however, this rule might perform poorly if in the future an incumbent experienced a cost reduction or a more efficient firm entered the market. We discuss this more in Section 4.3.

Continuing with the case of high discount factors, Dynamic PDP again introduces a trade-off between lower prices but less variety.

Proposition 5. *Suppose δ is sufficiently high that absent PDP firms would fully collude. Compared to the case where all n firms are shown to consumers, Dynamic PDP increases consumer surplus and total output if and only if*

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

Dynamic PDP can benefit consumers and increase output in a wide variety of circumstances when firms are patient and markets are cartelized. (Note that $n < \tilde{n}$ holds for many parameterizations—including those used in our later experiments—even for very large n.) We also find that revenue can increase. Although it is again hard to provide general analytical conditions, revenue increases if for example $n = 2$, $a - c < a_0$, and μ is small. This contrasts with Propositions 1 and 3 which showed that regular PDP with $k = 1$ can perform badly.

2. A Multi-Agent Reinforcement Learning Approach

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

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

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

2.1. **Q-learning with a Single Agent.**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Each algorithm maintains its own Q-matrix and updates it over time in the manner described in Section 2.1. Implementing Q-learning requires an initial "time zero" Q-matrix, which we

build as follows. Fixing an agent and state, for each action available to that agent we derive the within-period payoff that would be expected if all other agents uniformly randomized their actions. We then divide this value by $1-\delta$ so that the Q-matrix indeed contains an initial estimate of the total future payoffs of taking different actions today.

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

We run these algorithms until the induced strategy of each agent does not change for 100,000 periods.[7] In other words, for each agent in each period we take that agent's Q-matrix and determine, for each possible state, which action is associated with the highest payoff in terms of the Q-matrix. This procedure induces a policy function for each agent in each period. If, for any 100,000 period horizon, this policy function is stable for each agent then we say that the algorithms have converged. We then compute payoffs and other relevant metrics by averaging over this 100,000 period horizon.

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

3. Price-Directed Prominence: Experimental Results

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

3.1. **Higher Product Differentiation.**

To set a baseline for $\mu = 1/4$, Table 1 reports results for when price-directed prominence is not in effect ($\gamma = 0$). The Bertrand-Nash price is what theory predicts both firms would charge in the absence of collusion, and is given by 1.473. The collusive price maximizes the joint profits of the firms, with both firms charging the same price 1.925 (Lemma 2).[8] Table 1 also reports the (share-weighted) AI prices that the algorithms actually charge, given by 1.682. Thus, our algorithms typically converge to a price exceeding the Bertrand-Nash price, in line with what we would expect from Calvano, Calzolari, Denicolò, and Pastorello (2020).

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

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

Bertrand-Nash Price	Collusive Price	AI Price
1.473	1.925	1.682

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

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

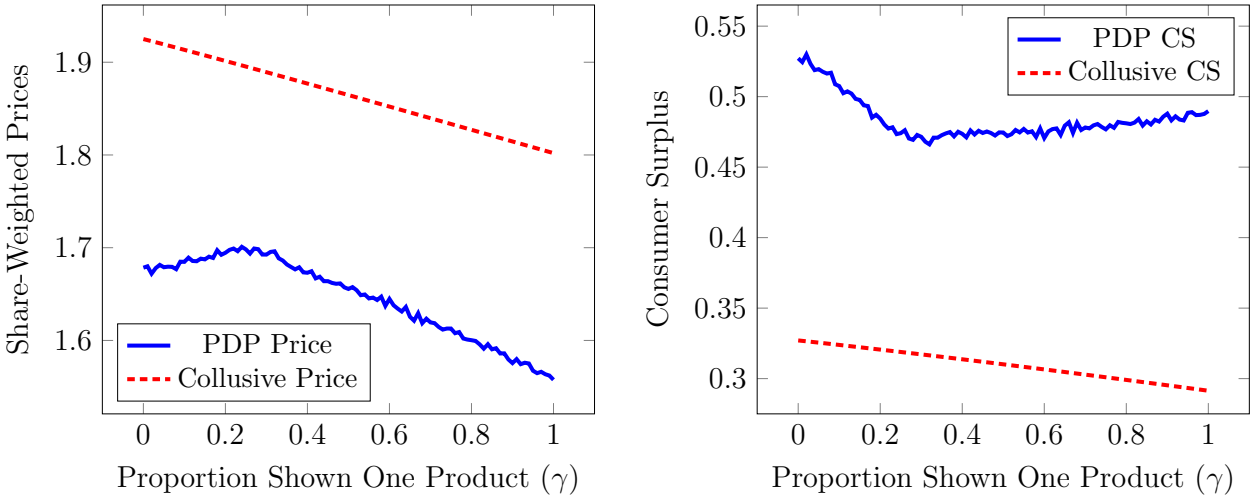


FIGURE 1. The effect of price-directed prominence on prices (left panel) and consumer surplus (right panel), with differentiation $\mu = 1/4$.

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

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

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

Bertrand-Nash Price	Collusive Price	AI Price
1.100	1.893	1.737

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

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

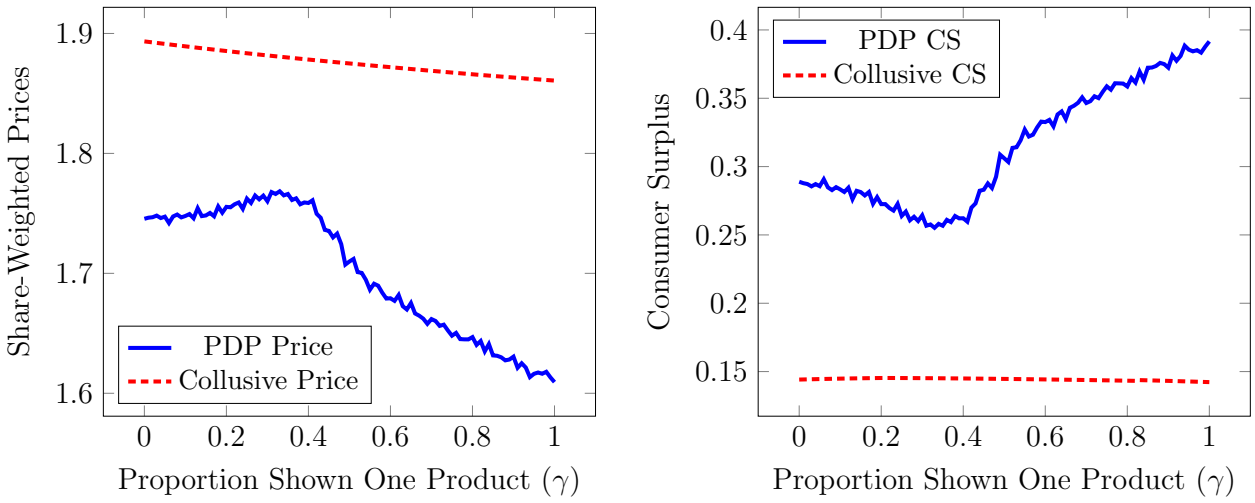
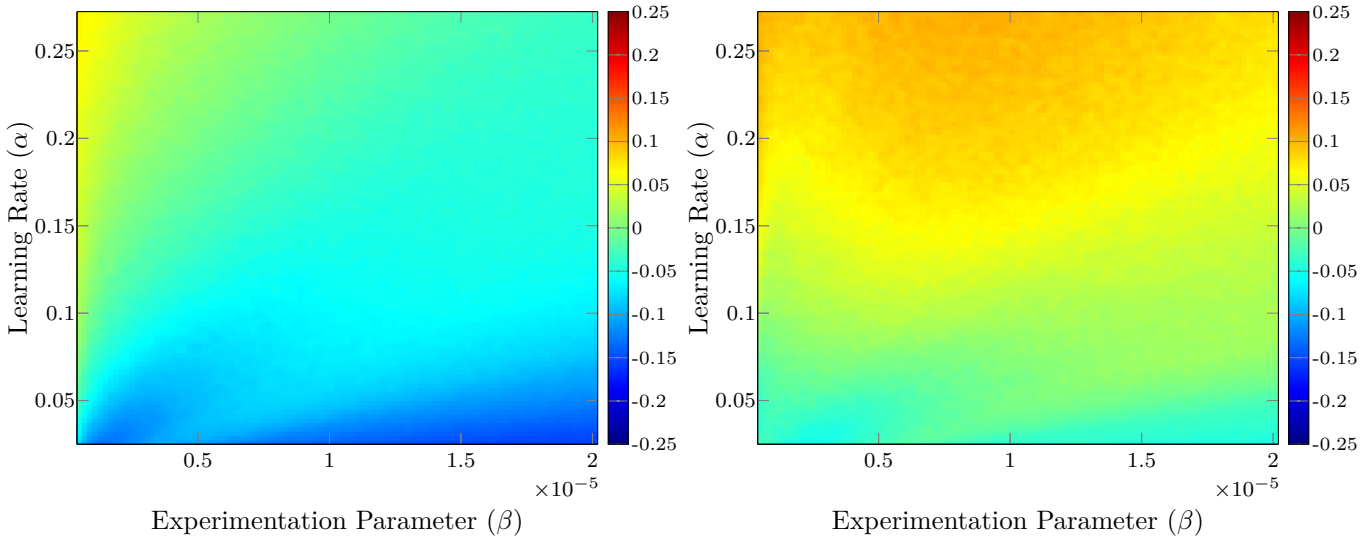


FIGURE 2. The effect of price-directed prominence on prices (left panel) and consumer surplus (right panel), with differentiation $\mu = 1/20$.

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

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



(a) PDP, $\mu = 1/4$

(b) PDP, $\mu = 1/20$

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

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

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

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

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

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

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

From these heatmaps two facts are apparent. First, changes in the learning parameters can have significant effects on the outcome even for fixed μ. Second, over a broad range of learning parameters, PDP appears more effective at raising consumer surplus when differentiation is low. Precisely, for $\mu = 1/20$ consumer surplus increases in 81.86% of the considered learning-parameter pairs, versus 13.75% of the pairs when $\mu = 1/4$. Similarly, when PDP has a positive effect, the magnitude of the effect appears larger when μ is smaller.

4. Dynamic PDP: Experimental Results

Although our experiments with price-directed prominence show some success in lowering prices and benefiting consumers, algorithmic prices remain high and consumers are often harmed when $\mu = 1/4$. Therefore, we now implement our policy of Dynamic PDP experimentally, recalling that Propositions 4 and 5 predict that this policy can improve consumer welfare especially when firms are very patient and the market is cartelized.

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

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

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

However, there is a sharp decrease in consumer surplus and increase in prices at $\gamma = 1$. It appears that this value presents a learning challenge for the algorithms. To see why this may

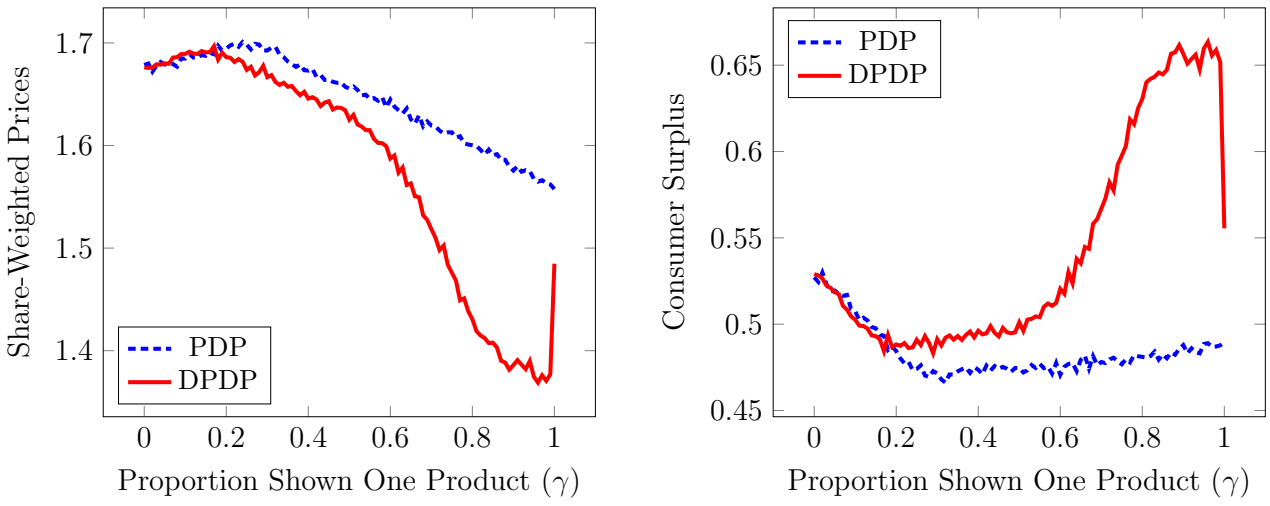


FIGURE 4. The effect of Dynamic PDP on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/4$.

be, observe that when $\gamma = 1$ one firm will have zero demand. The additional presence of a pricing advantage means that it may be that this firm's demand and profits are completely insensitive to prices across a significant range. In contrast, when $\gamma < 1$ each firm has positive demand and each firm's profit is always affected by marginal price changes.

Figure 5 presents results for the case of $\mu = 1/20$ (comparison prices and consumer surplus for regular PDP are the same as in Figure 2). The strongest effects occur at $\gamma = 0.99$, leading to about a 14% price decrease and a 75% increase in consumer surplus compared to not using PDP at all. We again see that for $\gamma = 1$ there is a sharp decrease in consumer surplus and increase in prices, which we attribute to the learning challenge mentioned above.

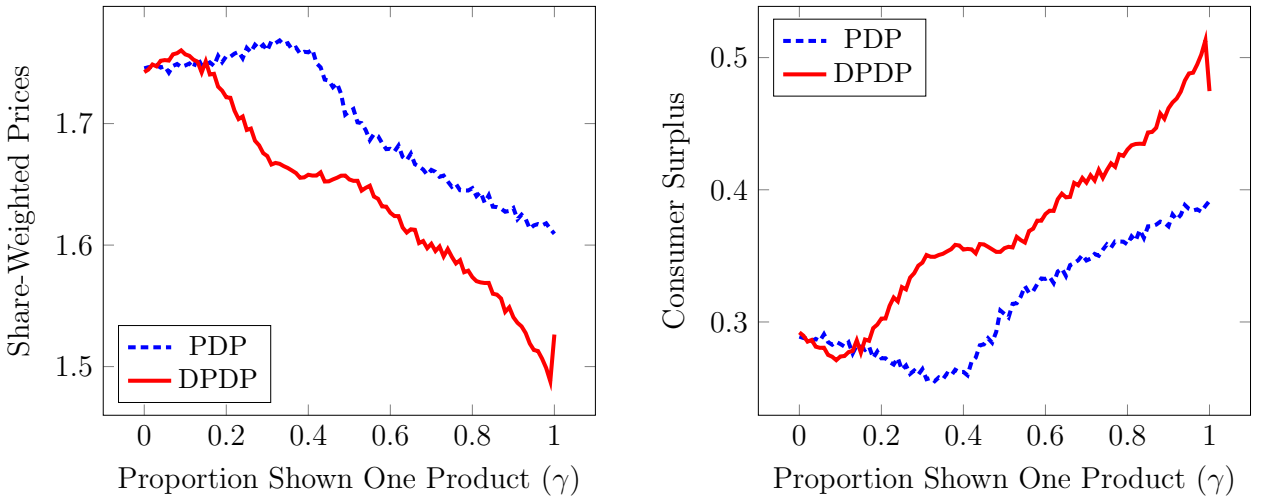
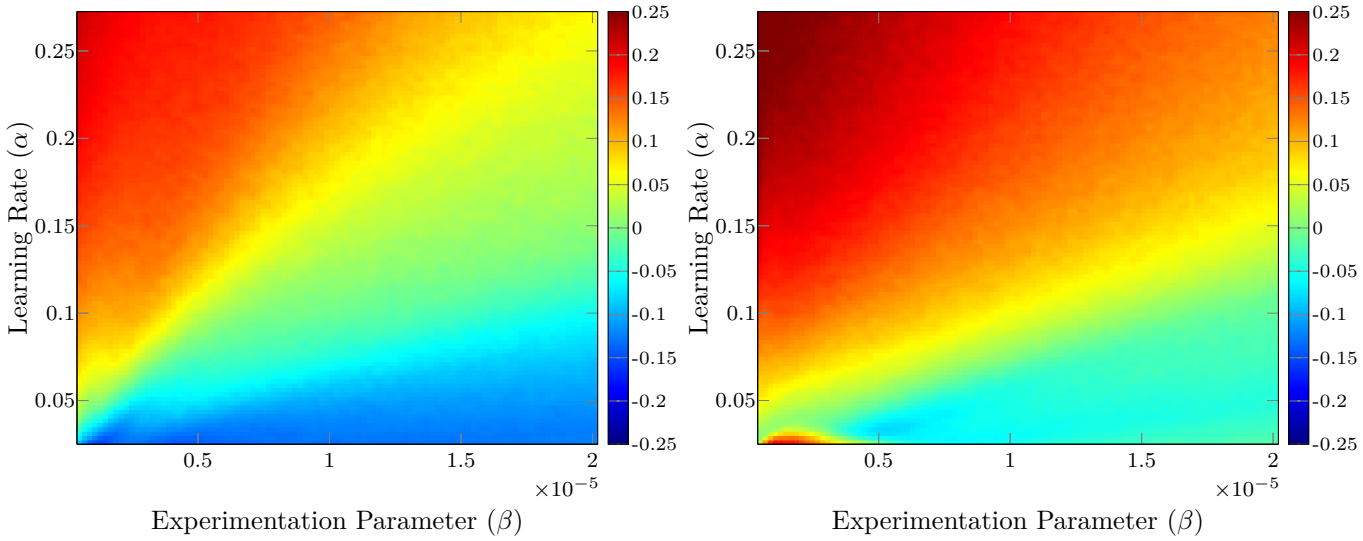


FIGURE 5. The effect of Dynamic PDP on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/20$.



(a) Dynamic PDP, $\mu = 1/4$

(b) Dynamic PDP, $\mu = 1/20$

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

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

Overall, Dynamic PDP appears to perform well in terms of lowering prices and benefiting consumers, although consumers do not always benefit from it. When $\mu = 1/4$, consumer surplus increases in 67.36% of the learning-parameter pairs, versus 80.46% of the pairs when $\mu = 1/20$. For $\mu = 1/4$ this is a significant difference compared to the 13.75% of cases where consumer surplus increased with regular PDP. For $\mu = 1/20$ the percentage of cases where consumer surplus increases is about the same as for regular PDP. However, the level of consumer surplus increases is much higher, so that DPDP still has a larger effect on consumer surplus compared to PDP when $\mu = 1/20$.

4.1. **Smarter AI: Lower Prices and Higher Consumer Surplus.** Compared to PDP, our implementation of Dynamic PDP presents an additional learning challenge.[9] Specifically, a firm's optimal price in period t may depend on whether it has the advantage or not, but that information is not included in the state space observed by the algorithms (only the previous period's prices are). Thus, as firms update their Q-matrices at time t, they incorporate

[9]We have already discussed one learning challenge of Dynamic PDP, that of reduced performance at $\gamma = 1$.

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.

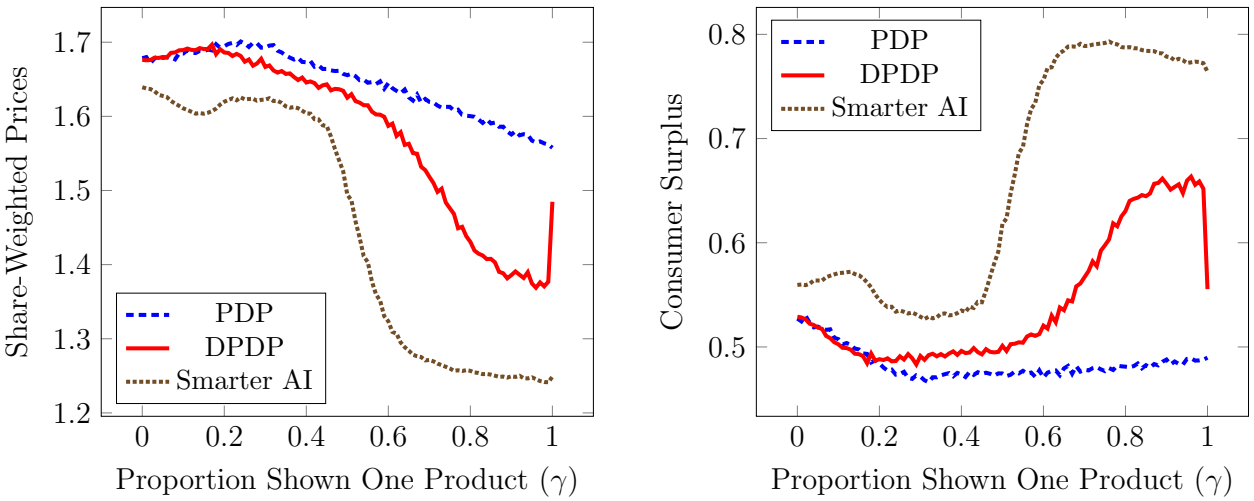


FIGURE 7. The effect of a smarter AI that includes in its state space the identity of the firm with the pricing advantage, on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/4$.

It is plausible that real-world algorithms would be designed to incorporate such additional information. To investigate, we alter our algorithms so that they can track the identity of the firm that was preferentially displayed to consumers in the previous period (that firm has the pricing advantage in the current period), in addition to prices from the previous period.

Allowing the AI to be "smarter" in this sense has a substantial effect on prices and consumer surplus. The case where $\mu = 1/4$ is shown in Figure 7 (similar results emerge for $\mu = 1/20$). Recall that in this case (for our baseline learning parameters), regular PDP harms consumers. With smarter AI, consumer surplus is highest at $\gamma = 0.76$, representing a 50% increase in consumer surplus compared to no PDP, and a 19% increase in consumer surplus compared to the best performance of Dynamic PDP without the extended algorithm (at $\gamma = 0.96$).[10] We also note that the smarter AI does not exhibit the sharp decrease in consumer surplus near $\gamma = 1$ that we observe for DPDP.

4.2. **Different Levels of the "Pricing Advantage".** Here we explore how the level of the pricing advantage ADV affects outcomes. We consider values of ADV in the interval $[0, 1]$, in increments of 0.01. Given that we have restricted our algorithms' prices to lie in $[1, 2.1]$ and that $c = 1$, even moderate values in this range are large enough that our theory

[10]We note that when $\gamma = 0$ the smarter AI delivers different prices and consumer surplus than PDP or DPDP do. The reason is that adding an additional element to the state space of the algorithms may influence their learning even if that element is not payoff relevant.

predicts large gains for consumers. Indeed, at $ADV = 0.3$, theory predicts marginal cost pricing for $\mu = 1/4$; for $\mu = 1/20$ a value of slightly more than $ADV = 0.4$ is sufficient.

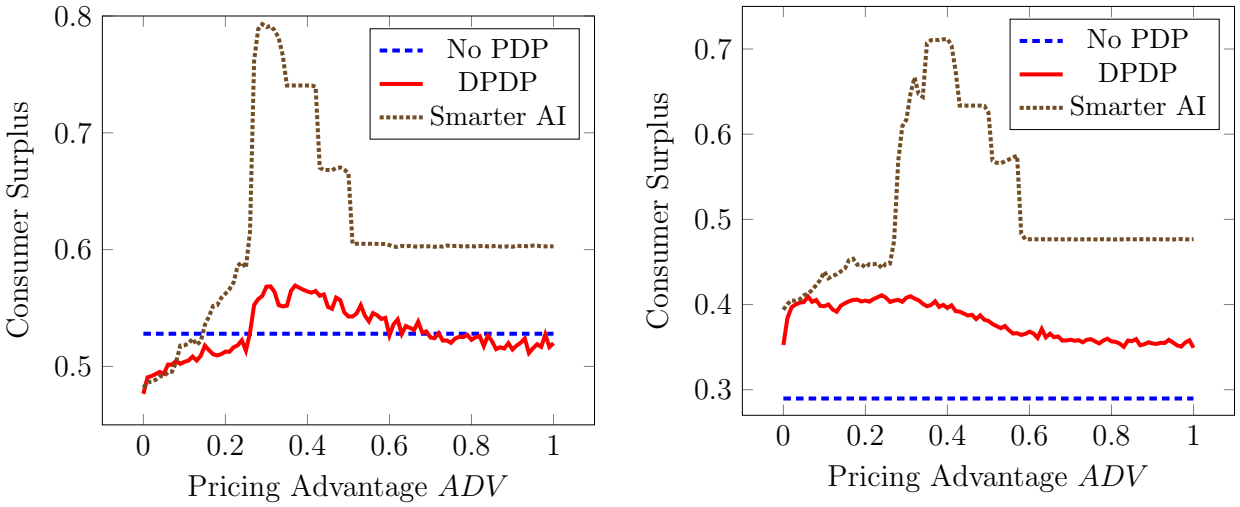
We will assess changes to ADV when $\gamma = 0.7$. Thus, in all of our results here we will compare the outcome of Dynamic PDP with $\gamma = 0.7$ to the case of no PDP at all. Note that when $ADV = 0$, Dynamic PDP corresponds to regular PDP in our experimental setting (and so there may be effects on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

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

The main difference between theory and our experiments is that consumer surplus is non-monotone in ADV, whereas theory predicts that increasing ADV beyond the level required to induce marginal cost pricing should have no further effect on consumer surplus. This is not entirely surprising given that our game theoretic results lean on backwards induction whereas our algorithmic agents do not learn in that manner. Larger values of ADV exacerbate a learning challenge discussed earlier: when ADV is large any firm without the pricing advantage will, for most or even all prices that it can charge, be competing only for the $1-\gamma = 0.3$ consumers who see all n products, and thus the AI may see no gains from setting lower prices. Similarly, larger values of ADV cover most or all of the prices that firms would set even if ADV were lower, suggesting that further increases should have little effect.

4.3. **Discussion.** Designing a marketplace that benefits consumers must take seriously both nuances in how algorithms learn and behave as well as likely non-stationarities in the marketplace. Learning challenges associated with Dynamic PDP, discussed above, include performance at $\gamma = 1$ and also partial observability of the state space when the algorithms only track past prices and not which firm was displayed to consumers in previous periods.

We can imagine how learning challenges and also non-stationarities in the marketplace might matter if, for example, the platform implemented a simpler version of Dynamic PDP that simply awarded 100% of all future demand to the firm that priced lowest in the initial period. This would effectively transfer all future competition to the first period and also, according to simple theory, lead to marginal-cost pricing. However, that version of Dynamic PDP might perform poorly, for several reasons. First, it might be very difficult for the algorithms to "discover" the marginal-cost pricing equilibrium given that they do not learn by backwards



(a) $\mu = 1/4$ (b) $\mu = 1/20$

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

induction, similar to challenges when $\gamma = 1$. It is possible that one firm might price lower than other firms initially but still quite high. If this occurred then potentially the market would be stuck forever with high prices. Similarly, if new, more-efficient entrants arrived or some incumbents experienced reductions in their marginal costs, these new competitive forces might not matter if these firms cannot displace whichever firm currently has the advantage.

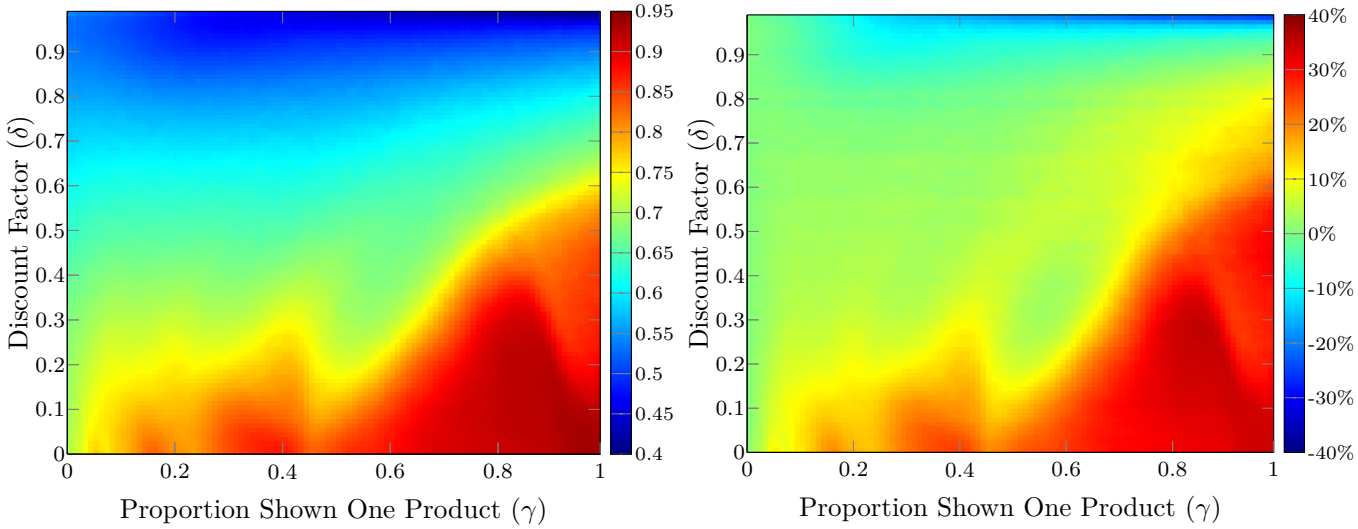
Dynamic PDP attempts to preserve robustness while also encouraging low prices. It preserves robustness by allowing firms to displace the rival that currently has an advantage by charging ADV less than the firm that has the advantage. At the same time, it must be remembered that the existence of the advantage itself is what ultimately drives lower prices; getting rid of the advantage would make it even easier for future firms to displace the advantaged firm but would also weaken the collusion-limiting nature of Dynamic PDP.

5. Less Patient Firms

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

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

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



(a) PDP and δ, raw values

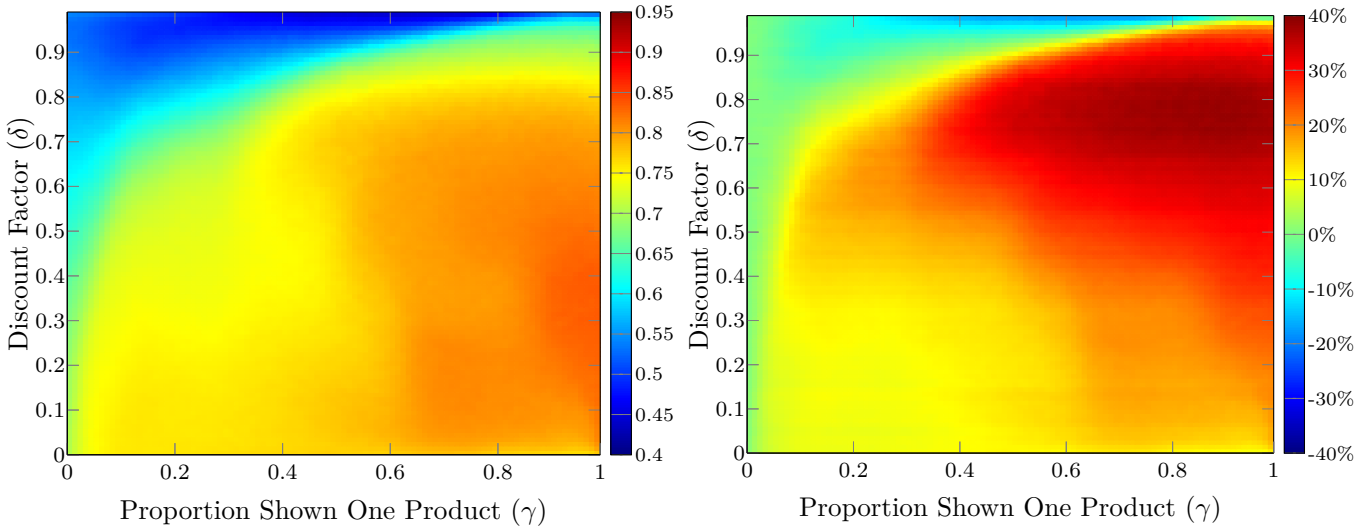
(b) PDP and δ, percentage change

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

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

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

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



(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 $\mu = 1/4$ and $ADV = 0.3$. Percentage change is calculated relative to $\gamma = 0$ for each given δ value.

each δ. We see that DPDP continues to work very well across a broad range of δ. In fact, DPDP raises consumer surplus for all δ values considered except for the highest ($\delta = 0.99$). We note that the percentage increases are not necessarily higher when δ is lower. This is because at $\gamma = 0$ for lower values of δ prices are already much lower than at higher δ values; there is less room for prices to drop. Another reason is that for many values of δ the prices at high values of γ are similar (and low, around 1.18).

6. Three Firms

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

Heatmaps covering a range of learning parameters tell a similar story. Figure 12 shows the effects of PDP (left panel) and Dynamic PDP (right panel). PDP always lowers consumer surplus, similar to the $n = 2$ case in which PDP only raised consumer surplus in 13.75% of the cases (Figure 3a). Although we do not display the data here, we also find that PDP

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

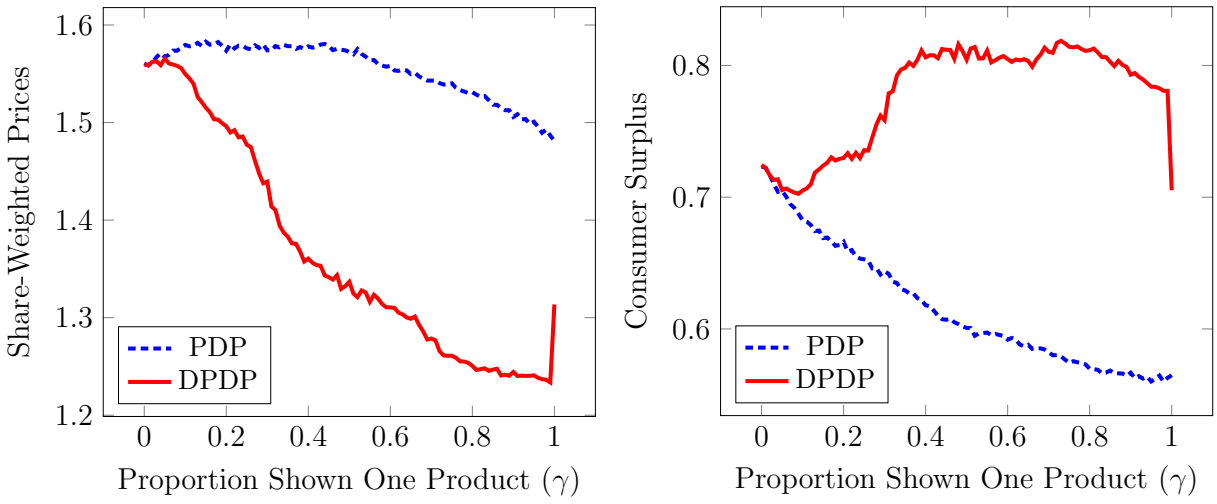


FIGURE 11. For $n = 3$, the effect of Dynamic PDP on prices (left panel) and consumer surplus (right panel), with $ADV = 0.3$ and differentiation $\mu = 1/4$.

almost always lowers consumer surplus even if $\mu = 1/20$, unlike the case with $n = 2$. Thus, at least when δ is high, our predictions from Proposition 3 hold true when $n = 3$: prices fall when PDP is imposed, but not enough to benefit consumers.[12]

In contrast to PDP, Dynamic PDP performs extremely well, increasing consumer surplus in 97.5% of the cases in Figure 12b. Although we do not display the data here, varying ADV leads to the same qualitative effects as with two firms, as discussed in Section 4.2.

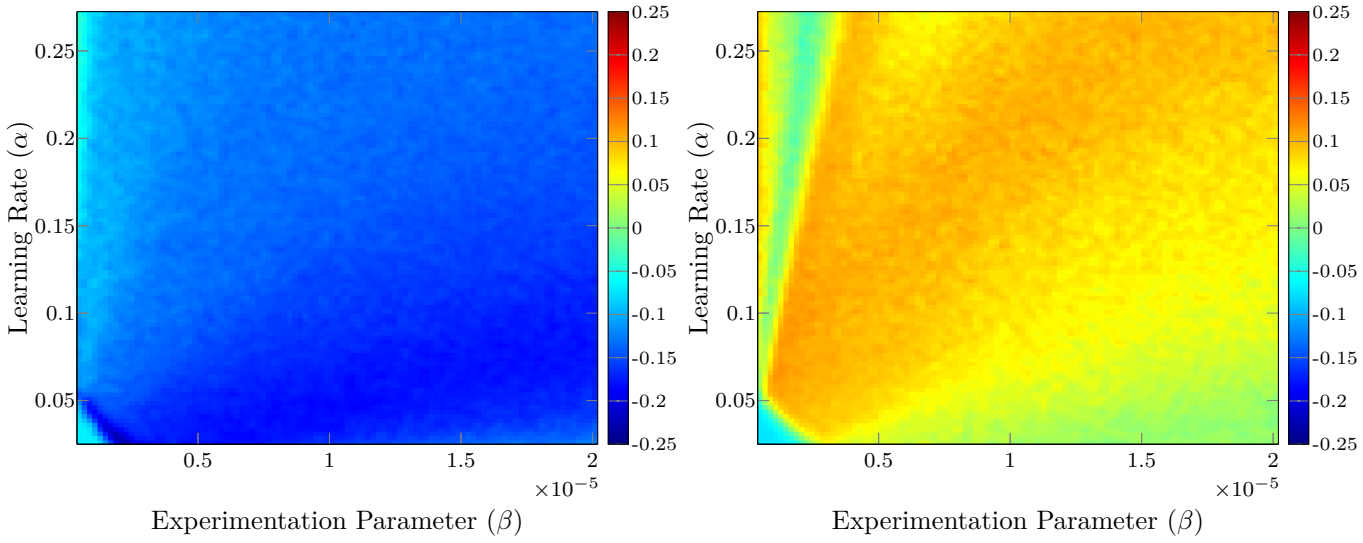
7. Platform Profits

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

Many online marketplaces impose fees on sellers. The most common fees are fixed shares of revenue and per-unit charges. Some platforms utilize both types of fees. For example, on the Amazon Marketplace sellers pay a proportion of the revenues they earn. But some categories have additional per-unit fees (or minimum per-sale fees from revenue sharing, which is equivalent to a per-unit fee when binding), and some sellers adopt a pricing plan offered by Amazon that also involves a per-unit fee. Additionally, some platforms such as Amazon offer fulfillment services to sellers, which (for Amazon) involve per-units fees.[13]

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

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



(a) PDP, $\mu = 1/4$

(b) Dynamic PDP, $\mu = 1/4$

FIGURE 12. Heatmaps of the effects of PDP and Dynamic PDP on consumer surplus, for $n = 3$, $\gamma = 0.7$, and $ADV = 0.3$. Consumer surplus increases in 0% of the cases in (a) and 97.54% of the cases in (b).

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

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

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

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

we look at various settings in which consumer surplus increases but per-customer revenue decreases, for $\gamma = 0.7$, including cases with high ($\delta = 0.95$) but also low ($\delta = 0.2$) patience, and cases with $n = 2$ and $n = 3$ firms. We suppose the initial market size is one. For each setting, we compute the percentage increase in market size which would keep overall revenues constant, and then divide that by the computed percentage increase in consumer surplus of moving from $\gamma = 0$ to $\gamma = 0.7$. We call the final number the "Critical Growth Elasticity" for revenue and present results in Table 3. For example, under PDP with $\mu = 0.05$, $\delta = 0.95$ and $n = 2$, the platform would break even on revenue if a one percentage point increase in consumer surplus led to a 0.297% increase in the number of customers visiting the platform.

Specification	Critical Growth Elasticity ($\gamma = 0.7$)
PDP ($\mu = 0.05, \delta = 0.95, n = 2$)	0.297
DPDP ($\mu = 0.05, \delta = 0.95, n = 2$)	0.293
DPDP ($\mu = 0.25, \delta = 0.95, n = 2$)	2.002
PDP ($\mu = 0.05, \delta = 0.2, n = 2$)	1.044
PDP ($\mu = 0.25, \delta = 0.2, n = 2$)	0.864
DPDP ($\mu = 0.05, \delta = 0.95, n = 3$)	0.392
DPDP ($\mu = 0.25, \delta = 0.95, n = 3$)	1.712

Effects of Policies on Revenue

TABLE 3. Critical Growth Elasticities for revenue for different polices and μ and δ values, using $\gamma = 0.7$. If a one percentage point increase in consumer surplus increases market size by at least this amount, then overall revenues increase.

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

We present results on output in Table 4. The second column gives the percentage of γ values such that consumer surplus is strictly higher at that value compared to $\gamma = 0$. Similarly, the third column gives the percentage of γ values such that total output is strictly higher at that value compared to $\gamma = 0$. For example, when $\mu = 0.05, \delta = 0.95$, $n = 2$, and PDP is used, consumer surplus is higher for 54% of the values of γ in (our grid on) $[0.01, 1]$, and output is higher for 22% of those values. The fourth and final column reports the critical growth elasticity as defined above except looking at output rather than revenue, again focusing on $\gamma = 0.7$ (values of 0.000 indicate that output is already higher at $\gamma = 0.7$ than at $\gamma = 0$). We see that these tend to be smaller than the required values for revenue. For example, under PDP with $\mu = 0.05$, $\delta = 0.95$ and $n = 2$, the critical growth elasticity for output is now only 0.032; this follows because PDP with $\gamma = 0.7$ increases consumer surplus by 19.93% and because the needed market growth to break even on total output is 0.64%.

Specification	% of γ values for which:		Critical Growth Elasticity
	CS Up	Output Up	($\gamma = 0.7$)
PDP ($\mu = 0.05, \delta = 0.95, n = 2$)	54%	22%	0.032
DPDP ($\mu = 0.05, \delta = 0.95, n = 2$)	83%	3%	0.045
DPDP ($\mu = 0.25, \delta = 0.95, n = 2$)	38%	24%	0.268
PDP ($\mu = 0.05, \delta = 0.2, n = 2$)	77%	37%	0.000
PDP ($\mu = 0.25, \delta = 0.2, n = 2$)	100%	92%	0.000
DPDP ($\mu = 0.05, \delta = 0.95, n = 3$)	71%	69%	0.000
DPDP ($\mu = 0.25, \delta = 0.95, n = 3$)	84%	69%	0.000

Effects of Policies on Output

TABLE 4. The second and third columns give the percentage of γ values such that consumer surplus or output increases, compared to $\gamma = 0$. The final column gives the Critical Growth Elasticities for output for different polices and μ and δ values, using $\gamma = 0.7$. If a one percentage point increase in consumer surplus increases market size by at least this amount, then overall quantity increases.

8. Conclusion

Online retail platforms often incorporate design features that steer demand towards particular sellers. Both economic theory and evidence from algorithmic experiments suggest that platform design decisions can benefit consumers and raise the platform's profits.

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 prices as the strategic variable. But in the real world, if steering were based entirely on price, then sellers might have incentives to push inefficiently low-cost and low-quality products. And even if price is the primary consideration, there are many potential policies that we have not considered that may work better, either in theory or practice.

When algorithms set prices we identified some challenges to improving market outcomes. First, when firms are patient, steering policies that condition only on current prices may not be sufficient to destabilize algorithmic collusion. Rather, a more subtle policy that directly attacks the foundations of collusion may be required. Second, because of particularities that we have documented in how algorithms learn, and because this process is not the same as economic theory typically assumes, successful platform design cannot be motivated entirely by theory but must also accommodate the learning challenges that algorithms may face.

Despite such challenges and limitations, our work is proof of concept that steering policies may be pro-competitive, both in competitive and cartelized markets, 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). 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. We now prove there is no Nash equilibrium in which $p_{k+1} > c$.

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

We use the following lemma in some subsequent proofs.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

It is convenient to rewrite Δr as

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

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

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

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

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

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

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

At the optimum (10) should equal zero for each $i = 1, \ldots, k$—which is only possible if $p_1 = p_2 = \cdots = p_k$. Recall that monopoly pricing and full collusion are equivalent. Therefore under full collusion it is (weakly) optimal to charge the same price on each product.

Substituting $p_1 = p_2 = \cdots = p_k = p^m(k)$ in Equation (10) and setting it to zero, $p^m(k)$ satisfies

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

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

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

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

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

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

Because $\pi^m(k)/(1 - \delta)$ is the industry's discounted value of colluding, and because each of the n firms could deviate from full collusion and assure itself of $\max_p (p - c)\tilde{D}(p, k)$ (or arbitrarily close to it for $k = 1$, in which the optimal deviation is to undercut $p^m(1)$ by an arbitrarily small amount), it follows that full collusion is sustainable if and only if

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

Next, let $\widehat{\delta}_k$ be the unique δ such that (12) holds with equality. Note that $\pi^m(k)$ is strictly increasing in k. Hence to prove that $\widehat{\delta}_1 > \widehat{\delta}_2 > \cdots > \widehat{\delta}_{n-1}$ it is sufficient to prove 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 (11) as

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

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

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

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

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

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

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

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

We will use the following result when proving Proposition 4.

Lemma 4. *Under PDP the critical discount factor for full collusion when $k = 1$ firms are shown is $\widehat{\delta}_1 = 1 - 1/n$.*

Proof. In the proof of Proposition 2 we showed that full collusion is sustainable if and only if (12) holds, and we also argued that $\arg\max_p(p-c)\tilde{D}(p,k) = p^m(k)$, which implies that $\max_p(p-c)\tilde{D}(p,k) = \pi^m(k)$. The claim then follows. ■

Proof of Proposition 4: Consider a pure-strategy subgame perfect Nash equilibrium (SPNE), and let $\widehat{p}_t$ denote the transaction price in period t along the equilibrium path. Let $\mathcal{U}$ denote period 0 as well as all periods $t > 0$ in which $\widehat{p}_t > \widehat{p}_{t-1}$, and let $\mathcal{D}$ denote all periods $t > 0$ in which $\widehat{p}_t \leq \widehat{p}_{t-1}$.

It is straightforward to prove that $\tilde{\pi}(p)$ is strictly increasing in p between c and $p^m(1)$. Let ADV^* be the unique ADV which solves $\tilde{\pi}(ADV) = \pi^m(1)/n$, and define $\widehat{\delta}$ as

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

Note that $\widehat{\delta}$ is weakly increasing in ADV, and that Lemma 4 implies $\widehat{\delta} \geq \widehat{\delta}_1$. Assume throughout the proof that $\delta < \widehat{\delta}$.

We first prove that $\widehat{p}_t \leq p^m(1)$ for all t. On the way to a contradiction suppose there exists a $t \in \mathcal{U}$ such that $\widehat{p}_t > p^m(1)$. Any firm could charge $p^m(1)$ in period t and win the advantage, and then i) if $ADV \leq c$, charge c in all future periods, and ii) if $ADV > c$, charge ADV in all future periods and keep the advantage. Therefore in period t the n firms' combined discounted payoff is at least

$$n\left[\pi^m(1) + \frac{\delta\tilde{\pi}(\max\{c, ADV\})}{1-\delta}\right]. \tag{15}$$

It is straightforward to show that this strictly exceeds $\pi^m(1)/(1-\delta)$ because $\delta < \widehat{\delta}$. However this is a contradiction since the joint profit in each period cannot exceed $\pi^m(1)$. Therefore $\widehat{p}_t \leq p^m(1)$ for all $t \in \mathcal{U}$. This further implies that $\widehat{p}_t \leq p^m(1)$ in any period $t \in \mathcal{D}$ as well. Hence $\widehat{p}_t \leq p^m(1)$ for all t.

Thus the supremum $\bar{p} \leq p^m(1)$ over transaction prices exists. We now prove that $\bar{p} = c$

We start by proving $\bar{p} = c$ for the case $ADV > c$. We do this in two steps.

In the first step we prove that $\bar{p} \leq ADV$. This follows automatically from earlier arguments if $ADV = p^m(1)$, so for this part of the proof consider $ADV < p^m(1)$. On the way to a contradiction suppose that $\bar{p} \in (ADV, p^m(1)]$. Let $\bar{t}$ be the first period in which $\widehat{p}_t > \bar{p} - \Delta$ for $\Delta \in (0, \bar{p} - ADV)$, and note that $\bar{t} \in \mathcal{U}$. Any firm could charge $\bar{p} - \Delta$ in period $\bar{t}$ and win the advantage, and then keep the advantage by charging ADV in all future periods. Therefore starting from period $\bar{t}$, along the equilibrium path the firms' combined discounted profits are at least

$$n\left[\tilde{\pi}(\bar{p} - \Delta) + \frac{\delta\tilde{\pi}(ADV)}{1-\delta}\right]. \tag{16}$$

In addition, since $\bar{p}$ is the supremum over transaction prices, joint profit in any period cannot exceed $\tilde{\pi}(\bar{p})$. Therefore (16) must be weakly less than $\tilde{\pi}(\bar{p})/(1-\delta)$, or equivalently

$$n\left[\tilde{\pi}(\bar{p} - \Delta) + \frac{\delta\tilde{\pi}\left(ADV\right)}{1-\delta}\right] \leq \frac{\tilde{\pi}(\bar{p})}{1-\delta}. \tag{17}$$

However (17) does not hold when $ADV \geq ADV^*$, because $\tilde{\pi}(\bar{p} - \Delta) > \tilde{\pi}(ADV)$ and because $ADV \geq ADV^*$ implies $n\tilde{\pi}(ADV) \geq \pi^m(1) \geq \tilde{\pi}(\bar{p})$. Similarly (17) does not hold when $ADV \in (c, ADV^*)$ and Δ is sufficiently small. To see why, note that (17) can be written as

$$\delta \geq \frac{n\tilde{\pi}(\bar{p} - \Delta) - \tilde{\pi}(\bar{p})}{n[\tilde{\pi}(\bar{p} - \Delta) - \tilde{\pi}\left(ADV\right)]}, \tag{18}$$

and also note that the righthand side is continuous in Δ and equal to $\left(1 - \frac{1}{n}\right)\frac{\tilde{\pi}(\bar{p})}{\tilde{\pi}(\bar{p}) - \tilde{\pi}(ADV)} \geq \widehat{\delta}$ at $\Delta = 0$. Therefore for any $\delta < \widehat{\delta}$ there exists Δ sufficiently small such that (18) fails. Summarizing, there exists Δ such that for any $ADV > c$ the condition (17) fails. We therefore have a contradiction, and conclude that $\bar{p} \leq ADV$.

In the second step (still for the case $ADV > c$) we prove that $\bar{p} = c$. On the way to a contradiction suppose that $\bar{p} \in (c, ADV]$. Again let $\bar{t}$ be the first period in which $\widehat{p}_t > \bar{p} - \Delta$ for $\Delta \in (0, \bar{p} - c)$, and note that $\bar{t} \in \mathcal{U}$. In period $\bar{t}$ any firm could charge $\bar{p} - \Delta$ and win the advantage, and then keep the advantage by charging $\bar{p} - \Delta < ADV$ in all future periods.

Hence the n firms' combined discounted profit along the equilibrium path from period $\bar{t}$ onwards is at least

$$n\left[\frac{\tilde{\pi}(\bar{p}-\Delta)}{1-\delta}\right].$$

However for Δ sufficiently small this is strictly above $\tilde{\pi}(\bar{p})/(1-\delta)$ which is a contradiction. Hence $\bar{p} \leq c$. But since $\bar{p} < c$ is impossible we must have $\bar{p} = c$.

We now prove that $\bar{p} = c$ for the case $ADV \leq c$. On the way to a contradiction suppose that $\bar{p} \in (c, p^m(1)]$. Let $\bar{t}$ be the first period in which $\widehat{p}_t > \bar{p} - \Delta$ for $\Delta \in (0, \bar{p} - c)$, and note that $\bar{t} \in \mathcal{U}$. Any firm could charge $\bar{p} - \Delta$ in period $\bar{t}$ and sell for sure. Therefore the sum of the n firms' combined discounted profits along the equilibrium path starting in period $\bar{t}$ is at least $n\tilde{\pi}(\bar{p}-\Delta)$. However for any $\delta < \widehat{\delta}$ there exists Δ sufficiently small such that $n\tilde{\pi}(\bar{p}-\Delta)$ strictly exceeds $\tilde{\pi}(\bar{p})/(1-\delta)$, which contradicts $\bar{p}$ being the supremum. Hence $\bar{p} \leq c$. But since $\bar{p} < c$ is impossible it must be that $\bar{p} = c$.

We have therefore established for all ADV satisfying $0 < ADV \leq p^m(1)$ and all $\delta < \widehat{\delta}$ that $\bar{p} = c$ in any pure strategy SPNE. The final step is to construct a SPNE where firms charge c in each period. Consider the following strategy:

(1) In period 0 all firms charge c.
(2) Suppose that the firm that won or kept the advantage 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 won or kept the advantage in period t charged strictly more than c in period t. Then in period $t+1$ all firms not holding the advantage charge c, while the firm holding the advantage charges the minimum of $\min\{c + ADV, p^m(1)\}$ and its price from period t.

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

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

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

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

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

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

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

■

References

Axelrod, R., and W. D. Hamilton (1981): "The evolution of cooperation," *Science*, 211(4489), 1390–1396.

Bloembergen, D., K. Tuyls, D. Hennes, and M. Kaisers (2015): "Evolutionary dynamics of multi-agent learning: A survey," *Journal of Artificial Intelligence Research*, 53, 659–697.

Brown, Z. Y., and A. MacKay (2019): "Competition in Pricing Algorithms," *Working Paper*.

Buşoniu, L., R. Babuška, and B. De Schutter (2010): "Multi-agent reinforcement learning: An overview," in *Innovations in multi-agent systems and applications-1*, pp. 183–221. Springer.

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2020): "Artificial Intelligence, Algorithmic Pricing and Collusion," *American Economic Review (forthcoming)*.

Chen, L., A. Mislove, and C. Wilson (2016): "An empirical analysis of algorithmic pricing on amazon marketplace," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 1339–1349.

CMA (2018): "Pricing Algorithms: Economic working paper on the use of algorithms to facilitate collusion and personalised pricing," *Competition Market Authority*.

Dal Bó, P., and G. R. Fréchette (2018): "On the determinants of cooperation in infinitely repeated games: A survey," *Journal of Economic Literature*, 56(1), 60–114.

Dana, J. D. (2012): "Buyer groups as strategic commitments," *Games and Economic Behavior*, 74(2), 470–485.

de Bruin, T., J. Kober, K. Tuyls, and R. Babuška (2015): "The importance of experience replay database composition in deep reinforcement learning. Deep Reinforcement Learning Workshop," *Advances in Neural Information Processing Systems (NIPS-DRLWS)*.

De Corniere, A., and G. Taylor (2019): "A model of biased intermediation," *The RAND Journal of Economics*, 50(4), 854–882.

Deck, C. A., and B. J. Wilson (2003): "Automated pricing rules in electronic posted offer markets," *Economic Inquiry*, 41(2), 208–223.

Dinerstein, M., L. Einav, J. Levin, and N. Sundaresan (2018): "Consumer price search and platform design in internet commerce," *American Economic Review*, 108(7), 1820–59.

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

Ezrachi, A., and M. E. Stucke (2017): "Artificial intelligence & collusion: When computers inhibit competition," *University of Illinois Law Review*, pp. 1775–1810.

Gómez-Losada, Á., and N. Duch-Brown (2019): "Competing for Amazon's Buy Box: A Machine-Learning Approach," in *International Conference on Business Information Systems*, pp. 445–456. Springer.

Hagiu, A., and B. Jullien (2011): "Why do intermediaries divert search?," *The RAND Journal of Economics*, 42(2), 337–362.

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

Inderst, R., and M. Ottaviani (2012): "Competition through commissions and kickbacks," *American Economic Review*, 102(2), 780–809.

Jaakkola, T., M. I. Jordan, and S. P. Singh (1994): "Convergence of stochastic iterative dynamic programming algorithms," in *Advances in neural information processing systems*, pp. 703–710.

Klein, T. (2019): "Autonomous Algorithmic Collusion: Q-Learning Under Sequential Pricing," *Working Paper*.

Kovacic, W. E., R. C. Marshall, L. M. Marx, and M. E. Raiff (2006): "Bidding rings and the design of anti-collusion measures for auctions and procurements," *Handbook of procurement*, 15.

Li, Y. (2017): "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*.

Lin, L.-J. (1992): "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, 8(3-4), 293–321.

Mehra, S. K. (2015): "Antitrust and the robo-seller: Competition in the time of algorithm," *Minnesota Law Review*, pp. 1323–1375.

Miklós-Thal, J., and C. Tucker (2019): "Collusion by algorithm: Does better demand prediction facilitate coordination between sellers?," *Management Science*, 65(4), 1552–1561.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. (2015): "Human-level control through deep reinforcement learning," *Nature*, 518(7540), 529–533.

O'Connor, J., and N. Wilson (2019): "Reduced Demand Uncertainty and the Sustainability of Collusion: How AI Could Affect Competition," *Working Paper*.

OECD (2017): "Algorithms and Collusion: Competition Policy in the Digital Age," *OECD*.

Salcedo, B. (2015): "Pricing Algorithms and Tacit Collusion," *Working Paper*.

Sandholm, T. W., and R. H. Crites (1995): "On multiagent Q-learning in a semi-competitive domain," in *International Joint Conference on Artificial Intelligence*, pp. 191–205. Springer.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. (2016): "Mastering the game of Go with deep neural networks and tree search," *Nature*, 529(7587), 484.

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

Teh, T.-H., and J. Wright (2020): "Intermediation and steering: Competition in prices and commissions," *American Economic Journal: Microeconomics*.

Tesauro, G., and J. O. Kephart (2002): "Pricing in agent economies using multi-agent Q-learning," *Autonomous Agents and Multi-Agent Systems*, 5(3), 289–304.

Tsitsiklis, J. N. (1994): "Asynchronous stochastic approximation and Q-learning," *Machine learning*, 16(3), 185–202.

Waltman, L., and U. Kaymak (2008): "Q-learning agents in a Cournot oligopoly model," *Journal of Economic Dynamics and Control*, 32(10), 3275–3293.

Watkins, C. J. (1989): "Learning from delayed rewards," *Thesis Chapter, King's College, Cambridge*.

Watkins, C. J., and P. Dayan (1992): "Q-learning," *Machine learning*, 8(3-4), 279–292.