



AI pricing algorithms under platform competition

J. Manuel Sanchez-Cartas¹ · Evangelos Katsamakos²

Accepted: 29 January 2024
© The Author(s) 2024

Abstract

Platforms play an essential role in the modern economy. At the same time, due to advances in artificial intelligence (AI), algorithms are becoming more widely used for pricing and other business functions. Previous literature examined algorithmic pricing, but not in the context of network effects and platforms. Moreover, platform competition literature has not considered how algorithms may affect competition. We study the performance of AI pricing algorithms (Q-learning and Particle Swarm Optimization) and naïve algorithms (price-matching) under platform competition. We find that algorithms set an optimal price structure that internalizes network effects. However, no algorithm is always the best because profitability depends on the type of competing algorithms and market characteristics, such as differentiation and network effects. Additionally, algorithms learn autonomously when an equilibrium is unstable and avoid it. When algorithm adoption is an endogenous strategic decision, several algorithms can be adopted in equilibrium; we characterize the conditions for the various outcomes and show that the equilibrium and platform profits are sensitive to algorithm design changes. Overall, our research suggests that AI algorithms can be effective in the presence of network effects, and platforms are likely to adopt a variety of algorithms. Lastly, we reflect on the business value of AI and identify opportunities for future research at the intersection of AI algorithms and platforms.

Keywords Algorithmic pricing · Artificial intelligence · Competition · Platforms · Q-learning · Reinforcement learning · Particle swarm optimization (PSO)

JEL Classification L41 · L12 · L13 · D43 · D21

✉ J. Manuel Sanchez-Cartas
jmcartas@ucm.es
Evangelos Katsamakos
katsamakos@fordham.edu

¹ Universidad Complutense de Madrid, Campus de Somosaguas, 28223 Madrid, Spain

² Gabelli School of Business, Fordham University, 140 W. 62nd Street, New York, NY 10023, USA

1 Introduction

Using algorithms instead of humans to make decisions is becoming more widespread in business. Algorithms are crucial for platforms such as social media (Facebook, TikTok), the sharing economy (Uber, Airbnb), video games, e-marketplaces, and many others. Platforms rely on algorithms to set prices, among other tasks. For example, Uber's ride fares may vary depending on the exact hour, destination, and weather conditions. The price of a Facebook advertisement varies depending on the audience and the ad format.

As digital platforms become prominent in the economy [1–3], understanding algorithmic pricing in the context of platforms is valuable for many companies. However, despite the vast research literature on platforms [4–6] and a growing literature on algorithmic pricing [7, 8], there is very little work on platform competition when platforms use pricing algorithms.

Platform pricing poses a challenge for algorithms because they must deal with network effects, and it is unclear whether algorithms can do that effectively. They must learn to set an optimal price structure to solve the coordination problem and get both sides on board [9–12]. Therefore, it is essential to understand how pricing algorithms behave under platform competition and whether the literature on pricing algorithms is robust in platform settings because one-sided intuitions often do not apply to multisided platforms. Moreover, it is unclear whether algorithms are effective in the presence of multiple equilibria. Most importantly, it is unclear what factors digital platforms need to consider when adopting pricing algorithms and what the competitive impact of those algorithms is.

This research aims to analyze the behavior of pricing algorithms used by competing platforms and how those algorithms affect platform competition outcomes (prices and profits). Our primary research questions are as follows: What is the performance of pricing algorithms when used by competing platforms? What factors should a platform consider when adopting a pricing algorithm? What algorithm should a platform choose when the choice of algorithm becomes an endogenous strategic decision?

To answer those questions, we build a computational model of two competing platforms grounded in game-theoretic economic literature. We use the classic Armstrong platform competition model [11] as a benchmark. We let the platforms use algorithmic pricing in a computer-simulated market and compare the prices and profits with the benchmark. Specifically, we analyze the pricing behavior of two advanced AI algorithms (Particle Swarm Optimization and Q-learning) and a naïve one (price-matching algorithm). We analyze what happens when platforms adopt the same algorithm or different algorithms. We also consider the choice of algorithm as a strategic platform decision: we let the platforms decide which algorithm to adopt and characterize the resulting equilibrium. We emphasize the firm perspective, focusing on how platform firms can do well using algorithms, but we also discuss potential competition policy implications.

Our framework allows us to consider the role of differentiation, network effects, and the possibility of multiple equilibria. In this regard, we analyze cases

with a unique and stable equilibrium, those with multiple equilibria, and those with asymmetric network effects. We evaluate various scenarios, allowing for a comprehensive evaluation of pricing algorithm performance under platform competition.

The article contributes to the literature on algorithmic pricing by showing that the algorithms learn the presence of network effects and set prices accordingly. It also contributes to the platform competition literature by showing that platform profitability depends on the algorithms used by competing platforms and market characteristics (differentiation and network effects). Moreover, it characterizes what algorithm platforms are expected to adopt when the choice of algorithm is endogenous, in which case platforms compete on algorithms and could differentiate themselves by choosing an algorithm that differs from the competitor. In addition, an extension explores how algorithmic memory (i.e., conditioning its play to the immediate past actions of the competitors) matters in algorithmic platform competition. Lastly, by evaluating AI algorithms, the article derives nuanced insights into the business value of AI and identifies opportunities for future research at the intersection of AI algorithms and platforms.

The rest of the article is organized as follows. Section 2 reviews related literature on algorithmic pricing and platforms. Section 3 presents the model we use as a framework, while Sect. 4 discusses the algorithms and parametrization. Section 5 presents computational experiments and results. Section 6 presents a sensitivity analysis, and Sect. 7 studies the algorithm adoption decisions of platforms. Section 8 discusses an extension with algorithmic memory. The discussion and conclusion are in Sects. 9 and 10.

2 Literature review

We review related literature on algorithmic pricing and platform competition.

2.1 Algorithmic pricing

Algorithmic pricing is a method of automatically setting prices to maximize the profits of a firm. In some sectors, such as e-marketplaces, pricing algorithms are used extensively [13]. Pricing algorithms benefit firms by lowering the cost of setting prices, but they can also reduce prices and increase consumer surplus [14, 15]. They may lead to more contestable markets, better service, better product availability, and an improved customer experience [16]. Although collusion is possible [7, 17, 18], it is unlikely in practice [19] and algorithmic design can help mitigate it [17]. However, it is more likely to find supracompetitive prices when firms use price-matching algorithms [20, 21] or when the training of algorithms is incomplete [22].

The current algorithmic pricing literature is primarily focused on Q-learning because it is simple and can be fully characterized by just a few parameters with economic interpretation [7]. Q-learning is a reinforcement learning algorithm

which is used to find the optimal action-selection policy in a given environment. Another option that has also attracted interest is the Particle Swarm Optimization (PSO) algorithm. PSO is a stochastic optimization technique that generates random points in a multidimensional space (particles) that move towards an optimal solution by sharing information about which points perform better [23]. This concept can easily be extended to price competition by assuming that each firm may test a limited set of prices (particles) before going to the market [24, 25]. PSO can also be characterized by just a few parameters and has the advantage of not suffering the dimensionality problem of Q-learning. Therefore, PSO would be a good option for multi-sided platforms that need to consider multiple sides and competitors.

Q-learning and PSO are sophisticated algorithms. However, some firms may utilize naïve algorithms such as price-matching, which has been studied since the early days of electronic commerce. Price-matching algorithms commit themselves to match (or beat) competitor prices. Therefore, they are easy to implement and allow setting prices at the same frequency as competitors when the monitoring is continuous. However, price-matching algorithms also raise policy concerns because they tend to tacitly collude [20, 21]. Despite this, their simplicity and wide availability make them ubiquitous in online marketplaces, as they are offered by many companies, such as Netrivals or Prisync.

There are many other algorithm options with varying degrees of sophistication and economic intuition. However, not all of them have been applied to economic problems (such as pricing) or are currently under development, with results that are not yet robust enough. Therefore, this paper focuses on Q-learning, PSO, and price-matching algorithms.

Although the literature has concentrated on just a few algorithms, the experimental evidence on algorithmic pricing is still limited. Only a few studies have adopted sophisticated AI algorithms to set prices in controlled environments [7, 8, 18, 26–29]. One key feature of this literature is the focus on a single algorithm, although comparative studies, which is another area to which we contribute, are beginning to attract attention [30–32]. Another feature is the focus on the possibility of supracompetitive prices that can emerge from various causes, such as collusion, miscoordination, imperfect training, or poor design. One specific aspect that has also attracted attention recently is the market framework, as the performance of an algorithm may vary in different markets [8, 33]. However, an aspect that has not been considered so far is the role of network effects that characterize many digital products and services, especially digital platforms.

The lack of literature on algorithmic platform pricing is explained by the development of algorithmic pricing literature, which mainly focuses on collusion. This phenomenon is much less understood and more complex in platform markets [34]. Therefore, the literature prefers to adopt simpler market frameworks. On the other hand, the experimental literature addresses the interaction between humans and algorithms but uses simple models for similar reasons [18]. Thus, the first contribution of this paper is to account for the impact of these algorithms in markets with network effects.

2.2 Platform competition and algorithmic pricing

A platform provides an infrastructure that facilitates the interaction between two (or more) groups, or sides, of platform participants, such as drivers and riders, in the case of ride-sharing platforms like Uber. A platform firm must consider the cross-side (indirect) network effects between the sides and set a price structure and level that maximizes the platform profit [11, 12, 35]. These network effects introduce a complexity in pricing, as platforms have to consider not only how competitors react to price changes, but also where those changes occur (on which side) and how competitors react to changes in the pricing structure (which side is cheaper). This problem appears because the platform firm faces a coordination (chicken-and-egg) problem: the agents' decision to join depends on what the other side does. For instance, consumers will not use a ride-sharing app with no drivers. The firm must solve the coordination problem between the two sides to ensure platform survival and growth, which involves deciding between charging both sides, letting one side access the platform for free, or subsidizing one side.

A vast platform economics literature considers several strategic issues such as price structure [9, 36], openness [37–40] and many others. However, the platform economics literature does not consider pricing algorithms used by platforms in platform competition settings [6] as we aim to do in this article.

It is important to clarify that some recent work analyzes a slightly different problem than ours: what happens when platform participants adopt algorithmic pricing. For instance, [41] considers a platform that shows sellers to users, and it finds that sellers using Q-learning set supracompetitive prices. Related empirical research on Amazon finds that repricing algorithms may reduce welfare [42]. There is also evidence of higher prices when algorithms compete on Bol.com [43].

However, we still lack a proper theoretical analysis of how algorithms deployed by platforms set prices. Without such a theoretical framework, we do not know whether the benchmark for comparison is too low or too high, as the equilibrium behavior of algorithms may deviate from that of classical models. In other words, we do not know how algorithms set platform prices and how their behavior is affected by the presence and strength of indirect network effects. Additionally, it is unclear whether all algorithms behave similarly enough to derive global insights. Thus, this paper aims to fill these two gaps: it illustrates how pricing algorithms set prices when used by competing platforms and gives an account of the algorithms' characteristics that influence those prices.

3 Model

Our model considers two platform firms that compete using pricing algorithms. We adopt the seminal two-sided market model of Armstrong [11]. There are two groups (sides) of agents whose population is normalized to one and are uniformly distributed on a line with unit length, and the two platforms are at locations 0 and 1 of the unit line. Each agent participates only in one platform; each agent on each side compares the two platforms and chooses the one with higher

utility. Agents value the presence of the other group on a platform. An agent on side j located at x receives the following utility from joining platform i :

$$u_j^i = v + \alpha_j n_{-j}^i - p_j^i - t |x_j - l_i| \tag{1}$$

Parameter t represents the mismatch cost and measures the intensity of competition between the competing platforms. A smaller value of t implies a higher competition intensity (lower level of differentiation). The expression $t |x_j - l_i|$ reflects the distance between the agent and the platform weighted by the mismatch cost and represents the heterogeneity of agent tastes. The parameter v captures the intrinsic value of the platform; we assume v is high enough to guarantee that all agents participate. The parameter αn_{-j}^i captures the network effect (i.e., how valuable is the presence of the opposite group on the platform), where n_{-j}^i is the number of agents on the other side of the platform i , and α represents the valuation of an extra agent on the other side, which we assume is $\alpha \in [-1, 1]$. Intuitively, positive values represent those cases where one side values more users on the other side. For example, sellers are better off with more buyers in a market and vice versa. Conversely, negative values represent those cases where one side prefers fewer users on the other side. For example, users may dislike advertising and prefer platforms with fewer advertisers [44, 45]. Lastly, p_i represents the price paid by agents. Formally, the side j demand for platform i is as follows:

$$n_j^i = \frac{1}{2} + \frac{1}{2} \frac{\alpha_j (p_{-j}^{-i} - p_{-j}^i) + t (p_j^{-i} - p_j^i)}{t^2 - \alpha^2} \tag{2}$$

In all the scenarios, the profit of platform i is $\pi^i = (p_j^i - c_j^i) n_j^i + (p_{-j}^i - c_{-j}^i) n_{-j}^i$, where c_j^i is the marginal cost on side j of platform i (i.e., the cost of attracting an additional user). The theoretical equilibrium in this model is defined by $p_j^i = c_j^i + t - \alpha_{-j}$ and equilibrium profits equal to $\pi^i = t - \frac{\alpha_j + \alpha_{-j}}{2}$. For simplicity and without loss of generality, we assume a zero marginal cost ($c_j^i = c_{-j}^i = 0$), thus the equilibrium is defined by two parameters, the mismatch cost (t) and network effects (α). This theoretical equilibrium provides the benchmark prices and profits for our analysis that follows. The theoretical market model allows us to test the algorithms in a controlled environment where the effect of all model components is known, facilitating the interpretation of results. Furthermore, the Armstrong model is used widely in platform economics and, therefore, constitutes a robust framework to test novel strategies such as the introduction of algorithmic pricing. Finally, note that the algorithms will play this game repeatedly, but this does not make it a repeated game because the algorithms cannot condition their play on the competitor’s previous play.

4 Algorithms and parametrization

This research considers Q-learning and PSO, two sophisticated AI-enabled pricing algorithms. We selected these two because (i) they are widely used in the literature, (ii) both are characterized by parameters that allow a clear interpretation of the results, and (iii) they are linked to economic concepts. Additionally, Q-learning and PSO have been studied extensively in experimental economic research, which provides us with a framework for comparison [7, 24–26, 46].

4.1 Q-learning

Q-learning is a method for finding an optimal policy with no prior knowledge of the inherent structure of the game. The method works by iteratively estimating the Q-function $Q_i(s, a_i)$, which represents the cumulative discounted payoff of taking action $a_i \in A$ in state $s \in S$ by agent i . Q-learning starts from an arbitrary value, which is updated at each iteration. After choosing a_i^t in state s^t , the algorithm observes the payoff π_i^t , the next state s^{t+1} , and updates $Q_i(s, a_i)$ for $(s, a_i) = (s^t, a_i^t)$ following the learning equation:

$$Q_i^{t+1}(s, a_i) = (1 - \alpha)Q_i^t(s, a_i) + \alpha \left[\pi_i^t + \delta \max_{a \in A} Q_i^t(s^{t+1}, a_i) \right]$$

The weight $\alpha \in [0, 1]$ is the learning rate and δ the discount rate, or how past performance influences current Q values. The algorithm chooses the action with the highest Q-value in the current state with probability $1 - \epsilon$ (exploitation mode) and randomizes uniformly across all possible actions with probability ϵ (exploration mode). At the start, given the lack of knowledge about the game, the algorithm should explore widely, but over time, the algorithm must start exploiting the best outcomes it has found. To reproduce such a behavior, we posit a time-declining exploration rate: $\epsilon = (1 - \beta)^t$, where $\beta > 0$ is a parameter. The algorithm will start by randomly selecting actions (exploration). The larger the β , the faster the exploration vanishes and the larger the likelihood of choosing the best action found so far (exploitation).

4.2 PSO

PSO generates random points in a multidimensional space (particles) that move towards an optimal solution by sharing information about which points perform better. In our application, we assume that each firm will consider a set of k potential prices, where k is the number of particles. The position of each particle in the real numbers represents a price. The algorithm will test each price and share which performs best (higher profits) with all the particles. Over time, the position of each particle changes as it moves toward those positions that perform best. This evolution is given by $p_{i,t} = p_{i,t-1} + v_{i,t-1}$, where $v_{i,t}$ is the evolutionary

velocity, which is determined by the best position the particle has found before (p_i^l) and the best position any other particle has found (p^g):

$$v_{i,t-1} = wv_{i,t-2} + l_1u_1(p_i^l - p_{i,t-1}) + l_2u_2(p^g - p_{i,t-1})$$

where w is an inertia weight factor that represents how past actions (prices) influence the current action (price); l_1 and l_2 are learning parameters and are called self-confidence and swarm confidence factors, respectively; and u_1 and u_2 are $U(0,1)$ random numbers. Like in Q-learning, there is a trade-off between exploration and exploitation controlled by the inertia weight. We assume a similar time-declining exploration rate represented by a declining inertia weight given by $w_t = (1 - w_o)^t$ where w_o is a constant initial decrease parameter.

4.3 Parametrization

We adopt the baseline parametrization of related literature [7, 8, 47]. This facilitates the comparison with other studies, and there is no additional theoretical justification. A novelty of our approach, simplifying the problem at hand, is that algorithms cannot condition their play on the past play of their opponents. In other words, algorithms cannot recall the actions taken in the previous period, which is the basis for sustaining punishment strategies. Instead, algorithms observe other's actions, learn which prices work best, and set prices accordingly. For example, if algorithms set prices equal to 1 and gain profits 2 in iteration t , they use this information to update the payoff matrix but, in this update, they do not consider what they did and earned in $t - 1$. This assumption simplifies the computational complexity but does not remove the strategic interactions between the platforms, as the price set by a platform depends on the price set by its competitor, which affects the profits. Furthermore, each platform pursues its own objectives (profit maximization) uncoordinated and independently, which implies that there are simultaneous learning processes that influence each other.

This assumption is important to separate the effects that the possibility of implementing additional strategies (such as tit-for-tat or grim trigger strategies) may have on the ability to set prices in a multi-sided market. The strategic interdependence between the platforms is still present but we only remove collusive behavior issues that stem from the possibility of conditional play (algorithmic memory) [7, 30] that are outside the main focus of this research. The main focus is understanding the interaction and performance of the studied pricing algorithms under platform competition. Formally, if we define memory as the set of all past prices in the last k -periods: $s_t = \{p_{t-1}, \dots, p_{t-k}\}$, Where k is the length of the memory. Our assumption is equivalent to $k = 0$. In addition, Sect. 8 provides an extension that relaxes that assumption.

This simplification has the consequence that parameters do not play the same role. Therefore, we must reproduce the same conditions as those in the literature in our framework (same exploration and learning rate, for example). Then, the baseline parametrization we use is as follows. For prices, we take the minimum

($p_{min} = -1.5$) and maximum ($p_{max} = v = 1.75$) feasible prices, and we build a set A of feasible prices to be given by m equally spaced points in the interval $[p_{min}, p_{max}]$, where $m = 50$. For the learning and exploration parameters, we assume $\alpha = 0.15$ and $\beta = 0.01$, meaning that each cell in the Q-matrix is visited almost 20 times just by random exploration. Finally, we assume $\delta = 0.95$. On the other hand, the baseline PSO algorithm consists of 5 particles ($k = 5$) with $l_1 = l_2 = 1.75$, and $w_0 = 0.025$. We also limit the evolutionary velocity range, $v_i \in [-0.3; 0.3]$ to avoid jumping between corner solutions [25].

To test the robustness of the results found under this parameterization, we perform a sensitivity analysis in Sect. 6, where the impact and intuition of those parameters are further analyzed.

For each parameter combination, we run 30 simulations (experiments) to remove any stochastic noise. Each simulation is initialized with random prices in the action set. Each simulation runs 50,000 iterations, enough to reach a stationary state in all cases analyzed (no deviation is present in the last 10,000 iterations). All the results we present are an average of 1000 last iterations of each experiment. For all parameter combinations, this is sufficient to ensure a thorough exploration and subsequent exploitation of the results. We compare these simulated results with the theoretical equilibrium defined previously. In all simulations, we observe that the algorithms converge to this equilibrium (or to corner solutions), convergence being understood as reaching these points and remaining there indefinitely.

5 Computational experiments and results

Our model considers two competing platforms, each using one of three pricing algorithms: Q-learning, PSO, and price matching. Therefore, we need to compare platform prices and profits in six cases corresponding to all the pair-wise combinations of pricing algorithms used by the two platforms.

First, we present the results for the two different regions of the model. The first region has a unique stable equilibrium. The second region has corner solutions, and it is unclear how the algorithms may react in such a situation. Then, we explore the role of asymmetric network effects.

5.1 Algorithmic platform pricing with a unique interior equilibrium

The Armstrong model has a unique interior equilibrium when the mismatch cost is high enough compared to network effects. In what follows, we compare the differences between the simulated and benchmark prices and profits when this condition holds.

Figures 1 and 2 show the pairwise comparison of the three algorithms. Two results stand out from these simulations. First, price-matching algorithms lead to supracompetitive prices even without the possibility of conditioning its play on the opponent's past play. On the one hand, the possibility of tacit collusion with these types of algorithms seems to be robust to the presence of network effects. Price

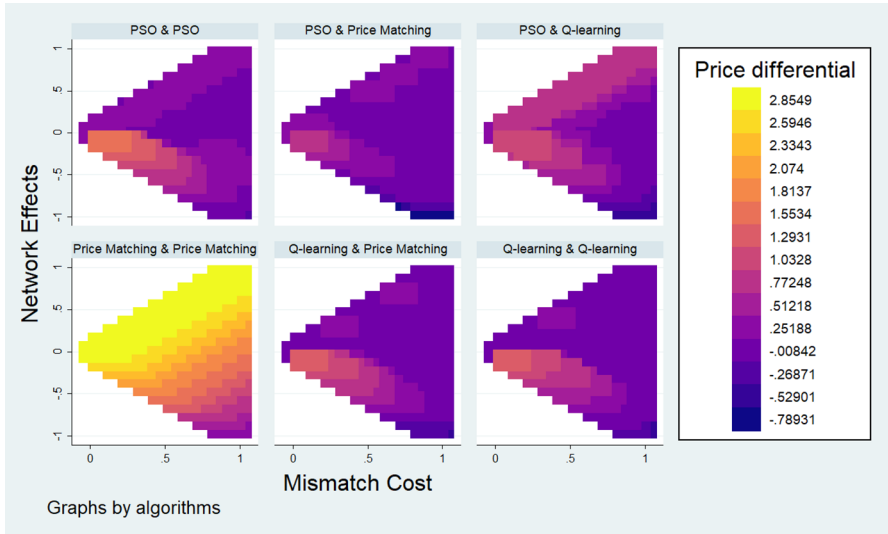


Fig. 1 Difference between simulated and benchmark platform prices

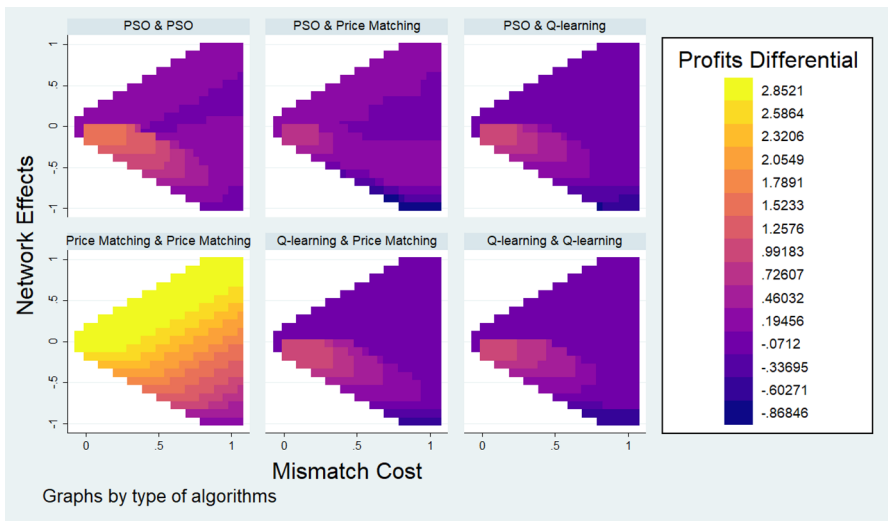


Fig. 2 Difference between simulated and benchmark platform profit

undercutting only happens if the market is growing [20]. Intuitively, by undercutting, platforms may create sufficient incremental sales to compensate for the lost revenue from its installed base. However, once the market is covered, this incentive disappears, and prices are expected to remain as high as possible. On the other hand, Q-learning is influenced by the discretization of space, and this is especially relevant when network effects increase or mismatch costs decrease. In these cases,

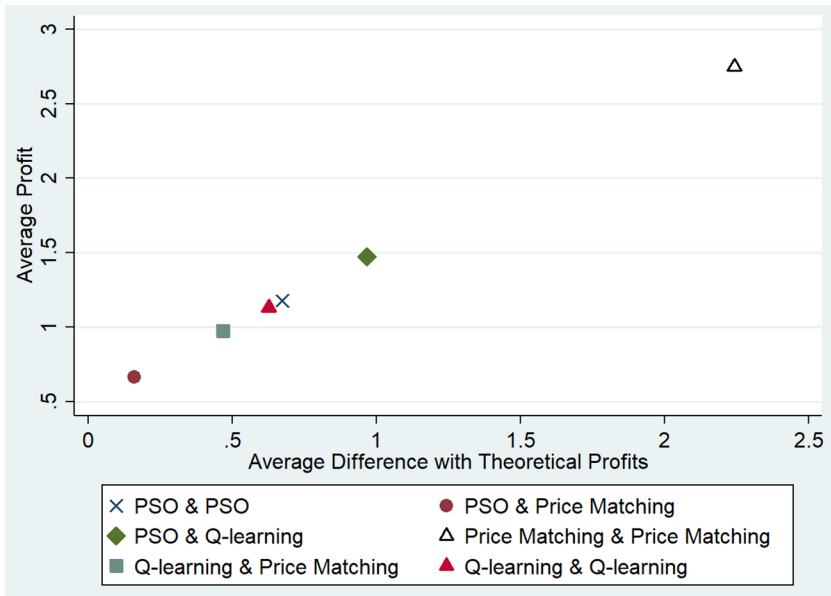


Fig. 3 Average profit by deviation from theoretical profit

the market becomes more sensitive to feedback loops between sides (i.e., more competitive), and there is downward pressure on prices that algorithms do not recognize. Discretizing the state space implies that price changes are less informative, and the algorithm is less able to learn this feedback. In contrast, PSO does not suffer from this problem, but since it does not operate at an infinitesimal level, this error is present but mitigated. Nonetheless, when different algorithms are combined, they are more likely to experiment differently, but it does not guarantee a better internalization of these feedback loops, which leads to small supracompetitive prices.

Second, our simulation outcomes are close to the benchmark results in most cases. Therefore, even the simplest forms of these algorithms learn to set prices in environments where consumer decisions are interrelated. However, when network effects are negative (i.e., cases in which users prefer to join platforms with smaller users bases, e.g. congested platforms), algorithms exhibit deviations from the benchmark. If mismatch cost is low, these deviations are positive, indicating that prices and profits are higher than the benchmark. In contrast, we find almost no deviation when network externalities are positive. Note that prices should increase when negative network effects are present. Intuitively, if platforms are trying to avoid attracting people, they use prices as a deterrent. If mismatch costs are low, this effect needs to be greater to be an effective deterrent. What we observe here is the effect of the imperfect learning of network effects, which is more pronounced in this case as the direction of the effect calls for a higher price and the tendency to price above the optimum is slightly reinforced.

Figure 2 shows some deviations but does not clearly indicate which combination of algorithms leads to higher profits or when. Figure 3 compares the average

profit and the average difference with theoretical profits generated by each pair of algorithms. The first result is the dominance of price-matching algorithms when combined with other price-matching algorithms. However, if price-matching algorithms face another type of algorithm, all the supracompetitive profits disappear, and the simulated results are close to benchmark values. This result suggests that price-matching algorithms may not be a good option when competitors use other algorithms. However, this is not the case with other algorithms, such as Q-learning or PSO, whose behavior is more stable.

Despite the small price differences, we observe that the algorithms deviate positively when considering their impact on profits, highlighting that they capture extra profits. This suggests that, on average, the implementation of algorithmic pricing is positive for platforms, as it is more profitable than traditional pricing. The reasons why this happens are manifold and depend on each combination of algorithms. For example, price-matching algorithms tacitly collude when it is not possible to expand the market [20]. In the case of Q-learning and PSO, as feedback loops become more relevant and algorithms learn more imperfectly, supracompetitive pricing becomes more prevalent, leading to higher profits. When two algorithms that exhibit this problem are combined (PSO vs Q-learning), learning does not improve but worsens.

We also find that simulated profits are close to the benchmark profits in all cases in which price-matching algorithms compete with other types of algorithms. This suggests that the presence of a price-matching algorithm increases price competition even when sophisticated algorithms are present.

Proposition 1 *Algorithms can lead to supracompetitive prices in platform markets. However, such deviations are slightly above the theoretical benchmark, suggesting that the algorithms make small upward errors that may generate profits greater than or equal to those of a rational agent.*

5.2 Algorithmic platform pricing with multiple equilibria

When network effects are extreme, one platform can dominate the market by attracting all consumers. However, both platforms may coexist with zero profits if they imitate each other in pricing. Those two outcomes are possible, but we do not know which is more likely. This situation of multiple equilibria challenges algorithms as they need to coordinate in one equilibrium. Figures 4 and 5 show the differences between simulated and interior equilibrium to facilitate a comparison with the previous section. Significant deviations are apparent this time because the algorithms do not always coordinate in the same way.

In contrast to the previous section, prices differ significantly in each case, suggesting that algorithms do not always coordinate (Fig. 4). Although the results vary according to the algorithms used, there is a common theme. Prices go in the opposite direction of what the benchmark suggests. In the benchmark model, positive network effects call for a price reduction, and the opposite is true if network effects are negative. However, algorithmic pricing does not follow these

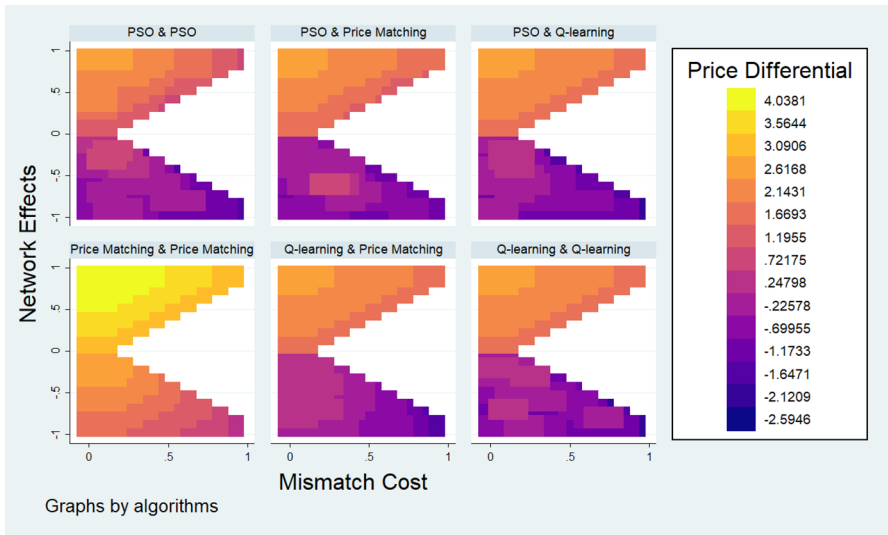


Fig. 4 Difference between simulated prices and interior equilibrium platform prices

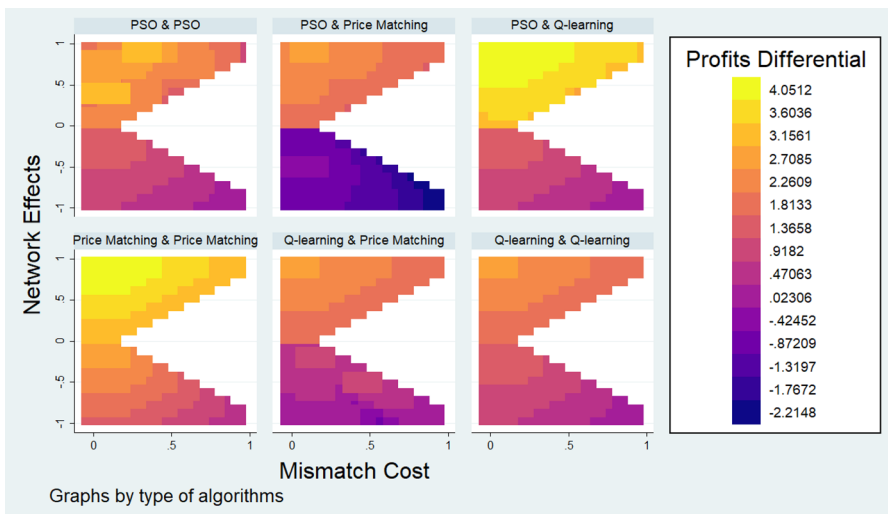


Fig. 5 Difference between simulated profits and interior equilibrium platform profits

intuitions, and we note a clear distinction between the regions with positive and negative network effects. In other words, algorithms autonomously learn to deviate to other (more stable) equilibria, such as the corner solutions. Note that, in all these cases, competition is extreme in the sense that network effects are stronger than differentiation (mismatch costs). This implies that price undercutting creates feedback loops that can attract all consumers on one side. In these cases,

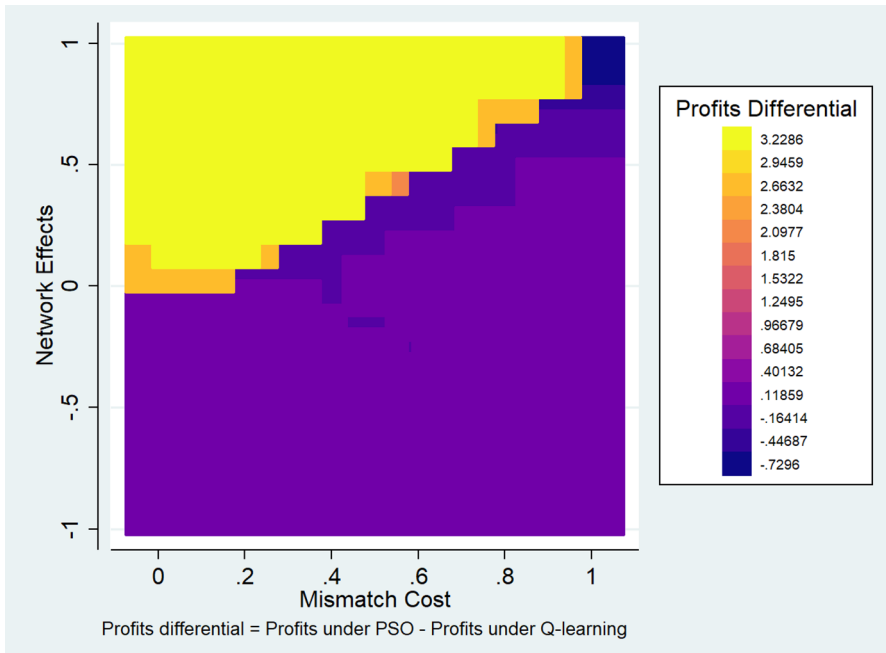


Fig. 6 Profit differential between PSO and Q-learning when competing against each other

the interior equilibrium is unstable as there is an incentive to deviate. What we observe here are the consequences of this process. Once a platform conquers the entire market, it sets prices without considering network effects as it has driven its competitor out of the market. This effect is stronger in the case of PSO because its granularity helps it learn feedback loops better than Q-learning, whose discretization of the state space makes it harder.

Considering the two regions together, we discover an interesting insight regarding the algorithm characteristics. In our simulations, PSO only overperforms Q-learning when network effects are positive, and there are multiple equilibria (yellow area), as shown in Fig. 6. In all other cases, Q-learning matches PSO or overperforms it. This situation highlights that the performance we observe in Fig. 3 is not fixed, and depending on whether we consider specific cases, such as the yellow area of Fig. 6 or all the potential cases of the benchmark model, the profitability of the algorithms will vary. This is a consequence of how each algorithm learns about network effects. When those network effects are crucial to determining the winner, the algorithm with the highest granularity has an advantage (PSO). However, when the strength of the network effects is moderate, other algorithms may have an advantage because their coexistence is not at risk (Q-learning), allowing supracompetitive prices to be sustained.

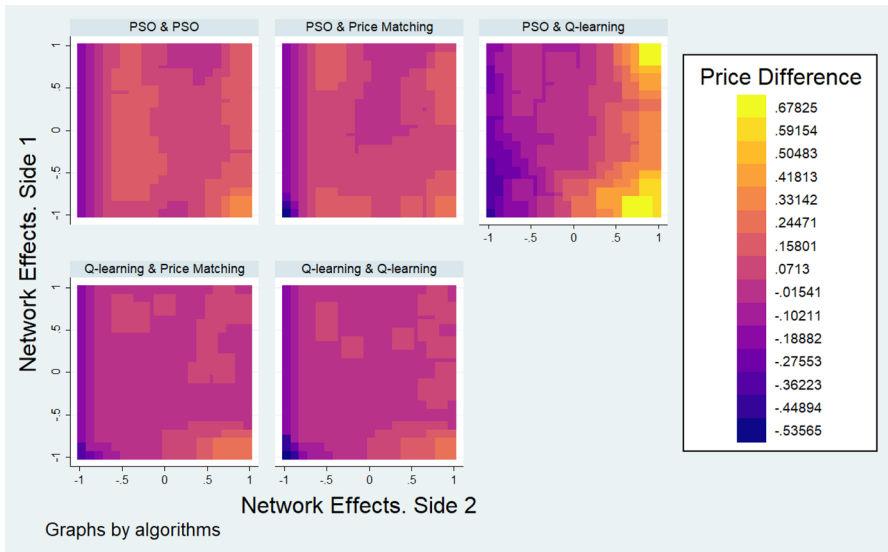


Fig. 7 Difference between simulated and benchmark platform prices. We do not depict the results for pair “Price Matching & Price Matching” as it reproduces the collusive price scheme and does not allow us to compare the rest of the cases easily

Proposition 2 *As network effects increase, algorithms with greater granularity in their action space have an advantage over other algorithms because they can set prices that drive out competitors.*

5.3 Algorithmic platform pricing with asymmetric network effects

In previous scenarios, we considered symmetric sides with respect to network effects ($\alpha_j = \alpha_{-j}$). In what follows, we examine asymmetric network effects to determine the robustness of prior insights and whether the algorithms can set the optimal price structure (Fig. 7). We set the mismatch cost equal to one to focus on the impact of asymmetric network effects. The intuitions are qualitatively the same for other values of mismatch cost.

Figure 7 shows that prices differ little from the benchmark. This result implies that algorithms learn to set the correct price structure. A few exceptions in which algorithms make significant errors cluster around the figure’s corners. That suggests that algorithms have problems setting the correct price structure only in extreme cases of strong network effects, especially when Q-learning and PSO compete, which is the case with the largest price differences. These results highlight that it does not matter having asymmetric network effects; the algorithms can deal with that. What matters is the relationship between network effects and mismatch costs that we observed in the previous sections.

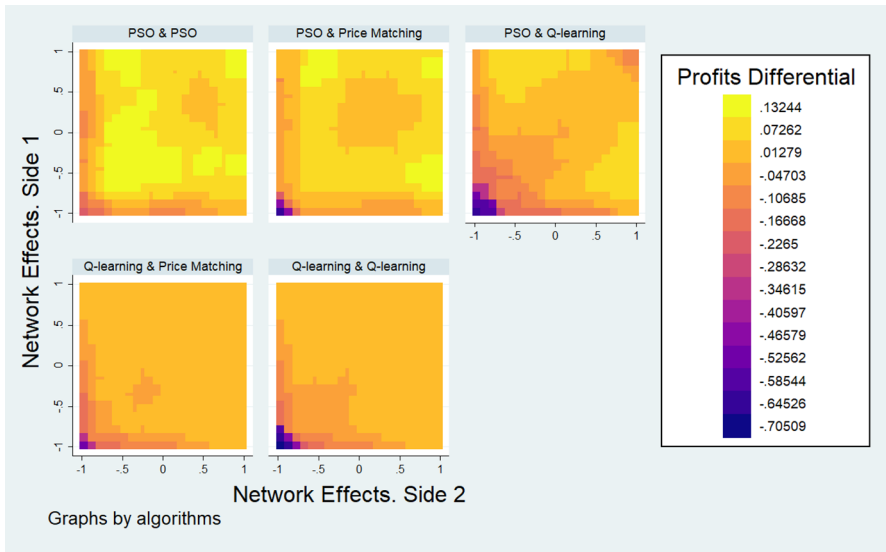


Fig. 8 Profit differences with asymmetric network effects

This insight is better shown in Fig. 8, reinforcing the idea that algorithms learn to handle network effects. However, we find significant deviations when both network effects are extremely negative, in which profits are significantly lower than in the benchmark. This could be the case for platforms that suffer from congestion. For example, e-commerce platforms that rely on shipping services that quickly become saturated can generate growth aversion on either side of the platform, and algorithmic pricing can worsen this situation. Only in these situations may prior insights require nuances, but it will depend on whether we have the case with stable interior equilibrium or not. In experiments with other mismatch costs, we observe that these cases tend to corner solutions where only one platform survives, as noted in the previous section.

Proposition 3 *Algorithms can learn about asymmetric network structures and set prices accordingly.*

Although algorithms may make some errors in setting the correct price structure, those errors are minimal. Figure 9 shows that profits are indistinguishable from the benchmark model, suggesting almost no deviation from the benchmark. This result also implies that asymmetries in network effects are not the source of price and profit deviations documented previously (Fig. 3) but the interaction between mismatch costs and network effects. This insight is especially relevant for companies seeking to implement algorithmic pricing, as it suggests that market characteristics determine which algorithm is more profitable. However, as we illustrate in the following sections, this is not the only aspect to consider.

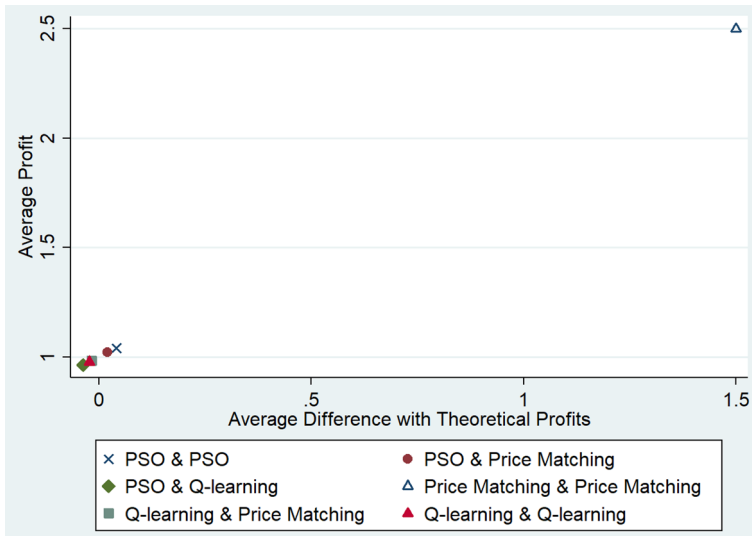


Fig. 9 Average profit by deviation from benchmark profit with asymmetric network effects

6 Sensitivity to algorithm parameters

Parameterization can affect the results in terms of convergence, accuracy, or computational complexity, among others [48, 49]. In this sensitivity analysis, we explore the parameter grid of Q-learning and PSO to show (i) that parameterization matters but is not critical in terms of equilibrium prediction and (ii) some parameters are more relevant than others in defining price behavior.

6.1 Q-learning

Like other reinforcement-learning algorithms, Q-learning adapts its behavior to past experiences, taking actions that have proven to be successful more often. To be able to do that, it learns by experimentation. As defined previously, two parameters control these two activities: α controls learning and β controls experimentation.

In principle, α may range from 0 to 1 but it is well known that high values of α disrupt the learning process because the algorithm quickly forgets what it has learned [7]. Accordingly, we consider 20 equally spaced points in the interval $[0, 0.1; 0.28]$. The experimentation parameter (β) suffers from a similar problem in that experimentation is needed, but too much may introduce noise that complicates the learning process. To understand what this parameter represents, let us consider how likely exploration is after several iterations. In our baseline scenario, all cells are expected to be randomly visited almost 20 times by the end of the simulation. To account for the impact of exploration, we consider 20 equally



Fig. 10 Sensitivity analysis: Q-learning parameters

spaced points in the interval $[0.01;0.15]$, equivalent to assuming that all cells are expected to be visited 20 to 2 times on average.

Although these two parameters have attracted the most attention lately [7, 26], the action space size is also essential. To illustrate this point, we consider three cases in which we subdivide the set of feasible prices into 10, 20, and 30 spaces ($m = 10, 20, 30$).

Figure 10 shows all combinations of the previous parameters and depicts the deviations in the simulated profits with respect to theoretical results in all the market simulations analyzed in previous cases. Although we observe variations, these are relatively small and tend to be mitigated the larger the action space.¹ Similar results can be found in other works [7, 26]. Although there is evidence that insufficient exploration can lead to seemingly collusive results [30, 50], this effect is significantly mitigated because we assume that algorithms cannot condition their play. Therefore, we can trust the robustness of previous insights regarding Q-learning.

6.2 PSO

Although the PSO algorithm has multiple variations, a common feature is its reliability in solving multidimensional optimization problems. Since our problem is unidimensional (prices), choosing a particular version is not critical. However, the algorithm’s performance is highly dependent on the chosen parameter values, as was the case for

¹ We also performed a similar experiment with the discount parameter, which can take values from 0 to 1, but the variability of the results is even lower. The reason is that platforms cannot condition their play on competitor’s past actions, a result already highlighted by [7]

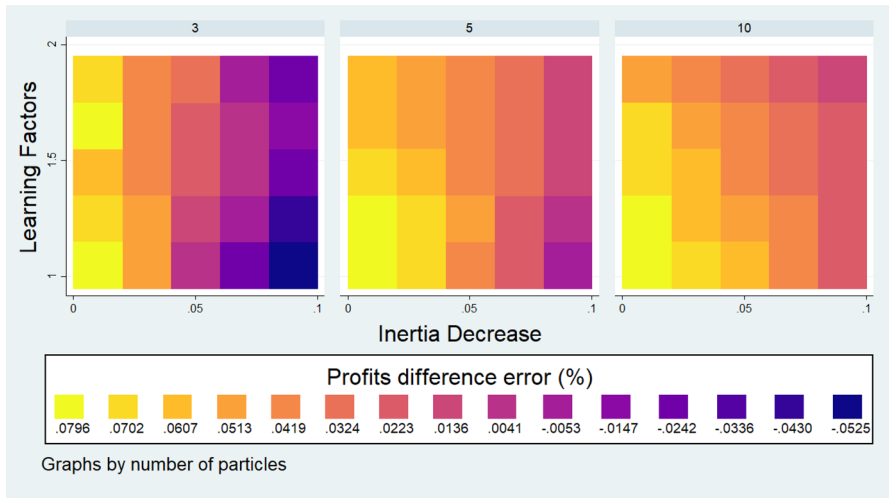


Fig. 11 Sensitivity analysis: Particle Swarm Optimization algorithm

Q-learning. In this sense, the learning parameters and the inertia weight, which controls the trade-off between exploration and exploitation, are the most relevant [49].

An interesting feature of PSO that contrasts with other approaches is that the exploration/exploitation parameter must be chosen along with the learning factors and should not be chosen individually. In this sense, we consider 10 equally symmetric spaced intervals between 1 and 2 for the learning parameters, $l_1, l_2 \in [1, 2]$ [51]. Following classical models, inertia weight linearly decreases over time [52]. We consider 10 equally spaced values between 0.01 and 0.09 for this declining parameter, $w_0 \in [0.01; 0.09]$. Like the action space in Q-learning, the number of particles in PSO is also critical. Therefore, we consider three additional experiments with 3, 5 and 10 particles ($k = 3, 5, 10$).

We find little variation in the PSO results when comparing different parameter variations, see Fig. 11. Note that these results aggregate the multiple scenarios from the previous sections. However, it is interesting to highlight that the decrease in inertia plays a key role, especially when the number of particles is small. This highlights the need to consider the parametrization of PSO altogether [49]. In general, the insights from prior sections are robust, given the small variation in results.

However, there is no guarantee that this apparent robustness will hold in other market frameworks; previous evidence in Q-learning and PSO experiments has highlighted that the performance of algorithms also depends on the problem at hand.

7 Platforms choose algorithms

What algorithm should a platform adopt to maximize profit in a competitive setting? To explore that question, we make the pricing algorithm choice endogenous. In particular, we study a two-stage game in which platforms choose algorithms

Table 1 Platform profits in a reduced-form algorithmic pricing game. All games

Firm 1	Firm 2					
	F2: Q-learning		F2: PSO		F2: Price-matching	
F1: Q-learning	0.68	0.68	0.82	0.65	0.71	0.76
F1: PSO	0.65	0.82	0.82*	0.82*	0.67	0.71
F1: Price-Matching	0.76	0.71	0.71	0.67	2.7*	2.7*

*Denotes an equilibrium outcome. The {Price-Matching, Price-Matching} pair is computed following [20] and [21]

simultaneously in the first stage, and then they compete by setting prices in the second stage. Table 1 summarizes the profits for each pair of algorithms considering all the cases of Sect. 5.1, which are the most widely studied in the platform literature. For this exercise, we do not consider how costly each algorithm is to implement. We focus only on the average profits they generate. Although this is a partial view, the objective here is to illustrate how the difference in pricing behavior can translate into differences in profits, which can influence the adoption decision.² Buchali et al. [31] performs a similar exercise with different pricing rules for logit and linear demands.

Table 1 highlights the presence of two Nash equilibria in pure strategies, one in which both firms adopt *PSO* and one in which both use *Price-Matching*. The result predicts that both sophisticated and naïve algorithms are likely to be used by platforms. Furthermore, this result shows that even if we make extreme simplifications, such as considering a theoretical model and addressing only average profits, we have multiple equilibria. It may seem surprising that Q-learning is not an equilibrium in this game, given that it is the most studied algorithm in the literature so far. However, there is preliminary evidence that such an algorithm would not be the optimal choice in multiple cases [30, 31].

The {*Price-Matching*, *Price-Matching*} equilibrium gives the highest profits to the platform firms, although price matching is a naïve algorithm. If one looks beyond the context of platforms, retailers in the US widely use price matching, and it can be an equilibrium when these rules compete with Q-learning in linear and logit markets [31]. On the other hand, PSO is also a potential equilibrium, and its widespread use justifies its analysis and suitability as a pricing algorithm [53].

Interestingly, there is no asymmetric equilibrium with firms adopting different algorithms. This result highlights that asymmetric equilibrium may require firm heterogeneity or other types of algorithms. All in all, what is clear is that even in simple environments, platforms face a coordination problem in choosing algorithms.

While this comparison is not exhaustive, our analysis shares some insights with [30] that found that Continuous-Actor-Critic (CAC) algorithms outperform Q-learning. Similarly, in our setting, the adoption of Q-learning seems suboptimal.

² Implementation cost can differ greatly between algorithms and is a dimension that must be considered. However, since we focus on addressing the performance of the algorithms and not their implementation, we abstract from this aspect. We thank one of the reviewers for raising this point.

Table 2 Platform profits by pair of algorithms when algorithms have memory

Firm 1	Firm 2					
	F2: Q-learning		F2: PSO		F2: Price-matching	
F1: Q-learning	0.94*	1*	0.8	0.77	0.75	0.75
F1: PSO	0.77	0.8	0.75	0.76	0.74	0.74
F1: Price-Matching	0.75	0.75	0.74	0.74	2.7*	2.7*

*Denotes an equilibrium outcome. The {Price-Matching, Price-Matching} pair is computed following [20] and [21]

Moreover, our result that Price-Matching and PSO are the best algorithms is unlikely robust. For instance, it would be reasonable to assume that the adoption decision also occurs in a repeated game setting, where the manager must decide whether to continue using a specific algorithm or change it in certain periods. This simple modification will likely alter our conclusions about how many equilibria exist and which algorithms are part of an equilibrium.³

Proposition 4 *There is no one algorithm that is always best, and multiple equilibria can be found. Platforms face a coordination problem.*

Note that this section focused on the region with a unique and global price equilibrium. This coordination problem persists in cases with multiple equilibria. If we consider the cases of Sect. 5.2, the results are robust, and the conclusion is the same.

8 Extension: conditional play

In our previous analysis, we assume that algorithms could not condition their play on competitor's past play (also known as algorithmic memory). We introduced such an assumption to separate out the effects that the possibility of implementing additional strategies (such as tit-for-tat or grim trigger strategies) may have on our results. In other words, do the results change if the algorithms can condition their play?

Assuming that the algorithms can condition their play does change the results. First, the new profit levels are generally higher than in the previous cases (Table 2). This happens because algorithms learn that setting higher prices may be profitable even in the face of deviations. Even if competitors deviate and reduce their prices, in the long run, it may be profitable to set higher prices and reduce them only if the competitor does so. Algorithms seem to learn to play a tit-for-tat strategy. In other words, we observe that prices are above benchmark levels, and if the competitor deviates to lower prices, the algorithm follows. Conversely, if the competitor raises its price, the algorithm also follows. Whether this is an actual tit-for-tat strategy is beyond the scope of this paper, but it is a possibility [7].

³ We thank one of the reviewers for providing us this example.

Second and most importantly, there is a change in the equilibria. In particular, the {PSO, PSO} strategy is no longer an equilibrium; instead {Q-learning, Q-learning} becomes one. Algorithmic memory makes Q-learning more prone to focus on equilibria other than one-shot. In contrast, PSO does not markedly deviate from the one-shot equilibrium. This result further emphasizes that the optimal set of algorithms in a platform market depends on the market structure and algorithmic characteristics. Firms interested in adopting pricing algorithms should consider that the profitability of introducing algorithmic pricing depends on the market structure (differentiation, network effects) but also on the type of algorithm chosen (PSO, Q-learning, price matching), its characteristics (e.g., conditional play) and what competitors do.

The analysis of this section represents a reduced-form game in which two firms must choose their pricing algorithms simultaneously and have complete information. This assumption may seem strong, but digital firms likely have the knowledge and expertise to know how each algorithm performs in real situations. On the other hand, the idea of simultaneous moves represents the lack of coordination between different firms. Nevertheless, comparing Tables 1 and 2 provides valuable insight into the incentives to adopt AI pricing solutions, and future research could relax some of the assumptions. Note that we do not consider adoption costs. While algorithms with more complex behavior (such as conditional play) are more likely to be an option, they are also likely to be more expensive. Although more sophisticated algorithms can implement more complex strategies, our work should reassure managers interested in simple algorithms because we show that they are capable of pricing competitively. In our framework, all algorithms generate profits greater than or equal to the theoretical model.

9 Discussion

We discuss theoretical and managerial implications and identify opportunities for future research.

9.1 Theoretical implications

This research analyzes how platform adoption of pricing algorithms affects competition outcomes (prices and profits). We consider three algorithms (PSO, Q-learning, and price-matching algorithm) used by platforms and compare all the pair-wise combinations. At the same time, the theoretical prices and profits of the Armstrong model provide a benchmark.

We find that algorithms can effectively set prices close to the benchmark values in most cases. Notably, the results differ significantly in the region with negative network effects. This is especially relevant for platforms subject to negative network effects, such as social media platforms where the sharing of fake news is extensive, as it reduces the value of new users to advertisers, and users may view advertisers as promoters of that fake news, creating negative cross-network effects. Or an e-commerce platform that may experience indirect negative network effects that call

for fewer users and sellers if the quality of shipping and delivery declines as the platform grows. In these cases, caution is advised. Furthermore, we find that when different algorithms compete with each other, the algorithms with greater granularity in their action space have an advantage because they can set prices that may drive out competitors if price competition is extreme.

When considering multiple equilibria, algorithms do not always coordinate. The outcome depends on the characteristics of the market, the competitors' algorithms, and the parameterization. Therefore, determining the optimal algorithm for a platform in all cases is challenging. We illustrate that the profitability of algorithms depends on which cases we consider. Platforms will not choose the most profitable algorithm if they do not adequately analyze all the potential cases.

Lastly, we address which algorithm is more likely to be adopted by profit-maximizing platform firms. Our results suggest that two equilibria are possible. One in which platforms adopt PSO and another in which platforms adopt price-matching. However, this result is altered with the introduction of algorithms that condition their play to the immediate past actions of the competitors (i.e., algorithmic memory). In the latter case, PSO is no longer an equilibrium, but Q-learning is. This result emphasizes that choosing the best algorithm is challenging and depends on market characteristics, algorithmic features, and competitors' choices.

9.2 Managerial and policy implications

This paper provides a general framework based on Armstrong's abstract platform competition framework. The goal of our framework is not to capture a particular platform in detail. Instead, it is about abstracting away from the details to identify important platform market parameters and characterize the behavior of pricing algorithms. A manager using our model should first consider what parameter values are closer to their business situation and then check the relevant results. Some general insights are outlined next.

We discuss insights for managers considering algorithmic pricing in their platform strategies. It is clear that algorithms can effectively set prices. Algorithms can autonomously learn the presence and strength of network effects in platform markets and coordinate the two sides accordingly. This insight is encouraging for platform firms that seek more and more automation. It also suggests that the algorithms could be used as an exploratory tool to learn which sides to subsidize. However, profits vary significantly when considering different algorithms, although generally higher than or equal to those expected in theory.

Interestingly, when facing multiple equilibria, algorithms learn autonomously to deviate from those that may not be stable, which further encourages the use of algorithms as pricing tools. Only when platforms face negative network externalities, such as congestion, pricing algorithms may not perform well.

Furthermore, the choice of the algorithm can significantly impact profitability. For instance, price-matching algorithms generate the highest profits but only when facing other price-matching algorithms. Price-matching algorithms are simple to implement, but profits can be lower if competitors adopt different algorithms. In this

sense, companies must also consider what their competitors do. One algorithm can outperform another, but a slight change in the market characteristics may turn the tables. Therefore, companies must consider the market structure, the type of algorithm chosen, its characteristics, and what competitors do when deciding whether and how to adopt algorithmic pricing. Selecting a pricing algorithm is similar to the platform making a strategic investment—a careful evaluation is therefore required.

We now reflect on the business value of AI motivated by our analysis of AI-enabled pricing algorithms. First, the popular business press often talks about algorithms, or AI more general, like it is one homogeneous object with well-defined effects [54]. Our analysis suggests this perception is misleading, and more nuance is needed. In particular, we show that in the context of platforms, the effects of pricing algorithms differ depending on the type of algorithm, the market characteristics, and the design parameters of the competing algorithms. Managers need to take a more nuanced approach that accounts for algorithmic design diversity that can drive diverse market outcomes. Besides managers, policymakers should refrain from enforcing general rules on algorithmic pricing for platforms since the effect of algorithms on competition is far from homogenous. Moreover, sophisticated AI algorithms, which are costly to implement, are not always better than following simple rules like price matching. This again suggests caution about the drive to use AI everywhere in business.

9.3 Limitations and future research

This research evaluates three types of algorithms: Q-learning, PSO, and price-matching algorithms. Future research could compare more types of algorithms and more algorithm design choices. Moreover, future work could analyze additional platform competition frameworks. It could also study more dimensions of algorithmic platform competition beyond algorithmic pricing.

A key assumption in our study and the related literature is the lack of adoption costs. Implementing sophisticated AI algorithms is likely more costly than simple price-matching rules. If sophisticated AI algorithms are prohibitively expensive to adopt, firms will not consider them. However, advances in AI and computing bring those costs lower over time, and more empirical research is needed to understand the structure of those costs. In summary, modeling adoption costs introduces additional complexity that requires a separate future study.

This article assumes that agents are rational and can forecast that a price change on the other side will impact them. However, agents may not be aware of price changes on the other side or may not be entirely rational. In those cases, prices may likely be higher due to the demand specification, but more research is needed. Overall, the growing importance of platforms and algorithms in the economy suggests a strong need for more research into related issues. More broadly, we recommend more research at the intersection of platforms and artificial intelligence.

10 Conclusion

The article builds an agent-based model that studies algorithmic platform pricing under competition. The model integrates algorithmic pricing into the classic Armstrong platform competition model. We evaluate three types of algorithms: Q-learning, PSO, and price-matching. We consider various scenarios, including multiple equilibria, asymmetric network effects, endogenous choice of pricing algorithm, and the possibility of conditioning algorithm play to competitor's past actions. Our simulation study, grounded on game-theoretic economic logic, contributes to the platform literature and the algorithmic pricing literature.

We summarize three main findings. First, the algorithms learn the presence of network effects and set prices accordingly. We find that the interaction of network effects and mismatch (differentiation) costs is crucial to algorithmic pricing. Second, profitability depends on market characteristics and algorithmic features like the possibility of playing conditional strategies. In this sense, the managerial considerations in the traditional economics literature apply, but algorithms also present additional ones. For example, strong network effects are related to the possibility of winner-take-all; when considering algorithms, we should realize that those algorithms with a richer action space have an advantage in terms of profitability. Third, there is no unique equilibrium in which only one algorithm is chosen. In fact, the set of optimal algorithms changes when considering different algorithmic characteristics. Even if we abstract from implementation costs and reduce the decision to a simple game in normal form, we find multiple equilibria. Moreover, this implies that platforms may face an additional coordination challenge, which provides an answer to why multiple pricing algorithms based on different technologies are currently available. In summary, while algorithmic pricing can be a valuable tool for companies seeking to optimize their pricing strategies, it is crucial to recognize the limitations of algorithms and the potential for mismatches between theoretical and actual profits. Companies should carefully consider the market structure, the type of algorithm used, and the potential costs and benefits of implementing algorithmic pricing before making any decisions.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Parker, G., Van Alstyne, M., & Choudary, S. (2016). *Platform revolution*. W.W. Norton & Company.
2. Cusumano, M. A., Yoffie, D., & Gawer, A. (2020). The future of platforms. *MIT Sloan Management Review*, 61(3), 46–54. <https://doi.org/10.4324/9781315598949-15>
3. Hagiü, A. (2014). Strategic decisions for multisided platforms. *MIT Sloan Management Review*, 55(2), 71–80.
4. Rietveld, J., & Schilling, M. A. (2021). Platform competition: A systematic and interdisciplinary review of the literature. *Journal of Management*, 47(6), 1528–1563. <https://doi.org/10.1177/0149206320969791>
5. Jia, X., Cusumano, M. A., & Chen, J. (2021). Multisided platform research over the past three decades: A bibliometric analysis. *International Journal of Technology Management*, 87(2–4), 113–144. <https://doi.org/10.1504/IJTM.2021.120937>
6. Sanchez-Cartas, J. M., & León, G. (2021). Multisided platforms and markets: A survey of the theoretical literature. *Journal of Economic Surveys*, 35(2), 452–487. <https://doi.org/10.1111/JOES.12409>
7. Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10), 3267–3297. <https://doi.org/10.1257/aer.20190623>
8. Sanchez-Cartas, J. M., & Katsamakas, E. (2022). Artificial intelligence, algorithmic competition and market structures. *IEEE Access*, 10, 10575–10584. <https://doi.org/10.1109/access.2022.3144390>
9. Bakos, Y., & Katsamakas, E. (2008). Design and ownership of two-sided networks: Implications for internet platforms. *Journal of Management Information Systems*, 25(2), 171–202. <https://doi.org/10.2753/MIS0742-1222250208>
10. Evans, D. S. (2003). Some empirical aspects of multi-sided platform industries. *Review of Network Economics*, 2(3), 191–209. <https://doi.org/10.2202/1446-9022.1026>
11. Armstrong, M. (2006). Competition in two-sided markets. *The RAND Journal of Economics*, 37(3), 668–691. <https://doi.org/10.1111/j.1756-2171.2006.tb00037.x>
12. Rochet, J. C., & Tirole, J. (2006). Two-sided markets : A progress report. *RAND Journal of Economics*, 37(3), 645–667.
13. Chen, L., Mislove, A., & Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. In *25th International World wide web conference, WWW 2016*. <https://doi.org/10.1145/2872427.2883089>
14. Miklós-Thal, J., & Tucker, C. (2019). Collusion by algorithm: Does better demand prediction facilitate coordination between sellers? *Management Science*, 65(4), 1552–1561. <https://doi.org/10.1287/mnsc.2019.3287>
15. Martin, S., & Rasch, A. (2022). Collusion by algorithm: The role of unobserved actions. *SSRN Electronic Journal*, (March). <https://doi.org/10.2139/ssrn.4060524>
16. Forbes. (2019). In the future, everything will be dynamically priced. Retrieved December 15, 2023, from <https://www.forbes.com/sites/forbestechcouncil/2019/09/10/in-the-future-everything-will-be-dynamically-priced/?sh=4913c873f8f1>
17. Eschenbaum, N., Mellgren, F., & Zahn, P. (2022). *Robust algorithmic collusion*. Retrieved from <http://arxiv.org/abs/2201.00345>
18. Werner, T. (2021). Algorithmic and human collusion. *SSRN Electronic Journal*, (November), 1–61. <https://doi.org/10.2139/ssrn.3960738>
19. Bernhardt, L., & Dewenter, R. (2020). Collusion by code or algorithmic collusion? When pricing algorithms take over. *European Competition Journal*, 16(2–3), 312–342. <https://doi.org/10.1080/17441056.2020.1733344>
20. Zhang, Z. J. (1995). Price-matching policy and the principle of minimum differentiation. *The Journal of Industrial Economics*, 43(3), 287–299. <https://doi.org/10.2307/2950581>
21. Lu, Y., & Wright, J. (2010). Tacit collusion with price-matching punishments. *International Journal of Industrial Organization*, 28(3), 298–306. <https://doi.org/10.1016/j.ijindorg.2009.10.001>
22. Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2023). Algorithmic collusion: Genuine or spurious? *International Journal of Industrial Organization*, 90, 102973. <https://doi.org/10.1016/j.ijindorg.2023.102973>

23. Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science* (pp. 39–43). IEEE. <https://doi.org/10.1109/MHS.1995.494215>
24. Zhang, T., & Brorsen, B. W. (2009). Particle swarm optimization algorithm for agent-based artificial markets. *Computational Economics*, 34(4), 399–417. <https://doi.org/10.1007/s10614-009-9171-9>
25. Maschek, M. K. (2015). Particle swarm optimization in agent-based economic simulations of the cournot market model. *Intelligent Systems in Accounting, Finance and Management*, 22(2), 133–152. <https://doi.org/10.1002/isaf.1367>
26. Klein, T. (2021). Autonomous algorithmic collusion: Q-learning under sequential pricing. *The RAND Journal of Economics*, 52(3), 538–558. <https://doi.org/10.1111/1756-2171.12383>
27. Assad, S., Calvano, E., Calzolari, G., Clark, R., Denicolò, V., Johnson, J., Wildenbeest, M. (2021). *Autonomous algorithmic collusion: Economic Research and policy implications*.
28. Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2021). Algorithmic collusion with imperfect monitoring. *International Journal of Industrial Organization*. <https://doi.org/10.1016/j.ijindorg.2021.102712>
29. Asker, J., Fershtman, C., & Pakes, A. (2023). The impact of artificial intelligence design on pricing. *Journal of Economics & Management Strategy*. <https://doi.org/10.1111/jems.12516>
30. Abada, I., Lambin, X., & Tchakarov, N. (2022). Collusion by mistake: Does algorithmic sophistication drive supra-competitive profits? *SSRN Electronic Journal*, 1–64. <https://doi.org/10.2139/ssrn.4099361>
31. Buchali, K., Grüb, J., Muijs, M., & Schwalbe, U. (2023). *Strategic choice of price-setting algorithms*.
32. Sanchez-Cartas, J. M., & Katsamakas, E. (2022). *Effects of algorithmic pricing on platform competition*. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4027365>
33. Leisten, M. (2022). *Algorithmic competition, with humans*. Retrieved from https://www.researchgate.net/profile/Matthew-Leisten/publication/349681786_Algorithmic_Compensation_with_Humans/links/639753dc484e65005b0541e5/Algorithmic-Competition-with-Humans.pdf
34. Lefouili, Y., & Pinho, J. (2020). Collusion between two-sided platforms. *International Journal of Industrial Organization*, 72, 102656. <https://doi.org/10.1016/j.ijindorg.2020.102656>
35. Rochet, J. C., & Tirole, J. (2003). Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4), 990–1029. <https://doi.org/10.1162/154247603322493212>
36. Parker, G., & Van Alstyne, M. (2005). Two-sided network effects: A theory of information product design. *Management Science*, 51(10), 1449–1592. <https://doi.org/10.1287/mnsc.1050.0400>
37. Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs. devolving control. *Management Science*, 56(10), 1849–1872. <https://doi.org/10.1287/mnsc.1100.1215>
38. Economides, N., & Katsamakas, E. (2006). Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science*, 52(7), 1057–1071. <https://doi.org/10.1287/mnsc.1060.0549>
39. Parker, G., & Van Alstyne, M. (2018). Innovation, openness, and platform control. *Management Science*. <https://doi.org/10.1287/mnsc.2017.2757>
40. Niculescu, M. F., Wu, D. J., & Xu, L. (2018). Strategic intellectual property sharing: Competition on an open technology platform under network effects. *Information Systems Research*, 29(2), 498–519. <https://doi.org/10.1287/isre.2017.0756>
41. Johnson, J., Rhodes, A., & Wildenbeest, M. R. (2020). Platform design when sellers use pricing algorithms. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3691621>
42. Musolf, L. (2022). Algorithmic pricing facilitates tacit collusion. In *Proceedings of the 23rd ACM conference on economics and computation* (pp. 32–33). New York, NY, USA: ACM. <https://doi.org/10.1145/3490486.3538239>
43. Wieting, M., & Sapi, G. (2021). Algorithms in the marketplace: An empirical analysis of automated pricing in E-commerce. *SSRN Electronic Journal*, (September). <https://doi.org/10.2139/ssrn.3945137>
44. Filistrucchi, L., Geradin, D., & van Damme, E. (2013). Identifying two-sided markets. *World Competition*, 36(Issue 1), 33–59. <https://doi.org/10.54648/WOCO2013003>
45. Wilbur, K. C. (2008). A two-sided, empirical model of television advertising and viewing markets. *Marketing Science*, 27(3), 356–378. <https://doi.org/10.1287/mksc.1070.0303>
46. Waltman, L., & Kaymak, U. (2008). Q-learning agents in a Cournot oligopoly model. *Journal of Economic Dynamics and Control*, 32(10), 3275–3293. <https://doi.org/10.1016/j.jedc.2008.01.003>

47. Sánchez-Cartas, J. M., Tejero, A., & León, G. (2021). Algorithmic pricing and price gouging. consequences of high-impact, low probability events. *Sustainability*, 13(5), 2542. <https://doi.org/10.3390/SU13052542>
48. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
49. Engelbrecht, A. P. (2007). *Computational intelligence: An introduction*. Wiley. <https://doi.org/10.1002/9780470512517>
50. Hansen, K. T., Misra, K., & Pai, M. M. (2021). Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms. *Marketing Science*, 40(1). <https://doi.org/10.1287/mksc.2020.1276>
51. Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: An overview. *Soft Computing*, 22(2), 387–408. <https://doi.org/10.1007/s00500-016-2474-6>
52. Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *IEEE International conference on evolutionary computation proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* (pp. 69–73). IEEE. <https://doi.org/10.1109/ICEC.1998.699146>
53. Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering*, 29(5), 2531–2561. <https://doi.org/10.1007/s11831-021-09694-4>
54. Acemoglu, D., & Johnson, S. (2023). *Power and progress: Our thousand-year struggle over technology and prosperity*. PublicAffairs.