

Artificial Intelligence, Algorithmic Competition and Market Structures

J. MANUEL SANCHEZ-CARTAS¹ AND EVANGELOS KATSAMAKAS²

¹Universidad Politecnica de Madrid, Campus de Montegancedo, 28223 Madrid, Spain

²Gabelli School of Business, Fordham University, New York, NY 10023, USA

Corresponding author: J. Manuel Sanchez-Cartas (juanmanuel.sanchez@upm.es)

ABSTRACT The use of artificial intelligence (AI) in the form of pricing algorithms to increase profits is becoming ubiquitous. However, the literature has focused on specific markets and algorithms so far, but it is unclear what happens across algorithms and markets. To analyze the business and economic impact of pricing algorithms, we build a computational model that considers two sophisticated AI algorithms (Q-learning and Particle Swarm Optimization) competing in prices in three different market structures (Logit, Hotelling, and linear demand models). From a social perspective, we find that PSO outperforms Q-learning, which tends to set supracompetitive prices. However, small changes in the algorithm designs may drive them to set more competitive prices, implying that a proper analysis of algorithmic competition requires considering the details of the algorithms and the market structure. When firms compete on algorithms, algorithms may generate price dispersion. Additionally, when facing a traditional competitor that uses a best-response function, algorithms tend to set supracompetitive prices, and both firms earn extra profits, but the traditional competitor benefits the most. Overall, the article contributes to understanding algorithmic competition, discusses implications for managers and policymakers, and identifies opportunities for future research.

INDEX TERMS Artificial intelligence, algorithmic competition, computational economics, market simulation, pricing.

I. INTRODUCTION

Artificial Intelligence (AI) is a powerful technology that is transforming today's business. However, in many cases, the business and economic impact of AI is not clear. A case in point is the use of AI-enabled algorithmic pricing.

Algorithmic pricing involves using algorithms that automatically set prices without human supervision. Companies use algorithmic pricing to compete more effectively and increase their profits. Algorithmic technology is not exclusive to big tech companies like Uber or Amazon. Software and services that allow small retailers to use such a technology are becoming more affordable and ubiquitous. For example, between 2014 and 2015, 500 out of 1,600 of the best-sellers products on Amazon were priced using algorithms, which allowed those sellers to win the Amazon "Buy Box" most of the time and outperform non-algorithmic competitors in Amazon's rankings [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Venkateshkumar M.

At first sight, it may seem to be a desirable outcome. Intelligent pricing algorithms may lead to more contestable markets as the frequency of price changes increases, may bring better service, better product availability, or an improved customer experience [2], [3]. On the other hand, they also pose a threat. Several authors and authorities have raised concerns about the capacity of those algorithms to collude autonomously and automatically (conscious parallelism). The Competition and Markets Authority (CMA) has highlighted that, although authorities have tools to deal with many forms of collusion, there is a chance that some collusive practices may fall out of the scrutiny of the institution [3].

Moreover, in the case of suspicious collusion, it is unclear where lies the burden of proof, whether the company should explain the algorithm, or whether the authorities should prove that the algorithm colluded [4]. Competition authorities have limited tools to deal with a scenario of "conscious parallelism" of algorithms [5]. At the same time, automated and autonomous collusion is a real threat, because a simple Q-learning algorithm may learn to collude [6]–[8].

However, some authors are more skeptical about real cases of algorithmic collusion. For instance, maybe the algorithm homogeneity assumption is too strong, and the collusive results are not robust ([3] and [5]). Other limitations for collusion may be the preference specification, the algorithm scalability, or the market structure [9].

This article aims to analyze the business and economic impact of pricing algorithms and algorithmic competition. We construct AI pricing agents and let them interact in computer-simulated markets. The advantage of this approach is that it allows for rigorous computational experiments without the need to recreate all those scenarios in the real world. We consider two sophisticated artificial intelligence pricing algorithms: Q-learning and Particle Swarm Optimization algorithms (PSO). Firms using algorithmic pricing compete in three different market structures: Logit, Hotelling, and Linear demand. The Logit model has become a standard in the algorithmic pricing literature [6], [10], [11]. The Hotelling and the linear demand model are well-known in operations research and economic modeling [12], [13].

In the main setup, two companies compete on prices using either Q-learning or PSO. We show that Q-learning leads to significant supracompetitive prices in some environments, but the outcome depends on the market structure. A priori, these results may raise concerns for both competition authorities and firms, which may devote significant resources to these cases.

We also examine how changes in algorithm design may affect the outcome of the algorithmic competition. We show that a slight change in the design of Q-learning and PSO may improve their performance. This finding raises a new concern because the algorithms are essentially the same, but they achieve more competitive results in some cases. This situation may suggest that competition authorities may require a case-by-case approach when dealing with price algorithms, which also requires access to the algorithm architecture and implies devoting significant resources. Moreover, firms should be aware that small mistakes in the code may lead to significantly different pricing behaviors.

We also consider a setting in which firms compete on algorithms: A firm uses PSO and the competitor Q-learning. We show a trade-off between exploration and exploitation. Our results suggest that the last algorithm exploiting its environment has an advantage similar to a Stackelberg leader. Intuitively, it sets prices knowing how its algorithmic competitor would react. Interestingly, in all cases, we find algorithms generate price dispersion. When an algorithm competes against a traditional best-response firm, we also find supracompetitive prices. The algorithmic firms earn extra profits but less than the best-response firm, and it is the best-response firm that benefits the most. Overall, the article makes the following contributions:

- 1) We explore the business and economic impact of AI in the form of algorithmic pricing. Firms use algorithms to compete more effectively and increase their profits, but

this raises the risk of intervention by public authorities concerned about anti-competitive behavior.

- 2) We develop a computational model of algorithmic competition that considers several algorithms (Q-learning and PSO) and market structures (Logit, Hotelling, linear demand).
- 3) We examine how algorithm design changes may affect market outcomes.
- 4) We examine an extension setup of firms competing on algorithms (instead of competing on pricing using algorithms).
- 5) We examine an extension setup of an algorithm competing against a traditional best-response function firm.

The following section provides a background on algorithmic pricing. Section III develops the computational model, while Section IV presents the computational experiments and results. Lastly, we summarize the results, discuss managerial and policy implications, and outline opportunities for future research.

II. ALGORITHMIC PRICING: Q-LEARNING & PARTICLE SWARM OPTIMIZATION

Algorithmic pricing is a method of automatically setting prices to maximize a firm's profit. Simple automated rules, such as discounts, may not satisfy this definition because they may not guarantee profit maximization. In 2011, two algorithms increased the Amazon prices of the book "The Making of a Fly" to \$24 million.¹ Those algorithms do not satisfy our definition because the objective was to generate marginally more revenue than the competitor. The Competition and Markets Authority (CMA) takes a much more inclusive definition that also considers monitoring, recommendation, and ranking algorithms as pricing algorithms [3]. We focus on price-setting algorithms.

A pricing algorithm should maximize expected profit, and it must be complex enough to guarantee that it cannot go out of control. Many algorithms can fulfill these two conditions, and the architecture of pricing algorithms is a well-kept secret of companies.² Furthermore, the analysis of algorithms based on complex artificial intelligence (AI) techniques is even more challenging due to the well-known black-box problem of AI.

We are interested in simple algorithms characterized by a few parameters to obtain a clear interpretation of the results, making it possible to keep arbitrarily modeling choices to a minimum. Two candidates that fulfill this requirement are Q-learning and Particle Swarm Optimization (PSO) algorithms. Both algorithms are used in experimental economic problems [6], [7], [14], [15], [16]. The Q-learning algorithm is part of the Reinforcement Learning (RL) literature [17], while the PSO is part of the Evolutionary Algorithms (EA) [18].

¹<https://www.wired.com/2011/04/amazon-flies-24-million/>

²<https://www.vice.com/en/article/v7gyp4/uber-drivers-sue-to-gain-access-to-its-secret-algorithms>

The Q-learning algorithm is the workhorse in current algorithmic pricing research. Q-learning is highly popular among computer scientists, is simple and can be fully characterized by just a few parameters, and shares the same architecture as the more sophisticated programs that have recently obtained spectacular successes in playing Go and chess [6]. Additionally, the core of the Q-learning algorithm is based on Bellman’s value function, which is widely used in economics and business research. However, Q-learning suffers a dimensionality problem [19]. To set prices, it must keep track of all the potential actions (prices) it can take; thus, the complexity grows exponentially with multi-product companies. A large price range may be infeasible computationally, and a short range may miss the optimal prices.

PSO does not suffer from the dimensionality problem [20]. It is suitable to address multidimensional optimization problems because it does not require keeping track of all the potential actions. Additionally, *the more ambitious the goals of some RL researcher, the more he/she will get drawn towards methods for [EA]*, [19]. However, PSO is not as intuitive or closely linked to economic modeling as Q-learning.

Compared with Q-learning, PSO algorithms do not “learn” strategies, but they “select” good outcomes. In other words, if we are interested in analyzing strategies, PSO is not a good option. However, if we are interested in analyzing just the output (optimal price), PSO can be a good option. More technically, another advantage of the PSO arises when the objective function is non-convex. In such a case, RL may get stuck at local optima while EA, just by virtue of sampling from a large population, would converge to the global solution easily. Additionally, PSO does not require a differentiable problem. In the following, we provide a short description of the algorithms and explain how they are implemented in our research.

A. 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 . This Q-function may be defined recursively as follows,

$$Q_i(s, a_i) = E(s, a_i) + \delta E[s^t, a_i^t] \tag{1}$$

In this framework, we assume that A and S are finite, and A is not state-dependent. Therefore, the Q-function becomes a $|A| \times |S|$ matrix. To estimate this matrix, Q-learning starts from an arbitrary initial matrix Q_i^0 , 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 the cell $Q_i(s, a_i)$ for $(s, a_i) = (s^t, a_i^t)$ following the learning equation:

$$Q_i^{t+1}(s, a_i) = (1 - \alpha) Q_i^t(s, a_i) + \alpha \left[\pi_i^t + \delta Q_i^t(s^{t+1}, a_i) \right] \tag{2}$$

For all $(s, a_i) \neq (s^t, a_i^t)$ the Q-values do not change, and the update of the cell $Q_i(s, a_i)$ is a convex combination of the previous value and the current reward plus the discounted value of the state that is reached next. The weight $\alpha \in [0, 1]$ is the learning rate, which we assume is constant.

Initially, to approximate the true Q-matrix, the algorithm must experiment by choosing actions that may be sub-optimal. As it is common in algorithmic pricing, at the beginning is desirable to explore the space and later to exploit the best results [6], [15]. We use a greedy model of exploration. 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 as time goes by, the algorithm must start exploiting the best outcomes it has found. To reproduce such a behavior, we posit a time-declining exploration rate:

$$\epsilon = e^{-\beta t} \tag{3}$$

where $\beta > 0$ is a parameter. The algorithm will start by randomly selecting actions. The larger the β , the faster the exploration vanishes.

B. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a stochastic optimization technique. It generates random points in a multidimensional space (particles) that move towards an optimal solution by sharing information about which points perform better. This concept is extended to price competition by assuming that each company may test a limited set of prices (particles) before going to the market. Then, companies choose those prices that perform better (higher profits) and remove those that perform worse. After repeating this operation multiple times, companies can set the best price given specific market or regulatory conditions.

Initially, 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. Thus, firms can evaluate the performance of each particle (price) in terms of profits. In the first iteration, the initial positions are random draws from a $U(0, 1)$ distribution. As time passes, the position of each particle will change as new information about the best positions is available.

In other words, the position of each particle depends on the locations of the best particles (those that provide the largest profits), and such an influence is called “evolutionary velocity ($v_{i,k}$)”, which determines the change of its position. Thus, a particle position is determined by the best position it has found before (p_i^t) and the best position any other particle in its swarm (or in the global swarm if there is only one swarm) has found before (p_i^g). Intuitively, a firm may test several prices, which is equivalent to controlling several particles. Therefore, p_i^g represents the best return found by any of the tested prices. In other words, the best position any controlled particle by that firm has found. Formally, the price $p_{i,t}$ at a time t is

updated as follows:

$$p_{i,t} = p_{i,t-1} + v_{i,t-1} \tag{4}$$

$$v_{i,t-1} = wv_{i,t-2} + l_1u_1(p_i^l - p_{i,t-1}) + l_2u_2(p_i^g - p_{i,t-1}) \tag{5}$$

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. In economic games, the payoff of a firm also depends on the prices of other companies. Thus, an optimal price in a previous iteration may not perform well in the current iteration and vice versa. Thus, p_i^l and p_i^g may change over time. In fact, at each iteration, we may have new different values for these parameters. In this sense, each firm will have a vector $M_s, s = l, g$ of size m that represents the memory of each firm. In this vector, firms will record the last m values of p_i^l and p_i^g , and among them, they will choose those with the best performance (largest profits).

Lastly, the inertia weight w is critical for the PSO's convergence behavior [21]. There is a trade-off between exploration and exploitation, like in Q-learning. Thus, we choose a similar model of exploration that vastly explores at the beginning and, as time goes by, it starts exploiting the best outcomes

$$w_t = (1 - w_o)^t \tag{6}$$

where w_o is a constant initial decrease parameter.

III. COMPUTATIONAL MODEL

We consider two firms that compete using pricing algorithms enabled by AI. First we define demand and market structures (Logit, Hotelling, and Linear demand). Then, we explain the model parameterization.

A. MARKET STRUCTURES

Following the recent literature on algorithmic pricing, we adopt a simple model of price competition with Logit demand and constant marginal costs [22]. Each company will face this demand function at each moment t . This price competition game assumes that there are n differentiated products and an outside good. Formally, the demand for product i is as follows:

$$q_i = \frac{e^{\frac{a_i - p_i}{\mu}}}{\sum_{j=1}^2 e^{\frac{a_j - p_j}{\mu}} + e^{\frac{a_0}{\mu}}} \tag{7}$$

Parameter a_i is the product quality of product i , an index that captures vertical differentiation. Product 0 is the outside good, so a_0 is an inverse index of aggregate demand. μ captures how different the products in the consumers' eyes are; thus, it is an index that represents the horizontal differentiation. The case of perfect substitutes is obtained in the limit as $\mu \rightarrow 0$. Each product is supplied by a different firm, so n is also the number of firms. Lastly, the profits of each company

are $\pi_i = (p_i - c) q_i$, where c is the marginal cost and p_i is the price.

We also adopt the Hotelling model, which is well known in economic and business research. We assume there are two firms situated at locations 0 and 1 on a line with unit length. The consumers are uniformly distributed along the line and demand one unit of product. A consumer's utility for each product is the value that the consumer derives from the consumption subtracting the price and the disutility from the mismatch between the firm's and the consumer's locations. The consumer compares two final products and chooses the one with higher utility. The utility for the consumer- j located at x from purchasing firms- i ' products are

$$u_j^i = v - p_i - \theta |x_j - k_i| \tag{8}$$

Essentially, θ captures the level of horizontal differentiation between the competing firms and measures the intensity of competition. A smaller value of θ implies a lower level of differentiation and higher competition intensity. The mismatch cost $\theta |x_j - k_i|$ captures such consumer heterogeneity. The parameter v captures the intrinsic value of the company that we assume high enough to guarantee that all consumers participate. Finally p_i represents the price paid by consumers. Thus, the demand for firm i of as follows:

$$q_i = \frac{1}{2} + \frac{1}{2} \frac{(p_j - p_i)}{\theta} \tag{9}$$

We also assume $\frac{3}{2}\theta < v$, which guarantees that all consumers buy at least from one firm.

Finally, the third model we consider is the classical linear demand model

$$q_i = a_i - p_i - dq_j \tag{10}$$

where a_i is the intrinsic value of the firm i , p_i is the price for the product of firm i , q_j is the quantity produced by a competitor j , and $d \in [0, 1]$ is a parameter that controls the level of differentiation, where 1 is the Bertrand model and 0 is the monopoly model.

Among all the variables of interest in these models, we focus on the comparison between prices (average simulated prices versus monopoly and Nash-equilibrium prices) and average profit gain (Δ), which is defined as:

$$\Delta \equiv \frac{\pi - \pi^N}{\pi^M - \pi^N} \tag{11}$$

where π is the average profit in the last 1.000 iterations, π^M is the profit under full collusion (monopoly), and π^N is the profit in the Nash equilibrium. Thus, $\Delta = 0$ corresponds to the competitive outcome and $\Delta = 1$ to the perfectly collusive outcome. The main reason for focusing on Δ is that it can be compared across different economic settings.

B. BASELINE PARAMETERIZATION

To facilitate comparisons with other works, we adopt the baseline parametrization of [6], [11]. The main reason to

avoid carrying out several experiments with different parameters is that those works did extensive sensitivity analysis of the parameter universe. In this way, we can focus our attention on the effect of the market environment and different algorithmic designs. Our baseline environment consists of a symmetric duopoly ($n = 2$) with $c_i = 1$, $a_i - c = 1$, $a_0 = 0$, and $\mu = 0.25$ in the Logit demand case, and $\theta = 1$, $c = 0$ and $v = 1.75$, in the Hotelling model, and $a_i = 1$, $d = 0.25$, $c = 0$, in the linear demand model.

In the case of Q-learning, it requires a finite action space. Following [6], we compute both the Nash-equilibrium of the one-shot games and the monopoly prices, P^N , and P^M respectively. Then, we consider that the set of feasible prices (\mathbf{A}) is given by 15 equally spaced points in the interval $[P^N - 0.15(P^M - P^N), P^M + 0.15(P^M - P^N)]$. Note that discretizing the action space implies that the exact Nash and monopoly prices may not be feasible, or even that new ones appear. For example, there is only one equilibrium in the continuous Bertrand game, but there are two in the discrete Bertrand one. Nonetheless, there may be mixed-strategy equilibria, and our algorithms that play pure strategies will oscillate around a target when it is not feasible. To ensure that the state space is finite, we posit a bounded memory. We assume that such memory lasts 1 period. Therefore, each firm has $|A| = 15$ and $|S| = 15^{2+1} = 3.375$. Regarding exploration, we assume the ϵ -greedy exploration model with a time-declining exploration rate, $\epsilon_t = e^{-\beta t}$, where we assume $\beta = 1.5 \times 10E - 4$. Finally, we assume that $\alpha = 0.15$ and $\delta = 0.95$ to let the algorithm discount future profits and let the initial Q_i^0 have all its elements equal to zero at $t = 0$. On the other hand, the baseline PSO algorithm consists of 5 particles ($k = 5$) with $l_1 = l_2 = 1.75$, $w_0 = 0.025$, and $m = 1$. We also limit the range of evolutionary velocity, $v_i \in [-0.3, 0.3]$ to avoid jumping between corner solutions. Similar parameterization appears in other algorithmic pricing research [16].

IV. COMPUTATIONAL EXPERIMENTS AND RESULTS

This section explores how each one of the algorithms performs when competing in prices in different market environments. We consider six different cases in which firms operate either with Q-learning or PSO algorithms. We assume firms update the prices at the same time, and no algorithm has any frequency advantage. In all cases (Logit, Hotelling, linear model), the set of feasible prices includes both the Nash-equilibrium and monopoly prices. The inclusion/exclusion of some prices has extreme relevance for this algorithm. It may be realistic to assume that some companies may let some prices out of this interval, for example, prices below marginal cost. However, if companies constraint too much the price interval, optimal prices may remain out-of-the-scope of algorithms.

In Table 1, we observe that Q-learning systematically sets supracompetitive prices. In all cases, profits are higher than competitive levels. Interestingly, the Q-learning obtains

TABLE 1. Average prices and Δ s. Two competing Q-learning algorithms.

Q-Learning	Firm 1		Firm 2		Extra-profits	
	Avg. Price	Theo. Price	Avg. Price	Theo. Price	Δ_1	Δ_2
LOGIT	1.70	1.47	1.69	1.47	0.66	0.61
Hotelling	1.24	1	1.25	1	0.15	0.15
Linear Demand	0.46	0.43	0.45	0.43	0.05	0.04

The small difference between cases that we observe in Table 1 and II is a consequence of the intrinsic randomness of both algorithms. All the results presented are the average of 100 rounds of experiments. Although both firms are symmetric, they may not experiment/exploit outcomes symmetrically, as such a decision is probabilistic.

TABLE 2. Average prices and Δ s. Two competing PSO algorithms.

PSO	Firm 1		Firm 2		Extra-profits	
	Avg. Prices	Theo. Prices	Avg. Prices	Theo. Prices	Δ_1	Δ_2
LOGIT	1.49	1.47	1.49	1.47	0.05	0.08
Hotelling	1.17	1	1.19	1	0.09	0.11
Linear Demand	0.43	0.43	0.43	0.43	0.05	0.04

approximately 66% and 15% extra profit in the Logit and Hotelling cases, respectively, compared to the static Nash equilibrium. In the linear demand case, the higher prices lead to earning relatively lower profits. Nonetheless, this case presents a large dispersion. At the end of the simulations, prices oscillate between the competitive and monopoly outcomes.

These results suggest that the Q-learning may not tend to set supracompetitive prices by itself, and it may be more a consequence of specific market features. Nonetheless, a priori, the existence of non-competitive prices does not tell us anything about its nature (i.e. whether they are collusive). In some cases, it may be genuine collusion, as in [6], but it may also be the case that Q-learning just learns to play a different kind of equilibrium, [23]. The key insight is that, in the three markets considered, Q-learning tends to set supracompetitive prices. This result is in line with [6], which shows the same result for other market parameters of the Logit model. However, the degree of supracompetitiveness is not constant, and it may depend on specific market characteristics. For example, in the Hotelling model with asymmetric firms ($v_2 = v_1 + 0.25$), we find that $\Delta_1 = 0.065$ and $\Delta_2 = 0.015$, and in the Linear model without differentiation (Bertrand’s model), we find that only one firm is active with oscillating but supracompetitive prices, $p_2 \in [0.06; 0.24]$ ($\Delta_2 \in [0; 0.738]$) while the competitor is oscillating between entering and exiting. These results emphasize that, although some algorithms may show a tendency to set supracompetitive prices, the market structure cannot be ignored.

Table 2 shows a contrasting result. The PSO sets prices close to the Nash equilibria, and the divergence that we observe is likely a consequence of its stochastic nature. Thus, in the three cases, the PSO leads to more competitive outcomes than Q-learning. We have also considered other market parameters for these three models, but in all cases, PSO tends to set lower prices than Q-learning.

Interestingly, in the linear demand model, PSO leads to a solution like Q-learning. This is a consequence of the model itself, which has the monopoly and duopoly outcomes close to each other. Thus a small variation in decimal precision translates into a large change in Δ .

TABLE 3. Average prices and Δs. Two competing Q-learning algorithms. 2nd Design.

Q-Learning (2nd)	Firm 1		Firm 2		Extra-profits	
	Avg. Prices	Theo. Prices	Avg. Prices	Theo. Prices	Δ ₁	Δ ₂
LOGIT	1.68	1.47	1.60	1.47	0.60	0.46
Hotelling	1.22	1	1.17	1	0.13	0.09
Linear Demand	0.45	0.43	0.45	0.43	0.05	0.04

Independently of the nature of those supracompetitive prices, what can we do to avoid such solutions? We propose to pay attention to algorithm designs that are more “competitive” or better resemble economic intuition. The idea is to focus on how the different algorithms simulate decision-making and modifying those features that represent the decision-making process of a rational agent. However, this usually is easier to say than done because the intuition of how each module behaves and interacts with the rest becomes blurry as algorithms become complex. Nonetheless, many algorithms share a similar architecture, which may help in focusing on specific modules. In the following subsection, we address two modifications of the Q-learning and PSO algorithms that lead to more competitive outcomes.

A. ALTERNATIVE ALGORITHMIC DESIGNS

Technically, Q-learning firms can choose any price at each state given that there is no constraint regarding their own actions; additionally, any combination of two prices (own price and competitor’s) is possible given that there is no way of knowing which price the competitor will choose. However, it is possible to consider an alternative scenario where the Q-learning firm can choose any price at each state but assumes that the competitor will keep its current prices in the following state. For example, in the previous section, if both Q-learning algorithms set a price level of 1.70, they could transition to any of the 225 pairs of potential prices in the next stage. However, in this design, if both Q-learning algorithms set 1.70, they can only transition to one of the 15 pairs of potential prices because the competitor’s price is taken as given at a level of 1.70. This way of behaving resembles a best-response function. Intuitively, the firm takes as given the action of the competitors and chooses the best feasible price. In terms of Q-learning, this modification only requires a constraint in the transition states. In this regard, the different specifications of the state space may change the way algorithms learn, and this change will lead to more competitive results, given that it resembles a best-response framework.

In Table 3, we observe that such modification drives down profits, and in some cases significantly. The Δ₂ column depicts the results with this new design, while the Δ₁ column depicts the results with the same design as the previous section. Interestingly, we observe that Δ₂ is 30% times smaller than Δ₁ in the Logit and Hotelling cases, which shows that non-competitive prices may be less likely with this modification. In contrast, in the linear demand model, there is almost no change. This result is likely a consequence of the model, which has the monopoly and duopoly outcomes close to each other.

TABLE 4. Average prices and Δs. Two competing PSO algorithms. 2nd Design.

PSO (2nd)	Firm 1		Firm 2		Extra-profits	
	Avg. Prices	Theo. Prices	Avg. Prices	Theo. Prices	Δ ₁	Δ ₂
LOGIT	1.5	1.47	1.5	1.47	0.07	0.09
Hotelling	1.11	1	1.10	1	0.06	0.06
Linear Demand	0.43	0.43	0.43	0.43	0.00	0.00

Therefore, even changes in the design may not affect all markets equally, and some may be more influenced than others depending on which algorithmic design we choose. Another possibility would be to assume that each firm considers a different set of prices or may have different memory lengths. These changes in the state space are also likely to influence prices, since they modify the state space more than the one considered in this section. However, analyzing the sensitivity of Q-learning firms to changes in the state space goes beyond the scope of this paper, but it is an interesting research venue with many unknowns so far. However, the limited evidence so far shows that state space plays a key role in understanding price patterns.

In the PSO case, we can introduce a new learning parameter (l_3) that is influenced by an exogenous price level, p^G and we assume $l_1 = l_2 = l_3 = 1$ to avoid volatile behavior in the PSO. This modification implies that this new price would be an “attractor”. Thus, it could be used either to attract a price toward its Nash equilibrium or to repel it. Formally,

$$v_{i,t-1} = wv_{i,t-2} + l_1u_1(p_i^l - p_{i,t-1}) + l_2u_2(p^g - p_{i,t-1}) + l_3u_3(p^G - p_{i,t-1}) \quad (12)$$

where $u_3 \sim U[0, 1]$. In this case, we assume that the price p^G is the most profitable price that any company has found. This case may represent those situations in which a company sets an offer that turns out to be quite profitable, and the rest of the companies imitates that offer. This new parameter would let the algorithm look for better prices but with a tendency to look in the neighborhood of that price. Other possibilities include p^G being imposed by authorities at marginal cost or the Nash equilibrium. This assumption is fundamental, and assuming other prices may completely change the results. For example, if instead, we assume that p^G is equal to marginal costs, PSO would drive all prices toward that point, which could be suboptimal in some frameworks.

In contrast with Q-learning, the introduction of this modification does not lead to significantly different results. Their results were already competitive, so this modification should not modify the previous result.

B. EXTENSION: COMPETING ON ALGORITHMS

Up to now, we analyzed markets where both firms adopted the same algorithm and they competed on pricing. But an interesting extension is a setting in which different firms adopt different pricing algorithms. In that setting firms compete on algorithms. Following the previous sections, we compare the case when a company uses Q-learning, and the competitor uses PSO. This setup may represent reality better. Moreover,

TABLE 5. Average prices and Δ s. PSO vs Q-learning algorithms.

PSO vs Q-learning	PSO Firm		Q-learning Firm	
	Avg. Prices Firm 1	Δ_1	Avg. Prices Firm 2	Δ_2
LOGIT	1.61	0.41	1.51	0.11
Hotelling	1.08	0.11	1.19	0.05
Linear Demand	0.48	-0.15	0.45	0.07

“if players use different pricing algorithms, each of which could be learning over time, it will increase the complexity and difficulty of establishing coordination”, [3]. We adopt the stateless Q-learning algorithm. This version of the Q-learning is memoryless, and its state space is just the set of prices that it can choose. This assumption is necessary because other algorithms may choose prices that are not in the Q-learning price interval. In this case, they may not be part of the state-space that Q-learning takes into account. On the contrary, taking into account all the values that the Best Response function or the PSO can take implies transforming the Q-learning into its continuous version, which complicates the analysis and implies further assumptions. Thus, to keep it simple, we adopt the stateless version of the Q-learning algorithm, whose inner workings is a simpler version of the one depicted in the previous sections.

In Table 5, we observe that, on average, PSO leads to higher prices in the LOGIT model this time, but the opposite is true with the Q-learning algorithm. However, in the Hotelling model, both algorithms perform better this time, but Q-learning keeps leading to higher prices than PSO, and, in the Linear demand model, both perform worse, and this time, PSO leads to higher prices. Interestingly, in Hotelling’s case, the Q-learning’s Δ s are lower than in previous sections despite higher prices. This is because Q-learning faces a smaller demand, which leads us to this counterintuitive result. Nonetheless, in all cases, we observe some degree of price dispersion (see Table 8) and prices higher than the competitive outcome.

We find that both algorithms set supracompetitive prices, but it is Q-learning the one that sets the lower prices in two out of three. This is a consequence of how these two algorithms interact. Both of them explore and exploit results, but they do it following different rules and timings. Therefore, one algorithm may start exploiting results sooner while the other explores the environment more, in this case, the one that explores more may end exploiting an optimal position as a response to a suboptimal position of the first algorithm. This is likely what we observe here because PSO converged in all simulations faster than Q-learning, which explored the environment more. [13] also finds a similar effect when algorithms can be outsourced and demand shocks are absent. In such a case, *commitment to a pricing algorithm is equivalent to commitment to a particular price. Thus, adoption of the third party’s pricing algorithm creates a sequential-move price game.*

A key observation is that when different algorithms are used against each other, the one with the largest exploration phase may have an advantage. Intuitively, our main setup could be compared with a Cournot model where all

TABLE 6. Average prices and Δ s. Q-learning vs. Best Response.

Q-Learning vs Best response	Q-learning Firm		Best-Response Firm	
	Avg. Prices Firm 1	Δ_1	Avg. Prices Firm 2	Δ_2
LOGIT	1.64	0.14	1.55	0.63
Hotelling	1.16	0.05	1.09	0.01
Linear Demand	0.45	0.02	0.43	0.02

algorithms make decisions at the same time, and they take as given the competitor’s, but this extension is more similar to a Stackelberg game where the first mover would be the one with the largest exploration phase. [24] also find this price pattern in their theoretical model and argue that this asymmetry between technologies may be a source of price dispersion.

C. EXTENSION: ALGORITHMS COMPETING WITH BEST-RESPONSE FIRMS

We compare the performance of Q-learning and PSO when they face a traditional competitor, in other words, a firm that sets prices using a best-response function. This set of scenarios represents a likely situation where some companies use pricing algorithms while some competitors rely on more traditional price-setting mechanisms [1]. We assume that both the algorithm and the traditional firm set prices with the same frequency. Note that the best-response function will always dominate a metaheuristic by definition. However, the interesting point here is understanding how an algorithmic agent may modify the prices set by a more traditional agent. Tables 6 and 7 compare the cases when Q-learning and PSO face a best-response function, respectively. Similar to our main setup, we observe that the case with Q-learning leads to larger prices than PSO. It is interesting that Q-learning does not decrease their prices too much compared to those of the main setup, but their Δ s are significantly smaller. Interestingly, in this case, we observe that the degree of price dispersion and magnitude of supracompetitive prices depend on the algorithm. Although prices are lower with PSO, they are more disperse, Table 8.

The first interpretation of this result is that algorithms do not pose a significant anti-competitive threat when facing competition because, even if they set larger prices, a best-response firm may capture part of the extra-profits, thus penalizing the algorithmic pricing. This case may resemble what has happened on Amazon during the COVID-19 pandemic. Six months after the price surge of sanitary products, many products remain overpriced [25]. However, as the comparison among cases shows, this effect would depend on the market structure. If we pay attention to the PSO, we obtain a similar insight, although in general, the markets are more competitive. Nonetheless, these results may not hold if pricing algorithms can set prices at a higher frequency and the algorithm is outsourced. In such a case, prices may be more volatile and even show the same sensitivity as monopoly prices without being as high. Additionally, if demand variation is high, it may be optimal to adopt algorithms by all players, thus making this scenario suboptimal, [13].

TABLE 7. Average prices and Δ s. PSO vs. Best Response.

PSO vs Best Response	PSO Firm		Best-Response Firm	
	Avg. Prices Firm 1	Δ_1	Avg. Prices Firm 2	Δ_2
LOGIT	1.51	0.03	1.49	0.15
Hotelling	1.03	0.00	1.00	-0.01
Linear Demand	0.43	0.70	0.43	0.92

TABLE 8. Price Dispersion Measures.

Cases		Standard Deviation	Quartiles		Range	
			25	75	Min	Max
Q-Learning vs PSO	Logit	0.11	1.42	1.57	1.33	1.76
	Hotelling	0.15	0.98	1.18	0.79	1.36
	Linear Demand	0.1	0.44	0.54	0.27	0.67
Q-Learning vs Best Response	Logit	0.02	1.64	1.64	1.594	1.686
	Hotelling	0.03	1.09	1.16	1.09	1.16
	Linear Demand	0.01	0.44	0.46	0.42	0.46
PSO vs Best Response	Logit	0.07	1.46	1.54	1.37	1.75
	Hotelling	0.18	0.96	1.14	0.43	1.26
	Linear Demand	0.09	0.42	0.43	0.36	0.5

V. DISCUSSION

Companies use AI in the form of algorithmic pricing to increase their profits and compete more effectively. However, this may attract the attention of public authorities who are concerned about threats to competition. In order to provide insight into these issues, this research analyzes how two well-known AI algorithms, Q-learning and Particle Swarm Optimization, set prices in three different market frameworks. We find that Q-learning leads to supracompetitive prices, while PSO tends to set competitive prices in several market structures.

To address whether those results are robust to changes in the algorithmic design, we consider two alternative designs of the Q-learning and PSO algorithms. In the Q-learning case, we observe that such a modification leads to more competitive outcomes. Thus, the algorithmic design seems to be essential when addressing algorithmic competition, and in some instances, supracompetitive prices may be a consequence of an ill-designed algorithm. Therefore, this situation implies that competition authorities should devote more resources to address algorithm behavior, and it would probably lead to a case-by-case approach. From the firms' perspective, small mistakes in the code may lead to significantly different pricing behaviors. Firms may face anti-competitive charges or public uproar when some of their prices are tagged as non-competitive despite being a simple "bug" in the code.

Moreover, we address the case when firms compete using different algorithms, and we find that, in such cases, there is a novel issue regarding the timings of exploration versus exploitation. Our results show that if we let the algorithms interact indefinitely, the one with the most extensive exploration phase will have an advantage over the competitor intuitively similar to a Stackelberg leader. Additionally, we observe price dispersion, which is another dimension that may raise concerns independently of its nature.

Lastly, when instead of an algorithmic competitor, firms face a best-response firm, prices are reduced compared to other frameworks but remain higher than those of the Nash equilibrium. Interestingly, in this case, the best-response firm benefits the most because it can capture part of the extra profits generated by the algorithmic distortion.

This result may suggest that algorithmic pricing may artificially raise prices and keep them above competitive levels, which would be consistent with the anecdotal evidence of price increases of sanitary products [25], [26], or the retail gasoline market [27].

A. IMPLICATIONS FOR POLICYMAKERS AND MANAGERS

Our results suggest that algorithmic pricing should concern policymakers, but the outcome depends crucially on the adopted algorithms, the algorithmic design, and the market structure. We should also note that our results emerge in duopolies with market power and when the number of competitors increases, algorithms tend to set more competitive prices [6], [23]. Thus, in a market where companies have limited influence on the global market outcome, it is likely that algorithmic pricing would set competitive prices. For example, if companies are small enough, and they cannot influence market prices, or the market is big enough, and one company's decision is marginal, algorithms face a quasi-stationary environment. Thus, it is more likely that they fulfill the conditions to converge to the competitive equilibrium. In these cases, we would retain all the benefits of algorithmic pricing (more contestable markets, increased competition, or better product availability) without the price distortions that we have observed in previous sections. In addition, better demand forecasting through algorithms can lead to lower prices in some cases [28]. Similarly, algorithmic consumers, who compare multiple firms using AI algorithms, may reduce the firms' ability to set supracompetitive prices.

Companies that consider adopting algorithmic pricing solutions face two problems a priori: their algorithms must identify the market structure and learn how to behave in it. There are multiple frameworks, software, and algorithms that can be used to set prices. The most interesting ones are those based on Artificial Intelligence (AI) that can autonomously and automatically identify the market structure and set prices. However, although these AI algorithms may solve the a priori problems, they may generate new ones like supracompetitive prices or price gouging. In this confusing environment, managers must be aware that there is no one-size-fits-all solution. An algorithm that performs well in a specific market may not work well in a different one. For example, [6] and [29] use Q-learning to set prices in different market environments with opposite results. Likewise, an algorithm that performs well in training environments may perform poorly once it has contact with reality and other firms. Moreover, it matters how competitors set prices. In this sense, a clear insight for managers is that although algorithmic pricing may be profitable, it is not robust. It would require human supervision and human intervention in some cases. Uber is a clear example of this behavior. They make extensive use of algorithms to set prices but, in emergencies, they usually suspend algorithmic pricing.³

³<https://abcnews.go.com/Business/uber-chooses-surge-price-cap-emergencies/story?id=28494303>

Another critical challenge for both practitioners and policymakers is the significant role that different designs and features may have in the market outcomes. This is especially relevant in markets where algorithmic pricing research advances the traditional dynamic pricing research, such as airlines or electricity markets, where real-time pricing has been considered the most efficient pricing policy [30]. Small changes in the code may lead to entirely different results with dramatic consequences like unintended supracompetitive prices. This circumstance also fuels the necessity of a different kind of market oversight from public authorities. Public authorities should not just aim to enforce competition but also detect unusual price patterns that may be a consequence of “bugs”. However, the use of anti-competitive rules in the algorithm might not be straightforward, which may increase the costs and efforts of designing and monitoring the remedy [31].

Collusion is not the only threat that algorithms pose. The role of algorithms in price gouging is another concern. Uber faced a great deal of uproar due to charging up to eight times the usual fares after the 2013 heavy storm in New York and the 2017 terror attacks in London. More recently, competition authorities in Spain, Romania, Italy, and Greece announced investigations into price hikes regarding sanitary products during the COVID-19 pandemic [26]. In the US, some Amazon sellers were fined for price gouging.⁴ Algorithms may get trapped in price gouging solutions, which may even drive up the prices of non-algorithmic firms [11]. Another challenge that firms may face is whether they should develop their own algorithms or outsource them. Recent evidence shows that outsourcing may not reduce competition, but it may reduce welfare [13].

Overall, our results suggest that algorithmic pricing creates new benefits and challenges for firms and competition authorities alike. Both firms and competition authorities should refrain from simplistic and one-sided approaches that label algorithmic pricing as good or bad. Instead, nuance and careful analysis is needed.

B. OPPORTUNITIES FOR FUTURE RESEARCH

Algorithmic pricing and competition is an underexplored area, and there are many opportunities for future research. For instance, all our models assume complete information, but it is not clear what happens in cases of imperfect information [32]. Another new topic is how algorithms perform in markets with multiple equilibria. It is also unclear what happens when algorithmic pricing interacts with algorithmic consumers. Moreover, there is very little empirical research on the presence and effects of algorithmic pricing [27]. Algorithmic pricing raises ethical issues that could be studied more in future work [33]. In the context of algorithmic management, people are concerned about surveillance, little transparency, and lack of human interaction [34]. As algorithms become

more complex, influential, and ubiquitous over time, this will create even more research questions.

ACKNOWLEDGMENT

The authors declare that they have no conflict of interest.

REFERENCES

- [1] L. Chen, A. Mislove, and C. Wilson, “An empirical analysis of algorithmic pricing on Amazon marketplace,” in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 1339–1349.
- [2] A. Shartsis. (2019). *In the Future, Everything Will be Dynamically Priced*. Accessed: Feb. 6, 2021. [Online]. Available: <https://www.forbes.com/sites/forbestechcouncil/2019/09/10/inthe-future-everything-will-be-dynamically-priced/>
- [3] “Pricing algorithms. Economic working paper on the use of algorithms to facilitate collusion and personalised pricing,” Competition and Markets Authority, London, U.K., Tech. Rep. CMA94, 2018.
- [4] U. Schwalbe, “Algorithms, machine learning, and collusion,” *J. Competition Law Econ.*, vol. 14, no. 4, pp. 568–607, Dec. 2018.
- [5] S. Van Uytsel, “Artificial intelligence and collusion: A literature overview,” in *Robotics, AI and the Future of Law*, Singapore, 2018, pp. 155–182.
- [6] E. Calvano, G. Calzolari, V. Denicolo, and S. Pastorello, “Artificial intelligence, algorithmic pricing, and collusion,” *Amer. Econ. Rev.*, vol. 110, no. 10, pp. 3267–3297, 2020.
- [7] T. Klein, “Autonomous algorithmic collusion: Q-learning under sequential pricing,” *RAND J. Econ.*, vol. 52, no. 3, pp. 538–558, Sep. 2021.
- [8] E. Calvano, G. Calzolari, V. Denicoló, and S. Pastorello, “Algorithmic collusion with imperfect monitoring,” *Int. J. Ind. Org.*, vol. 79, Dec. 2021, Art. no. 102712.
- [9] A. Ittoo and N. Petit, “Algorithmic pricing agents and tacit collusion: A technological perspective,” in *Chapter L’Intelligence Artificielle et le Droit*, H. Jacquemin and A. De Stree, Eds. Bruxelles, Belgium: Larcier, 2017, pp. 241–256.
- [10] J. Johnson, A. Rhodes, and M. R. Wildenbeest, “Platform design when sellers use pricing algorithms,” Working Paper SSRN: 3753903, 2020.
- [11] J. M. Sánchez-Cartas, A. Tejero, and G. León, “Algorithmic pricing and price Gouging. Consequences of high-impact, low probability events,” *Sustainability*, vol. 13, no. 5, p. 2542, Feb. 2021.
- [12] P. Belleflamme and M. Peitz, *Industrial Organization: Markets and Strategies*. Cambridge, U.K.: Cambridge Univ. Press, 2015.
- [13] J. Harrington, “The effect of outsourcing pricing algorithms on market competition,” Dept. Bus. Econ. Public Policy, Wharton School Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep. SSRN 3798847, 2021.
- [14] L. Waltman and U. Kaymak, “Q-learning agents in a Cournot oligopoly model,” *J. Econ. Dyn. Control*, vol. 32, no. 10, pp. 3275–3293, Oct. 2008.
- [15] T. Zhang and B. W. Brorsen, “Particle swarm optimization algorithm for agent-based artificial markets,” *Comput. Econ.*, vol. 34, no. 4, p. 399, 2009.
- [16] M. K. Maschek, “Particle swarm optimization in agent-based economic simulations of the Cournot market model,” *Intell. Syst. Accounting, Finance Manage.*, vol. 22, no. 2, pp. 133–152, Apr. 2015.
- [17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
- [18] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford, U.K.: Oxford Univ. Press, 1996.
- [19] J. Schmidhuber, “Evolutionary computation versus reinforcement learning,” in *Proc. 26th Annu. Conf. IEEE Ind. Electron. Soc., (IECON) IEEE Int. Conf. Ind. Electron., Control Instrum., 21st Century Technol.*, vol. 4, Oct. 2000, pp. 2992–2997.
- [20] A. P. Engelbrecht, *Computational Intelligence: An Introduction*. Hoboken, NJ, USA: Wiley, 2007.
- [21] A. Chatterjee and P. Siarry, “Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization,” *Comput. Oper. Res.*, vol. 33, no. 3, pp. 859–871, Mar. 2006.
- [22] K. E. Train, *Discrete Choice Methods With Simulation*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

⁴<https://www.cnbc.com/2020/11/17/amazon-sellers-fined-for-coronavirus-price-gouging-hand-sanitizer.html>

- [23] J. Asker, C. Fershtman, and A. Pakes, "Artificial intelligence and pricing: The impact of algorithm design," Nat. Bur. Econ. Res., Cambridge, MA, USA, Working Paper 28535, 2021.
- [24] Z. Y. Brown and A. MacKay, "Competition in pricing algorithms," Nat. Bur. Econ. Res., Cambridge, MA, USA, Working Paper 28860, 2021.
- [25] G. Brombach and T. Murray, "High COVID-19 prices persist on Amazon," U.S. Public Interest Res. Groups Educ. Fund, Washington, DC, USA, Tech. Rep. No, Sep. 2020.
- [26] OECD, "Exploitative pricing in the time of COVID-19," Organisation Econ. Co-Operation Develop. (Tackling Coronavirus (COVID-19)), Paris, France, Tech. Rep. No, May 2020.
- [27] S. Assad, R. Clark, D. Ershov, and L. Xu, "Algorithmic pricing and competition: Empirical evidence from the German retail gasoline market," CESifo Working Papers, Munich, Germany, Tech. Rep. 8521, 2020.
- [28] J. Miklós-Thal and C. Tucker, "Collusion by algorithm: Does better demand prediction facilitate coordination between sellers?" *Manage. Sci.*, vol. 65, no. 4, pp. 1552–1561, Apr. 2019.
- [29] A. Collins and L. Thomas, "Comparing reinforcement learning approaches for solving game theoretic models: A dynamic airline pricing game example," *J. Oper. Res. Soc.*, vol. 63, no. 8, pp. 1165–1173, Aug. 2012.
- [30] G. Dutta and K. Mitra, "A literature review on dynamic pricing of electricity," *J. Oper. Res. Soc.*, vol. 68, no. 10, pp. 1131–1145, Oct. 2017.
- [31] F. Beneke and M.-O. Mackenrodt, "Remedies for algorithmic tacit collusion," *J. Antitrust Enforcement*, vol. 9, no. 1, pp. 152–176, Apr. 2021.
- [32] H. Varian, "Artificial intelligence, economics, and industrial organization," Nat. Bur. Econ. Res., Cambridge, MA, USA, Working Paper No. 24839, 2018.
- [33] P. Seele, C. Dierksmeier, R. Hofstetter, and M. D. Schultz, "Mapping the ethicality of algorithmic pricing: A review of dynamic and personalized pricing," *J. Bus. Ethics*, vol. 170, no. 4, pp. 697–719, May 2021.
- [34] M. Möhlmann and O. Henfridsson, "What people hate about being managed by algorithms, according to a study of Uber drivers," *Harvard Bus. Rev.*, vol. 30, pp. 1–7, Aug. 2019.

J. MANUEL SANCHEZ-CARTAS received the B.S. degree in economics from the University of Extremadura, Spain, in 2013, the M.S. degree in applied economics from Complutense University & Alcala University, Madrid, Spain, in 2015, and the Ph.D. degree in economics and telecommunication engineering from the Universidad Politecnica de Madrid & the European Institute of Technology (EIT), in 2018. Since 2019, he has been a Senior Research Economist at Zio Analytics. His research interests include industrial organization, digital markets, computational economics, and platform competition. His research has been published in the *Journal of Economic Surveys*, *Journal of Economics*, and *International Journal of Economic Theory* among others.

EVANGELOS KATSAMAKAS received the B.Eng. degree in computer science and computer engineering from the University of Patras, Greece, in 1998, the M.Sc. degree from the London School of Economics, and the Ph.D. degree in information systems from the Stern School of Business, New York University. He is currently a Professor in information, technology & operations with the Gabelli School of Business, Fordham University, where he worked as the Department Chair, from July 2012 to June 2018. His research analyzes the strategic and economic impact of digital technologies focusing on digital transformation, platforms, network effects, and fintech. His research interests include analytical modeling, machine learning, and computational modeling of complex business systems. His research has appeared in *Management Science*, *PLOS One*, *Journal of Management Information Systems*, *International Journal of Medical Informatics*, *Electronic Commerce Research & Applications*, *System Dynamics Review*, and in multiple other scholarly journals and conference proceedings (HICSS, IEEE). His research on digital innovation received the 2016 SIM Best Paper Award.

• • •