

Article

Algorithmic Pricing and Price Gouging. Consequences of High-Impact, Low Probability Events

Juan Manuel Sánchez-Cartas * , Alberto Tejero  and Gonzalo León

Campus de Montegancedo, Universidad Politécnica de Madrid, 28223 Madrid, Spain;
alberto.tejero@upm.es (A.T.); gonzalo.leon@upm.es (G.L.)

* Correspondence: juanmanuel.sanchez@upm.es

Abstract: Algorithmic pricing may lead to more efficient and contestable markets, but high-impact, low-probability events such as terror attacks or heavy storms may lead to price gouging, which may trigger injunctions or get sellers banned from platforms such as Amazon or eBay. This work addresses how such events may impact prices when set by an algorithm and how different markets may be affected. We analyze how to mitigate these high-impact events by paying attention to external (market conditions) and internal (algorithm design) features surrounding the algorithms. We find that both forces may help in partially mitigating price gouging, but it remains unknown which forces or features may lead to complete mitigation.

Keywords: pricing algorithms; artificial intelligence; sustainable business models; digital products; PSO; competition; price gouging

JEL Classification: L13; L86; D43; D83



Citation: Sánchez-Cartas, J.M.; Tejero, A.; León, G. Algorithmic Pricing and Price Gouging. Consequences of High-Impact, Low-Probability Events. *Sustainability* **2021**, *13*, 2542. <https://doi.org/10.3390/su13052542>

Academic Editor: Evangelos Katsamakos

Received: 22 January 2021
Accepted: 21 February 2021
Published: 26 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In 2017, London suffered a terror attack. Given the memories of the metro attacks of 2005, many Londoners tried to reach home with ride-hailing services, such as Uber. However, some users were charged up to 2.5 times the normal fare to get home [1]. This rise in fares was a consequence of a pricing algorithm that was not capable of handling this emergency. This was not the first time that Uber faced such an event. In 2013, when New York was in the midst of a storm, Uber increased its fares up to eight times the usual fares [2]. Both events caused a great deal of uproar and led to price gouging accusations. More recently, in March 2020, during the first weeks of confinement because of the COVID-19 pandemic in Europe, sellers on Amazon and eBay were warned that price gouging would not be tolerated and could lead to permanent suspensions of accounts [3,4]. Such policies especially affect sellers that use pricing algorithms given that they are more sensitive to changing market conditions and change prices tens or even hundreds of times per day [5]. Only in Amazon's case, this policy has led to the suspension of almost 4.000 selling accounts just in the US in the first weeks of the pandemic [3]. Thus, independent of whether price gouging is necessary to bring balance to the market, it is certain that an accusation of price gouging may drive some firms out of business. Given that companies that set prices with autonomous algorithms are more exposed because of their high-frequency pricing, it becomes essential to understand how those high-impact, low-probability (HILP) events influence algorithms that set prices [6].

Algorithmic pricing is a method of automatically setting prices using preprogrammed instructions to maximize sellers' profits, and as traditional businesses start their digital transformation, interest in those algorithms rises because of their potential to lead to higher profits, cost reductions, efficiency gains, and more competitive markets [7], which may result in more sustainable business models [8]. However, in the light of the previous examples, some questions arise. To what extent can HILP events influence algorithmic

pricing? Are there market conditions that make some markets more sensitive to HILP events? The previous examples show that such events must be taken seriously. Amazon and eBay have a “zero tolerance” policy regarding price gouging. Thus, avoiding price gouging is essential for third-party sellers to keep doing business on those platforms. Other related questions are how to mitigate such price gouging events and what can be done to prevent such raises.

To continue the progress made in analyzing the societal and economic impact of pricing algorithms, we take a simulation approach and analyze how a particle swarm optimization (PSO) algorithm implemented to set prices in a simple canonical market environment may be affected by HILP events. To do so, we construct artificial intelligence (AI) pricing agents and let them interact in computer-simulated markets. In this framework, we address how HILP events influence the price set by algorithms in differentiated and undifferentiated markets and whether market forces or specific algorithmic designs can mitigate their impact.

We find that the PSO works quite well in setting prices near the optimal one and that such a behavior is robust to different parameters, as other works have shown [9,10]. Then, we analyze how a HILP event distorts the optimal price predicted by the PSO and how competition with a seller that does not use pricing algorithms and a change in the design of the PSO may mitigate its impact. To the best of our knowledge, this is the first time that the effect of HILP events in pricing algorithms is addressed. Additionally, this work also contributes to the recent literature on algorithmic pricing policies by addressing how different factors (market forces and design) influence the algorithm performance. In this regard, we are not aware of similar works that address these factors.

We find that, after a HILP event, only in differentiated markets do prices return to normal but, in undifferentiated markets, prices remain higher than optimal, which may raise price gouging concerns. Additionally, neither factor is capable of completely mitigating the HILP event, although they partially mitigate it. In both cases, the final price remains higher than optimal, which would explain why the only solution found to deal with such events in the last decade was either to suspend the algorithmic pricing or to set price caps [11,12]. In this regard, a key warning for sellers that use algorithmic pricing is to test HILP events in their algorithms. The sustainability of many digital businesses also relies on avoiding unnecessary price gouging accusations, which can be generated by pricing algorithms unintentionally.

2. E-Commerce, Digital Transformation, and Algorithmic Pricing

The development of e-commerce has followed a growing trend in recent decades, where many companies have changed the way they connect not only with their customers and suppliers but also with other stakeholders: administration, financial entities, etc. [13]. Something that has contributed significantly to the growth of electronic commerce in recent years has been the commitment of many organizations to including a digital transformation within their strategies [14].

Digital transformation is based on the application of technology to increase productivity, value, and social well-being through the creation of new models, processes, software, and systems for business. Digital transformation is revolutionizing organizations in all sectors, breaking down the barriers between people, companies, and “things” [15]. From an economic perspective, the objectives of digital transformation are mainly found in the implementation of new and innovative business models as well as in the increase in income generation, the productivity of organizations, and the addition of value in the economy [16].

Within this new, highly competitive digital scenario, carrying out a good pricing strategy plays a fundamental role, both to maximize customers’ satisfaction and to create competitive advantages over competitors [17]. However, pricing digitally comes with some complexity. Price is a crucial factor for companies to be competitive in markets as transparent as today’s online. In these markets, e-commerce providers often have to adjust

their prices in short time intervals, for example, in order to take into account the prices that are frequently changed by their competitors [18], but they also have to be careful when defining the factors that affect prices [19].

Although there is indeed a burgeoning and growing literature on electronic commerce and digital transformation, there is still little research on something as important as algorithmic price competition and strategic behaviors [20]. This is precisely one of the objectives of this work. Under the label of algorithmic pricing, many optimization tools and techniques that help managers set prices could be included. The use of algorithms to set prices has its origin in the study of the convergence towards equilibrium in game theory. Originally, the interest was in iterative methods to solve games [21], the stability of a game (see [22] for an introduction), or how players may learn to play a game [23]. Recently, the attention has moved towards optimization techniques that could be used in combination with large datasets that are updated continuously [24]. Current evidence shows that algorithmic pricing may increase competition [25], which in turn may lead to more innovations [26]. We are interested in algorithms powered by AI that may be employed in practice. These AI solutions should share the same feature. They must try to find a set of optimal prices that maximizes some objective function, normally, a profit function.

However, pricing algorithms are a well-stored secret of companies, and although there are pending causes under investigation, little is known about the architecture of those algorithms [27]. Furthermore, when algorithms are based on artificial intelligence (AI) techniques, the well-known black-box syndrome introduces another complexity in the analysis process. Therefore, among all the potential frameworks, we are interested in those that are simple and can be characterized by just a few parameters to obtain a clear interpretation of the results. This requirement also makes it possible to keep arbitrarily modeling choices to a minimum and to conduct a comprehensive comparative statics analysis with respect to the parameters. Thus, under this requirement and to the best of our knowledge, two potential candidates are the Q-learning and the Particle Swarm Optimization (PSO) algorithms. Both have been successfully applied to experimental economic problems [9,10,20,28,29]. Additionally, we are also interested in AI solutions that could be applied by companies. Although little is known on the specific software that firms use, those algorithms should not be computationally expensive, and they should be able to deal with multi-dimensional optimization problems since many digital companies are multi-product companies. In this regard, the Q-learning algorithm seems not to be a good choice in its basic setting. Its downside is that the Q-learning algorithm suffers a dimensionality problem [30]. 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, and dealing with multidimensionality requires complex algorithms that may not be suitable for all companies. Additionally, the standard Q-learning algorithm applies only to discrete actions and states. Although some works have proposed ways to solve this limitation, such as [31], we are not aware of similar works in economics. A related problem is that we need to specify the space-state environment; in other words, we need to define the range of prices. A large price range may be unfeasible computationally, and a short-range may miss the optimal prices. For example, if it is optimal to give a product for free (and to set a positive price on another), that would not be accomplished by a Q-learning algorithm that only considers positive real numbers. Additionally, no one will wait eight minutes to compute an optimal price due to testing many different prices to know the fare of an Uber ride. Thus, algorithms must balance complexity and exhaustiveness.

In this regard, the PSO algorithm does not suffer the dimensionality problem, and in its basic form, it is suitable to address multidimensional optimization problems. More technically, another advantage of the PSO arises when the objective function is non-convex. In such a case, Q-learning algorithms may become stuck at local optima while PSO, just by virtue of sampling from a large population, would converge to the global solution easily. Additionally, PSO does not require keeping track of all the potential actions, which makes it simpler computationally. Nonetheless, a key drawback of PSO is that it is not as intuitive or

closely linked to economic modeling as Q-learning, which would explain its relative lack of use in economics. In general, Q-learning is more economically sound than PSO because the former is based on the Bellman equation, which is broadly used in economics. Lastly, there are also other limitations when using PSO. For example, in comparison 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, but if we are interested in analyzing outcomes (like in this work), this constraint is not binding. However, there are other reasons to focus on PSO. For example, from an economic point of view, there is evidence that Q-learning algorithms may lead to collusive outcomes [20,29], while there is no such evidence with PSO yet. Additionally, PSO does not require the problem to be differentiable, which makes its application to markets that show discontinuities possible. Finally, some readers that are familiar with this literature may question why we do not consider genetic algorithms (GA). On the one hand, PSO has the same ability to find a global optimum as genetic algorithms but can find optima faster than GA [32]. On the other hand, from an economic point of view, some modeling options may seem arbitrary but have a considerable impact on the outcomes [9]. Therefore, the best option to carry out our experiment is the PSO algorithm.

3. Materials and Methods

3.1. The Particle Swarm Optimization (PSO) Algorithm

PSO is a stochastic optimization technique that is based on generating random points in a multidimensional space (particles) that move towards an optimal solution by sharing information about which points perform better. The idea of PSO came from watching the way flocks of birds, fish, or other animals adapt to avoid predators and to find food by sharing information [33]. Thus, particles could share information with all other particles (one global swarm) or may be organized in different groups (swarms) that only share information among them. This concept can easily be extended to price competition by assuming that each company may control a “swarm of prices”. Another way to economically interpret PSO is to consider 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. By repeating this operation multiple times, companies can set the best price given certain market/regulatory conditions. Note that, to carry out those actions, companies do not need an algorithm, but the use of AI allows them to test a broader set of prices and faster.

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 randomly drawn from a $U(0, 1)$ distribution but, 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 is influenced by 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^l) 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^g). Formally, the price $p_{i,t}$ at a time t is updated as follows:

$$p_{i,t} = p_{i,t-1} + v_{i,t-1} \quad (1)$$

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

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 factor and swarm confidence factor, respectively; and u_1 and u_2 are $U(0, 1)$ random numbers. Note that Equation (1) can be rewritten as $\Delta p_i = v_{i,t}$, which resembles a gradient similar to the

best-response function of a classical economic model. Similarly, by rewriting Equation (2), we have $\Delta v_i = f(p_i, p^l, p^g, w)$, which resembles the slope of the best-response function. However, we have to be aware that this is only a resemblance and that the way PSO works differs from traditional economic modeling.

In economic games, the payoff of a firm also depends on the prices of other companies. Thus, a price that was optimal 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, Reference [34] stated that the inertia weight w in Equation (2) is critical for the PSO's convergence behavior. There is a trade-off between exploration and exploitation. As it is common in algorithmic pricing, an initial large w is desirable to explore the space that decreases over time to exploit the best results [9,20]. In this sense, we assume the following:

$$w_t = (1 - w_0)^t \quad (3)$$

where w_0 is a constant initial decrease parameter.

3.2. Market Environment

We use a simple model of price competition with LOGIT demand and, for simplicity's sake and without loss of generality, we assume constant marginal costs [35]. Other cost functions better resemble some of the motivating examples, especially the case of Uber. However, we are interested in comparing the theoretical equilibrium price with the algorithmic one, and the constant marginal cost assumption keeps the model simple. Nonetheless, other cost functions were tested, but the qualitative results remain robust. This model has been applied extensively in empirical work, and it also is a useful framework to address algorithmic price competition [20]. Following the literature on algorithmic pricing, each company will face this demand function at each time t . In the motivating examples, it may represent how sellers set prices on Amazon for each 20 or 30 min, or how Uber and Lyft set prices in specific areas. This price competition game assumes that there are n differentiated products and an outside good. Formally, the demand for product i is as follows:

$$q_i = \frac{e^{\frac{a_i - p_i}{\mu}}}{\sum_{j=1}^n e^{\frac{a_j - p_j}{\mu}} + e^{\frac{a_0}{\mu}}} \quad (4)$$

Parameter a_i is the product quality of product i , which can be thought of as 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 $\mu \rightarrow 0$. In our model, we assume two different scenarios, one where the horizontal differentiation is high and another where it is low. Each product is supplied by a different firm, so n is also the number of firms. Lastly, the profits of each company are $\pi_i = (p_i - c_i)q_i$, where c_i is the marginal costs.

3.3. Baseline Parametrization

PSO is nondeterministic. Thus, it is not guaranteed to return to the same solution in each run. Additionally, given the different parameters that are possible to modify, the speed and accuracy of the algorithm may differ. Therefore, it is fundamental to analyze the robustness by comparing the performance of the algorithm under alternative parameter values against the theoretical equilibrium.

Initially, we focus on a baseline economic environment that consists of a symmetric duopoly ($n = 2$) in two scenarios of low ($\mu_L = 0.01$) and high ($\mu_H = 1$) differentiation

with $a_0 = 0$, $a_i = 1.18$, and $c_{i,L} = 0.18$ or $c_{i,H} = 0$ in the scenarios, respectively. 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 $m = 5$. We also limit the range of evolutionary velocity, $v_i \in [-0.3, 0.3]$. This is a common assumption to avoid “jumping” between corner solutions. Similar parametrizations can be found in works where prices are also set by algorithms, such as [10] or [20]. The comparison between two scenarios with low and high differentiation is not trivial, as shown by [36], when using a Q-learning algorithm to set prices. Thus, we pay special attention to comparing both scenarios.

4. Results: Optimal Simulated Prices

In this section, we focus on the baseline scenario and explore the optimality of the PSO solution. This exercise aims to show that PSO can be implemented as a price-setting algorithm. To do so, we perform different experiments. For our purposes, an “experiment” consists of 1500 repeated iterations. Each experiment is run under the same set of parameters 30 times to reduce the stochastic noise. We choose 1500 iterations to guarantee that PSO is stable. Five hundred iterations are enough, but after the HILP event, we allow twice that number to guarantee that the shock does not generate instability. On the other hand, 30 repetitions of the same framework are enough to find patterns. We have tried more iterations, but there are no differences in the results.

In Figure 1, we depict the average price and demand of the base model at each one of the 1500 iterations. The PSO algorithm reproduces quite successfully the price and demand of the differentiated market. On the other hand, it sets a higher-than-optimal price in the undifferentiated market. This result is also found by [36] using a Q-learning algorithm. Nonetheless, demands are quite volatile in this market, as we observe in Figure 1. Thus, PSO may “learn” to avoid large confrontations by setting prices higher than optimal.

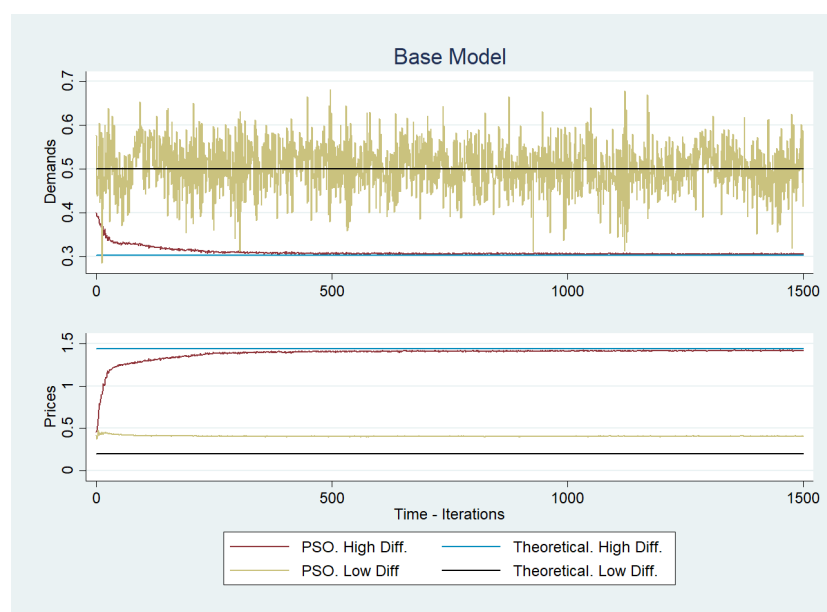


Figure 1. Simulated demands and prices over 1500 iterations in the two markets with high (red) and low differentiation (gold). In blue and black are their theoretical counterparts.

Although in Figure 1 the parameters are fixed, the results are robust to changes in the parameters. The stability analysis is available in Appendix A.

4.1. High-Impact, Low-Probability (HILP) Events

To simulate a HILP shock that reproduces something similar to the motivating examples, we assume that, at the 500th iteration, a shock occurs and each price “jumps” to a new position equal to the current price plus one and that the profits associated with

those new positions are twice the current ones. For example, if the best-found price at the 499th iteration was 0.25 and the associated profits were 1, after the shock, the best-found price and profits would be 1.25 and 2, respectively. We experimented under more mild conditions, i.e., with price jumps ranging from 0.25 to 1 and profits increasing from 25% up to 100%. In all the cases, the conclusions remained the same. In the end, those jumps must be significant to be noticeable and, more importantly, to raise concerns about their negative consequences (e.g., to enforce the price gouging policies of Amazon or eBay). In this sense, it is relevant to distinguish between peak-and-valley and HILP events. For example, energy markets face peak-and-valley events every day, but rarely do they face events that shut down the entire grid, which would be an example of HILP events. Those events are the ones we are interested in and the ones we try to simulate in this paper.

In Figure 2, we observe the consequences of a HILP event in how the PSO set prices. In the aftermath of the event, prices tend to decrease slowly towards the initial values but always remain higher than before. This process of convergence is slow, and it takes several iterations to drive the prices down, with almost 250 iterations in the differentiated market while almost 100 iterations in the undifferentiated market.

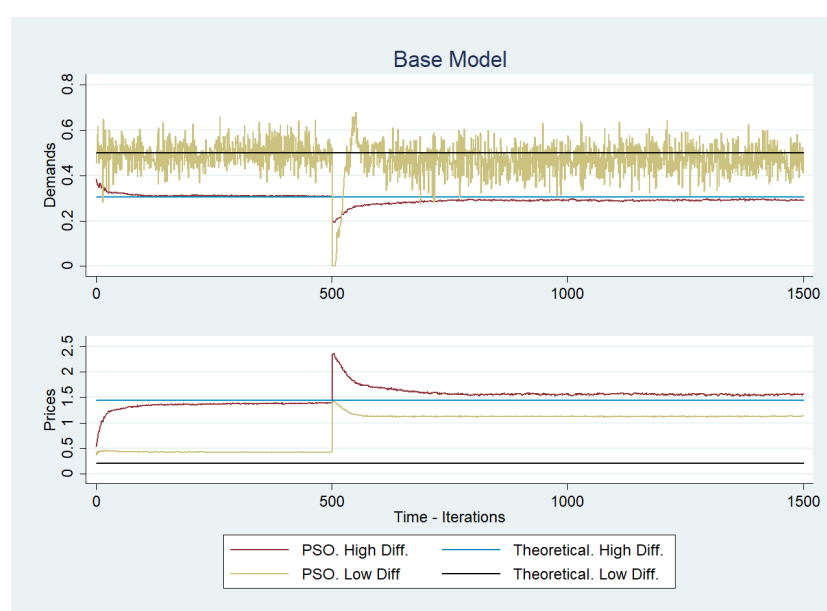


Figure 2. High-impact, low-probability (HILP) event on demands and prices in the two markets.

The algorithm interprets that a significant increase in prices is necessary to bring balance to the market. However, our results show that the effect is not homogeneous. In the differentiated market, the algorithm can return to the neighborhood of the original equilibrium price. In other words, the algorithm is capable of detecting unusual shock and of returning to the original price afterward. This could be the case of those cities that experienced HILP events but that have a differentiated supply of ride-hailing services or cases that are less extreme than HILP events. In those cases, surge pricing is short-lived. In the undifferentiated market, prices never return to previous levels. It is this case that may raise new concerns given that, after a HILP event, prices never return to the original equilibrium. This case represents situations in which products may be essential and undifferentiated in consumers' eyes. For example, after a terror attack on public transportation, people want to go home but do not care about whether you can choose the music played or the type of car. Uber is a great example of the impact of these HILP events. Although its algorithm can deal with peak-and-valley events such as storms, fairs, or conferences, HILP events such as terror attacks or heavy storms have led to situations so extreme that required suspension of the algorithmic pricing in the US, the UK, France, and Australia [1,11,37,38]. Similarly, retailers have experienced similar effects. A study by the US Public Interest Research Groups (PIRGs) has found that prices of essential products

on Amazon were more than 50% higher than average six months after the first hit of the COVID-19 pandemic in the US [39]. Although PIRGs are advocacy groups and their findings can be called into question, similar concerns have arisen in Spain, Romania, Italy, and Greece, where competition authorities have announced investigations into price hikes regarding sanitary products [40]. Thus, it is likely that platforms may consider that unusual high prices during extended periods violate their price gouging policies, which may get sellers suspended or banned and may drive some of them out of business. Even if selling platforms do not take steps to prevent price gouging, in some US states and European countries, consumer protection laws can be enforced, which may have even worse effects on the long-term sustainability of those sellers [40].

So far, these results show that, even in situations where there is only one global equilibrium, a significant shock may move prices away from equilibrium and keep them in supracompetitive levels when companies use pricing algorithms. Thus, an interesting question is what can we do to mitigate such an impact.

4.2. Mitigating Factors

The previous results highlight that a HILP event may completely distort the market. In this regard, an exhaustive analysis of the different methods, protocols, or forces that may mitigate such events is out of the scope of this work. The study of algorithmic regulation and policies is still in its infancy. To the best of our knowledge, the only work that analyses algorithmic pricing policies so far is [36] that focuses on policies to mitigate collusion. We focus on two factors that are general enough to be common to any market in which algorithmic pricing is in use: market forces and algorithmic design.

4.2.1. Market Forces

As classical industrial organization theory shows, strategic interactions with competitors play a key role in setting prices. In this regard, an interesting scenario to consider is when only one of the companies uses algorithmic pricing while the other sets prices following a best-response function, as in classical theoretical models. In Figure 3, we observe that the initial shock is less severe than before, which also implies a faster recovery. However, neither in the low differentiated nor in the highly differentiated markets do the prices completely returned to their previous levels. In fact, the relative increase in the low differentiated market is larger than in the highly differentiated one. Nonetheless, the return is faster than before. Thus, market forces may limit but not mitigate the effect of HILP events. Despite the competition, algorithms may be trapped in higher-than-optimal prices as a consequence of algorithmic pricing. This result is consistent with the anecdotal evidence of the motivating examples, given that the only solution to deal with such events was either to suspend the algorithmic pricing or to set price caps. This situation raises another concern for public authorities. Prices stay above competitive levels despite just one company using algorithmic pricing, which may raise concerns about tacit collusion. However, whether those high prices are consistent with collusive practices or a design feature or are a simple failure of the algorithm to learn the real equilibrium is unknown. Nevertheless, the message is clear. If not taken into account, HILP events may lead to higher-than-optimal prices, which may trigger the price gouging policies of selling platforms and may put sellers that use price algorithms at risk.

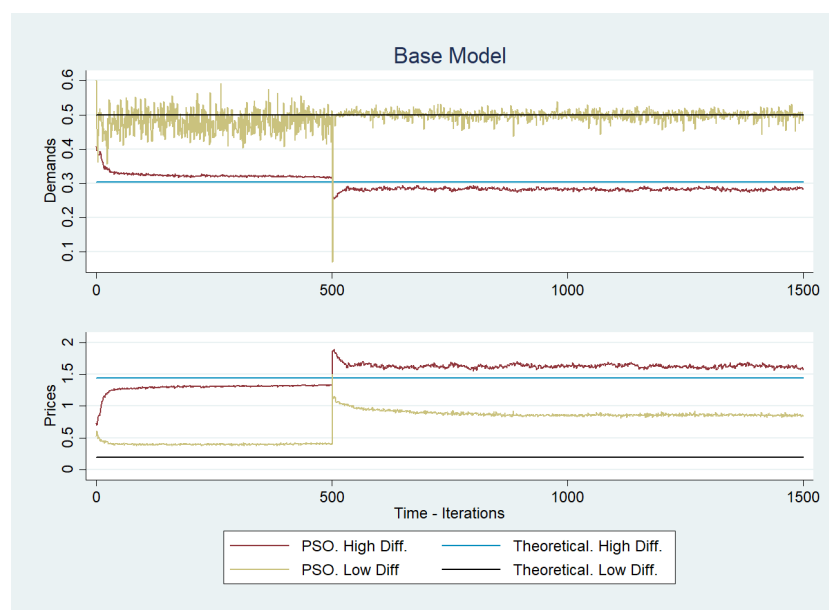


Figure 3. Impact on demands and prices: market forces.

4.2.2. Design

Another potential way to mitigate the impact of HILP events is to design the PSO in such a way that those events have limited effects. This may be accomplished by designing the algorithm to “tend” to set competitive prices. A simple way to introduce such a behavior is by introducing a new learning parameter (l_3) that is influenced by an exogenous price level defined by the company (p^G); technical details are reported in Appendix B. This last case may represent those situations in which the company knows that, historically, prices are normally around p^G . In this way, companies let the algorithm look for better prices but with a tendency to look in the neighborhood of that price. Another possibility is to assume that p^G is imposed by authorities. For example, that may be the case in regulated markets. Following the motivating examples, in an emergency, some algorithms may be trapped in price gouging solutions, but this learning parameter will “force” prices to move away from those solutions. To make this example simple, we assumed that p^G is exogenously set and that it is equal to the theoretical equilibrium prices, which may be a good representation of historical prices. However, other ways of setting p^G could be made, for which the impact may differ from the one analyzed here.

The interesting point of this modification is how it may deal with a HILP event. In Figure 4, we observe that, after the shock, it starts a correction phase. As in the previous section, the effect takes longer in the differentiated market than in the undifferentiated one. Intuitively, market contestability may play a key role in how PSO set prices. Note that, in comparison with the differentiated market, in the undifferentiated one, PSO sets proportionally larger prices after a shock, which may be a consequence of learning to avoid confrontation. In this regard, it seems that market forces and design may help to mitigate price gouging, but in the case of the PSO, they alone are not enough to remove it. Although it may look like a trivial problem, such events are exceptional, and it seems to be a defying challenge for algorithms, given that, since 2013, the only solution so far is to suspend the algorithm. In conclusion, the results show that concerns about price gouging will be more common with algorithmic pricing. This raises a new concern for companies that rely on selling goods on platforms such as Amazon or eBay because they could be banned if their algorithms get stuck on higher-than-optimal solutions. Thus, addressing HILP events should become an essential part of algorithmic testing. If not, the sustainability of digital sellers can be compromised.

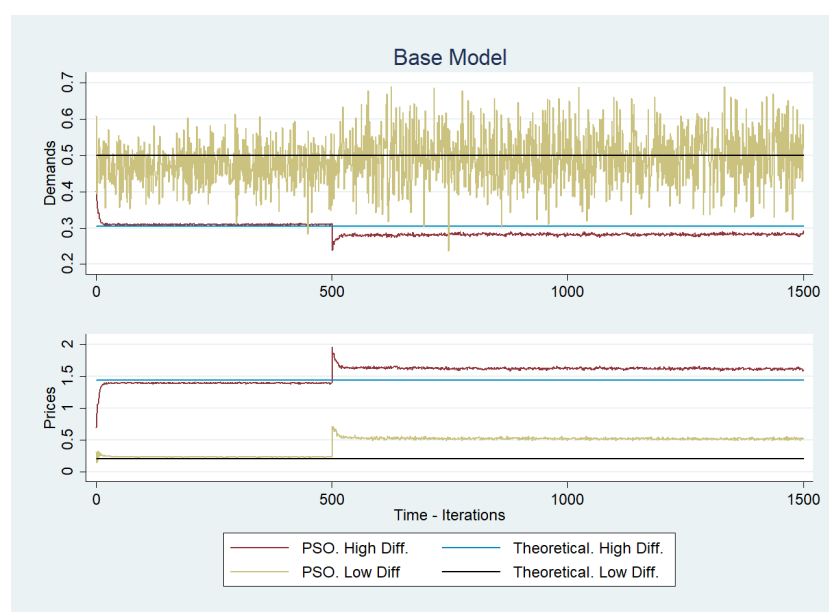


Figure 4. Impact on demands and prices: modified particle swarm optimization (PSO).

5. Conclusions

Algorithms setting prices for products are a reality in many businesses, such as ride-hailing services or web-based markets such as Amazon, and they are essential for the sustainability of long-term strategies of many companies. However, emergencies such as heavy storms, wildfires, or terror attacks have shown that such algorithms may lead to significant increases in prices that are sustained over time, which may backfire on companies in the form of price gouging accusations. Thus, it is increasingly relevant to pay attention to algorithmic pricing not only because of the life-saving consequences that a fair price can have in emergencies, such as a bottle of water after a hurricane [41] or airfares during an evacuation [42], but also because a price gouging accusation may get sellers banned from platforms, which may endanger their businesses.

The preliminary results presented in this work show that we cannot simply ignore the role of HILP events on price algorithms. Especially in undifferentiated markets, prices after a HILP event may remain higher than optimal, which would more likely induce price gouging accusations. Although competition and the introduction of specific instances in the design of the algorithm may help in mitigating those increases, they cannot be removed completely. Currently, the only way to mitigate those situations is to “disconnect” the algorithm. However, at that point, it may be too late. These results send a clear message. As the use of algorithms to set prices expands, it is more likely that we will experience price gouging after HILP events. In this regard, we recommend testing HILP events in sellers’ algorithms before going to the market. Unintentionally, pricing algorithms may generate price gouging episodes that may compromise the long-term sustainability of many digital sellers.

Nonetheless, this work presents several limitations worth highlighting. First, we assume a specific demand function that, although general, is by no means exhaustive to cover all potential markets. Similarly, we assume only one optimization algorithm but, in real markets, companies may use pricing algorithms that differ from the PSO. In this regard, the intuition of our results is that HILP events may cause a significant disruption even in simple markets. However, the exact disruption will depend on the market and the specific algorithms used. Finally, a key limitation is the lack of information regarding the specific algorithms in use in real markets. Algorithms are a well-stored secret and such a circumstance limits to what extent an empirical approach is possible.

Therefore, a future area of research is to address what happens in other market environments, such as those where consumers are not fully rational or those where the companies are multi-sided platforms, as in [43]. An underexplored case that also deserves attention is when there are more than two competing companies. To the best of our knowledge, there is no such work yet despite the significant influence that the number of competitors has in the markets. On the other hand, in this work, we have focused on the shock and posterior mitigation of a HILP event. Nonetheless, the identification of early warnings is also interesting, which could help in diminishing the effect of those shocks. In this sense, prevention is as relevant as mitigation and risk management could provide an interesting base to address these concerns. Empirically, it could be interesting to address whether social networks or search analytics tools can identify some early warnings that can “feed” the algorithms and that can react by setting a price cap to avoid price gouging, for example.

Author Contributions: Conceptualization, J.M.S.-C., A.T. and G.L.; methodology, J.M.S.-C.; software, J.M.S.-C.; validation, J.M.S.-C.; formal analysis, J.M.S.-C.; investigation, J.M.S.-C.; resources, G.L.; data curation, J.M.S.-C.; writing—original draft preparation, J.M.S.-C., A.T. and G.L.; writing—review and editing, A.T. and G.L.; visualization, J.M.S.-C.; supervision, A.T. and G.L.; project administration, A.T.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the UPM institutional project “Development of open innovation ecosystems” grant number RP16001831.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the editors and three anonymous referees whose comments helped improve this paper substantially.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
RL	Reinforcement Learning
EA	Evolutionary Algorithms
PSO	Particle Swarm Optimization

Appendix A. PSO Convergence and Stability

A key step when analyzing a pricing algorithm is its sensitivity to different parameters. In this regard, we conducted a set of experiments aimed at showing how different parameters influence the price predicted by the PSO and how it differs from the optimal one. In Figure A1, we analyze the difference between the theoretical price and the PSO predicted price. In Figure A1, a positive value represents that the PSO was not able to reach the optimal price. On the other hand, a negative value represents that the PSO price is higher than optimal. We observe that, in all the experiments, at least 75% of the simulations predict the optimal price with a deviation smaller than 0.2. Nonetheless, it is interesting to observe that differentiation may play a role in those errors. Surprisingly, as differentiation increases (from blue to red boxes), the deviation tends to favor lower-than-optimal prices. It is also worth highlighting that, when differentiation is high, some simulations may lead to extremely low prices while the opposite is true in the undifferentiated market. This result naturally raises collusive concerns. It would be possible that PSO “learns” that the best option for “survival” is to avoid confrontation. However, analyzing this concern is out of the scope of this work, but it is worth paying attention to, as other works have highlighted (see [20,44] or [29]), and it remains an open question for future work.

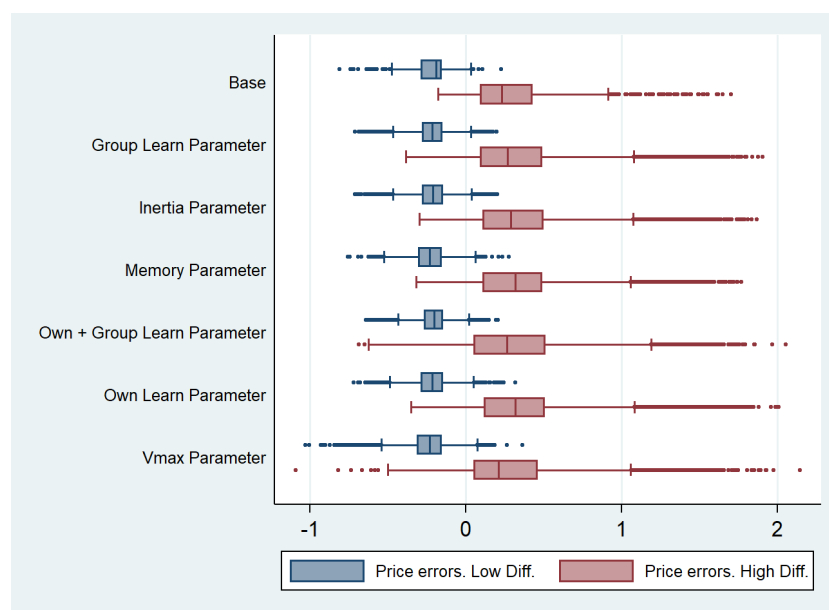


Figure A1. Prices, parameter sensitivity, and difference between optimal and simulated prices in 30 simulations.

Similarly, in Figure A2, we analyze the variability of the predicted demands with respect to the optimal ones. Although at first sight it may look like the PSO fails to estimate demand when differentiation is low, this extreme variability is common in models of price competition with no differentiation, as a small change in price may lead to great changes in demands. In this sense, the algorithm reproduces a key feature of the market. On the other hand, the demand when differentiation is high tends to be well predicted. Thus, different configurations of PSO parameters may lead to different price predictions, but in more than 75% of the cases, those predictions are near the theoretical optimum price (value zero in Figures A1 and A2). Note, however, that we have focused on two extreme cases, but when differentiation is neither too low nor too high, PSO predicts prices closer to the optimal one.

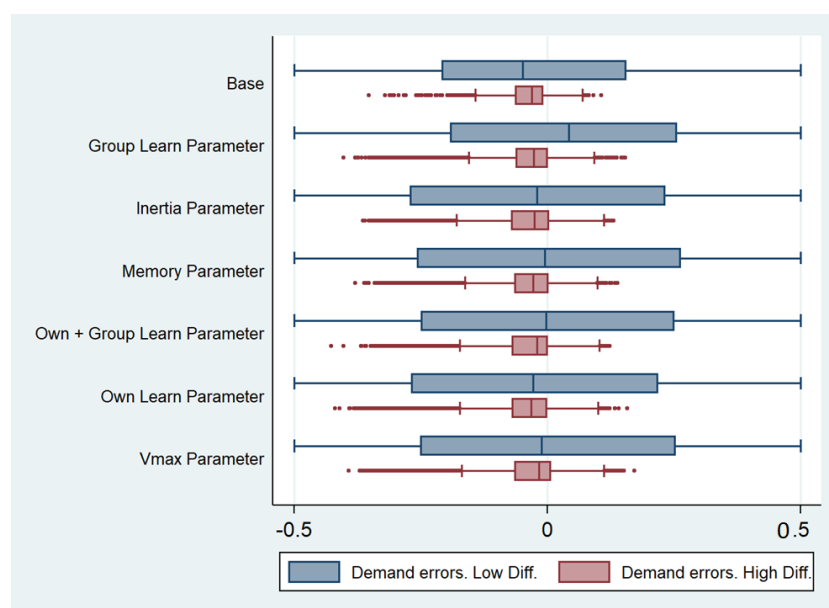


Figure A2. Demands, parameter sensitivity, and difference between optimal and simulated demands in 30 simulations.

Appendix B. PSO Design with Exogenous Price Levels

Formally, the modification of Section 4.2.2 requires including this new learning parameter in Equation (2), which now becomes

$$v_{i,t-1} = wv_{i,t-2} + l_1u_1(p_i^l - p_{i,t-1}) + l_2u_2(p^g - p_{i,t-1}) + l_3u_3(p^G - p_{i,t-1}) \quad (A1)$$

where $u_3 \sim U[0, 1]$. This modification is not trivial, and it modifies the internal behavior of the algorithm. However, it influences neither its stability nor the intuitions of previous sections. All the results, data, and code in this regard are available upon request. The more relevant change is that, on average, the model performs better this time, but this is a consequence of assuming that p^G is equal to the equilibrium prices. This assumption is fundamental, and assuming other prices may completely change the results. For example, in an early version of this experiment, it was assumed that p^G was equal to marginal costs, which drove prices in the differentiated market to become extremely low. However, the stability of the PSO remained similar. Other factors also influence the price of the service, such as quality or brand positioning. Although we focus on prices, all those features play a passive role in this framework via the LOGIT demand. However, it remains unknown how algorithms perform when those features play a more active role.

References

1. Meyjes, T. Uber Slammed for Putting Prices up during London Bridge Terror Attack. 2017. Available online: <https://metro.co.uk/2017/06/04/uber-slammed-for-putting-prices-up-during-london-bridge-terror-attack-6683200/> (accessed on 5 February 2021).
2. Weiner, J. Is Uber's Surge Pricing Fair? 2014. Available online: <https://www.washingtonpost.com/blogs/she-the-people/wp/2014/12/22/is-ubers-surge-pricing-fair/?noredirect=on> (accessed on 6 February 2021).
3. Amazon. Price Gouging Has no Place in our Stores. 2020. Available online: <https://www.aboutamazon.com/news/company-news/price-gouging-has-no-place-in-our-stores> (accessed on 6 February 2021).
4. eBay. Price Gouging Policy. 2021. Available online: <https://www.ebay.com/help/policies/prohibited-restricted-items/price-gouging-policy?id=5106> (accessed on 6 February 2021).
5. Chen, L.; Mislove, A.; Wilson, C. An empirical analysis of algorithmic pricing on amazon marketplace. In Proceedings of the 25th International Conference on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; pp. 1339–1349.
6. Lee, B.; Preston, F.; Green, G. *Preparing for High-impact, Low-probability Events: Lessons from Eyjafjallajökull*; Chatham House: London, UK, 2012.
7. Shartsis, A. In The Future, Everything Will Be Dynamically Priced. 2019. Available online: <https://www.forbes.com/sites/forbestechcouncil/2019/09/10/in-the-future-everything-will-be-dynamically-priced/> (accessed on 6 February 2021).
8. Westerman, G.; Bonnet, D.; McAfee, A. *Leading Digital: Turning Technology into Business Transformation*; Harvard Business Press: Boston, MA, USA, 2014.
9. Zhang, T.; Brorsen, B.W. Particle swarm optimization algorithm for agent-based artificial markets. *Comput. Econ.* **2009**, *34*, 399. [CrossRef]
10. Maschek, M.K. Particle Swarm Optimization in Agent-Based Economic Simulations of the Cournot Market Model. *Intell. Syst. Account. Financ. Manag.* **2015**, *22*, 133–152. [CrossRef]
11. Moore, M. Uber Slammed for Surge Pricing during London Attacks. 2017. Available online: <https://nypost.com/2017/06/04/uber-slammed-for-surge-pricing-during-london-attacks/> (accessed on 6 February 2021).
12. Kim, S. How Uber Chooses Its Surge Price Cap in Emergencies. 2015. Available online: <https://abcnews.go.com/Business/uber-chooses-surge-price-cap-emergencies/story?id=28494303> (accessed on 6 February 2021).
13. Fathian, M.; Sadjadi, S.J.; Sajadi, S. Optimal pricing model for electronic products. *Comput. Ind. Eng.* **2009**, *56*, 255–259. [CrossRef]
14. Hess, T.; Matt, C.; Benlian, A.; Wiesböck, F. Options for formulating a digital transformation strategy. *MIS Q. Exec.* **2016**, *15*, 123–139.
15. Schwertner, K. Digital transformation of business. *Trakia J. Sci.* **2017**, *15*, 388–393. [CrossRef]
16. Ebert, C.; Duarte, C.H.C. Digital Transformation. *IEEE Softw.* **2018**, *35*, 16–21. [CrossRef]
17. Cheng, H.K.; Tang, Q.C. Free trial or no free trial: Optimal software product design with network effects. *Eur. J. Oper. Res.* **2010**, *205*, 437–447. [CrossRef]
18. Bauer, J.; Jannach, D. Optimal pricing in e-commerce based on sparse and noisy data. *Decis. Support Syst.* **2018**, *106*, 53–63. [CrossRef]
19. Bombin, A.; Magerramov, P.; Silakova, L.; Shchekoldin, A. *Digital Product Pricing Algorithm Development*; Technical Report; EasyChair: Manchester, UK, 2020.
20. Calvano, E.; Calzolari, G.; Denicolò, V.; Pastorello, S. Artificial Intelligence, Algorithmic Pricing, and Collusion. *Am. Econ. Rev.* **2020**, *110*, 3267–97. [CrossRef]
21. Robinson, J. An iterative method of solving a game. *Ann. Math.* **1951**, *54*, 296–301. [CrossRef]

22. Vives, X. *Oligopoly Pricing: Old Ideas and New Tools*; MIT Press: Cambridge, MA, USA, 1999.
23. Fudenberg, D.; Drew, F.; Levine, D.K.; Levine, D.K. *The Theory of Learning in Games*; MIT Press: Cambridge, MA, USA, 1998; Volume 2.
24. Charpentier, A.; Elie, R.; Remlinger, C. Reinforcement Learning in Economics and Finance. *arXiv* **2020**, arXiv:2003.10014.
25. Brown, Z.; MacKay, A. Competition in pricing algorithms. *SSRN* **2020**. [[CrossRef](#)]
26. Vives, X. Innovation and competitive pressure. *J. Ind. Econ.* **2008**, *56*, 419–469. [[CrossRef](#)]
27. Ongwese, E. Uber Drivers Sue to Gain Access to its Secret Algorithms. 2020. Available online: <https://www.vice.com/en/article/v7gyp4/uber-drivers-sue-to-gain-access-to-its-secret-algorithms> (accessed on 6 February 2021).
28. Waltman, L.; Kaymak, U. Q-learning agents in a Cournot oligopoly model. *J. Econ. Dyn. Control* **2008**, *32*, 3275–3293. [[CrossRef](#)]
29. Klein, T. Autonomous Algorithmic Collusion: Q-Learning under Sequential Pricing. *Amst. Law School Res. Pap.* **2019**. [[CrossRef](#)]
30. Schmidhuber, J. Evolutionary computation versus reinforcement learning. In Proceedings of the 26th Annual Conference of the IEEE Industrial Electronics Society, IECON 2000, 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation, 21st Century Technologies, Nagoya, Japan, 22–28 October 2000; Volume 4, pp. 2992–2997.
31. Gaskett, C.; Wettergreen, D.; Zelinsky, A. Q-learning in continuous state and action spaces. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Sydney, Australia, 6–10 December 1999; Springer: Sydney, Australia, 1999; pp. 417–428.
32. Hassan, R.; Cohanin, B.; De Weck, O.; Venter, G. A comparison of particle swarm optimization and the genetic algorithm. In Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Austin, TX, USA, 18–21 April 2005; p. 1897.
33. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science (MHS'95), Nagoya, Japan, 4–6 October 1995; pp. 39–43.
34. Chatterjee, A.; Siarry, P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Comput. Oper. Res.* **2006**, *33*, 859–871. [[CrossRef](#)]
35. Train, K.E. *Discrete Choice Methods with Simulation*; Cambridge University Press: New York, NY, USA, 2009.
36. Johnson, J.; Rhodes, A.; Wildenbeest, M.R. Platform design when sellers use pricing algorithms. *SSRN* **2020**. Available online: <https://ssrn.com/abstract=3753903> (accessed on 1 November 2020). [[CrossRef](#)]
37. Hawkings, A. Tracing the Spread of Uber Rumors during the Paris Terrorist Attacks. 2015. Available online: <https://www.theverge.com/2015/11/16/9745782/uber-paris-terrorism-rumors-twitter-facebook> (accessed on 9 February 2021).
38. Corporation, B.B. Uber ‘Truly Sorry’ for Price Rise during Sydney Siege. 2014. Available online: <https://www.bbc.com/news/technology-30595406> (accessed on 9 February 2021).
39. Groups, U.P.I.R. *High COVID-19 Prices Persist on Amazon*; Technical Report; US PIRG Education Fund: Mountain View, CA, USA, 2020.
40. OECD. *Exploitative Pricing in the Time of COVID-19*; Technical Report; OECD: Paris, France, 2020.
41. Snider, M. Hurricane Irma: A Case of Water Sells for \$99.99 on Amazon as Residents Fear Price Gouging. 2017. Available online: <https://eu.usatoday.com/story/money/2017/09/06/hurricane-irma-case-water-sells-99-99-amazon-residents-fear-price-gouging/636893001/> (accessed on 6 February 2021).
42. Coorsh, K. Air Canada Addresses Accusations of Price-Gouging during Fort McMurray Evacuation. 2016. Available online: <https://www.ctvnews.ca/business/air-canada-addresses-accusations-of-price-gouging-during-fort-mcmurray-evacuation-1.2891081> (accessed on 6 February 2021).
43. Katsamakas, E.; Madany, H. Effects of user cognitive biases on platform competition. *J. Decis. Syst.* **2019**, *28*, 138–161. [[CrossRef](#)]
44. Schwalbe, U. Algorithms, machine learning, and collusion. *J. Compet. Law Econ.* **2018**, *14*, 568–607. [[CrossRef](#)]