Sanchez-Cartas *Complex Adapt Syst Model (2018) 6:2*
https://doi.org/10.1186/s40294-018-0053-7

Complex Adaptive Systems Modeling

METHODOLOGY Open Access

# Agent-based models and industrial organization theory. A price-competition algorithm for agent-based models based on Game Theory

Juan Manuel Sanchez-Cartas*

*Correspondence:
juanmanuel.sanchez@upm.es
Centre for Technology Innovation, Universidad Politecnica de Madrid, Campus de Montegancedo, Madrid, Spain

**Abstract**

**Purpose:** Simulating markets using agent-based models must consider pricing. However, the strategic nature of prices limits the development of agent-based models with endogenous price competition.

**Methods:** I propose an agent-based algorithm based on Game Theory that allows us to simulate the pricing in different markets. I test the algorithm in five theoretical economic models from the industrial organization literature.

**Results:** In all cases, the algorithm is capable of simulating the optimal pricing of those markets. It is also tested in two more cases: one in which the original work fails to predict the optimal outcome, and another one that is quite complex to solve analytically. Lastly, I present two potential extensions of this algorithm: one dynamic, and another one based on quantity competition.

**Conclusions:** This algorithm opens the door to the extensive inclusion of pricing in agent-based models, but also, it helps to establish a link between the industrial organization literature and the agent-based modeling.

**Keywords:** Agent-based models, Algorithmic game theory, Price optimization, Industrial organization

## Background

Prices play an essential role in any market and understanding how they are fixed is a fundamental part of the Economic Science. However, complex problems such as social networks or the launching of new digital platforms can set new challenges in understanding how those prices are fixed.

To tackle these complex problems, some researchers have adopted the agent-based modeling approach. But, there is a lack of integration between this approach and the industrial organization literature. This lack of integration is clearly depicted by the

absence of works that address prices in agent-based models (ABM), despite being considered an essential variable in markets.[1]

To address this issue, I propose an algorithm for agent-based models that simulates price competition among companies. This algorithm establishes a new link between the industrial organization literature and the agent-based modeling. It is based on Game Theory, and it guarantees the optimality of consumers' and companies' behavior without needing to use the equilibrium equations of any theoretical model, nor relying on maximizing (minimizing) any real function.[2] Intuitively, this algorithm resembles the best response map but without assuming any particular theoretical model or function. The algorithm encompasses two sub-algorithms, one for consumers and another one for companies. Both sub-algorithms encompass several behavioral rules that are combined to simulate the behavior predicted by Game Theory. In this sense, it is possible to address markets with heterogeneous decisions-makers, asymmetric information flows and levels, continuous or discontinuous behaviors, etc. and without assuming that decision-makers carried out complicated mathematical manipulations, and at the same time, it is also possible to guarantee the optimality and rationality of our results.[3]

We test the algorithm in five different theoretical models, and we prove that the algorithm reproduces the Nash equilibria of those models. We also consider two extended versions of two theoretical models that are quite complex to solve. We prove that the algorithm also works in those cases. Lastly, we consider two more cases. The first one is dynamic, and the second one is based on quantity competition. In those two cases, we show that the algorithm is easily adaptable to other frameworks that are not static price-competition games.

This work does not pretend to provide groundbreaking evidence that agent-based models are better than other alternatives. We only try to establish a bridge between agent-based modeling and the mainstream industrial organization literature. And to do so, we apply the agent-base modeling to well-known theoretical models.

#### Agent-based models and economic theory. An ongoing issue

The situation of agent-based models in Economics can be summarized as follows: *Despite the power of ABM, widespread acceptance and publication of this method in the highest-level journals has been slow. This is due in large part to the lack of commonly accepted standards of how to use ABM rigorously*, Rand and Rust (2011). This problem is not new, but although some advances are taking place, there is plenty of room for improvement.

Rand and Rust (2011) argues that, a common perception of agent-based models is that they are "toys" because of the lack of documentation, proper testing or theoretical background. Although some attempts have been made, theoretical economic models are rarely considered in the agent-based model literature.[4] Those works which consider a

[1] *In the current agent-based [literature] [...] research on the price, the most important attribute of a product, are very rare*, Diao et al. (2011).

[2] It works in both, discrete and continuous frameworks. If the optimum exists, the algorithm can identify it. If it does not exist, the algorithm chooses a second-best solution that maximizes the utility (profits) of users (companies) given the actions of companies (users) in the local area.

[3] We define a rational agent as the one who makes those choices that maximize their subjective utility given a specific set of information.

[4] In this work, we consider only microeconomic models because they tend to be simpler in the number of relationships they take into account than macroeconomic models.

theoretical framework are a minority, and we only can highlight some examples such as Rixen and Weigand (2014), Hamill and Gilbert (2016), Barr and Saraceno (2005) and Chang (2011). Nonetheless, they present two shortcomings:

- They rely on the equilibrium equations of the theoretical models, so the simulated markets are constrained by the theoretical assumptions.
- They tend to assume other interactions among the agents even when the equilibrium of the theoretical model does not consider such interactions, which if taken into account, may change the equilibrium. Therefore, there is no standard rule or procedure to consider how to implement such theoretical frameworks.

Finally, another relevant shortcoming is that they tend to consider competition in quantities, particularly, the Cournot model. For instance, Barr and Saraceno (2005) investigates how environmental and organizational factors affect the equilibrium outcome of a repeated Cournot model. Chang (2011) analyzes entry and exit in an industrial market characterized by turbulent technological processes and by quantity competition.

Rixen and Weigand (2014) analyzes the diffusion of smart meters and, although they try to relax some assumptions, they remain constrained by the Cournot model. Lastly, Hamill and Gilbert (2016) shows how Cournot models can be simulated using agent-based models, but it relies on the equilibrium outcomes of the basic Cournot model as in the previous works. There are also other works which do not assume theoretical frameworks, but assume exogenous and non-optimal prices such as Fuks and Kawa (2009), Zhang and Brorsen (2011) or Diao et al. (2011). To the best of our knowledge, only Leeuwen and Lijesen (2016) has considered a market with endogenous price competition but, it is limited to a Hotelling model, it is designed *ad-hoc*, it cannot be applied to other cases, it makes small but systematic errors when predicting prices, and it is not efficient when there are many consumers or companies.

Given this situation, two questions arise:

- Can theoretical models be simulated following a set of standard rules?
- Can price competition be implemented in agent-based models following a set of common rules in concordance with Game Theory and economic intuitions?

The first question represents the common problem that each economist faces when dealing with agent-based models and, probably, it will take some decades to achieve a convergence in standards and criteria. On the other hand, the second question is the one we are going to answer in this work. We analyze an agent-based algorithm that simulates the pricing behavior that is assumed in theoretical economic models. This algorithm allows us to identify each component individually and to test its validity against the theory. The relevant contribution of the algorithm is twofold: First, it does not rely on the differentiability of an aggregate equation, but on the optimization of the agents' decisions. Second, it guarantees the optimality of the users' and companies' decisions (buying decisions and prices respectively) in ABM.

## Experimental: the price-competition algorithm. Applications to theoretical models

### The algorithm

The algorithm is designed in a modular way. In this sense, each part of it can be used in different contexts and for different purposes. It consists in two sub-algorithms: The consumers' and the companies' algorithm. The consumers' reproduces the buying decisions of consumers and allows us to generate demands. The companies' sub-algorithm reproduces the process by which companies choose the price levels. The continuous interaction of both leads us to the equilibrium (or equilibria).

#### *The consumers' algorithm*

Our framework considers that each consumer at each moment of time $t$ has an utility function $U(\cdot)$. Each consumer considers the utility he/she obtains from each company he/she knows. If no company offers a worthy product, $U(\cdot) < 0$, the consumer abandons the market; but if there is a company which offers a valuable product, $U(\cdot) \geq 0$, the consumer will compare those products and will buy the one that maximizes its utility. All the consumers who buy the same product from the same company form the demand addressed to that company. In Fig. 1 is depicted a schematic version of the algorithm. Up to this point, this algorithm is not new, and similar algorithms can be found in other works. However, this algorithm is scalable. It presents four features that differentiate it from other proposals:[5]

- Its modularity. This algorithm can be used independently of the price competition.
- We do not impose any demand function, the demands are the emergent result of the consumers' decisions. It only matters how consumers make decisions.
- Consumers make decisions based on their utility functions, which can be discrete or continuous and have as many parameters as necessary. However, those functions are flexible, and each consumer has its own utility function. That implies the possibility of introducing multi-dimensional heterogeneity, externalities, etc. Also, we do not impose fixed utility functions, those utilities can be dynamic.
- Although we assume perfect information throughout the paper, the algorithm can be adapted to deal with those cases in which information is not perfect. For example, we can assume there is a network that connects consumers and by which information flows.

#### *The companies' algorithm*

This sub-algorithm (Fig. 2) is composed of four sub-sub-algorithms.[6]

1. In the first algorithm, each company considers a change in prices (increase or decrease). The parameter $\epsilon$ controls this change (iteration parameter). The algorithm requires a starting price that can be any real number.[7]

[5] The pseudo-code is available in the Annex. The consumer's algorithm corresponds to the first two sub-algorithms of the "Static situation of the market", i.e. utilities and demands.

[6] The pseudo-code of those sub-sub-algorithms is in the Annex.

[7] There is no range of prices previously fixed. The prices are free to vary, but it is important which is the initial price that we choose. It is recommendable to try several different prices with the only condition of not being so high that all the utilities are negative.

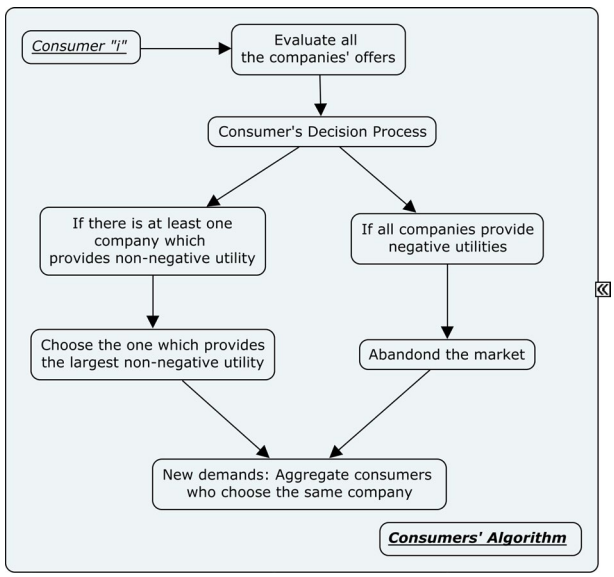


**Fig. 1** Consumers' algorithm

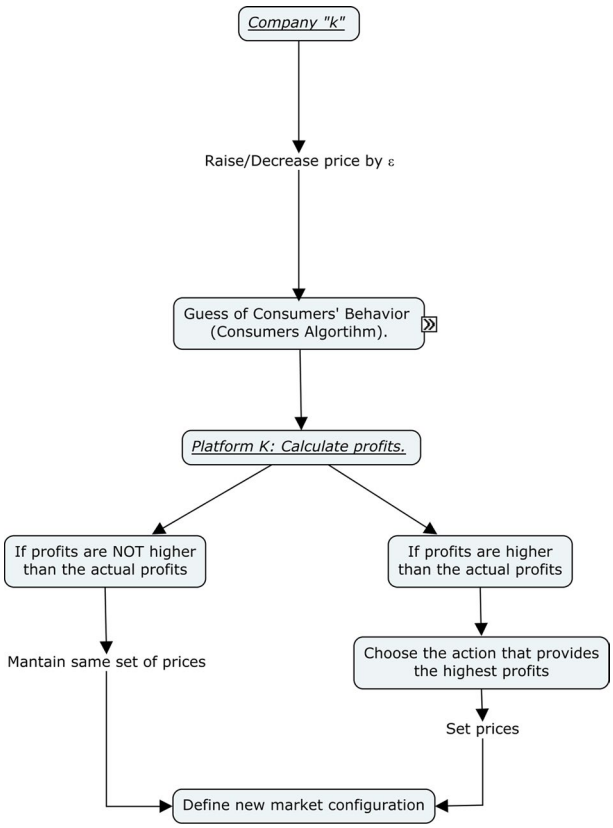


**Fig. 2** Companies' algorithm

2. In the second, each company considers how that change in prices will affect the demands. To estimate the impact of that change in the demands, they estimate the change in the utilities taking other companies' prices as given. Because all consumers compare the utility that each company's product provides them, the price decisions of companies influence the other companies. If we assume that all companies are rational and fully-informed. They will perfectly forecast the decision of consumers.[8] Nonetheless, the algorithm can consider other assumptions about how companies make their decisions.
3. The third algorithm is an optional one. It is designed to control externalities such as direct and indirect network effects. It addresses the case in which the change in prices affects the decisions of other consumers that, at the same time, influences other consumers and so on. In some cases, when a company changes their prices, that decision attracts other consumers that, at the same time, attract more consumers and so on. This algorithm simulates these feedback loops until this effect disappears (convergent feedback loops or the attraction of all the consumers).
4. Companies compare the three actions: increase, decrease or maintain the prices, and they choose the most profitable one.

As it was stated before, the algorithm resembles the best response map, so the convergence towards the price equilibria depends on the initial price and the size of the changes in prices ($\epsilon$).[9] Once the algorithm has reached an equilibrium, the algorithm will be continuously testing if any deviation is profitable. This algorithm can be considered as a helpful tool to look for local optima. To find global ones, we need to consider different $\epsilon$ -values and different suitable starting prices.

### Theoretical frameworks

To prove that the proposed algorithm is capable of reproducing theoretical models, we need to apply it in several frameworks. We choose five theoretical economic models. All of them share only one feature: companies compete in prices for consumers. This framework only requires knowing the utility functions, so we will not solve those models analytically. However, we provide references to those theoretical models. All the assumptions made in the following sections are the original assumptions of the models. Some of them are very restrictive, but those models are very well known in Economics, and it is easier to prove the algorithm in an environment that is well-known. Later, we will show how the algorithm is capable of dealing with those models but relaxing some of their assumptions.

#### *Horizontal differentiation: the Hotelling framework*

This model[10] assumes there is a large pool of small consumers that are uniformly distributed in a line. At the extreme of that line, there are two companies that compete in prices to attract consumers. Each consumer has to move from his/her position to the

[8] This is the case in the vast majority of theoretical models that we simulate in this work.

[9] This opens a discussion about how the algorithm tackles the convergence, but as long as this algorithm resembles the best response map of price competing markets, the convergence towards the equilibrium is similar to the one expected using Bertrand's Intuition or Paradox, [Tirole and Matutes (1990), Chapter 5].

[10] See [Belleflamme and Peitz (2015), Chapter 5.2].

company's position he/she prefers. In Economics, this heterogeneity among consumers is called "horizontal differentiation". At the same level of prices, some consumers prefer one company but other consumers prefer the competitor. For example, at the same price, some consumers prefer Coke over Pepsi. In the classical version of this model, the utility of a consumer $i$ buying the product of company $j, j \in 1, 2$ is

$$U_{i,j} = c^u + q_j - t * |l_j - x_i| - p_j \tag{1}$$

All consumers are identical with the exception that they are uniformly located ($x_i$) between 0 and 1. Consumers face a transportation cost ($t$) for reaching companies which are located at the extremes of the interval ($l_j \in [0, 1]$). The intuition of this parameter is the following: The distance between products and consumers in that interval can be interpreted as a "cost" because consumers have to go from their position (that represents their ideal product) to companies' position (that represents the position of the real product). Therefore, the term that controls the differentiation is $t * |l_j - x_i|$.

To guarantee that all consumers buy at least one platform, the theoretical model assumes that all consumers have an identical (and sufficiently high) reservation value, $c^u$. Lastly, we assume each company sells a product that has an exogenous quality level $q_j$, and they fix a price $p_j$.

#### *Vertical differentiation model*

In this case[11], consumers' preferences are

$$U_{i,j} = \theta_i * q_j - p_j \tag{2}$$

All consumers pay a price $p_j$ when consuming the product $j, j \in 1, 2$, which has a quality $q_j$. Without loss of generality, we assume $q_1 > q_2$. The parameter $\theta$ represents the taste for quality. It is uniformly distributed across the population of consumers between $\underline{\theta} \geq 0$ and $\overline{\theta} = \underline{\theta} + 1$ with density 1.

We make two extra assumptions that are common in literature to guarantee enough differentiated consumers and a covered market respectively:

**Assumption 1** $\overline{\theta} \geq 2\underline{\theta}$

**Assumption 2** $\frac{(\overline{\theta} - 2\underline{\theta})}{3}(q_1 - q_2) \leq \underline{\theta} q_1$

The intuition of the vertical differentiation is the following: at the same level of prices, all consumers prefer one company over the rest. For example, at the same price, all consumers prefer a Ferrari over a Fiat.

#### *Externalities: two-sided markets*

We consider two cases. Both of them are extensions of the previous models. In the first case, we consider the two-sided market proposed by Hagiu and Hałaburda (2014) in which two platforms compete in prices for users and developers. This model can be

[11] See [Belleflamme and Peitz (2015), Chapter 5.3].

considered an extension of the Hotelling's in which there are indirect network externalities between two independent groups of consumers: users and developers. Following the original work, we assume all consumers are rational and buy one and only one platform. In this case, the utility of a user $i$ consuming the platform $j, j \in 1, 2$ is given by[12]

$$U_{i,j} = c_i^u - t * |l_j - x_i| - p_j + \delta n_{-j} \tag{3}$$

Users are uniformly located in the interval [0,1], and they suffer a cost when going from their position to the platforms', like in the Hotelling model. However, instead of valuing exogenous qualities, users (developers) value the number of developers (users) they can meet in the platform ($n_{-j}$). We assume all users (developers) value the presence of the other group in the same way ($\delta$).

The second model is proposed by Gabszewicz and Wauthy (2004), which is another example of two-sided markets. However, it has two interesting features: first, it presents vertical differentiation. This characteristic is unusual among the two-sided market literature, mainly because it generates multiple equilibria. Second, there is no information about the stability of the equilibria, so we have the opportunity to test the algorithm in an environment in which several potential equilibria are possible.

It consists of two platforms that represent exhibition centers that compete for visitors and exhibitors. In this case, visitors' preferences are described by

$$U_{i,j} = \theta x_j^e - p_j \tag{4}$$

And exhibitors' preferences are

$$U_{i',j} = \gamma v_j^e - \pi_j \tag{5}$$

Parameters $\theta$ and $\gamma$ are best understood as a measure of how each visitor (exhibitor) values an additional exhibitor (visitor) in the exhibition centers. The intuition underlying the model is the following. From an exhibitor's point of view,[13] the willingness to rent a stand in the exhibition center depends on his personal value of an additional visitor ($\gamma$), on the number of additional sales this exhibitor may expect (that depends on the number of visitors, $v_j^e$),[14] and on the rental fee, $\pi_j$. At the same price, all exhibitors will prefer the exhibition center with more visitors.

In the original work, the authors assume that $\theta, \gamma, v_j^e, x_j^e \in [0, 1]$, in other words, they assume a normalized market.

#### *Warranties model*

This model is taken from [Belleflamme and Peitz (2015), Chapter 13]. In this model, we suppose that a firm offers a product that breaks down with probability $1 - \lambda$, and consumers are willing to pay for a product that works a quantity $r > 0$. The authors make four assumptions:

[12] The case of developers is symmetric. We omit it for simplicity's sake.

[13] The intuition is symmetrical in the case of visitors.

[14] Symmetrically, $x_j^e$ represents the number of expected exhibitors.

- There is a unit mass of homogeneous consumers
- There are two firms. A high-quality one ($\lambda_1$) and a low-quality one ($\lambda_2$). Therefore, $\lambda_1 > \lambda_2$
- Consumers do not know the quality of each company
- Both companies have constant marginal costs, $c$.

From consumers' point of view, a firm can be the high-quality one with probability $\rho$ and the low-quality one with probability $1 - \rho$. Thus, the expected utility of consumers is

$$U_i = (\rho\lambda_1 + (1 - \rho)\lambda_2)r - p \tag{6}$$

Up to now, the model does not consider warranties. Let's introduce full warranties (the company replaces the broken product by a new one). The expected cost of introducing a warranty is $c/\lambda_1$ for the high-quality one, and $c/\lambda_2$ for the low-quality one. The interesting point about this model is that consumers only observe prices and warranties. Warranties can play a decisive role in shaping prices and market shares because they provide information about which company could be the high-quality one. Companies can use warranties to "prove" users that its product is the high-quality one.[15] To guarantee the stability of the equilibrium, the authors assume $\lambda_2 r < c$.

## Results and simulation tests

We create a world with 314 consumers (in the case of the market with two-sided platforms, there are 628 consumers divided into two groups: users and developers, or visitors and exhibitors) and two companies. We run one thousand simulations for each case in NetLogo. In all those cases, we compare the theoretical and the simulated equilibria. We consider two $\varepsilon$-values, 0.1 and 0.05. For each case, we only consider a set of parameters values, for example, different transportation costs or quality levels, etc. to show how the algorithm is capable of reaching the theoretical equilibria. We can consider other cases with other parameters but, for simplicity's sake and without loss of generality, they are not included (although they are available upon request).[16]

### Horizontal differentiation: the Hotelling framework

We run the simulations considering different transportation costs and with symmetric quality levels ($q_1 = q_2$). As depicted in Fig. 3, depending on what $\varepsilon$-values we assume, we obtain different simulated prices that reproduce the theoretical prices properly.[17]

To test if simulated prices are able to reproduce the theoretical prices accurately, we run a linear regression in which the dependent variable is the theoretical price, and the explanatory variable is the simulated one. In Table 1, we observe that the r-squared is close to 99% in both cases. So, the simulated prices explain 99% of the variations in theoretical prices.

Let's consider the difference between theoretical and simulated prices. We test if those errors are normally distributed. To do so, we consider three normality test, all of them assume the null hypothesis of normality.

[15] The model predicts that only the high-quality company will be willing to introduce the warranties.

[16] The number of users and developers is arbitrarily selected, other numbers can be considered and conclusions will not change.

[17] The code can be downloaded at: https://goo.gl/XSV97N.

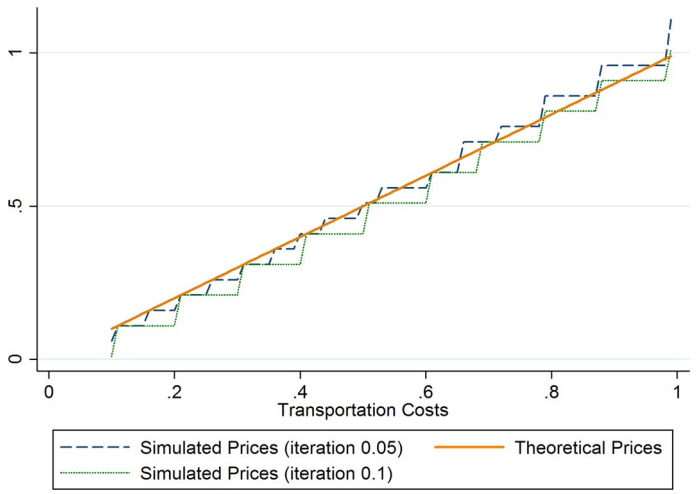


**Fig. 3** Simulated and theoretical prices. Hotelling's model

The first one is the D'Agostino's K-squared test, or the Skewness and Kurtosis test. This test is based on the kurtosis and skewness measures, it can be considered a default normality test. We also consider the Shapiro–Wilk test, which is another traditional normality test. Lastly, we consider the Shapiro–Francia test, which is indicated as the best test to detect deviation from normality in a recent work, Mbah and Paothong (2015).

In Table 2, we observe the p-values associated with those tests. Let's consider first the case in which $\varepsilon = 0.1$. In this case, at 95% confidence level, only in the D'Agostino's K-squared test we can reject the null hypothesis of normality. It is reasonable to think that errors are normally distributed and therefore, the average ($-$ 0.036) is a good representation of the simulations errors. In this case, it is an underestimation of 3.6%. However, given that the algorithm works considering 10% of change in prices ($\varepsilon = 0.1$), this error is negligible.

On the other hand, when $\varepsilon = 0.05$, all the tests suggest that the difference between simulated and theoretical prices is not normal. We run a variance-comparison test, and we find that, at 99% confidence level, the standard deviations between the cases with transportation costs in the interval [0.1;0.59] and in the interval [0.60;0.99] have different variances. Given that each simulation for each parameter is independent of the rest, we can analyze both sub-samples separately. The first one considers transportation costs between 0.1 and 0.59 and the other one between 0.60 and 0.99. In those intervals, at 95% confidence level, we cannot reject the null hypothesis of equal variance.

If we run the normality tests again, we cannot reject the null hypothesis of normality at 95% confidence level, Table 3. Therefore, the average error in simulated prices is an underestimation of 1.3% in the first part ($-$ 0.013), and an overestimation of 2% in the second part (0.02). But given that the algorithm works considering changes of 5% in prices, in this case, those errors are also negligible.

**Table 1 Hotelling model**

| Variable | Coefficient (std. err.) |
|---|---|
| (a) Hotelling, Iteration 0.1 | |
| Simulated price | 1.0459** (0.0073) |
| N | 90 |
| $R^2$ | 0.9957 |
| $F_{(1,89)}$ | 20,558.99 |
| (b) Hotelling, Iteration 0.05 | |
| Simulated price | 0.9821** (0.005) |
| N | 90 |
| $R^2$ | 0.9977 |
| $F_{(1,89)}$ | 38,519.85 |

**Table 2 Hotelling. Normality tests**

| $\epsilon$/p-values | Skewness and kurtosis test | Shapiro–Wilk test | Shapiro–Francia test |
|---|---|---|---|
| *0.10* | 0.0205 | 0.09909 | 0.60196 |
| *0.05* | 0.0000 | 0.0000 | 0.001 |

**Table 3 Hotelling model, Iteration at 0.05**

| Intervals | Skewness and kurtosis test | Shapiro–Wilk test | Shapiro–Francia test |
|---|---|---|---|
| [0.1;0.59] | 0.5595 | 0.23156 | 0.86565 |
| [0.60;0.99] | 0.2468 | 0.67779 | 0.68322 |

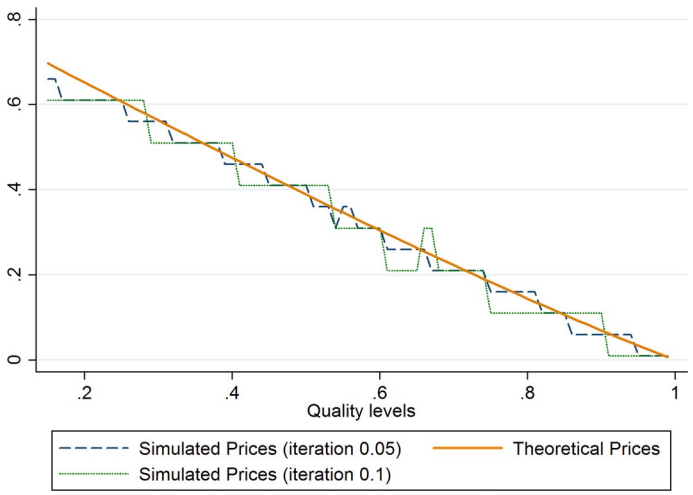


**Fig. 4** Simulated and theoretical prices. Vertical differentiation model

### Vertical differentiation model

In this case, we run the simulations by considering different quality levels for company 2. Nonetheless, we maintain $q_1 = 1$. In Fig. 4, we observe the simulated and the theoretical

equilibrium prices considering different quality levels (company 2). As in the previous case, the simulated prices reproduce the theoretical ones accurately.[18]

We run a linear regression for each $\varepsilon$. The dependent variable is the theoretical price, and the explanatory variable is the simulated price. We find the algorithm reproduces the behavior of the theoretical companies properly.

In Table 4, we observe that the r-squared is close to 99% in all cases. The simulated prices are able to explain 99% of the variations in the theoretical prices. As in the Hotelling model, simulations reproduce the optimizing behavior of companies accurately. Let's consider the differences between the theoretical and simulated prices. As before, we consider three normality tests.

In Table 5, we observe that all the normality tests point out that, at 95% confidence level, we cannot reject the null hypothesis of normality. Therefore, the average of the errors can be interpreted as the average error of the algorithm in this model. In this case, the algorithm yields an average error of $-$ 1.5% ($-$ 0.015) when $\varepsilon = 0.1$ and 1% (0.01) when $\varepsilon = 0.05$. Nonetheless, the algorithm considers changes in prices of 10 and 5% respectively, so these errors are negligible.

In this model, demands are not equal. In contrast with the rest of the models, companies will not equally share the market. Nonetheless, the algorithm is also capable of simulating demands. In Fig. 5, we observe demands are well reproduced at the beginning, but there are divergences at high-quality levels. This effect depends on the $\varepsilon$ we assume, and it is exclusive of vertically differentiated models.

When both qualities are similar, prices tend to be similar too but, because the algorithm works in "discrete jumps ($\varepsilon$)", when both qualities are similar, little increments of quality may not change the simulated prices but may change utilities, which leads to

**Table 4 Vertical differentiation model**

| Variable | Coefficient (std. err.) |
|---|---|
| (a) Iteration 0.1 | |
| Simulated price | 1.0385** (0.0104) |
| N | 85 |
| $R^2$ | 0.9917 |
| $F_{(1,84)}$ | 10066.75 |
| (b) Iteration 0.05 | |
| Simulated price | 1.0314 ** (0.0054) |
| N | 85 |
| $R^2$ | 0.9977 |
| $F_{(1,84)}$ | 36883.17 |

**Table 5 Vertical differentiation. Normality tests**

| $\varepsilon$ | Skewness and kurtosis test | Shapiro–Wilk test | Shapiro–Francia test |
|---|---|---|---|
| 0.1 | 0.3097 | 0.77471 | 0.94308 |
| 0.05 | 0.2197 | 0.22636 | 0.24974 |

[18] The code can be downloaded at: https://goo.gl/WnfPhk.

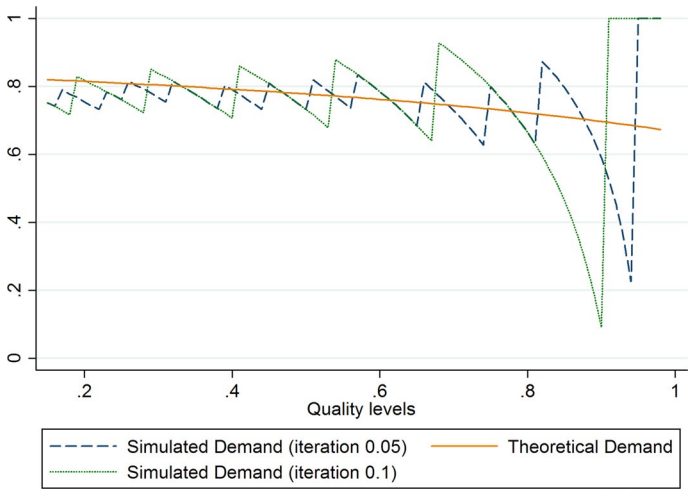


**Fig. 5** Simulated and theoretical demands. Vertical differentiation model

changes in demands. The more similar the platforms are, the more relevant are these changes. For that reason, those unusual behaviors are concentrated in the region where platforms are less differentiated. If we omit that region, we observe that the difference between theoretical and simulated demands (hereinafter, the errors) are normally distributed and, on average, the algorithm predicts without errors the demand level. However, we cannot omit the fact that some errors arise. To prove that those errors come from the $\varepsilon$-value assumed, we also consider the case in which $\varepsilon = 0.01$.

In Table 6, the $\Delta$ represents the difference between companies' quality levels at which the errors are normal. For instance, when $\varepsilon = 0.01$, only if the difference in quality is bigger than 0.05, the errors are normal. In parenthesis, there are the p-values that correspond to those cases in which we remove the observations with a differentiation less than $\Delta$. In Table 7, we observe the case in which $\varepsilon = 0.01$ is the best one in reproducing demands because prices are more sensitive to differences in qualities.

In contrast with the previous case, the lower the $\varepsilon$, the more accurate are the simulated prices and demands in all cases. Figure 6 depicts the comparison between the cases in which $\varepsilon = 0.05$, $\varepsilon = 0.1$ and $\varepsilon = 0.01$.

### Externalities: two-sided markets

#### *Hagiu and Halaburda's model*

This model considers that two platforms compete in prices for consumers (users and developers). Consumers value the presence of the other group so, the more, the better. For simplicity's sake and without loss of generality, we assume both sides are

**Table 6 Vertical differentiation. Normality tests. Demands**

| $\varepsilon$ ($\Delta$) | Skewness and kurtosis test | Shapiro–Wilk test | Shapiro–Francia test |
|---|---|---|---|
| 0.1(0.14) | 0.0037 (0.0674) | 0.00000 (0.13764) | 0.00001 (0.07698) |
| 0.05(0.09) | 0.0009 (0.0396) | 0.00001 (0.09888) | 0.00002 (0.04154) |
| 0.01 (0.05) | 0.000 (0.6848) | 0.00000 (0.69446) | 0.00001 (0.55921) |

**Table 7 Vertical differentiation models. Demands**

| Variable | Coefficient (std. err.) |
|---|---|
| (a) Iteration 0.1 | |
| Simulated demand | 0.9556** (0.0231) |
| N | 85 |
| $R^2$ | 0.9534 |
| $F_{(1,84)}$ | 1717.66 |
| (b) Iteration 0.05 | |
| Simulated demand | 0.9850** (0.0165) |
| N | 85 |
| $R^2$ | 0.9771 |
| $F_{(1,84)}$ | 3585.43 |
| (c) Iteration 0.01 | |
| Simulated demand | 0.9987** (0.0067) |
| N | 85 |
| $R^2$ | 0.9962 |
| $F_{(1,84)}$ | 22024.81 |

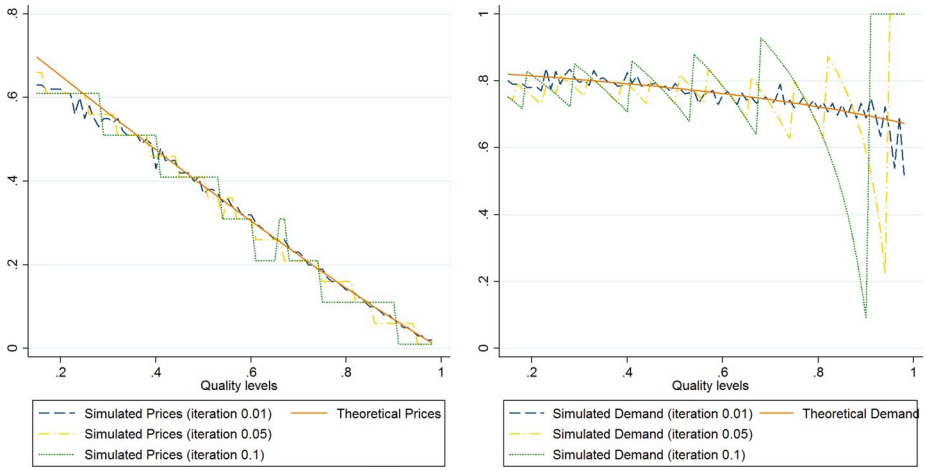


**Fig. 6** Comparison of $\varepsilon$-values. Vertical differentiation model

symmetrical, and we only analyze one of them, in this case, we focus on users. We run simulations considering different transportation costs. In Fig. 7, we observe that the simulated and theoretical prices behave in a similar way. As in the previous cases, the simulated prices reproduce the theoretical ones properly.[19]

If we regress the theoretical prices with the simulated ones, we find that, in all cases, the r-squared is superior to 99%, which shows that simulated prices predict quite well the theoretical prices, Table 8.

In Table 9, we analyze the differences between the simulated and the theoretical prices. We observe that when $\varepsilon = 0.1$ and at 95% confidence level, we cannot reject the null hypothesis of normality. On the other hand, given that the average is approximately

[19] The code can be downloaded at: https://goo.gl/RDwsDD.

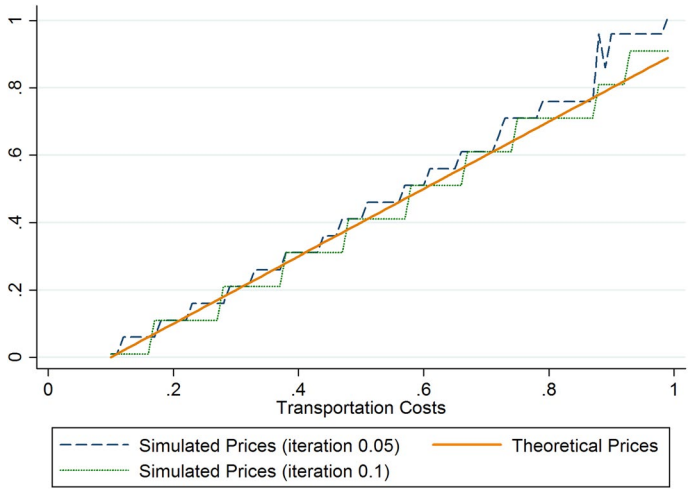


**Fig. 7** Simulated and theoretical prices. Hagiu and Halaburda's model

**Table 8 Hagiu and Halaburda's model**

| Variable | Coefficient (std. err.) |
|---|---|
| (a) Two-sided, Iteration 0.1 | |
| Simulated price | 0.9924** (0.0068) |
| N | 90 |
| $R^2$ | 0.9959 |
| $F_{(1,88)}$ | 21,373.90 |
| (b) Two-sided, Iteration 0.05 | |
| Simulated price | 0.9201** (0.0055) |
| N | 90 |
| $R^2$ | 0.9968 |
| $F_{(1,88)}$ | 27,492.02 |

**Table 9 Hagiu and Halaburda's model. Normality tests**

| $\varepsilon$ | Skewness and kurtosis test | Shapiro–Wilk test | Shapiro–Francia test |
|---|---|---|---|
| 0.1 | 0.2179 | 0.14804 | 0.77307 |
| 0.05 | 0.0000 (0.2568) | 0.0000 (0.3255) | 0.0000 (0.4511) |

zero, we can state that, in this case, the algorithm makes no error. However, when we consider $\varepsilon = 0.05$, we can reject the null hypothesis of normality at 99% confidence level. However, this rejection is a consequence of an unusual behavior at high transportation costs. When transportation costs are higher than 0.9, simulated prices show great differences with theoretical prices as depicted in Fig. 7.

If we omit that interval, in all normality tests, the null hypothesis cannot be rejected at 95% confidence level. The average is approximately equal to 0.03. So, we can state that when $\varepsilon = 0.05$, the algorithm makes an average error of 3% (3.8% if that interval is not omitted). As in previous cases, given that the algorithm works with percentages bigger than the average errors, those errors are negligible.

**Table 10 Gabszewicz and Wauthy's equilibria**

| Equilibria | Prices | Demands | Profits |
|---|---|---|---|
| Duopolistic equilibrium | $p_1 = \pi_1 = 2/49$<br>$p_2 = \pi_2 = 8/49$ | $x_1 = v_1 = 2/7$<br>$x_2 = v_2 = 4/7$ | $\Pi_1 = 0.0233$<br>$\Pi_2 = 0.1866$ |
| Monopoly corner eq. | $p_1 = 0, \pi_1 = 1/2$ | $x_1 = 1, v_1 = 1/2$ | $\Pi = 1/4$ |
| Monopoly equilibrium | $p_1 = \pi_1 = 1/2$ | $x_1 = v_1 = 1/2$ | $\Pi = 1/4$ |

**Table 11 Prices, demands and profits. Simulation of Gabszewicz and Wauthy's model**

| Equilibria | Passive beliefs | Responsive beliefs |
|---|---|---|
| $\epsilon = 0.1$ | $P = x = \Pi = 0$ | $P = 0.21\, x = 0.698\, \Pi = 0.293$ |
| $\epsilon = 0.05$ | $P = x = \Pi = 0$ | $P = 0.21\, x = 0.647\, \Pi = 0.291$ |
| $\epsilon = 0.01$ | $P = x = \Pi = 0$ | $P = 0.24\, x = 0.599\, \Pi = 0.287$ |
| $\epsilon = 0.005$ | $P = x = \Pi = 0$ | $P = 0.22\, x = 0.672\, \Pi = 0.296$ |
| $\epsilon = 0.001$ | $P = 0.215$<br>$x = 0.685$<br>$\Pi = 0.294$ | $P = 0.136\, x = 0.8375\, \Pi = 0.228$ |

***Gabszewicz and Wauthy's model***

In contrast with the previous model, the Gabszewicz and Wauthy's model[20] considers vertical differentiation between the platforms. However, this is not the only difference. The essential difference is that this model presents three different equilibria in the duopolistic framework. These equilibria are also different than the previous ones. They do not depend on parameters values. Each price-equilibrium is defined by constant prices and demands. Table 10 shows the different equilibria found by Gabszewicz and Wauthy. However, some of those equilibria are not stable, and the simulations are affected by this feature.

Let's prove that some of those equilibria are not stable. On the one hand, the duopolistic equilibrium is not stable. If any company fixes a zero price on one side and the monopoly price on the other side, this deviation is profitable. If any company deviates, the other one will abandon the market.[21] If we analyze the other equilibria, only the "Monopoly equilibrium" is stable.[22] This happens because Gabszewicz and Wauthy assume "passive beliefs". That implies that when a platform changes prices on one side, the other side does not change their expectations immediately. So, it is always profitable a deviation from the zero price because, given the passive beliefs, the platforms earn an extra profit from deviating. And once they have moved from that equilibrium and the expectations have changed, they will move to the "Monopoly equilibrium" to keep their profits.

In this case, the theoretical equilibrium is a constant value, so we present in Table 11 the simulation results with different $\epsilon$-values and the theoretical equilibrium values. As we stated in the "Vertical differentiation model" section, the vertical differentiation can

[20] This model can be found at https://goo.gl/S6i3rn.

[21] Technically, they will be indifferent, but we can only observe one reality, in this case, we assume it is the abandon.

[22] This makes sense because of the vertical differentiation. The network effects are so strong that there is an incentive towards concentration in only one platform.

have dramatic effects on the equilibrium because it may lead to big changes. In the case of indirect network effects and vertical differentiation, those changes can be even more dramatic. For example, when $\epsilon = 0.01$ or $\epsilon = 0.005$ the model collapses to the Bertrand equilibrium (which is another possibility considered by the authors). This result is a consequence of the extreme sensitivity of the model to changes, and the existence of several equilibria.

As we move towards small $\epsilon$-values, the monopoly equilibrium becomes the predominant outcome. However, in comparison with other models, it seems that there is a big divergence in demands but not in prices. This characteristic is a consequence of the simulation model. While the theoretical model considers an infinite amount of consumers uniformly distributed, our model only considers 314. That implies that one percent change in the $\gamma$ and $\theta$ parameters represents 1% of population in the theoretical model, but 1.3% of population in the simulated model. This small difference leads to a huge difference between the demands of the theoretical and the simulated model because the vertical differentiation. Again, these results show the relevance of testing different $\epsilon$-values. In comparison with previous models, testing different $\epsilon$-values does not change the equilibrium pointed out by the simulations. But in this case, different equilibria appear when choosing different $\epsilon$-values. All of them are equilibria of the theoretical model. So, testing different $\epsilon$-values helps us to identify global equilibria as it was argued before.

#### Warranties model

In this case, the theoretical model predicts two different equilibria when there are warranties and when there is no warranty. If there is no warranty, both companies fix the same price ($p = (\rho\lambda_1 + (1-\rho)\lambda_2)r$), but if warranties are used, consumers can identify the quality of each company and prices will be different ($p_1 \leq r$, $p_2 = \lambda_2 r$). However, in the last case, only the high-quality company remains in the market. For simplicity's sake and without loss of generality, we analyze the former case.[23]

In Fig. 8, we depict the theoretical prices and the average simulated ones with different values of $\lambda_1$. In Fig. 9, we observe the theoretical price and two simulated prices during the first 50 iterations. In this model, prices are not always stable at a specific point, but they oscillate around that point. This oscillating pattern is due to the discrete nature of the algorithm, that is applied in a continuous environment that is quite sensitive to small changes.

To test if simulated prices are able to reproduce the theoretical prices accurately, we run a linear regression in which the dependent variable is the theoretical price, and the explanatory variable is the simulated one. In Table 1, we observe that, in both cases, the r-squared is around 99%, so our simulated prices are able to explain 99% of the variations in theoretical prices. Nonetheless, in this case, we have seven observations only, so we cannot test the normality of these cases properly.[24] However, Figs. 8 and 9 leave little room for doubt that the algorithm is capable of simulating the theoretical optimal prices.

[23] Nonetheless, the code and the software to simulate both cases are available at: https://goo.gl/zFvY7t.

[24] We have only seven observation because we are working with the averages of simulated prices. The only test that we can properly run is the Wilk test and, in both cases, at 95% confidence level, we cannot reject the null hypothesis of normality.

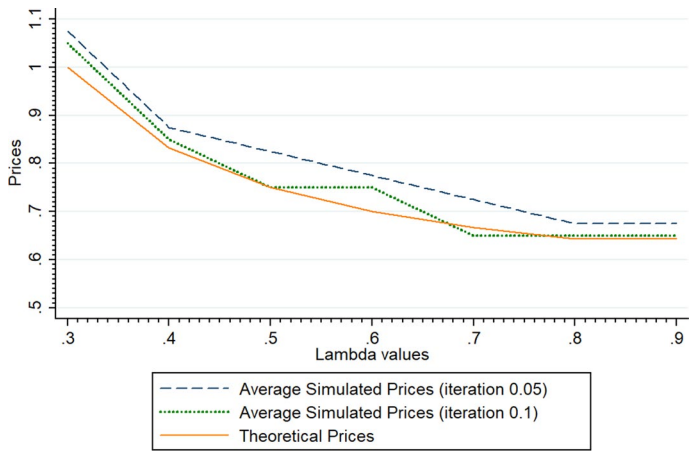


**Fig. 8** Average simulated and theoretical prices

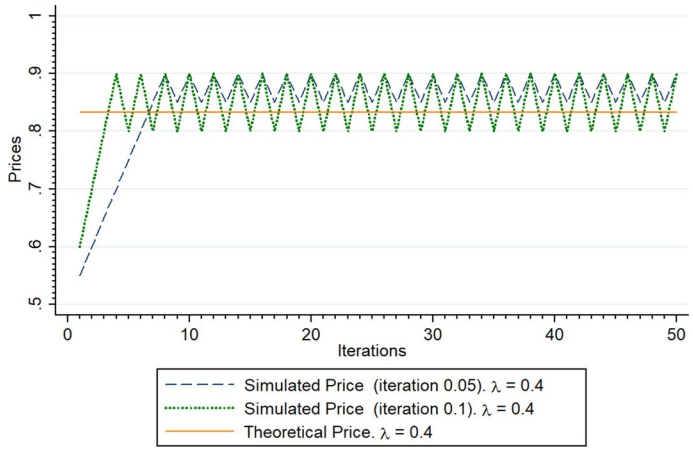


**Fig. 9** Simulated and theoretical prices

However, the algorithm presents two limitations in this model. First, sometimes the algorithm is not capable of reaching the equilibrium and predicts a situation in which all companies abandon the market. This happens because of the discrete nature of the algorithm, small changes in the parameters correct this situation, e.g. instead of using $\lambda_1 = 0.89$, use $\lambda_1 = 0.9$. Second, extreme cases such as $\rho = 0$ or $\rho = 1$ will not work. This is a limitation of the current model because it is not programmed to consider the cases in which the model converges to others with full information.

#### Relaxation of assumptions and extensions. New directions in agent-based modeling

Up to now, we have considered the theoretical models under their original assumptions. However, we can use the algorithm to address cases in which some assumptions are relaxed or even cases that are out of the reach of theoretical models. For instance, we can consider the market is not covered, utility functions are not linear, users are distributed following a normal, an exponential, or any other distribution, etc. In all of those cases, the algorithm works properly. We can even include new layers of complexity by

**Table 12 Warranties model**

| Variable | Coefficient (std. err.) |
|---|---|
| (a) Warranties, Iteration 0.1 | |
| Simulated price | 0.9756** (0.0111) |
| N | 7 |
| $R^2$ | 0.9992 |
| $F_{(1,6)}$ | 7621.27 |
| (b) Warranties, Iteration 0.05 | |
| Simulated price | 0.9304** (0.008) |
| N | 7 |
| $R^2$ | 0.9996 |
| $F_{(1,6)}$ | 14,831.19 |

assuming more than two companies, or by assuming that users are heterogeneous in several dimensions. In this section, we consider two different cases in which we relax such assumptions.

First, we simulate the Hagiu and Halaburda's model presented in "Hagiu and Halaburda's model" section, but this time we test the model outside of the parameter region defined by the authors. What is relevant about that region is that the theoretical model predicts negative profits, which is not realistic because platforms will prefer to fix zero prices in both sides than to lose money.

Second, we simulate the Hoteling's model presented in "Horizontal differentiation: the Hotelling framework" section, but this time we assume that not all the consumers enter the market at the same time. In fact, we assume that there is diffusion of information process that determines who enters the market and when.

Lastly, we briefly show how this algorithm can address other theoretical models in the industrial organization literature that are based on quantity competition or consider a two-stage competition (dynamic frameworks). I consider those models as extensions of the price competition algorithm, but not as extensions of the other theoretical frameworks.

#### *Relaxed assumptions: Hagiu and Halaburda's model*

One of the assumptions of this model to guarantee that *"all optimization problems with competing platforms are well-behaved"* is $t^2 > \delta^2$, Eq. 3. If we assume this condition does not hold, the theoretical model predicts negative profits. Obviously, this is not optimal because a better option is to fix zero prices or to abandon the market, which guarantees zero profits.

If we consider that such assumption does not hold and we run the algorithm, platforms always find that prices near zero are an equilibrium.[25] In Table 13, we consider two different starting prices for the algorithm ($-$ 0.05 and 0.25) and two different scenarios.[26] We find that simulated profits make more sense than their counterparts, the theoretical ones. The algorithm is capable of finding the best outcome in those cases in

[25] Prices equal to zero or less than the $\epsilon$ assumed.

[26] In other scenarios with other starting prices, we have the same qualitative result.

**Table 13 Comparison between theoretical and simulated models when conditions regarding the "well behavior" of equilibria are relaxed**

| Variables/cases | Starting price | Deltas | Transportation costs | Equilibrium prices users (developers) | Equilibrium profits |
|---|---|---|---|---|---|
| Simulated case | 0.25 | 0.65 | 0.15 | − 0.05 (0.15) | 0.05 |
| | − 0.05 | 0.65 | 0.15 | 0.05 | 0.05 |
| | − 0.05 | 0.8 | 0.4 | 0.05 (− 0.05) | 0 |
| Theoretical case | | 0.65 | 0.15 | − 0.5 | − 0.5 |
| | | 0.8 | 0.4 | − 0.4 | − 0.4 |

which the theoretical model provides unsatisfactory results. This result implies the algorithm can be used as a verification tool for theoretical models but also, it can be used to deal with cases that are theoretically complex or non-tractable.

#### *Relaxed assumptions: Hotelling's model*

This case represents a Hotelling model in which consumers are entering the market in a non-random way. We assume that only a small percentage of consumers can enter the market in the first iteration (all of them are less than one node away from one another). Then, those consumers can transmit the information about the product they consume to others neighbors in their networks in the following iterations.[27] For simplicity's sake, we assume that users are linked in a random network. We assume there are two companies, and consumers can be in one of these states: uninformed, totally informed or partly informed about one company. In other words, this case mixes the spread of information in a network with two price-competing companies trying to attract consumers.[28]

On the left side of Fig. 10, simulations show that prices are more volatile. This behavior is normal because new users are entering the market at each moment. Companies try to adjust prices to the expansion of the market, but companies are also competing, so they try to reduce prices. Prices only reach a stable position in the case with low differentiation because the competition is stronger than the expansion effect (new users).

On the right side of Fig. 10, simulations show the case with low differentiation is the only one that reaches a point where adoption is not growing anymore (the expansion effect is over, almost all the potential consumers are attracted). However, the other two cases are so volatile because the spread of information is less regular and slower than before.

We can observe too that high prices are a barrier that blocks other users from becoming consumers, which limits the adoption. When prices are high, some users do not consume the products although they know them. This limit the spread of information which leads to stopping the diffusion. On the other hand, users only know about the individual products, so there are cases in which users are only aware of one product and, because information initially spreads from clusters, there will be clusters that will never know

[27] Consumers only spread the information if they consume the product. In some cases, they may have the information but they do not spread it to their neighbors because they do not consume the product because their utility functions are negative at those prices. But, if they consume, they can "infect" their neighbors with the information about the product they consume (14% chance).

[28] We acknowledge that the whole process is not explained in a conscientious way. However, this case is further analyzed in Sanchez-Cartas and Leon (2017). We present this case only for illustrative purposes.

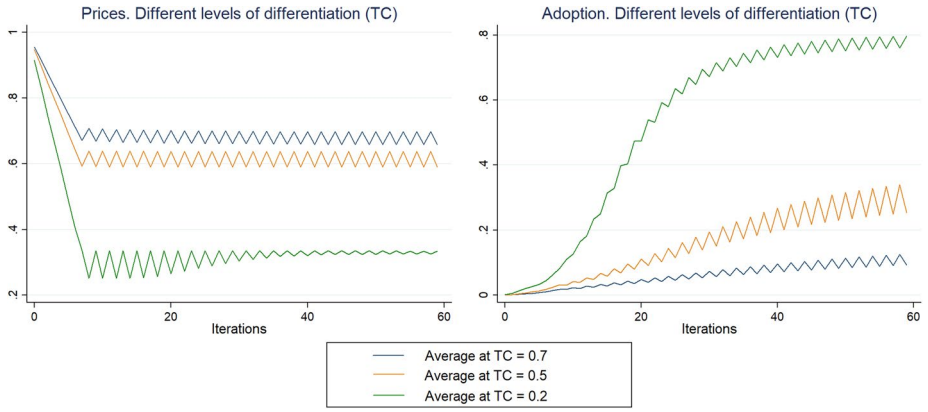


**Fig. 10** Users' prices. Information about products

about the existence of other companies. In this situation, there is volatility in prices and adoption because there is a trade-off between rising prices (because of the differentiation levels and larger profits) and reducing prices (because of the competition between platforms and the boosting effect that it has on the demands).

***Extension 1: the perfect competition framework. Competition in quantities***

Although we assume the algorithm can be applied to simulate price competition, it can also simulate quantity competition. To prove this point, we test a small modification of the algorithm. This is the simplest model of quantity competition. It assumes there is a large number of companies and consumers, all of them are so small with respect to the market that no one can influence the market. So, every participant is a price taker. In that sense, consumers have to choose how many products they want to buy (given a fixed level of rent), and companies have to choose how many products they will produce at given prices. For simplicity's sake, we consider only two companies (but we can consider any other number).

In this case, we consider the utility functions defined by [Belleflamme and Peitz (2015), Page 65]. In particular, the utility function takes the form

$$U(q_1, q_2) = 1 + aq_1 + aq_2 - (bq_1^2 + 2dq_1q_2 + bq_2^2)/2$$

We consider companies can produce natural quantities of each product (1,2,3,..), and they sell their products at a fixed price.[29] Depending on the values of $b$ and $d$, we can address several cases considering complements or substitutes. In Table 14, we compare different cases. In all of them, the algorithm points out that the consumers' decisions about their quantities are the optimal ones.[30]

For example, when $d = 0$ and $b = 1$, it is optimal for consumers to buy products from both companies. So, both companies will produce enough products for all the

[29] $q_1$ and $q_2$ refer to the quantity produced by company 1 and 2 respectively.

[30] Simply computation with the parameter values and the utility function prove this point. The model can be downloaded at https://goo.gl/bjDP5E.

**Table 14 Consumers choices. Perfect Competition**

| Parameter values at a = 1 | Quantities consumed | Parameter values at a = 1 | Quantities consumed |
|---|---|---|---|
| d = 0; b = 1 | $q_1 = q_2 = 1$ | d = 0.7; b = 1 | $q_1 = 1$ or $q_1 = 0$<br>$q_2 = 0$ or $q_2 = 1$ |
| d = b | $q_1 = 1$ or $q_1 = 0$<br>$q_2 = 0$ or $q_2 = 1$ | d = 0.7; b = 0 | $q_1 = q_2 = 1$ |
| d = − 1; b = 0 | $q_1 = q_2 = 1$ | d = 0.1; b = 1 | $q_1 = q_2 = 1$ |
| d = − 1; b = 1 | $q_1 = q_2 = 1$ | d = 0.2; b = 1 | $q_1 = q_2 = 1$ |

consumers. But when $d = b$, consumers only demand one product, and only one company will produce it.

***Extension 2: Milgrom–Roberts Models of barriers to entry***

The algorithm is flexible, and it can deal with dynamic frameworks. To test this statement, we chose the Milgrom–Roberts model.[31] The model assumes two stages: At the first stage, there is a monopoly that produces a good with a marginal cost that may be high $c_h$ or low $c_l$. The information about the cost is not available outside of the monopoly. At the second stage, an entrant has the opportunity to enter the market. However, his/her decision of entering depends on the marginal cost of the incumbent and on its own marginal costs $c_e$. If the cost of the incumbent is high, the entrant will have positive profits; if the cost is low, the entrant will have negative profits.

The original model considers two different equilibria: the separating one and the pooling one. For simplicity's sake and without loss of generality, we focus on the pooling one.[32] This equilibrium predicts that the incumbent will block the market forever. Independently of the high or low marginal costs. This happens if the Eq. 7 holds. This equation states that the duopoly profits of the entrant when the incumbent has high costs must be larger than zero. But they must be negative with low costs, [Tirole and Matutes (1990), Condition 9.8]

$$c_l < c_e < c_h \tag{7}$$

If this equation does not hold, for example, if the entrant has lower marginal costs than the incumbent with low marginal cost. The entrant will always enter the market, and two outcomes are possible:

- Competition: if both have low costs and profits are non-negative.
- Expulsion: the incumbent will be expelled from the market

In Table 15, we observe the outcome of the simulated model. The idea is to have an intuition of how works the algorithm in this model. We consider two cases. One in which the incumbent has high marginal costs. And another one in which it has low marginal costs. In all those cases, the result predicted by the agent-based model is the same than the one predicted by theory.

[31] This model is an adaptation of Milgrom and Roberts (1982) that can be found in [Tirole and Matutes (1990), Chapter 9]. Code at https://goo.gl/djiCWh.

[32] The Milgrom–Roberts model presents two different equilibria that are based on different assumptions. In contrast, the Gabszewicz and Wauthy's model has multiple equilibria under the same assumptions.

**Table 15 Milgrom–Roberts model**

| Parameters values when $c_{incumbent} = c_h$ | Market outcome | Parameters values when $c_{incumbent} = c_l$ | Market outcome |
|---|---|---|---|
| $c_l = 0.1; c_h = 0.84$ $c_e = 0.1$ | Incumbent is expelled | $c_l = 0.1; c_h = 0.84$ $c_e = 0.1$ | Duopoly competition |
| $c_l = 0.1; c_h = 0.84$ $c_e = 0.2$ | Incumbent is expelled | $c_l = 0.3; c_h = 0.84$ $c_e = 0.1$ | Incumbent is expelled |
| $c_l = 0.1; c_h = 0.84$ $c_e > 0.2$ | Block to entry | $c_l = 0.1; c_h = 0.84$ $c_e > 0.2$ | Block to entry |

## Discussion. Similarities and dissimilarities between the algorithm and other approaches

The proposed algorithm resembles the optimization programs that are used in Economic Theory or Game Theory. In these cases, it is common to compute an optimization program from which the first and second order conditions are derived to identify the reaction functions and the equilibria. The algorithm is closely related to the idea of reaction functions. However, there are relevant similarities and differences. On the one hand, the algorithm acts in a similar way than reaction functions; the output of both is the best reply to a specific situation, and the continuous interaction of reaction functions or agents with the algorithm leads to the Nash equilibria. If it does not exist, the algorithm can find corner solutions or second best equilibria.[33]

On the other hand, there are three relevant differences in comparison with the classical optimization programs used in Economics:

- First, the algorithm does not provide a stylized expression that represents the equilibria. It computes a numerical one.
- Second, tractability is not an issue. Traditionally, researchers assume some assumptions like linear demands, constant values, continuity of functions, concave profits functions, etc. that may not be realistic. This is done to guarantee tractability and well-behaved equilibria. However, that is not necessary with the algorithm. If there is no equilibrium, the algorithm chooses a second-best solution (one which produces the most favorable outcome for the agent, taking other agents' strategies as given). In some cases, several second-best solutions can be found, in such cases, the algorithm may pivot from one to another. But, if the equilibrium is global, the algorithm will prove that by showing always the same outcome.
- Third, we do not assume continuity or differentiability. The algorithm is discrete, and it approximates a continuous environment.

There are other features that make the algorithm interesting. For example, the algorithm puts emphasis in the process of reaching the equilibrium, and not in the equilibrium itself. This allows us to study the impact of shocks not only in the equilibrium but also, in the process of reaching it. Nonetheless, the most important contribution is the

[33] For example, in a Cournot model, the equilibrium can be reached using the reaction functions only. Those functions also reproduce an interactive behavior among companies that leads to the equilibrium. The algorithm works in a similar way.

possibility of relaxing assumptions and guarantying the optimality of results at the same time. In this way, we can introduce new dimensions that were quite complicated in theoretical models, such as introducing a network structure among the consumers.

Nonetheless, some researchers may argue that the algorithm is not different than solving the model numerically. However, this is not true. To solve a model numerically requires solving the theoretical model analytically in advance in most of the cases,[34] but with this algorithm, that is not necessary. The algorithm does not assume that agents make complicated mathematical manipulations. It consists of simple actions (consumers make their best decision, the one that maximizes their utility: buy one of the product or not; and companies increase, decrease or maintain prices, what it is more profitable at each time), and we prove that, with those simple actions, the theoretical equilibria predicted by many theoretical models can be reached.

I am not the first one in considering theoretical models for agent-based simulations. But this is the first time that an algorithm reproduces the optimizing behavior of agents like in theoretical economic models. So, it is possible to build agent-based models that are closely linked to traditional economics by using economic intuitions. I do not pretend to present the agent-based modeling as an alternative to the traditional industrial organization literature. Instead, this work presents the agent-based modeling as a complement of that literature. If we can agree that the rules presented in "The algorithm" section represent the same rules that we take as given (or we assume) in the consumer's and company's decision problems. Then, we can agree that the algorithm is a representation of the process of maximization of utility (profits) of users (and companies).

Nonetheless, the algorithm I propose is also criticizable. It is a tool designed to look for local equilibria. So, to address global equilibria, it requires different starting points, different number of iterations or even different discrete "jumps ($\varepsilon$)" in prices. If not, we are at risk of identifying local optima that are not the global ones.

## Conclusions

We develop an algorithm for agent-based models that simulates the behavior of price-competing companies. This algorithm considers two sub-algorithms: one for consumers, and another one for companies. The consumers' algorithm specifies that all users will choose that product that provides them with the highest utility. The companies' algorithm specifies that each company will change its prices by a small quantity if they believe that such change is profitable.

We test the price-competition algorithm in several theoretical frameworks such as the Hotelling model, a vertical differentiation model or a two-sided market. In all of them, we prove the algorithm is capable of reaching the equilibria predicted by theory. We also prove that the algorithm works in cases that were not considered by theory, or in parameter regions where the theoretical model was not suitable to be analyzed. These results open the door to implementing endogenous pricing in agent-based models but also, to

[34] We can argue that some models cannot be solved analytically, and it is required solving them numerically. But that is not the point here. We argue that even such models require a previous theoretical work, in which optimization programs are defined, and some expressions are derived.

test theoretical models in a new environment or even to teach theoretical models in a new way.

We conclude with a discussion of the current limitations of this research, and how this line of research is related to other approaches used in the Economic literature.

## Additional files

**Algorithm pseudo-code**

In this appendix, we provide the pseudo-code of the whole algorithm. It is divided into several procedures.

- The Sub-Algorithm 1 computes the situation of the market at any moment
- The Sub-Algorithm 2 provides a high-level vision of how the price competition is simulated. This sub-algorithm calls to procedure calls "DemandPrediction"
- The Sub-Sub-Algorithm 3 computes how each company considers that the change in prices will affect the market. During this procedure, three more procedures are called: an estimation of impact in utilities, an estimation of impact in demands, and the decision of platforms.
- The Sub-Sub-Algorithm 4 and the Sub-Sub-Algorithm 5 control the estimation of new utilities.
- The Sub-Algorithm 6 and the Sub-Sub-Algorithm 7 control the estimation of new demands.
- The Sub-Sub-Algorithm 8 controls indirect effects that some companies may cause on other platforms.
- The Sub-Sub-Algorithm 9 controls the final decision of companies with regard to prices.

**Algorithm 1** Compute Static Situation of Market

**procedure** UTILITIES
  **for** Each $Sides\ j$ **do**
    Users: Create $UtilityVector \longleftarrow \emptyset$
    **for** Users: Each Platform $i$ **do**
      Compute $\rightarrow$ users' utility in $i$ minus $Prices$
      $UtilityVector$[$i$] $\longleftarrow$ $i$-utility
    **end for**
    Users: $OptimalDecision$ = max($UtilityVector$)
    **if** Users: $OptimalDecision < 0$ **then**
      $OptimalDecision = \emptyset$
    **end if**
    Users w/ $OptimalDecision \neq \emptyset$ : $ChosenCompany \longleftarrow UtilityVector$.index($OptimalDecision$)
  **end for**
**end procedure**

**procedure** DEMANDS
  **for** Each $Sides\ j$ **do**
    Create $DemandVector \longleftarrow \emptyset$
    **for** Each Platform $i$ **do**
      Demand-$i$ = Count users w/ [$ChosenCompany = i$]
      $DemandVector$[$i$]$\longleftarrow$ Demand-$i$
    **end for**
  **end for**
**end procedure**

**procedure** PROFITS
  **for** Each Platform $i$ **do**
    **for** Each $Sides\ j$ **do**
      Platforms: $Profits \longleftarrow Profits$[$j$-1] + prices-$i$ * $DemandVector$[$i$]
    **end for**
  **end for**
**end procedure**

**Algorithm 2** Price Competition. High level vision

**procedure** PRELIMINARY ACTIONS
  **for** Each $Sides\ j$ **do**
    Create $\epsilon$ = change in prices
    Platforms: Create $PricesUp \longleftarrow Prices + \epsilon$
    Platforms: Create $PricesDown \longleftarrow Prices$ - $\epsilon$
  **end for**
**end procedure**
**procedure** PREDICTION
  $Actions \longleftarrow (PricesUp, PricesDown)$
  **repeat**
    DemandPrediction
  **until** number repeats = number platforms
**end procedure**

**Algorithm 3** Price Competition. Companies address the consequences of their changes in prices

```
1: procedure DemandPrediction
2:     for ( doEach Sides j)
3:         for Each Actions x do
4:             ConsiderPlatf ⟵ 0
5:             change ⟵ DemandVector[0]
6:             iteration ⟵ 0
7:             Users: Create PotentialUtilityVector ⟵ ∅
8:             Users: Create will ⟵ [none]
9:             Platforms: Create PotentialDemandVector ⟵ ∅
10:            for Users: Each Platform i do
11:                Compute → users' utility in i minus (Actions[x])
12:                PotentialUtilityVector[i] ⟵ i-utility
13:            end for
14:            while change ⩾ 0.01 do
15:                ControlChange ⟵ Profits of Platform[ConsiderPlatf]
16:                Users: Create ComparionUtilityVector ⟵ ∅
17:                procedure Evaluation of new utilities
18:                end procedure
19:                procedure Evaluation of new utilities. Feedback Loops Control
20:                end procedure
21:                procedure Generation of potential demands
22:                end procedure
23:                if Indirect Network effects exists then
24:                    procedure Feedback Loops
25:                    end procedure
26:                end if
27:                Platform[ConsiderPlatf]: PotentialProfits = (Actions[x]) *
   PotentialDemandVector[i] + prices-i(Group2) * PotentialDemandVector(Group2)[i]
28:                procedure Impact feedback loops on other platforms
29:                end procedure
30:                iteration = iteration + 1
31:                change = abs(ControlChange - PotentialProfits of platform[i])
32:            end while
33:        end for
34:    end for
35:    procedure Decision Making.Platform[ConsiderPlatf]
36:    end procedure
37:    ConsiderPlatf ⟵ ConsiderPlatf + 1
38:    if ConsiderPlatf = N then
39:        ConsiderPlatf ⟵ 0
40:    end if
41: end procedure
```

**Algorithm 4** Price Competition. Sub-Procedures(1/6). Utilities estimation

```
1: procedure Evaluation of new utilities
2:     if iteration ⟵ 0 then
3:         if ConsiderPlatf ⟵ 0 then
4:             Users: ComparionUtilityVector[0] ⟵ PotentialUtilityVector[0]
5:             Users: ComparionUtilityVector[1:N] ⟵ UtilityVector[1:N]
6:         end if
7:         if ConsiderPlatf ≠ 0 and ConsiderPlatf ≤ N-1 then
8:             Users: ComparionUtilityVector[0:i-1] ⟵ UtilityVector[0:i-1]
9:             Users: ComparionUtilityVector[i] ⟵ PotentialUtilityVector[i]
10:            Users: ComparionUtilityVector[i+1:N] ⟵ UtilityVector[i+1:N]
11:        end if
12:        if ConsiderPlatf = N then
13:            Users: ComparionUtilityVector[0:N-1] ⟵ UtilityVector[0:N-1]
14:            Users: ComparionUtilityVector[N] ⟵ PotentialUtilityVector[N]
15:        end if
16:    end if
17: end procedure
```

**Algorithm 5** Price Competition. Sub-Procedures(2/6). New utilities if externalities are present

**procedure** EVALUATION OF NEW UTILITIES. FEEDBACK LOOPS CONTROL
  **if** $iteration \neq 0$ **then**
    **if** $ConsiderPlatf = 0$ **then**
      Users: $ComparionUtilityVector[0] \longleftarrow PotentialUtilityVector[0]$
      Users: $ComparionUtilityVector[1{:}N] = IteratedUtilityVector[1{:}N]$
    **end if**
    **if** $ConsiderPlatf \neq 0$ **and** $ConsiderPlatf \leq$ N-1 **then**
      Users: $ComparionUtilityVector[0{:}i\text{-}1] \longleftarrow IteratedUtilityVector[0{:}i\text{-}1]$
      Users: $ComparionUtilityVector[i] \longleftarrow PotentialUtilityVector[i]$
      Users: $ComparionUtilityVector[i{+}1{:}N] \longleftarrow IteratedUtilityVector[i{+}1{:}N]$
    **end if**
    **if** $ConsiderPlatf =$ N **then**
      Users: $ComparionUtilityVector[0{:}N\text{-}1] \longleftarrow IteratedUtilityVector[0{:}N\text{-}1]$
      Users: $ComparionUtilityVector[N] \longleftarrow PotentialUtilityVector[N]$
    **end if**
  **end if**
**end procedure**

**Algorithm 6** Price Competition. Sub-Procedures(3/6). Estimation of potential demands

**procedure** GENERATION OF POTENTIAL DEMANDS
  Users: $PotentialOptimalDecision =$ Max$(ComparionUtilityVector)$.
  **if** Users: $PotentialOptimalDecision < 0$ **then**
    $PotentialOptimalDecision \longleftarrow \emptyset$
  **end if**
  Users w/ $PotentialOptimalDecision \neq \emptyset$: $PotentialChosenCompany \longleftarrow ComparionUtilityVector$.index$(PotentialOptimalDecision)$
  **for** Each Platform $i$ **do**
    PotentialDemand-$i$ = Count users w/ $PotentialOptimalDecision = i$
    $PotentialDemandVector[i] \longleftarrow$ PotentialDemand-$i$
  **end for**
**end procedure**

**Algorithm 7** Price Competition. Sub-Procedures(4/6). Estimation of potential demands if externalities are present

**procedure** FEEDBACKS LOOPS
  Users(Group2): Create $PotentialUtilityVector(Group2) \longleftarrow \emptyset$
  Users(Group2): Create $will(Group2) \longleftarrow$ [none]
  **for** Users(Group2): Each Platform $i$ **do**
    Compute $\rightarrow$ users' utility w/ $NetworkEffects$=Demand-$i$
    $PotentialUtilityVector(Group2)[i] \longleftarrow i$-utility
  **end for**
  Users(Group2) $PotentialOptimalDecision =$ max$(PotentialUtilityVector(Group2))$
  **if** Users: $PotentialOptimalDecision < 0$ **then**
    $PotentialOptimalDecision \longleftarrow \emptyset$
  **end if**
  Users: $PotentialChosenCompany \longleftarrow PotentialUtilityVector(Group2)$.index$(PotentialOptimalDecision)$
  **for** Each Platform $i$ **do**
    PotentialDemand-$i$(Group2) = Count Users(Group2) w/ $[PotentialOptimalDecision = i]$
    $PotentialDemandVector(Group2)[i] \longleftarrow$ Demand-$i$(Group2)
  **end for**
**end procedure**

**Algorithm 8** Price Competition. Sub-Procedures(5/6). Control for indirect effects on other companies

**procedure** IMPACT FEEDBACK LOOPS ON OTHER COMPANIES
  Users: Create $PotentialUtilityVector \longleftarrow \emptyset$
  Users: Create $IteratedUtilityVector \longleftarrow \emptyset$
  Users: Create $will \longleftarrow$ [none]
  Platforms: Create $PotentialDemandVector \longleftarrow \emptyset$
  **for** Users: Each Platform $i$ **do**
    Compute $\rightarrow$ users' utility in $i$ minus ($PricesUp$ **or** $PricesDown$) w/ $NetworkEffects$=Demand-$i$(Group2)
    $PotentialUtilityVector[i] \longleftarrow i$-utility
    Compute $\rightarrow$ users' PotentialUtility in $i$ minus $Prices$
    $IteratedUtilityVector[i]$ $i$-PotentialUtility
  **end for**
**end procedure**

**Algorithm 9** Price Competition. Sub-Procedures(6/6). Companies decision

**procedure** DECISION MAKING. PLATFORM[$ConsiderPlatf$]
  **if** $PotentialProfits$ at ($Actions$[x]) $>$ $Profits$ & $PotentialProfits$ at ($Actions$[x]) $>$ $PotentialProfits$ at ($Actions$[-x]) **then**
    Set $Prices \longleftarrow Actions$[x]
  **end if**
**end procedure**

**Authors' contributions**
JMS designed the algorithm, tested it, and drafted the manuscript. The author read and approved the final manuscript.

**Competing interests**
The author declares that he has no competing interests.

**Acknowledgements**
I appreciate the helpful comments and advice on this research supplied by the participants at the 23rd International Conference in Computing in Economics and Finance and at the 15th International Conference on Practical Applications of Agents and Multi-Agent Systems. I also want to stress my appreciation for the helpful comments of two anonymous reviewers that have significantly improved the final paper.

**Consent for publication**
Not applicable.

**Ethics approval and consent to participate**
Not applicable.

**Funding**
Not applicable.

## Publisher's Note



Received: 5 December 2017 Accepted: 23 February 2018
Published online: 12 March 2018

**References**

Barr J, Saraceno F (2005) Cournot competition, organization and learning. J Econ Dyn Control 29(1):277–295
Belleflamme P, Peitz M (2015) Industrial Organization: Markets and Strategies. Cambridge University Press, University Printing House, Cambridge
Chang M-H (2011) Entry, exit, and the endogenous market structure in technologically turbulent industries. East Econ J 37(1):51–84
Diao J, Zhu K, Gao Y (2011) Agent-based simulation of durables dynamic pricing. Syst Eng Proc 2:205–212

Fuks K, Kawa A (2009) Simulation of resource acquisition by e-sourcing clusters using netlogo environment. In: KES international symposium on agent and multi-agent systems: technologies and applications. Springer, Berlin. pp 687– 696
Gabszewicz JJ, Wauthy X (2004) Two-sided markets and price competition with multi-homing. CORE Discussion Paper no 2004/30
Hagiu A, Hałaburda H (2014) Information and two-sided platform profits. Int J Ind Organ 34:25–35
Hamill L, Gilbert N (2016) Agent-based modelling in economics. Wiley, Chichester
Mbah AK, Paothong A (2015) Shapiro–Francia test compared to other normality test using expected p-value. J Stat Comput Simul 85(15):3002–3016
Milgrom P, Roberts J (1982) Limit pricing and entry under incomplete information: an equilibrium analysis. Econometrica J Econ Soc 50(2):443–459
Rand W, Rust RT (2011) Agent-based modeling in marketing: guidelines for rigor. Int J Res Mark 28(3):181–193
Rixen M, Weigand J (2014) Agent-based simulation of policy induced diffusion of smart meters. Technol Forecast Soc Change 85:153–167
Sanchez-Cartas JM, Leon G (2017) On simulating the adoption of new products in markets with rational users and companies. In: International conference on practical applications of agents and multi-agent systems. Springer, Berlin. pp 83– 94
Tirole J, Matutes C (1990) La Teoría de la Organización Industrial, 1st edn. Ariel, Barcelona
van Leeuwen E, Lijesen M (2016) Agents playing hotelling's game: an agent-based approach to a game theoretic model. Ann Reg Sci 57(2–3):393–411
Zhang T, Brorsen BW (2011) Oligopoly firms with quantity-price strategic decisions. J Econ Inter Coord 6(2):157–170