Universitat Oberta de Catalunya

uoc.edu

Citation for published version

Van de Geer, R., Den Boer, Arnoud V. , Bayliss, C., Currie, C., Ellina, A., Esders, M., Haensel, A., Xiao, L., Maclean, K. D. S., Martínez Sykora, A., Riseth, A. N., Odegaard, F. & Zachariades, S. (2018). Dynamic pricing and learning with competition: insights from the dynamic pricing challenge at the 2017 INFORMS RM & pricing conference. Journal of Revenue and Pricing Management, (), 1-19.

DOI

https://doi.org/10.1057/s41272-018-00164-4

Document Version

This is the Submitted Manuscript version.
The version in the Universitat Oberta de Catalunya institutional repository, O2 may differ from the final published version.

Copyright and Reuse

This manuscript version is made available under the terms of the Creative Commons Attribution Non Commercial No Derivatives licence (CC-BY-NC-ND) http://creativecommons.org/licenses/by-nc-nd/3.0/es/, which permits others to download it and share it with others as long as they credit you, but they can't change it in any way or use them commercially.

Enquiries

If you believe this document infringes copyright, please contact the Research Team at: repositori@uoc.edu

Universitat Oberta de Catalunya

Research archive

Dynamic Pricing and Learning with Competition: Insights from the Dynamic Pricing Challenge at the 2017 INFORMS RM & Pricing Conference[1,2]

April 11, 2018

Ruben van de Geer[a,I], Arnoud V. den Boer[b,I], Christopher Bayliss[c,II], Christine Currie[c,II], Andria Ellina[c,II], Malte Esders[d,II], Alwin Haensel[e,II], Xiao Lei[f,II], Kyle D.S. Maclean[g,II], Antonio Martinez-Sykora[c,II], Asbjørn Nilsen Riseth[h,II], Fredrik Ødegaard[g,II], and Simos Zachariades[c,II]

[a]Department of Mathematics, Vrije Universiteit, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands; [b]Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Science Park 105-107, 1098 XG Amsterdam, The Netherlands, and Amsterdam Business School, University of Amsterdam, Plantage Muidergracht 12, 1018 TV Amsterdam, The Netherlands; [c]Mathematical Sciences, University of Southampton, SO17 1BJ, UK; [d]Faculty IV Electrical Engineering and Computer Science, Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin; [e]Haensel AMS, Advanced Mathematical Solutions, Ritterstr. 2a, 10969 Berlin, Germany; [f]Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027; [g]Ivey Business School, Western University, 1255 Western Road, London, ON N6G 0N1, Canada; [h]Mathematical Institute, University of Oxford, OX2 6GG, UK.

[I]dynamic pricing challenge organizer, lead author;

[II]dynamic pricing challenge participant, co-author.

arXiv:1804.03219v1 [q-fin.GN] 20 Mar 2018

Abstract

This paper presents the results of the Dynamic Pricing Challenge, held on the occasion of the 17th INFORMS Revenue Management and Pricing Section Conference on June 29-30, 2017 in Amsterdam, The Netherlands. For this challenge, participants submitted algorithms for pricing and demand learning of which the numerical performance was analyzed in simulated market environments. This allows consideration of market dynamics that are not analytically tractable or can not be empirically analyzed due to practical complications. Our findings implicate that the relative performance of algorithms varies substantially across different market dynamics, which confirms the intrinsic complexity of pricing and learning in the presence of competition.

Keywords: dynamic pricing, learning, competition, numerical performance

[1]Chris Bayliss and Christine Currie were funded by the EPSRC under grant number EP/N006461/1. Andria Ellina and Simos Zachariades were part funded by EPSRC as part of their PhD studentships. Asbjørn Nilsen Riseth was partially funded by EPSRC grant EP/L015803/1.

[2]Corresponding author: Ruben van de Geer, e-mail: r.vande.geer-at-vu.nl, phone: +31(0)644810227.

1 Introduction

1.1 Motivation

It is becoming increasingly common in today's online marketplaces that sellers' pricing decisions are determined by algorithms. The most striking example is arguably Amazon.com, which made more than 2.5 million price changes each day during 2013—a staggering figure that—most likely—has only increased ever since.[3] Even the price of the Bible—not the most obvious candidate for dynamic pricing—changes dozens of times each year,[4] which reveals that algorithmic pricing has gained a strong foothold in today's business practice. The complexities of optimally adjusting prices in response to competitors' prices, changing market circumstances, interactions between products in the seller's own portfolio, consumer reviews, incomplete information about consumers behavior, and many more factors that affect demand and revenue are obviously huge. To address these complexities, a large stream of scientific literature has emerged that designs pricing algorithms and analyzes their performance. A particularly large research area has evolved around the question of *learning*: how should a seller price its products to optimize profit when the price-demand relation is unknown upfront, and therefore has to be learned from accumulating sales data?

In recent years, a large number of studies have appeared that address this question from a monopolist's perspective (see literature review below). These research efforts have led to an understanding of the structure of optimal pricing strategies in a monopolist setting, in particular into the question of how much effort a seller should put into price experiments in order to strike the right balance between 'exploration' (conducting price experiments in order to learn the price-demand relation) and 'exploitation' (utilizing statistical knowledge to maximize profit).

For pricing and learning in a competitive environment, the picture is rather different. It turns out to be very difficult to give a useful qualitative assessment of a pricing strategy, for the simple reason that its performance depends on the (unknown) pricing behavior of competitors. A particular strategy may work very well when used against simplistic pricing rules, but perform much worse against sophisticated algorithms. Even the right performance measure is not clear (can one, e.g., improve upon the full information Nash equilibrium?). Not only is theoretical understanding limited; there also does not appear to be an extensive numerical study that compares the practical performance of different algorithms.

Thus, there is a serious lack of understanding of the structure of well-performing pric-

[3] https://www.profitero.com/2013/12/profitero-reveals-that-amazon-com-makes-more-than-2-5-million-price-changes-every-day/, visited on December 12, 2017.

[4] https://camelcamelcamel.com/Holy-Bible-James-Version-Burgundy/product/0718015592, visited on December 12, 2017

ing strategies with learning and competition, while at the same time understanding these pricing strategies is increasingly important from a practical viewpoint. This motivated the organizers of the INFORMS Revenue Management & Pricing Section Conference 2017 to organize a dynamic pricing contest, in order to get insights into the numerical performance of different pricing strategies in a competitive environment with incomplete information, and so to gain insight into the properties of well-performing pricing policies. The results of this contest are reported in this paper.

1.2 Literature

The literature on 'learning and earning' from a monopolist's perspective has gained much attention in recent years: see, e.g., Araman and Caldentey (2009), Besbes and Zeevi (2009), Farias and van Roy (2010), Harrison et al. (2012), Broder and Rusmevichientong (2012), Chen and Farias (2013), Cheung et al. (2013), den Boer and Zwart (2014), Keskin and Zeevi (2014), den Boer and Zwart (2015), and Johnson Ferreira et al. (2016). A recent review of these and related references is provided by den Boer (2015). A main take-away from this strand of literature is the importance of having the 'right' amount of *price experimentation*.

The importance of incorporating competition into these learning-and-earning models, and the potential detrimental effect of ignoring competition, has been demonstrated by Schinkel et al. (2002), Tuinstra (2004), Bischi et al. (2004, 2007), Isler and Imhof (2008), Cooper et al. (2014), and Anufriev et al. (2013), building forth on earlier work by Kirman (Kirman, 1975, 1983, 1995, Brousseau and Kirman, 1992).

Various approaches have been adopted to incorporate competition into learning-and-earning problems. Bertsimas and Perakis (2006) consider least-squares learning in an oligopoly with finite inventories and linear demand function, and propose an algorithm for estimation and pricing. Kwon et al. (2009), Li et al. (2010), Chung et al. (2012) adopt the framework of differential variational inequalities to study a capacitated oligopoly, propose an algorithm to solve these equations, and estimate unknown parameters via Kalman filtering. Perakis and Sood (2006) (see also Friesz et al. (2012)) take a robust-optimization approach to the dynamic oligopoly pricing problem, and study Nash equilibrium policies. Fisher et al. (2017) conduct a field experiment with randomized prices to estimate a consumer-choice model that does not require competitor sales observations, design a best-response pricing strategy, and test it with a field experiment for a leading Chinese online retailer.

A sample from the extensive economics and econometrics literature that study asymptotic behavior of pricing strategies in competitive environments is Cyert and DeGroot (1970), Kirman (1975), Aghion et al. (1993), Mirman et al. (1993), Fishman and Gandal (1994), Harrington (1995), Bergemann and Valimaki (1996), Gallego (1998), Alepuz

and Urbano (1999), Rassenti et al. (2000), Belleflamme and Bloch (2001), Keller and Rady (2003), Dimitrova and Schlee (2003). These papers typically assume that a particular learning method is used by the competitors, and study whether the price process converges to a Nash equilibrium.

The computer science literature also proposes several pricing-and-learning algorithms, see e.g. Greenwald and Kephart (1999), Dasgupta and Das (2000), Tesauro and Kephart (2002), Kutschinski et al. (2003), Könönen (2006), Jumadinova and Dasgupta (2008, 2010), Ramezani et al. (2011). For a further discussion of these and other relevant papers, we refer to Section 6.2 of den Boer (2015).

Finally, several simulation platforms have been designed to assess the performance of pricing policies, see, e.g., DiMicco et al. (2003) or Boissier et al. (2017).

1.3 Contributions

This paper presents the results of the Dynamic Pricing Challenge, held on the occasion of the 17th INFORMS Revenue Management and Pricing Section Conference on June 29-30, 2017, at the Centrum Wiskunde & Informatica, Amsterdam, The Netherlands. For this challenge, participants were invited to submit pricing and learning algorithms that would compete for revenue in a broad range of simulated market environments in both duopoly and oligopoly settings. The extensive simulations that we ran allow us to describe the numerical performance of various pricing and learning algorithms and provides insight into the performance and properties of several types of policies. Given that the participants submitted a wide variety of algorithms—such as bandit-type models, customer choice models, econometric regression models, machine learning models, and greedy ad-hoc approaches—we are able to relate the performance of a broad range of algorithms to different market structures.

Hence, this paper offers a framework to analyze various paradigms from the field of pricing and learning with competition and allows us to consider market dynamics that are analytically intractable and can not be empirically analyzed due to practical complications. As such, this paper presents the results of a controlled field experiment that improve our understanding of pricing and learning with competition and helps to guide future research. Our most important findings are:

- Relative performance varies substantially across different market dynamics. Some algorithms perform well in competitive environments, whereas others are better at exploiting monopolist-like environments. None of the considered algorithms is able to dominate the others in all settings.

- Relative performance varies substantially across oligopoly and duopoly markets. For example, algorithms based on linear demand models perform very well in duopoly competitions, whilst performing poorly in oligopolies.

- The algorithms that generate most revenue are very reliant on price-sensitive customers, making them vulnerable to intensified competition. Other algorithms are more robust in a sense that they were able to generate revenue from various types of customers and attract more loyal customers.
- A greedy algorithm that follows the lowest-priced competitor in a tit-for-tat fashion proves very difficult to outperform.
- Ignoring competition is increasingly harmful when competition is more fierce, i.e., when the number of competitors in the market is large and/or price sensitivity of the customers is high.
- The amount of exploration needs careful consideration as too much exploration hurts performance significantly.

The organization of the rest of this paper is as follows. In Section 2 we describe the experimental design of this study. In Section 3 and 4 the results are presented and discussed, respectively. Finally, in Section 5 some concluding remarks are provided.

2 Experimental Design

2.1 Experimental Setting

This experiment was designed to resemble a market in which the competitors all sell a single product to a group of heterogeneous customers. The competitors have no information a priori on either the demand mechanism or the behavior of the other competitors and are required to post a price before each (discrete) time period. Furthermore, it was assumed that competitors can monitor each other's prices, but only observe their own sales (we refer to sales as the number of items sold). This is true for many markets in reality, especially in online retailing, where retailers can monitor competitors' prices without much effort. Thus, the participants of this experiment were required to design an algorithm that would accept as input their own realized sales and the historical prices of all competitors and, subsequently, as output returns their price for the period to come. In addition, we assumed there are no inventory restrictions and, to give the participants some direction, the following domain knowledge was made available: "it seems unlikely that posting prices higher than 100 is optimal". For convenience and to prevent compatibility issues, all participants were required to submit their pricing policy in Python 2.7.x or 3.x and no restrictions on the use of libraries were put in place.

To evaluate the performance of all submitted policies we ran 5000 simulations, where a single simulation consists of two different settings of competitive market environments:

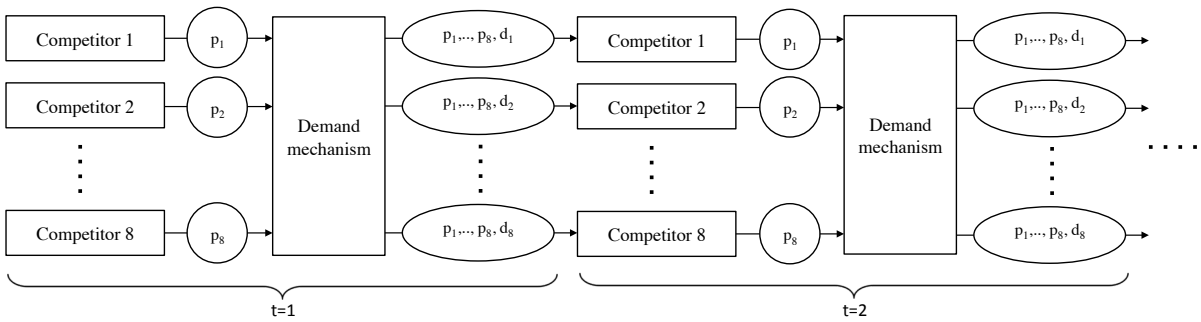


Figure 1: Overview of one simulation (oligopoly competition only); p_i and d_i denote the price and sales quantity of competitor i, respectively.

- Duopoly competition: all competitors compete in a round-robin setup, i.e., each competitor competes with all other competitors in one-vs-one contests.
- Oligopoly competition: all competitors compete simultaneously against each other.

Thus, if m is the number of participating competitors, each simulation consists of $\binom{m}{2}$ duopoly competitions and one oligopoly competition (in this experiment there were eight competitors, thus $m = 8$ and each simulation consists of $\binom{8}{2} = 28$ duopolies and one oligopoly). The oligopoly competition is especially insightful as it allows us to evaluate the competitors in a very competitive environment, whereas the duopoly competition can help us understand what the relative strength of the different competitors is.

In a single simulation, each of the duopoly competitions and the oligopoly competition consists of 1000 discrete time periods. This means that for $t \in \{1, \ldots, 1000\}$, each competitor posts a price for $t + 1$ and, subsequently, sales quantities for period $t + 1$ are generated from the undisclosed demand mechanism and all competitors earn revenue accordingly. This iterative process for the oligopoly competition is illustrated in Figure 1, whereas for the duopoly competitions, the scheme is the same except for the number of competitors, which is then two.

Finally, it was communicated to all participants that we would construct a ranking and determine the winner as follows. In each simulation, for each competitor we compute its revenue share of that particular simulation by averaging

- the competitor's share of total revenue earned in the oligopoly competition and
- the competitor's share of total revenue earned in the duopoly competitions.

The latter is computed by aggregating all the revenue earned in the $\binom{m}{2}$ duopoly competitions and computing the competitors' revenue shares accordingly. More precisely, if x_{ij} is the revenue earned by competitor j in the oligopoly competition of simulation i and y_{ijk} is the revenue earned by competitor j in the duopoly competition versus competitor

k in simulation i, then competitor j's revenue share of simulation i is equal to

$$\frac{1}{2}(\bar{x}_{ij} + \bar{y}_{ij}), \text{ where } \bar{x}_{ij} := \frac{x_{ij}}{\sum_{k=1}^{m} x_{ik}} \text{ and } \bar{y}_{ij} := \frac{\sum_{k=1}^{m} y_{ijk}}{\sum_{u=1}^{m}\sum_{k=1}^{m} y_{iuk}}. \tag{1}$$

Thus $\bar{x}_{ij}$ and $\bar{y}_{ij}$ are the oligopoly and duopoly revenue shares, respectively, of competitor j in simulation i. Note that, as such, a competitor's duopoly revenue share also depends on the revenue earned in competitions between other competitors. Consequently, it is not beneficial to earn a high revenue share in a duopoly competition where relatively little revenue is earned. The final score for competitor j is simply the average over all its revenue shares, i.e., its final score equals $\frac{1}{5000}\sum_{i=1}^{5000}\frac{1}{2}(\bar{x}_{ij} + \bar{y}_{ij})$.

This way of constructing a final score is a design choice of the pricing contest; clearly, numerous alternative ways to measure performance are conceivable.

2.2 Competitor Algorithms

In Table 1 summaries of all the competing algorithms are provided (see appendix for more detailed descriptions). The competitors rely on a wide range of demand models:

- linear models: OLS and WLS (ordinary and weighted least squares, respectively),
- bandit models: B-GRID, B-BUCKET, and B-MODEL (bandits where the arms pertain to price points in a grid, price buckets, and demand models, respectively),
- customer choice modeling: LOGIT,
- machine learning: ML,
- greedy price-matching heuristic: GREEDY.

All competitors randomize prices in the first periods and all competitors except LOGIT and GREEDY also engage in exploration later on to capture possible non-stationary dynamics (which was not the case for the markets that we simulated). Regarding the modeling of competitor behavior, most competitors use variants of exponential smoothing to predict competitor prices and optimize own prices accordingly. Others model competitors' prices as multivariate normal random variables, use the median of historical prices as predictor, or ignore competition altogether. All non-bandit models use a line search to optimize their own revenue with respect to price, except for WLS, which optimizes own revenue relative to the revenue earned by the competition.

Name	Demand Model	Pricing	Competitors' prices
LOGIT [e,1,226]	Finite mixture logit	During the first 100 periods the price is set as the minimum of the prices observed in the previous period. Then prices are optimized w.r.t. the demand model through a line search.	Multivariate normal distribution on historical prices
OLS [f,2,114]	Four linear regression models taking all combinations of price/log(price) and demand/log(demand). Select the model with the highest R^2.	During the first 40 periods and subsequently each time period with 5% probability a price is uniformly sampled from the interval $(0, 100)$. With probability 1% price is set to 0. Otherwise, prices are optimized through a line search.	Ignored
B-GRID [c,1,81]	ε-greedy multi-armed bandit algorithm with ten arms pertaining to the prices $10, 20, \ldots, 100$.	With probability $\varepsilon = 0.2$ an arm is chosen randomly, otherwise the arm with the highest average revenue so far is selected.	Ignored
B-BUCKET [c,2,160]	ε-greedy multi-armed bandit algorithm with ten arms pertaining to the price buckets $(0, 10], (20, 30], \ldots, (90, 100]$.	With probability 0.2 an arm is chosen randomly, otherwise the arm with the highest average revenue for the predicted competitor price (duopoly) or the mode of the competitors' prices (oligopoly) is chosen.	Exponential smoothing
B-MODEL [c,2,1010]	ε-greedy multi-armed bandit algorithm with four arms pertaining to three different demand models, each capturing different consumer behavior, and an epsilon-greedy algorithm in the fourth arm.	During the first 100 periods prices are set randomly, thereafter with probability 0.2 an arm is chosen randomly; otherwise offer the price suggested by the arm expected to generate the highest revenue, given a prediction of the competitors' prices.	Exponential smoothing
ML [d,1,325]	Least-squares, ridge, Lasso, Bayesian ridge, and stochastic gradient descent regression, and random forest (model selection through cross-validation).	Exploration cycles are initiated regularly and consist of 40 time periods in which prices are set according to a cosine function around the mean price level observed. Otherwise prices are optimized w.r.t. the demand model through a line search.	Exponential smoothing
GREEDY [d,1,11]	None	Price is set as minimum price observed in the previous period. If this price is lower than the 10% percentile of all the prices observed in the last 30 time periods, then the price is set as the maximum of this percentile and 5.	Ignored
WLS [g,1,387]	Weighted least squares where weights capture time-dependent aspects of demand.	Prices are set to maximize own revenue relative to competition. Prices are randomized during the first 10 periods prices and when own prices are constant for three subsequent periods.	Median of historical prices over variable window.

Table 1: The variables in brackets after each competitor name corresponds to affiliation (see title page), team size, and lines of code, resp.

2.3 Demand Mechanism

The design of the demand mechanism that we propose is built on the belief that it should resemble a competitive market with a heterogeneous customer base, as is often observed in practice. Meanwhile, we had to carefully manage the complexity of the demand mechanism to allow for evaluation, interpretation, and further analysis following the contest. In doing so, we assumed the arrival process and demand mechanism to be time-independent within a simulation, i.e., during time steps $1, \ldots, 1000$ the arrival process is homogeneous and the demand function is static. Furthermore, we assumed that the market consists of three segments, namely loyal customers, shoppers, and scientists, who all have their own parameterized demand functions, as we will explain in the following sections. We emphasize once more that the participants of this competition were not aware of any of the aforementioned assumptions regarding the market structure. All in all, we realize that the outcomes inevitably depend on the ground truth that is constructed here, but it is intended to be versatile enough to reward the competitors that are best able to learn various types of demand dynamics.

2.3.1 Arrivals and population composition

We assume Poisson customer arrivals over time with mean arrivals per time period equal to λ_i in the ith simulation, where $\lambda_i \sim U(50, 150)$. Here and throughout, if we write $x \sim F$, we mean that x is sampled from (and not distributed as) F, i.e., x is a realization. Furthermore, we denote the shares of the three customer segments, i.e., shoppers, loyals, and scientists, by θ_i^{sho}, θ_i^{loy}, and θ_i^{sci} in the ith simulation. In addition, the scientist segment is assumed to consist of two subsegments, namely PhDs and professors, with respective shares of γ_i^{phd} and γ_i^{prof}. The sampling of arrivals is summarized in the code block titled 'Arrival Process', where n is the number of arriving customers in a certain period,

Arrival Process

```
for i ∈ {1, ..., 5000} do
    Sample arrival rate λ_i ~ U(50, 150)
    Sample segment shares θ_i^sho, θ_i^loy, θ_i^sci
    Sample subsegment shares γ_i^phd, γ_i^prof
    for t ∈ {1, ..., 1000} do
        Sample arrivals n ~ Poisson(λ_i)
        Sample segment arrivals n^sho, n^loy, n^sci ~ Multinom(n, (θ_i^sho, θ_i^loy, θ_i^sci))
        Sample subsegment arrivals n^phd, n^prof ~ Multinom(n^sci, (γ_i^phd, γ_i^prof))
    end for
end for
```

consisting of n^{sho}, n^{loy}, and n^{sci} shoppers, loyal customers, and scientists, respectively. The scientist segment, consists of n^{phd} PhDs and n^{prof} professors. Thus, the number of

arriving customers n equals $n^{\text{sho}} + n^{\text{loy}} + n^{\text{sci}}$ and the number of arriving scientists n^{sci} equals $n^{\text{phd}} + n^{\text{prof}}$.

2.3.2 Demand of loyal customers and shoppers

The WTP of shoppers is assumed to be exponentially distributed with mean β_i^{sho} in simulation i, where $\beta_i^{\text{sho}} \sim U(5, 15)$. In each time period in simulation i, we sample a WTP for each arriving shopper from the exponential distribution with mean β_i^{sho} and compare these WTPs to the lowest price offered in the market. Each shopper for whose WTP exceeds the lowest price offered, buys from the competitor that offers the lowest price and otherwise leaves without buying anything. Ties are broken randomly.

For the loyal customers, we assume their WTP is exponentially distributed as well, but with mean β_i^{loy} in simulation i, where $\beta_i^{\text{loy}} = u \cdot \beta_i^{\text{sho}}$ and $u \sim U(1.5, 2.0)$, making the loyal customers relatively price-insensitive compared to the other customer groups, as one would expect from a loyal customer segment. In the simulation, each loyal customer is assigned randomly to a competitor, and they will buy from this competitor if their WTP exceeds the price the competitor is offering; otherwise they leave without making a purchase. Subsequently, we sample for all loyal customers a WTP and compare this to the price offered by the competitor that they are loyal to. In case their WTP exceeds the price offered, they buy, otherwise they leave without buying anything.

2.3.3 Demand scientists

The demand of the scientists is assumed to follow a finite mixture logit model, or latent class logit model, where the mixture comprises professors and PhDs. Thus, the professors and PhDs both choose according to a logit model with their own parameters. Let the posted prices in period t be equal to p_t and define the probability that an arriving PhD purchases from competitor k during simulation i as follows

$$q_k^{\text{phd}}(p_t) = \frac{\exp\left(\alpha_i^{\text{phd}} - \beta_{n,i}^{\text{phd}} \cdot p_{k,t}\right)}{1 + \sum_{j=1}^{n} \exp\left(\alpha_i^{\text{phd}} - \beta_{n,i}^{\text{phd}} \cdot p_{j,t}\right)}$$

where $n = 2$ in the duopoly case and $n = m(= 8)$ in the oligopoly case. This probability is defined similarly for professors. The parameters α_i^{phd}, $\beta_{n,i}^{\text{phd}}$, α_i^{prof}, and $\beta_{n,i}^{\text{prof}}$ of simulation i are sampled so that the prices that would be optimal for the shoppers, loyals, PhDs, and professors in isolation are of the same order of magnitude. This is done to ensure that no unrealistically large differences in optimal prices between (sub)segments occur. In doing so, for the PhDs we set α_i^{phd} equal to β_i^{sho} and, subsequently, set $\beta_{n,i}^{\text{phd}}$ such that the optimal price for the PhDs is within 50% of the optimal price for the shoppers (which

is equal to β_i^{sho}). This is achieved as follows

$$\alpha_i^{\text{phd}} = \beta_i^{\text{sho}}$$
$$p_i^{\text{phd}} := \beta_i^{\text{sho}} \cdot u, \text{ where } u \sim U(0.50, 1.50)$$
$$\beta_{n,i}^{\text{phd}} = \frac{W\left(ne^{\alpha_i^{\text{phd}}-1}\right)+1}{p_i^{\text{phd}}}$$

where W is the Lambert function, i.e., $W(xe^x) = x$, which is uniquely defined in this case as $ne^{\alpha_i^{\text{phd}}-1} > 0$. Thus, we set the parameters so that if the market consisted solely of PhDs, then, overall revenue would be maximized if p_i^{phd} was set by all competitors, where p_i^{phd} is within a reasonable distance of the average WTP of shoppers. Thus, the parameters are set so that

$$\arg\max_{p \in \mathbb{R}_+^n} \sum_k p_k q_k^{\text{phd}}(p) = \mathbb{1}_n p_i^{\text{phd}}$$

where $\mathbb{1}_n p^{\text{phd}}$ is understood to be an n-vector with each element equal to p_i^{phd}. Similarly, we set

$$\alpha_i^{\text{prof}} = \alpha_i^{\text{phd}} \cdot u, \text{ where } u \sim U(1.00, 1.25)$$
$$p_i^{\text{prof}} := p_i^{\text{phd}} \cdot u, \text{ where } u \sim U(1.00, 1.50)$$
$$\beta_{n,i}^{\text{prof}} = \frac{W\left(ne^{\alpha_i^{\text{prof}}-1}\right)+1}{p_i^{\text{prof}}}$$

so that the optimal price in a market consisting of only professors would be higher than in a market that consists solely of PhDs, while keeping the price levels in line.

2.3.4 Demand mechanism summary

Summarizing, in simulation $i \in \{1, \ldots, 5000\}$, in each period we see in expectation $\lambda_i \cdot \theta_i^{\text{sho}}$ arriving shoppers, $\lambda_i \cdot \theta_i^{\text{loy}}$ arriving loyals, $\lambda_i \cdot \theta_i^{\text{sci}} \cdot \gamma_i^{\text{phd}}$ arriving PhDs, and $\lambda_i \cdot \theta_i^{\text{sci}} \cdot \gamma_i^{\text{prof}}$ arriving professors. Each of these customer types chooses according to its own parameterized demand function (of which the parameters are constant throughout the simulation) as described in the previous sections. This all implies that in some simulations, $\theta_i^{\text{loy}} \approx 1$ meaning that all competitors are essentially monopolists, which should theoretically lead to higher prices, lower sales, and relatively high revenues. Another extreme, when $\theta_i^{\text{sho}} \approx 1$, resembles a market in which there is perfect competition—each competitor offers the same product to arriving customers that purchase the cheapest alternative available. In this case, one would expect prices to spiral down over time. In addition, the share of loyal customers in the market ($\frac{1}{3}$ on average) is independent of the number of competitors in

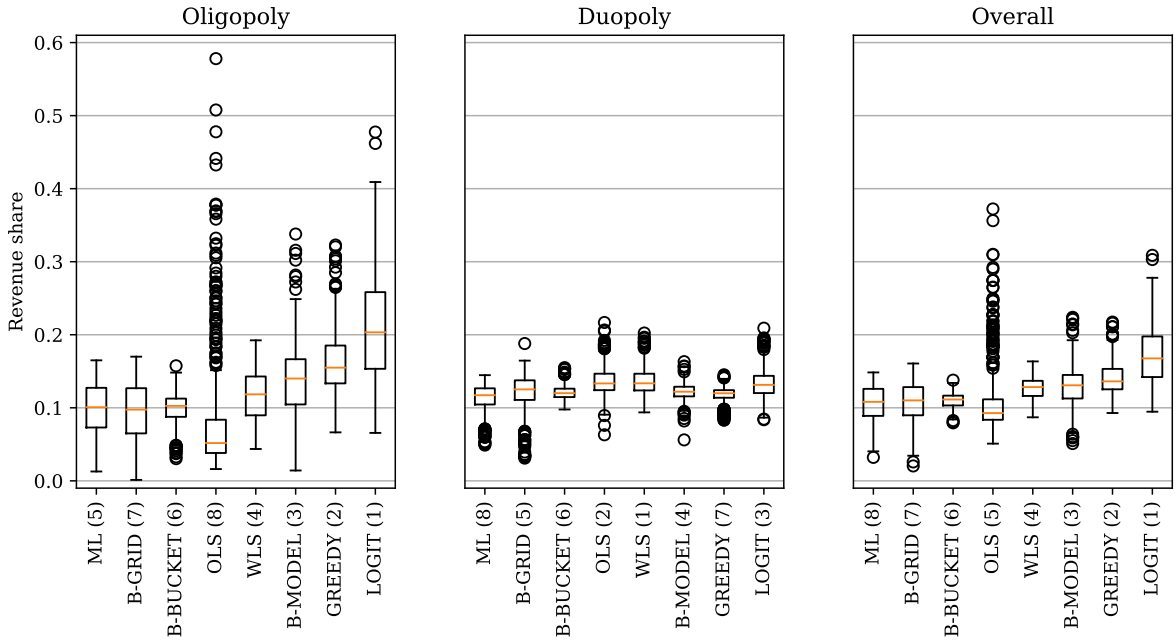


Figure 2: Boxplots of the revenue shares for the oligopoly competition, the duopoly competitions, and the overall competition. In parenthesis after the names of the algorithms are the rankings per competition part (e.g., WLS was the winner of the duopoly part). Only the first 500 of 5000 simulations are used to construct the boxplots for the sake of readability.

the market. Therefore, in a duopoly, an arriving customer is loyal to a specific competitor with probability $\frac{1}{2} \cdot \frac{1}{3}$, whereas, in the oligopoly, this probability equals $\frac{1}{m} \cdot \frac{1}{3}$. Therefore, by construction, we anticipate the market to be much more competitive in the oligopoly setting than in the duopoly case, as one would expect from economic theory.

3 Results

First, in Section 3.1 a summary of the overall results is presented and then, in Sections 3.2 and 3.3, the results of the oligopoly and duopoly competitions, respectively, are considered in greater detail. Thereafter, in Section 4, a discussion is provided on the observations presented here.

3.1 Overall Results

In Figure 2 boxplots of the revenue shares for the oligopoly, the duopoly, and the overall competition are provided. Thus, the boxplots in the left panel are based on the $\bar{x}_{ij}$'s, in the middle panel on the $\bar{y}_{ij}$'s, and in the right panel on $\frac{1}{2}(\bar{x}_{ij}+\bar{y}_{ij})$'s, all defined in (1). The figure reveals that LOGIT was the eventual winner of the competition and that its success was primarily due to its superior performance in the oligopoly competitions. Overall,

the differences in performance in the oligopoly are substantial, whereas in the duopoly competitions it has proven to be much harder to outperform one another. Nevertheless, we observe remarkable differences in relative performance across the oligopoly and duopoly parts. For example, OLS nearly earned the highest mean revenue share in the duopoly competitions, while its performance in the oligopoly competition was on average the worst amongst all competitors. Note however, that despite this poor average, OLS did perform particularly well on some occasions (earning almost 60% of all revenue on one occasion). The other way around, we observe that competitor GREEDY performs poorly in the duopoly competitions, but that its performance in the oligopoly competition is relatively good, being second placed only after LOGIT. Furthermore, we observe that B-BUCKET, B-GRID, and ML are consistently outperformed in both duopoly and oligopoly competitions, as they are amongst the bottom four in both parts of the competition. In the following sections, we analyze the aforementioned observations in greater detail and make more detailed comparisons between the different pricing strategies in the market.

3.2 Oligopoly Competition

In this section, we uncover what causes the substantial differences in performance in the oligopoly competition, which are observable in the first panel in Figure 2. This is done by analyzing how the competitors differ in terms of realized sales and prices posted and how their performance varies as the market composition, i.e., the shares of segments, differs across simulations.

First of all, in Figure 3 we present the price and revenue realizations of a single simulation to demonstrate what a simulation typically looks like. Although a single simulation is not representative for performance in general, we found that the characteristics that can be observed in Figure 3 are demonstrative for most of the scenarios that we visually inspected. In particular, it can be observed that the prices of LOGIT, B-MODEL, GREEDY, and WLS converge (after engaging in price exploration), whilst the other competitors show more erratic price paths. For example, B-BUCKET and B-GRID put a lot of emphasis on exploration throughout the simulation without eventually converging to a small price range, which can be explained by the fact that these algorithms engage in active price exploration with a 20% probability in each period. Furthermore, we see that OLS initially seems to converge to a competitive price, but that from around period 300 onwards it fluctuates around a relatively high price level, earning hardly any revenue. Finally, regarding ML we can observe that it initiates many exploration cycles around a cosine function, which affects its performance negatively in this stationary environment.

To analyze how the competitors differ in sales generated across segments, in Figure 4 (a) the mean sales per time period per segment for each competitor is presented (note that sales is understood to be a quantity here and throughout). From this figure it can

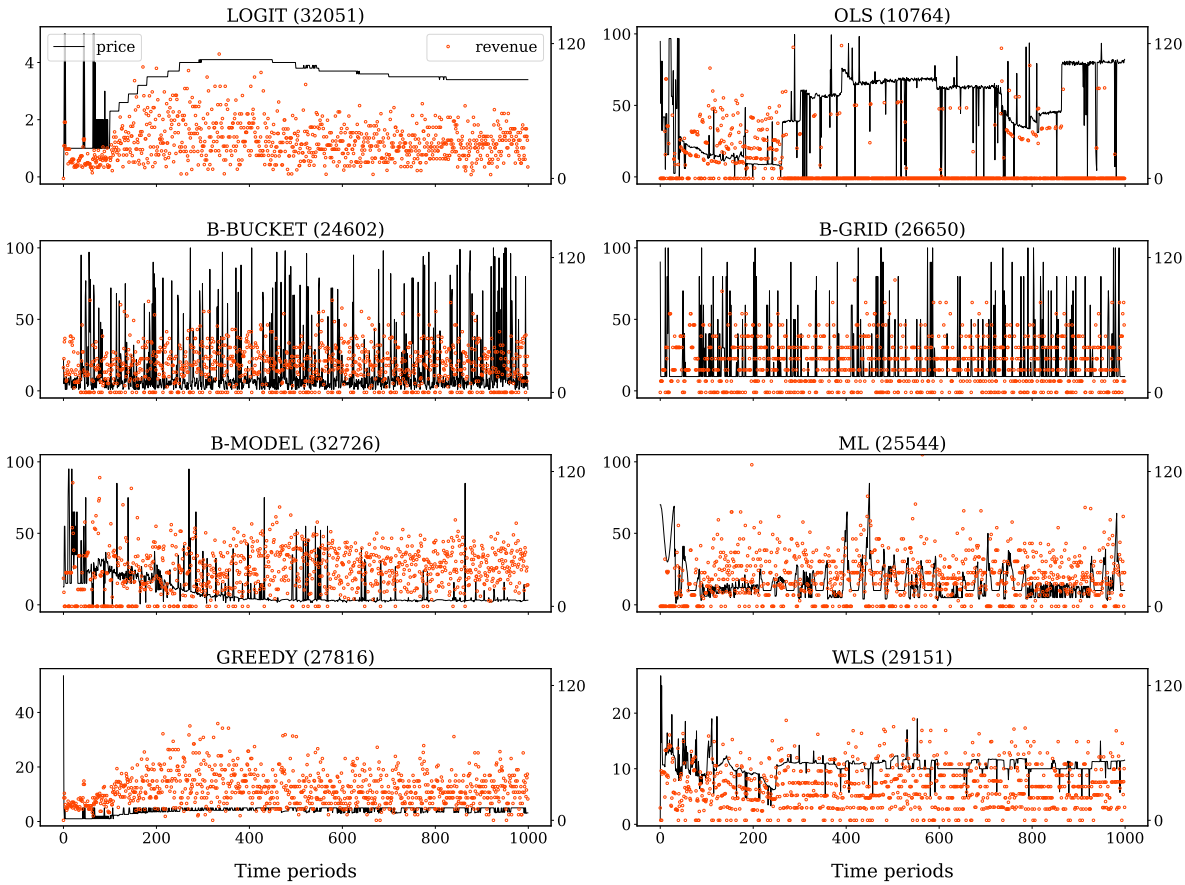
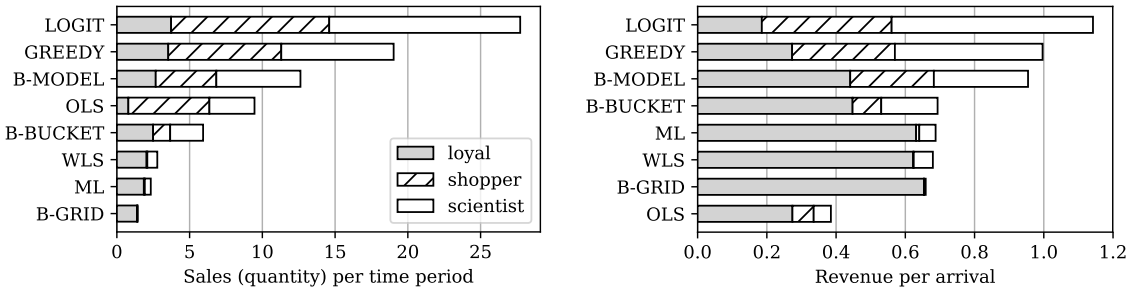


Figure 3: Realizations of prices and revenue for simulation 4955. The solid line represents the prices and pertains to the left axis. The red dots represent the revenue earned and pertain to the right axis. In parenthesis is the total accumulated revenue. In this simulation the segment shares were equal to $\theta^{\text{sho}} = 0.05$, $\theta^{\text{loy}} = 0.72$, and $\theta^{\text{sci}} = 0.23$.



(a) (b)

Figure 4: For the oligopoly competition (a) mean sales per time period and (b) mean revenue per arriving customer, split out over the three customer segments.

be observed that the three best-performing algorithms in the oligopoly, LOGIT, GREEDY, and B-MODEL, are also able to generate the highest sales. Furthermore, LOGIT not only generates the highest total sales, but it is also able to generate the highest sales per customer type. This could well be due to the fact that if a competitor generates high sales in the shopper segment, this means that it is frequently the lowest priced competitor in the market, which, in turn, means that the scientists and its loyal customers are also likely to buy. Thus, high sales in the shopper segment leads automatically to relatively high sales in the other two segments. In addition, it is remarkable that WLS generates much lower sales than OLS, while, according to Figure 2, WLS performs significantly better in terms of revenue generation. Arguably, this is due to the fact that WLS sells predominantly to the high-paying loyal segment, whereas OLS generates its revenue primarily from the shopper and scientist segments, which are more price sensitive. Overall, the competitors that are able to reach the shoppers and scientists are capable of generating high sales, which concurs with the observation that for each competitor the potential sales from shoppers and scientists is much larger than from loyal customers (as was discussed in the final part of Section 2.3.4).

In Figure 4 (b) the mean revenue per arriving customer for each competitor is illustrated (e.g., LOGIT makes on average about 0.30 from an arriving shopper, whereas B-GRID practically earns nothing in this case). The figure reveals that the competitors' distributions of earnings over the customer segments varies substantially. This can be seen when considering the top performers (LOGIT, GREEDY, and B-MODEL), which make the majority of their income from shoppers and scientists. For the other algorithms, especially ML, WLS, and B-GRID, it holds that they earn hardly anything from shoppers and scientists, but are able to earn relatively much from incoming loyal customers. To relate these observations to the prices posted in the market, we provide in Figure 5 boxplots of the prices posted and in Table 2, amongst other things, the fraction of time periods in which a competitor was the lowest or highest priced competitor in the market. In general,

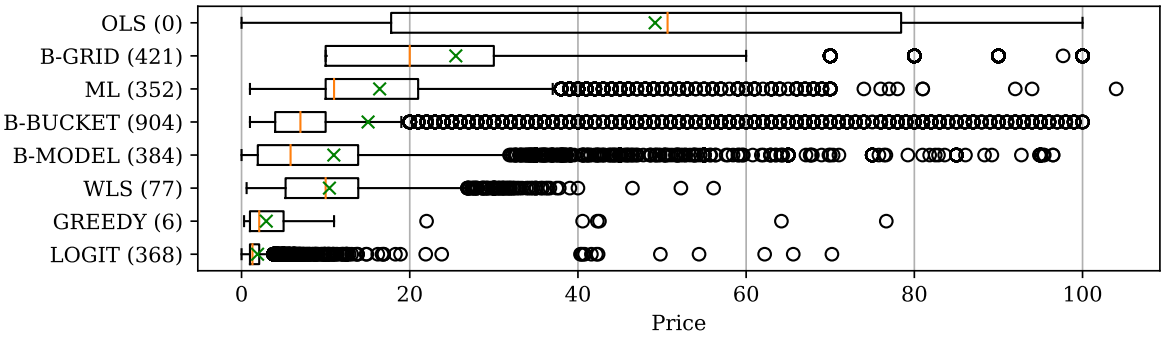


Figure 5: Boxplot based on the prices posted in 5000 randomly chosen time periods (out of 5000×1000 time periods in total). The bars and crosses denote the median and mean, respectively. In brackets are the corresponding number of outliers.

	Price		Sales		Revenue	
	Low	High	Low	High	Low	High
LOGIT	<u>0.55</u>	0.00	0.01	<u>0.37</u>	0.02	<u>0.23</u>
OLS	0.13	<u>0.60</u>	<u>0.69</u>	0.13	<u>0.68</u>	0.14
B-BUCKET	0.08	0.09	0.27	0.07	0.27	0.07
B-GRID	0.00	0.19	0.47	0.00	0.44	0.10
B-MODEL	0.20	0.06	0.23	0.19	0.24	0.15
ML	0.01	0.08	0.36	0.01	0.35	0.08
GREEDY	0.46	0.00	0.06	0.27	0.07	0.15
WLS	0.00	0.01	0.17	0.00	0.16	0.09

Table 2: Fraction of time periods (of 5000×1000 time periods total) in which a competitor was lowest/highest priced, generated the lowest/highest sales, and earned the lowest/highest revenue. The column sums are greater than one because ties occur.

we observe that the successful algorithms set relatively low prices, earn most revenue from shoppers and scientists, and relatively little revenue from the loyal customers. It is by no means immediate that LOGIT's low-pricing strategy works well under all circumstances—its dependency on shoppers and scientists makes LOGIT vulnerable in case the market consists of more competitors that are aggressive on price. Thus, in a sense is B-MODEL more robust by being less reliant on the price sensitive segments. This is also illustrated by B-GRID, which only serves loyal customers and generates the least sales, but does not earn substantially less than WLS, ML, and, B-BUCKET and performs substantially better than OLS.

Regarding price experimentation, we observe from Figure 5 that OLS's policy induces a very wide price range, with its first quartile around twenty and the third quartile just below eighty, resembling a Gaussian distribution of prices. In addition, GREEDY does not engage in much price experimentation, as one would expect from its construction. The other boxplots indicates that the price distributions all have a heavy right tail; these

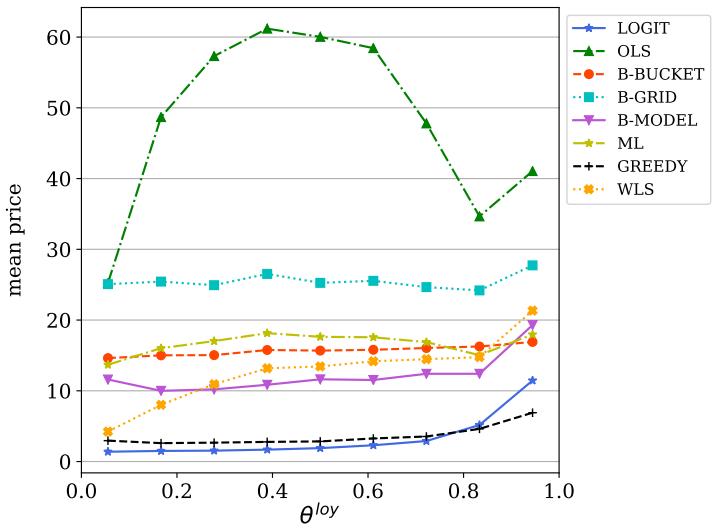


Figure 6: Mean prices for various levels of the share of loyal customers θ^{loy}.

competitors engage in exploration coincidentally, while pricing around a relatively small interval for most of the time.

In Figure 6 the mean prices for different values of θ^{loy} are given. This is insightful, since when the share of loyal customers increases from zero to one, the market moves from a very competitive market to a market in which every competitor is essentially a monopolist. One would theoretically expect that competitors post higher prices once their pricing power increases. However, from the figures, we can observe that the prices of three of the four worst-performers, namely OLS, B-GRID, and ML, do not increase in the share of loyal customers. On the other hand, the prices of LOGIT, GREEDY, WLS, B-BUCKET, and B-MODEL, do increase in the share of loyal customers. This indicates that these competitors are better capable of identifying the market structure and improve pricing decisions accordingly.

Finally, Figures 7 and 8 illustrate the mean revenue per time period for various segment shares of loyal customers, and scientists, respectively.[5] From Figure 7 we observe that, in general, revenues increase as the share of loyal customers increases, as one would theoretically expect. However, some competitors are better capable of exploiting the increase in pricing power than others—for example WLS's relative performance improves substantially as θ^{loy} increases, while LOGIT's relative performance deteriorates. On the other hand, in Figure 8 we observe that performance across competitors diverges as the share of scientists increases. Most notably, LOGIT's relative performance increases substantially, which can be attributed to the fact that LOGIT's demand specification is able to closely resemble the demand function of the scientists (which is a finite mixture of logit demand functions).

[5]The figure for the shopper segment is very similar to Figure 7 except for that the revenues decrease in the shopper share — figure is omitted to save space.

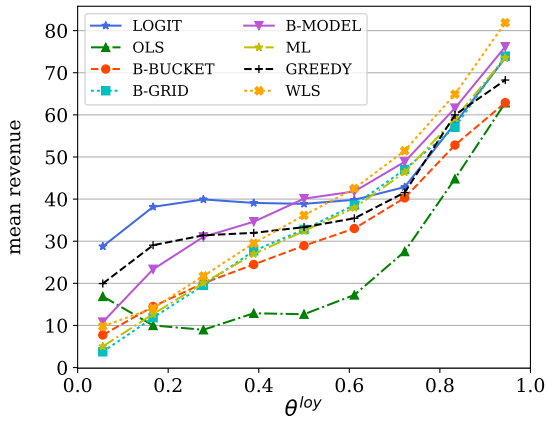


Figure 7: Mean revenue per time period for various levels of θ^{loy}.

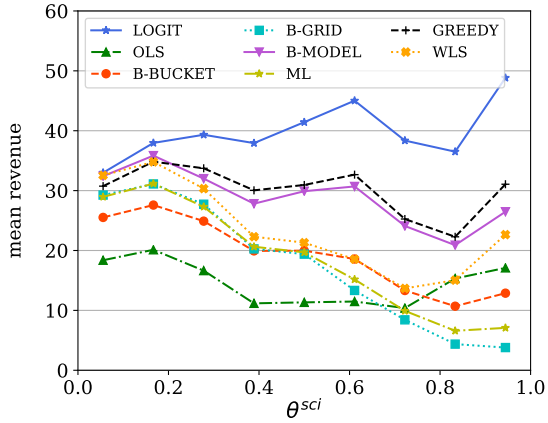


Figure 8: Mean revenue per time period for various levels of θ^{sci}.

3.3 Duopoly Competition

From Figure 2 it follows that in the duopoly competitions the differences in performance are less substantial than in the oligopoly competition. Two reasons can be identified why this is the case. First of all, the share of loyal customers is relatively large in the duopoly, so that both competitors have in general more pricing power whilst only having to consider one other competitor, making it harder to outperform one another. Second of all, the performance is not transitive in a sense that "if A beats B and B beats C, then A beats C", so that differences in performance tend to cancel out over the duopolies. Nonetheless, the duopolies are interesting to analyze the relative performance of the various pricing policies.

For a single simulation, in Figure 9 we provide the realized prices and revenue of duopolies in which WLS, the top performer in the duopoly part, was involved.[6] The figure reveals that in the duopoly in which WLS and LOGIT compete relatively little revenue is generated since both algorithms price very low—this is not only the case in this example, but is a structural property, as will be shown below. It is interesting to see that GREEDY's performance is weakened by its mechanism that resets prices to 5 if prices get too low (these are the bumps that are visible in the plot). Finally, as was the case in the oligopoly, OLS is in this case not able to find a competitive price level consistently (however it was still able to perform well in the duopolies).

In Table 3 the mean revenue per time period for all the duopolies are provided. The last column contains the row-wise averages, which indicate how much revenue each algorithm makes on average overall. Similarly, the final row contains column-wise averages, which indicate how much revenue other algorithms were able to make against the corresponding algorithm (thus, e.g., LOGIT makes on average 234 per time period, while the other competitors make on average 218 when competing with LOGIT). The table confirms

[6] A plot of WLS vs B-GRID is omitted to save space but is very similar to the plot of WLS vs B-BUCKET.

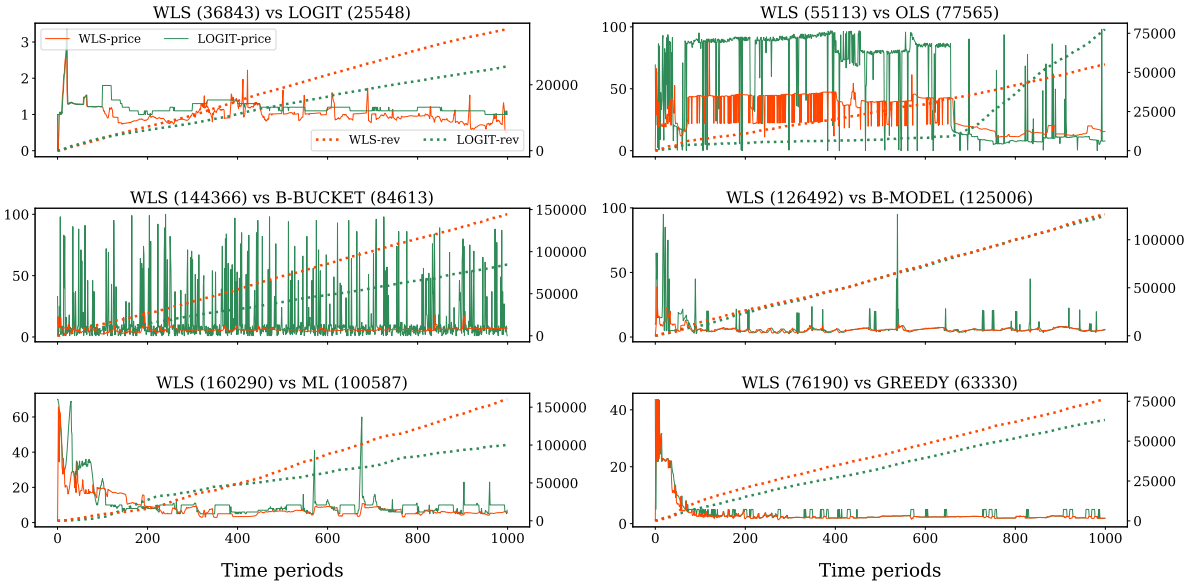


Figure 9: Realizations of prices and revenue for simulation 2285. The red and green solid lines represent the prices of WLS and the corresponding opponent, respectively, and pertains to the left axis. The red and green dotted lines represent the cumulative revenue earned by WLS and the corresponding opponent, respectively, and pertains to the right axis. In parenthesis is the total accumulated revenue. In this simulation the segment shares were equal to $\theta^{\text{sho}} = 0.33$, $\theta^{\text{loy}} = 0.29$, and $\theta^{\text{sci}} = 0.38$.

	LOGIT	OLS	B-BUCKET	B-GRID	B-MODEL	ML	GREEDY	WLS	average
LOGIT		252	221	319	238	242	266	96	234
OLS	235		172	265	210	295	256	249	240
B-BUCKET	175	249		256	207	272	198	175	219
B-GRID	196	247	169		239	306	274	181	230
B-MODEL	250	231	185	241		270	176	218	224
ML	249	221	177	236	203		216	186	213
GREEDY	267	260	206	273	196	209		123	219
WLS	153	257	234	361	219	317	140		240
average	218	245	195	279	216	273	218	175	

Table 3: The pairwise mean revenue per time period for all duopolies. Each cell indicates how much revenue the algorithm in the corresponding row was able to earn against the algorithm in the corresponding column on average per time period. Green (red) indicates if the amount was higher (lower) than that of the corresponding opponent.

Heatmap of mean prices and price dispersion

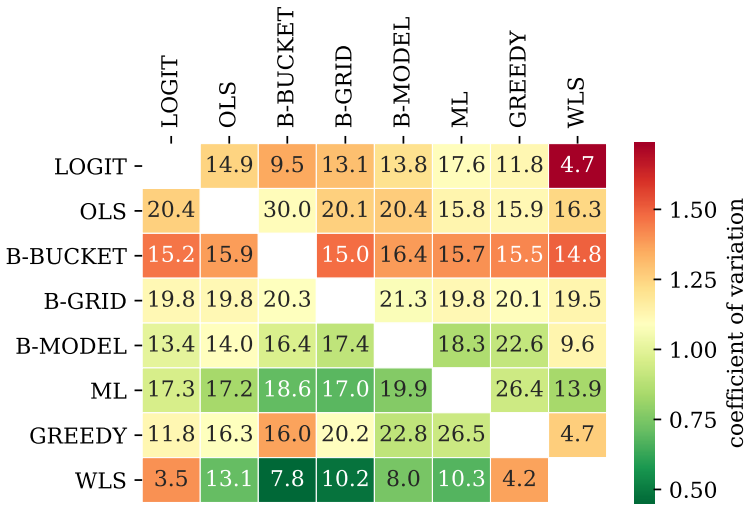


Figure 10: The number in each cell indicates the mean price that the algorithm in the corresponding row posted against the algorithm in the corresponding column. The color of each cell pertains to the coefficient of variation (i.e., the ratio of the standard deviation to the mean) and corresponds to the color bar on the right.

the observation from Figure 9 that the market between WLS and LOGIT is on average the smallest ($153 + 96 = 249$ revenue per time period). In fact, WLS proves to be very hard to generate revenue against, since on average competitors earn only 175 per time period when competing with WLS. Meanwhile, WLS is able to generate substantial profits with an average revenue per time period of 240, and as a result WLS was able to beat all other competitors based on revenue earned. This is remarkable, as WLS performs poorly in the oligopoly, as was discussed in the previous section.

Furthermore, we observe that the results are very mixed and that there is no competitor that loses against all other competitors. Even the worst performer, ML, is able to defeat other competitors, namely, LOGIT and GREEDY, which is remarkable as both perform very well overall. In addition, based on Table 3 we observe that GREEDY has a very steady performance, never earning much more (or much less) than its opponents, as one would expect—it simply follows its opponent's actions, without exploiting the opponent's weakness. The other top performer in the duopoly part, namely OLS, only beats B-GRID and ML according to Table 3. Nonetheless, we observe that OLS on average earns the same amount of revenue, namely 240, as WLS.

All in all, Table 3 reveals that the performance is very much dependent on the competitor's policy and that some algorithms that perform well in the oligopoly (e.g., GREEDY), struggle in the duopolies and vice versa (e.g., OLS). This, once more, confirms the intrinsic complexity of pricing and learning with competition.

Finally, to gain insight in the pricing levels of the competitors in the duopoly competitions, in Figure 10 a heatmap of the mean prices and the coefficient of variation (the ratio

of the standard deviation to the mean) is provided. The numerical values correspond to the mean prices posted and the color indicates the amount of dispersion in the prices. We observe that WLS consistently has the lowest mean price and that OLS, which performed equally well according to Figure 2, posted on average substantially higher prices than WLS. Thus, OLS is able to remain competitive while maintaining a higher overall price level. Other algorithms, e.g., B-GRID, also maintain a high mean price level, but are less successful in generating profit. This can well be explained by the fact that the price level of B-GRID is much less dependent on its competitor, i.e., its mean price posted is always around 20, whereas OLS varies its pricing level across its competitors, thereby being able to generate more profit. Regarding the dispersion of prices, we observe that WLS, B-MODEL, and ML are the least experimental in setting prices and that especially B-BUCKET is very experimental, which is in correspondence with Figures 3 and 9.

4 Discussion

In terms of overall performance, LOGIT has proven to be the most effective algorithm in this competition—it earned most revenue in the oligopoly part and was also competitive in the duopoly part, where it finished in third place, close behind OLS and WLS. Its success can partly be explained by the fact that its demand specification closely resembles the demand function of the scientist segment, which led to the highest revenue per arriving scientist (see Figures 4 (b) and 8). However, the fact that LOGIT was able to identify and exploit the revenue potential of the shopper segment by often pricing low was undoubtedly the key reason for its success. This strategy negatively affected performance when pricing power increased (see Figure 7), but proved beneficial overall. Arguably, if GREEDY would not have had arrangements in place to prevent a 'race to the bottom', LOGIT's revenue might have been substantially lower. This indicates that LOGIT's low-pricing strategy will not perform well in all circumstances—its dependency on shoppers and scientists makes LOGIT vulnerable in case the market consists of more competitors that are aggressive on price. In terms of robustness, B-MODEL performs well as it is less reliant on the price-sensitive segments, whilst still earning substantial revenue.

Regarding the two linear models (WLS and OLS), the difference in their performance between the duopoly and oligopoly parts is striking. In the duopoly setup, WLS and OLS are the two top performers, thereby confirming that the use of linear approximations for demand can give a simple and robust way to model demand for local price changes. The design of WLS, which anticipates its competitor's revenue by maximizing the difference of own and the competitor's revenue, has caused WLS to be very difficult to earn revenue against (see Table 3). On the other hand, OLS ignored competition altogether, which worked surprisingly well in the duopoly competitions and resulted in OLS posting much higher prices than WLS according to Figure 10 (but generating roughly the same amount

of revenue). In the oligopoly setup, however, WLS and especially OLS struggled, which indicates that ignoring competition is increasingly harmful as competition increases and that, regarding WLS, explicitly taking into account all competitors' anticipated revenues has proven ineffective. Similarly, the relative performance of B-GRID, which also ignored competition, was worse in the oligopoly.

Two of the bandit models, namely B-GRID and B-BUCKET, performed poorly due to a defect in their designs, which only allowed prices in very crudely discretized price sets and prevented them to set low prices consistently thereby greatly hindering their performance. The algorithms could be improved by making them more adaptive by, e.g., allowing poorly performing arms to be eliminated or adding additional arms close to the current optimal value to allow the algorithm to focus in on profitable prices. It would also be interesting to assess the performance of existing continuous armed-bandit algorithms.

However, the other bandit model, B-MODEL, proved to be competitive and finished third overall. It was designed to cope with different customer behaviors (e.g., bargain hunters and quality seekers, see the appendix for details) by assigning a different demand model to each of its arms. The relative success of the B-MODEL appears to have been its ability to generate above average revenues from each customer segment as, e.g., illustrated in Figure 4 (b). One of its pitfalls has been its high level of exploration of price points as illustrated in, e.g., Figures 3 and 5, especially in comparison to LOGIT and GREEDY.

The approach of ML heavily relied on machine learning methods to model the demand characteristics and was designed to cope with non-stationarities, such as changes in the price elasticity over time or changes in the overall demand volume. In doing so, ML persisted in engaging in exploration cycles, which hindered its performance in the stationary environment that was simulated. This confirms the notion that one should only experiment if the anticipated long-term revenue of doing so outweighs the short-term cost.

Finally, certainly the most simple strategy, namely GREEDY's ad-hoc approach of matching the lowest price in the market, turned out to be very effective in the oligopoly. Generally, it followed whoever was the lowest priced (mostly LOGIT and B-MODEL) and was thereby able to generate substantial revenue especially from scientists and shoppers. The arrangements that GREEDY put in place to prevent downward price spirals were initiated frequently (see, e.g., Figure 9), which could otherwise have led to even lower prices and presumably deteriorating revenues. In the duopoly part, no competitor was able to significantly outperform GREEDY, as one would expect, however, GREEDY was not able to exploit competitors' weaknesses either, leading to average performance.

5 Conclusion and Managerial Insights

This paper presented the results of the Dynamic Pricing Challenge, held on the occasion of the 17th INFORMS Revenue Management and Pricing Section Conference on June 29-

30, 2017, at the Centrum Wiskunde & Informatica, Amsterdam, The Netherlands. The participants of this pricing challenge submitted a wide variety of pricing and learning algorithms of which the numerical performance in a simulated market environment with competition was analyzed. As such, this paper presented a framework in which various paradigms from the field of pricing and learning with competition are analyzed by means of a controlled field experiment. This has allowed us to consider market dynamics that are not analytically tractable or can not be empirically analyzed due to practical complications.

Our analysis has revealed a number of interesting insights both from practical and scientific points of view. First of all, it is shown that relative performance varies substantially across oligopoly and duopoly markets and across different market dynamics, which confirms the intrinsic complexity of pricing and learning in the presence of competition. Most notably, none of the considered algorithms is able to consistently outperform the market—each algorithm meets its Waterloo at some point in the competition. This reveals that algorithm design needs careful consideration and that the structure and dynamics of the market need to be taken into account to determine which algorithm is the best fit. Second, a greedy algorithm that followed the lowest-priced competitor in a tit-for-tat fashion has proved very difficult to outperform. Especially in the oligopolistic markets it was able to attract substantial revenue from price-sensitive customers, whilst—as expected—showing average performance in the duopoly competitions. Third, although the eventual winner was determined by revenue earned, the results reveal that some algorithms are better capable of attracting customers from different segments, thereby being less reliant on one specific segment, and, therefore, being more robust. The winning algorithm, e.g., is predominantly dependent on price-sensitive customers that can easily be targeted by competitors, whilst other competitors earned reasonable revenue from a more loyal customer base. Fourth, the results reveal that ignoring competition is increasingly harmful when competition is more fierce, i.e., when the number of competitors in the market is large and/or price sensitivity of the customers is high. Finally, the analysis reveals that too much exploration can hurt performance significantly.

Possible extensions to this study that could enhance its generalizability is to impose more complex market dynamics in the simulations, such as temporal dependencies or strategic customer behavior. Although it is appealing (and not too difficult) to do this, we chose not to do so, since it makes it more cumbersome to relate the algorithms and their performance to the dynamics of the market.

6 Appendix

6.1 Competitor Algorithms

6.1.1 Competitor LOGIT

This competitor models the demand according to a finite mixture logit model, where the mixture is taken over the number of possible customer arrivals. Thus, a probability distribution over the the number of arrivals in a single period is estimated and for each possible number of arrivals, a different multinomial logit model is estimated as well. Each multinomial logit model here, induces a probability distribution over the competitors, i.e., it specifies with which probability an arriving customer purchases from each competitor (including a no-purchase option). In doing so, it is assumed that the utility of buying from competitor i is of the form $a - bp_i$, where p_i is the price posted by competitor i and a and b are assumed to differ across the mixture components.

In practice, this competitor uses the first 100 time periods to estimate the maximum number of arriving customers in a single time period. This is done by setting a price of 0 for the first period and for each of the following 99 periods of this exploration phase, the price is set as the minimum of the prices observed in the previous period. After these 100 periods, an upper bound on the number of arrivals in a single period is taken as the maximum realized demand in a single period multiplied by $(m + 1)$, i.e., the number of competitors plus one. Subsequently, an Expectation-Maximization algorithm is used to estimate a probability distribution over the number of arrivals, as well as the parameters of the multinomial logit models. All these parameters are updated every 20 time periods.

To optimize prices, in every period the competitors' prices for the period to come are predicted. For this purpose, it is assumed that the sorted prices of the competitors follow a multivariate normal distribution, where the sorted prices are used to mitigate the effect of price symmetries. Subsequently, 1000 competitor prices are sampled from the multivariate normal distribution and the revenue function is approximated by averaging over these realization. To optimize the price, a crude line search over a discretization of the assumed price space $(0, 100)$ is executed and the price with the highest revenue is chosen.

6.1.2 Competitor OLS

The approach of this competitor to pricing is to favor simplicity. The view is taken that competitors' actions cannot be controlled and that for all intents and purposes, they are random. Thus, they are modeled as an aggregate source of random "noise" and the focus is on how the competitor's own price influences demand in this environment. The algorithm is split into an exploration segment and a "running" segment. The exploration segment lasts for the first 40 periods and the running segment lasts for the rest of the 960

periods.

In the exploration segment, the algorithm explores the field to ensure sufficient variation in data. In each period, a price is sampled uniformly from the interval $(0, 100)$. After the exploration period, the algorithm enters the running segment. In the running segment, the majority of the time consists of estimating a demand curve based only on the competitor's own historical prices and optimizing accordingly. To do so, four linear regression models are fit, taking all combinations of log-transformation of both independent (price) and dependent (demand) variables, and the model with the highest R^2 value is chosen (OLS is an acronym for ordinary least squares). Using this model, the price is optimized using a crude line search and, subsequently, a small perturbation is added to the price for further exploration.

Finally, in each period in the running segment there is a 5% chance of further exploration and a 1% chance of "competitive disruption". Here, "competitive disruption" is an action designed to intentionally confuse competitors who attempt predict competitor prices or who use competitor prices in their model. When this action is initiated the model sets the price to zero in an attempt to confuse competitors via extreme actions.

6.1.3 Competitor B-GRID

B-GRID is adapted from the ε-greedy multi-armed bandit algorithm (Sutton and Barto, 1998). It assumes a bandit framework with ten arms, where the arms pertain to the prices $10, 20, \ldots, 100$ (B-GRID is an acronym for bandit on a grid). Thus selecting the first arm means posting a price of 10. This algorithm neglects competition and simply keeps track of the average revenue under each arm. With probability ε, an arm is selected randomly, whereas with probability $1 - \varepsilon$, the arm that has the highest observed average revenue is selected. The exploration parameter ε is set to 0.2, so that on average 200 time periods are used for exploration and 800 for exploitation.

6.1.4 Competitor B-BUCKET

This competitor considers the problem of learning and pricing in a multi-armed bandit framework similar to that of B-GRID. In doing so, the optimal price is assumed to be contained in the interval $(0, 100]$, which is split into ten intervals of even length, i.e., it is split into price buckets $(0, 10], (10, 20], \ldots, (90, 100]$. Each of these price buckets pertains to one arm and selecting a specific arm means posting a price that is uniformly sampled from the corresponding price bucket (B-BUCKET is an acronym for bandit with buckets).

To incorporate the competitors' prices, it is assumed that the arms' values, i.e., revenues, depend on the prices posted by the other competitors. More precisely, in each time period the competitors' modal price bucket is forecast using exponential smoothing. The modal price bucket is the bucket that is predicted to contain most of the competitors'

prices. We assume that the optimal choice of price to offer is dependent on this modal price bucket.

In practice this works as follows. At each time step, with probability ε an exploration step is performed in which an arm is selected randomly. Alternatively, with probability $1-\varepsilon$, an exploitation step is undertaken. In this case, the algorithm selects a price from the price bucket with the highest observed average revenue for the predicted modal price bucket. The exploration parameter ε is set to 0.2, so that on average 200 time periods are used for exploration and 800 for exploitation.

6.1.5 Competitor B-MODEL

This competitor advocates a bandit formulation of the problem as well, although its design differs conceptually from that of B-BUCKET and B-GRID. Where the aforementioned two competitors assign prices (or prices buckets) to arms, here, an arm pertains to a demand model (B-MODEL is an acronym for bandit with models). The demand models that constitute the four arms are the following:

- **Demand Model 1** (Bargain hunters) assumes that the distribution of customers' willingness to pay (WTP) is normally distributed and that customers select a competitor's price from the subset of prices that fall below their WTP with probability proportional to $\left(\frac{WTP-p_i}{WTP}\right)^b$, where p_i is the price being offered by competitor i and b is a parameter that influences customers' price sensitivity. High (low) values of $b > 1$ (< 1) capture customer populations that are highly (in)sensitive to prices close to their reserve price. In general this first demand model captures bargain hunters as in all cases customers will tend to choose low prices where possible.

- **Demand Model 2** (Quality seekers) is a variant of the first demand model but the reserve price of a customer is proportional to $\left(1-\frac{WTP-p_i}{WTP}\right)^c$. This model captures customers who use price as an indicator of quality. The parameter c has a similar interpretation to b above.

- **Demand Model 3** (Cheapest price subset) assumes that each customer sees a different random subset of the available prices. Customers are assumed to select the cheapest price that is visible to them. The subset is assumed to include a random number of options uniformly distributed between d and e, which are parameters that can be estimated from the demand data.

The fourth arm alone is used for the first 100 time periods with a relatively high exploration rate to provide sufficient data for estimating the parameters a, b, c, and d of the three demand models by means of simulated annealing. After 100 time periods the reward vectors are reset and the four-armed bandit assumes control of pricing. Similar to

the previous two bandit algorithms, with probability ε an arm is selected randomly and otherwise the most profitable arm is selected.

Optimal prices are chosen based on a forecast of the competitor price (duopoly) or the profile of competitor prices (oligopoly), where we define the competitor price profile to be an ordered list of competitors' prices. For the oligopoly the competitor price profile is forecast for the next time period using exponential smoothing with trend. In order to estimate the optimal price to charge under each demand model, the algorithm generates a set of potential prices and the projected revenue is evaluated at each price, for the forecast competitor price profile. The price with the highest predicted revenue is assumed to be the best price for this demand model. In an exploitation step, the algorithm selects the arm with the highest predicted revenue and offers the best price for this arm.

6.1.6 Competitor ML

The approach of this competitor is to rely on machine learning techniques to predict demand and optimize prices accordingly (ML is an acronym for machine learning). Much emphasis is put on learning the demand characteristics, as the algorithm dynamically switches back and forth from exploration to exploitation mode over time. In exploration mode, during forty time periods, prices are set according to a cosine function around the mean price level observed to test a variety of price levels and, possibly, confuse competitors. After this learning cycle, demand is modeled using own prices and the competitor prices as covariates by means of a variety of regression models (least-squares, ridge regression, Lasso regression, Bayesian ridge regression, stochastic gradient descent regression, and random forest) and the best model, in terms of demand prediction, is selected through cross-validation.

Subsequently, the model of choice is used during an exploitation cycle of variable length: the length is sampled uniformly between 70 and 150, however, if the revenue earned deteriorates too fast, then, immediately a new exploration cycle is initiated. The price is optimized by discretizing the price space and computing the revenue for all prices. When a new exploration cycle starts, so either when the exploitation cycle was finished or because the revenue deteriorated significantly, all historical data is disregarded for the benefit of capturing shifts and shocks in the market most adequately.

6.1.7 Competitor GREEDY

This competitor advocates a particularly simple strategy: set the price as the minimum price observed in the previous time period. To avoid a "race to the bottom" with another competitor, the following facility is implemented: if the minimum price observed in the previous period is lower than the 10% percentile of all the prices observed in the last 30 time periods, then the price for the coming period is set as the maximum of this percentile

and 5 (i.e., if this 10% percentile is smaller than 5, the price is set to 5).

6.1.8 Competitor WLS

The characterizing feature of this competitor is that it aims to maximize own revenue relative to its competitors. More precisely, it attempts to maximize own revenue minus the revenue of the competitor that earns the most revenue. In doing so, it is assumed that demand of competitor k equals $d(p_k, p_{\bar{k}})$, where p_k is the price of competitor k, $p_{\bar{k}}$ is the $(m-1)$-vector with the prices of the competitors of k, and where the notion of time is suppressed. In addition, it is assumed that $d(\cdot,\cdot)$ is independent of permutations in its second argument, i.e., in the vector $p_{\bar{k}}$. Thus, this algorithm aims to obtain the price that maximizes own profit compared to the competitors, that is to solve in each time step,

$$\max_{p_1}\{p_1 d(p_1, p_{\bar{1}}) - \max\{p_k d(p_k, p_{\bar{k}}) \mid k = 2, \ldots, m\}\}. \tag{2}$$

where the competitors are indexed 1 to m and WLS is indexed 1. Note that $p_1 \in p_{\bar{k}}$ for $k \in \{2, \ldots, m\}$.

The demand function is assumed to be of the form $d(x, y) = a + bx + c\sum_{k=1}^{m-1} y_k$ and the parameters a, b, and c are estimated using weighted least squares (hence the name WLS). To capture different time-dependent aspects of demand, various schemes for the weighting of observations are considered and evaluated based on the Median Absolute Error of their historical demand predictions. The best weighting scheme is used in (2) to optimize the price. For this purpose, the price for competitor k in the period to come is predicted based on the median of the historical prices over some window, where the window length is chosen to minimize the Median Absolute Error of historical price predictions.

Finally, for the purpose of exploration, during the first ten periods prices are randomized to guarantee sufficient variance in the observations to estimate the demand models. In addition, when after these ten periods this competitor's own price is constant for three subsequent periods, the prices are randomized for the next period to induce exploration.

References

P. Aghion, M. Paz Espinosa, and B. Jullien. Dynamic duopoly with learning through market experimentation. *Economic Theory*, 3(3):517–539, 1993.

M. D. Alepuz and A. Urbano. Duopoly experimentation: Cournot competition. *Mathematical Social Sciences*, 37 (2):165–188, 1999.

M. Anufriev, D. Kopányi, and J. Tuinstra. Learning cycles in Bertrand competition with differentiated commodities and competing learning rules. *Journal of Economic Dynamics and Control*, 37(12):2562–2581, 2013.

V.F. Araman and R. Caldentey. Dynamic pricing for nonperishable products with demand learning. *Operations Research*, 57(5):1169–1188, 2009.

P. Belleflamme and F. Bloch. Price and quantity experimentation: a synthesis. *International Journal of Industrial Organization*, 19(10):1563–1582, 2001.

D. Bergemann and J. Valimaki. Market experimentation and pricing. Cowles Foundation Discussion Paper 1122, Cowles Foundation for Research in Economics at Yale University, `http://cowles.econ.yale.edu/P/cd/d11a/d1122.pdf`, 1996.

D. Bertsimas and G. Perakis. Dynamic pricing: a learning approach. In *Mathematical and Computational Models for Congestion Charging*, pages 45–79. Springer, New York, 2006.

O. Besbes and A. Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420, 2009.

G. I. Bischi, C. Chiarella, and M. Kopel. The long run outcomes and global dynamics of a duopoly game with misspecified demand functions. *International Game Theory Review*, 06(03):343–379, 2004.

G. I. Bischi, A. K. Naimzada, and L. Sbragia. Oligopoly games with local monopolistic approximation. *Journal of Economic Behavior & Organization*, 62(3):371–388, 2007.

M. Boissier, R. Schlosser, N. Podlesny, S. Serth, M. Bornstein, J. Latt, J. Lindemann, J. Selke, and M. Uflacker. Data-driven repricing strategies in competitive markets: An interactive simulation platform. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 355–357, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4652-8. doi: 10.1145/3109859.3109979. URL `http://doi.acm.org/10.1145/3109859.3109979`.

J. Broder and P. Rusmevichientong. Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980, 2012.

V. Brousseau and A. Kirman. Apparent convergence of learning processes in mis-specified games. In B. Dutta, D. Mookherjee, T. Parthasarathy, T. E. S. Raghavan, D. Ray, and S. Tijs, editors, *Game Theory and Economic Applications*, pages 303–331. Springer-Verlag, Berlin, 1992.

Y. Chen and V.F. Farias. Simple policies for dynamic pricing with imperfect forecasts. *Operations Research*, 61 (3):612–624, 2013.

W. C. Cheung, D. Simchi-Levi, and H. Wang. Dynamic pricing and demand learning with limited price experimentation. Working paper, MIT. Availale at `http://web.mit.edu/wanghe/www/research.html`, 2013.

B. D. Chung, J. Li, T. Yao, C. Kwon, and T. L. Friesz. Demand learning and dynamic pricing under competition in a state-space framework. *Engineering Management, IEEE Transactions on*, 59(2):240–249, 2012.

W. L. Cooper, T. Homem de Mello, and A. J. Kleywegt. Learning and pricing with models that do not explicitly incorporate competition. Working paper, University of Minnesota. Available at `http://www.menet.umn.edu/~billcoop/competitive2014.pdf`, 2014.

R. M. Cyert and M. H. DeGroot. Bayesian analysis and duopoly theory. *Journal of Political Economy*, 78(5): 1168–1184, 1970.

P. Dasgupta and R. Das. Dynamic pricing with limited competitor information in a multi-agent economy. In P. Scheuermann and O. Etzion, editors, *Cooperative Information Systems*, volume 1901 of *Lecture Notes in Computer Science*, pages 299–310. Springer, Berlin, Heidelberg, 2000.

A. V. den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in Operations Research and Management Science*, 20(1):1–18, 2015.

A.V. den Boer and B. Zwart. Simultaneously learning and optimizing using controlled variance pricing. *Management Science*, 60(3):770–783, 2014.

A.V. den Boer and B. Zwart. Dynamic pricing and learning with finite inventories. *Operations Research*, 63(4): 965–978, 2015.

J. M. DiMicco, P. Maes, and A. Greenwald. Learning curve: a simulation-based approach to dynamic pricing. *Electronic Commerce Research*, 3(3-4):245–276, 2003.

M. Dimitrova and E. E. Schlee. Monopoly, competition and information acquisition. *International Journal of Industrial Organization*, 21(10):1623–1642, 2003.

V.F. Farias and B. van Roy. Dynamic pricing with a prior on market response. *Operations Research*, 58(1):16–29, 2010.

M. Fisher, S. Gallino, and J. Li. Competition-based dynamic pricing in online retailing: a methodology validated with field experiments. *Management Science*, Articles in advance:1–19, 2017.

A. Fishman and N. Gandal. Experimentation and learning with networks effects. *Economics Letters*, 44(1-2): 103–108, 1994.

T. L. Friesz, C. Kwon, T. I. Kim, L. Fan, and T. Yao. Competitive robust dynamic pricing in continuous time with fixed inventories. arXiv:1208.4374 [math.OC], 2012.

A. G. Gallego. Oligopoly experimentation of learning with simulated markets. *Journal of Economic Behavior & Organization*, 35(3):333–355, 1998.

A. R. Greenwald and J. O. Kephart. Shopbots and pricebots. In A. Moukas, C. Sierra, and F. Ygge, editors, *Agent Mediated Electronic Commerce II*, pages 1–23. Springer, 1999.

J. E. Harrington. Experimentation and learning in a differentiated-products duopoly. *Journal of Economic Theory*, 66(1):275–288, 1995.

J. M. Harrison, N. B. Keskin, and A. Zeevi. Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science*, 58(3):570–586, 2012.

K. Isler and H. Imhof. A game theoretic model for airline revenue management and competitive pricing. *Journal of Revenue and Pricing Management*, 7(4):384–396, 2008.

K. Johnson Ferreira, D. Simchi-Levi, and H. Wang. Online network revenue management using Thompson sampling. Working paper, 2016. URL `https://ssrn.com/abstract=2588730`.

J. Jumadinova and P. Dasgupta. Firefly-inspired synchronization for improved dynamic pricing in online markets. In *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, pages 403–412. IEEE, 2008.

J. Jumadinova and P. Dasgupta. Multi-attribute regret-based dynamic pricing. In W. Ketter, H. La Poutré, N. Sadeh, O. Shehory, and W. Walsh, editors, *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, volume 44 of *Lecture Notes in Business Information Processing*, pages 73–87. Springer Berlin Heidelberg, 2010.

G. Keller and S. Rady. Price dispersion and learning in a dynamic differentiated-goods duopoly. *The RAND Journal of Economics*, 34(1):138–165, 2003.

N. B. Keskin and Assaf Zeevi. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research*, 62(5):1142–1167, 2014.

A. Kirman. On mistaken beliefs and resultant equilibria. In R. Frydman and E. S. Phelps, editors, *Individual forecasting and aggregate outcomes*, pages 147–166. Cambridge University Press, New York, 1983.

A. P. Kirman. Learning by firms about demand conditions. In R. H. Day and T. Graves, editors, *Adaptive Economics*, pages 137–156. Academic Press, New York, 1975.

A. P. Kirman. Learning in oligopoly: Theory, simulation, and experimental evidence. In A. P. Kirman and M. Salmon, editors, *Learning and Rationality in Economics*, pages 127–178. Basil Blackwell, Cambridge, MA, 1995.

V. Könönen. Dynamic pricing based on asymmetric multiagent reinforcement learning. *International Journal of Intelligent Systems*, 21(1):73–98, 2006.

E. Kutschinski, T. Uthmann, and D. Polani. Learning competitive pricing strategies by multi-agent reinforcement learning. *Journal of Economic Dynamics and Control*, 27(11-12):2207–2218, 2003.

C. Kwon, T. L. Friesz, R. Mookherjee, T. Yao, and B. Feng. Non-cooperative competition among revenue maximizing service providers with demand learning. *European Journal of Operational Research*, 197(3):981–996, 2009.

J. Li, T. Yao, and H. Gao. A revenue maximizing strategy based on Bayesian analysis of demand dynamics. In D. A. Field and T. J. Peters, editors, *Proceedings of the 2009 SIAM Conference on Mathematics for Industry*, pages 174–181, Philadelphia, PA, 2010. Society for Industrial and Applied Mathematics.

L. J. Mirman, L. Samuelson, and A. Urbano. Duopoly signal jamming. *Economic Theory*, 3(1):129–149, 1993.

G. Perakis and A. Sood. Competitive multi-period pricing for perishable products: a robust optimization approach. *Mathematical Programming*, 107(1):295–335, 2006.

S. Ramezani, P. A. N. Bosman, and H. La Poutre. Adaptive strategies for dynamic pricing agents. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 323–328. IEEE, 2011.

S. Rassenti, S. S. Reynolds, V. L. Smith, and F. Szidarovszky. Adaptation and convergence of behavior in repeated experimental Cournot games. *Journal of Economic Behavior & Organization*, 41(2):117–146, 2000.

M. P. Schinkel, J. Tuinstra, and D. Vermeulen. Convergence of Bayesian learning to general equilibrium in mis-specified models. *Journal of Mathematical Economics*, 38(4):483–508, 2002.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

G. Tesauro and J. O. Kephart. Pricing in agent economies using multi-agent Q-learning. *Autonomous Agents and Multi-Agent Systems*, 5(1):289–304, 2002.

J. Tuinstra. A price adjustment process in a model of monopolistic competition. *International Game Theory Review*, 6(3):417–442, 2004.