

**Dynamic Pricing and Learning with Competition:
Insights from the Dynamic Pricing Challenge
at the 2017 INFORMS RM & Pricing Conference¹²**

August 28, 2018

Ruben van de Geer^{a,I}, Arnoud V. den Boer^{b,I}, Christopher Bayliss^{c,II}, Christine Currie^{d,II}, Andria Ellina^{d,II}, Malte Esders^{e,II}, Alwin Haensel^{f,II}, Xiao Lei^{g,II}, Kyle D.S. Maclean^{h,II}, Antonio Martinez-Sykora^{d,II}, Asbjørn Nilsen Riseth^{i,II}, Fredrik Ødegaard^{h,II}, and Simos Zachariades^{d,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*IN3 - Computer Science Dept., Universitat Oberta de Catalunya, Spain;* ^d*Mathematical Sciences, University of Southampton, SO17 1BJ, UK;* ^e*Faculty IV Electrical Engineering and Computer Science, Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin;*

^f*Haensel AMS, Advanced Mathematical Solutions, Ritterstr. 2a, 10969 Berlin, Germany;* ^g*Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027;* ^h*Ivey Business School, Western University, 1255 Western Road, London, ON N6G 0N1, Canada;* ⁱ*Mathematical Institute, University of Oxford, OX2 6GG, UK.*

^I*dynamic pricing challenge organizer, lead author;*

^{II}*dynamic pricing challenge participant, co-author.*

Ruben van de Geer is a PhD student at the Department of Mathematics of the Vrije Universiteit Amsterdam. Ruben's research is on pricing and on machine learning applied to operations research.

Arnoud V. den Boer is assistant professor at the mathematics department of the University of Amsterdam, and also affiliated to the Amsterdam Business School. His research centers around dynamic pricing under incomplete information.

Christopher Bayliss is a Postdoctoral researcher within the ICSO research group at the Open University of Catalonia where he is conducting research within the area of logistics and smart cities. His research interests include: Optimization under uncertainty, simulation-optimization, airline scheduling, packing problems, dynamic pricing and problems within logistics.

Christine Currie is an Associate Professor of Operational Research at the University Of Southampton, UK. She has worked on revenue management projects with organizations from a range of sectors and has particular expertise in dynamic pricing.

Andria Ellina is a third year PhD student at the University of Southampton. Andria's research is on exploration and exploitation methodologies with a particular interest in their application in revenue management.

¹Chris 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.

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

Malte Esders worked as a Machine Learning engineer at Haensel AMS in Berlin. He is now a PhD candidate in the Machine Learning group at Technical University of Berlin.

Alwin is the Founder and Managing Director of the Technology & Data Analytics company Haensel AMS - Advanced Mathematical Solutions. He holds a PhD in Applied Mathematics and studied in London, Berlin and Amsterdam. His main fields of interest are the modeling of customer purchasing behavior, data prediction and optimization under uncertainty.

Xiao Lei is a PhD student of Industrial Engineering and Operations Research at Columbia University. His research interest are in revenue management and business analytics.

Kyle D.S. Maclean is currently an assistant professor at the Ivey Business School at Western University. He obtained his PhD in Management Science from Western University.

Dr. Antonio Martinez-Sykora is a Lecturer at the Southampton Business School. He finished his PhD back in 2013 and he has been working on combinatorial optimization problems over 10 years, especially on cutting and packing, routing/logistics and revenue management problems.

Asbjørn Nilsen Riseth is a Norwegian mathematician studying problems in industrial mathematics. He focuses on decision making for random systems, differential equations, and scientific software development.

Fredrik Ødegaard is an Associate Professor in Management Science at Ivey Business School, with cross-appointment at the Department of Applied Mathematics, Western University. Dr. Ødegaard's general research areas are revenue management and health care. Dr. Ødegaard currently serves on the Editorial Board of the Journal of Revenue and Pricing Management.

Simos Zachariades is a PhD student in Mathematics at the University of Southampton. His research is focused on robust optimisation methods in network revenue management.

**Dynamic Pricing and Learning with Competition:
Insights from the Dynamic Pricing Challenge
at the 2017 INFORMS RM & Pricing Conference**

August 28, 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 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 pricing 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.

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

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:

- The relative performance of the pricing and learning algorithms that we consider, 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 all the others in all settings.
- The relative performance of the pricing and learning algorithms that we consider, 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 more 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, we present and discuss the results, respectively. In Section 5, we conclude with managerial insights and practical implications of our study for the industry.

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:

- 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.

In this experiment there were eight competitors, which means that 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 each of the 5000 simulations, the oligopoly competition and each of the duopoly competitions consists of 1000 discrete time periods. This means that for each $t \in \{1, \dots, 1000\}$, each competitor posts a price for $t + 1$ and, subsequently, we generate sales quantities for period $t + 1$ from the undisclosed demand mechanism, and all competitors earn revenue accordingly. This iterative process is illustrated for the oligopoly competition in Figure 1 (for a duopoly competition, the scheme is the same except for the number of competitors, which is then two).

We measure the performance of all competitors and determine the winner of the pricing challenge as follows. In each simulation, we compute for each competitor 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 28 duopoly competitions, and computing the competitors’ revenue shares accordingly. More precisely, for simulation

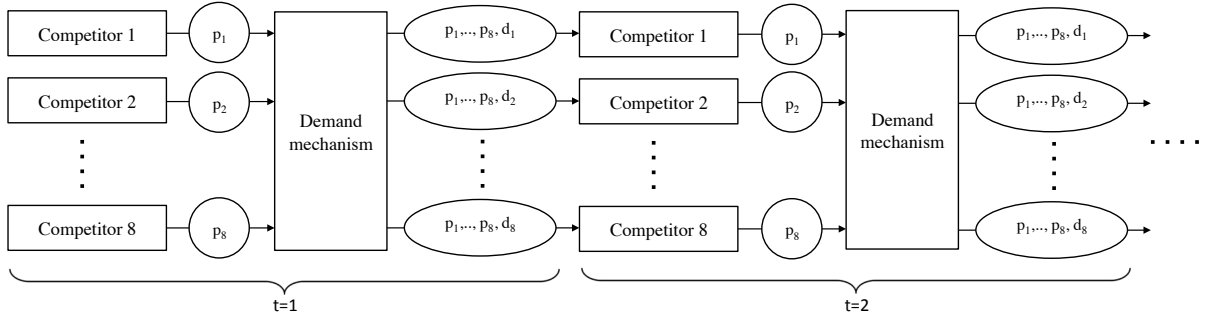


Figure 1: Overview of the first two time periods of the simulation of an oligopoly competition; p_i and d_i denote the price and sales quantity, respectively, of competitor $i \in \{1, \dots, 8\}$.

$i \in \{1, \dots, 5000\}$, let x_{ij} be the revenue earned by competitor $j \in \{1, \dots, 8\}$ in the oligopoly competition, and let y_{ijk} be the revenue earned by competitor $j \in \{1, \dots, 8\}$ in the duopoly competition versus competitor $k \in \{1, \dots, 8\}$. We define the revenue share of competitor j in simulation i as follows:

$$\frac{1}{2}(\bar{x}_{ij} + \bar{y}_{ij}), \text{ where } \bar{x}_{ij} := \frac{x_{ij}}{\sum_{k=1}^8 x_{ik}} \text{ and } \bar{y}_{ij} := \frac{\sum_{k=1}^8 y_{ijk}}{\sum_{u=1}^8 \sum_{k=1}^8 y_{iuk}}. \quad (1)$$

Here, we set $y_{ijj} = 0$ for all $i \in \{1, \dots, 5000\}$ and $j \in \{1, \dots, 8\}$. In (1), \bar{x}_{ij} and \bar{y}_{ij} represent the oligopoly and duopoly revenue share, respectively, of competitor $j \in \{1, \dots, 8\}$ in simulation $i \in \{1, \dots, 5000\}$. Note that, as such, a competitor's own duopoly revenue share depends on the revenue earned in competitions between other competitors. This means 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 $j \in \{1, \dots, 8\}$ 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

Table 1 summarizes the pricing and learning policies all the competing algorithms. In the appendix, we provide more detailed descriptions of the algorithms. 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, and
- 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. Re-

garding 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-GRUD [c,1,81]	ϵ -greedy multi-armed bandit algorithm with ten arms pertaining to the prices 10, 20, ..., 100.	With probability $\epsilon = 0.2$ 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], ..., (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 our demand mechanism 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 have to carefully manage the complexity of the demand mechanism to allow for evaluation, interpretation, and further analysis following the contest. To this end, we assume that the arrival process and demand mechanism are time-independent within a simulation. The customer population consists of three segments, namely loyal customers, shoppers, and scientists, who all have their own parameterized demand functions, as we describe 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 we construct 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

In simulation $i \in \{1, \dots, 5000\}$, we have the following arrival process and customer population composition. Customers arrive according to a Poisson process, with mean arrivals per time period equal to λ_i , 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 of F . Let θ_i^{sho} , θ_i^{loy} , and θ_i^{sci} , be the share of shoppers, loyals, and scientists, respectively, where $\theta_i^{\text{sho}} + \theta_i^{\text{loy}} + \theta_i^{\text{sci}} = 1$ and $\theta_i^{\text{sho}}, \theta_i^{\text{loy}}, \theta_i^{\text{sci}} \in (0, 1)$. The scientist segment consists of two subsegments, namely PhDs and professors, with respective shares of γ_i^{phd} and γ_i^{prof} , where $\gamma_i^{\text{phd}} + \gamma_i^{\text{prof}} = 1$ and $\gamma_i^{\text{phd}}, \gamma_i^{\text{prof}} \in (0, 1)$.

We summarize the sampling of arrivals in the code block ‘‘Arrival Process’’. Here, n is

Arrival Process

```

for  $i \in \{1, \dots, 5000\}$  do
  Sample arrival rate  $\lambda_i \sim U(50, 150)$ 
  Sample segment shares  $\theta_i^{\text{sho}}, \theta_i^{\text{loy}}, \theta_i^{\text{sci}}$ 
  Sample subsegment shares  $\gamma_i^{\text{phd}}, \gamma_i^{\text{prof}}$ 
  for  $t \in \{1, \dots, 1000\}$  do
    Sample arrivals  $n \sim \text{Poisson}(\lambda_i)$ 
    Sample segment arrivals  $n^{\text{sho}}, n^{\text{loy}}, n^{\text{sci}} \sim \text{Multinom}(n, (\theta_i^{\text{sho}}, \theta_i^{\text{loy}}, \theta_i^{\text{sci}}))$ 
    Sample subsegment arrivals  $n^{\text{phd}}, n^{\text{prof}} \sim \text{Multinom}(n^{\text{sci}}, (\gamma_i^{\text{phd}}, \gamma_i^{\text{prof}}))$ 
  end for
end for

```

the number of arriving customers in a given period, and consists of n^{sho} shoppers, n^{loy} loyal customers, and n^{sci} scientists, i.e., $n = n^{\text{sho}} + n^{\text{loy}} + n^{\text{sci}}$. The scientist segment consists of n^{phd} PhDs and n^{prof} professors, i.e., $n^{\text{sci}} = n^{\text{phd}} + n^{\text{prof}}$.

2.3.2 Demand of shoppers and loyal customers

In simulation $i \in \{1, \dots, 5000\}$, the willingness-to-pay (WTP) of shoppers is exponentially distributed with mean β_i^{sho} , where $\beta_i^{\text{sho}} \sim U(5, 15)$. This means that, in each time period in simulation i , we sample a WTP from the exponential distribution with mean β_i^{sho} for each

arriving shopper. Each shopper whose WTP exceeds the lowest price offered in the market, buys from the competitor that offers the lowest price, and otherwise leaves without buying anything. Ties are broken randomly.

The WTP of the loyal customers in simulation $i \in \{1, \dots, 5000\}$ is exponentially distributed as well, but with mean β_i^{loy} , where $\beta_i^{\text{loy}} = u \cdot \beta_i^{\text{sho}}$ and $u \sim U(1.5, 2.0)$. This makes the loyal customers relatively price insensitive compared to shoppers, as one would expect from loyal customers. In each time period, for each arriving loyal customer, we sample a WTP from the exponential distribution with mean β_i^{loy} , and assign this customer randomly to one of the competitors. This assignment determines to which competitor each customer is loyal to. We compare the WTP of each arriving loyal customer with the price set by the competitor that this customer is loyal to: they purchase if their WTP exceeds the price offered, and otherwise leave without making a purchase.

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 two components: professors and PhDs. This implies that the professors and PhDs both choose according to a logit model, but with their own respective parameters, which are set as follows.

Consider simulation $i \in \{1, \dots, 5000\}$ and let n be the number of competitors in the market, i.e., $n = 2$ in any duopoly competitions and $n = 8$ in the oligopoly competitions. In a given period, suppose the posted prices in the market are equal to $p = (p_1, \dots, p_n)$. We define the probability that an arriving PhD purchases from competitor $k \in \{1, \dots, n\}$ as follows:

$$q_{ki}^{\text{phd}}(p) = \frac{\exp\left(\alpha_i^{\text{phd}} - \beta_{ni}^{\text{phd}} \cdot p_k\right)}{1 + \sum_{j=1}^n \exp\left(\alpha_i^{\text{phd}} - \beta_{ni}^{\text{phd}} \cdot p_j\right)},$$

and we define $q_{ki}^{\text{prof}}(p)$ for professors in similar fashion.

We sample the parameters α_i^{phd} , β_{ni}^{phd} , α_i^{prof} , and β_{ni}^{prof} to ensure that, if you optimize the price for each of the segments separately, then no unrealistically large differences in optimal prices occur. In doing so, we prevent that some customers are willing to pay a price that is orders of magnitudes larger than the price that other customers are willing to pay. To this end, we set α_i^{phd} equal to β_i^{sho} , and set β_{ni}^{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,

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

where W is the Lambert function, i.e., $W(xe^x) = x$, which is uniquely defined in this case as $ne^{\alpha_i^{\text{phd}} - 1} > 0$ (Akçay et al., 2010). Here, p_i^{phd} is the price that maximizes revenue if the market

consists solely of PhDs (see Theorem 6 in Akcay et al. (2010)):

$$\arg \max_{p \in \mathbb{R}_+^n} \sum_{k=1}^n p_k q_{ki}^{\text{phd}}(p) = \mathbb{1}_n p_i^{\text{phd}},$$

where $\mathbb{1}_n p_i^{\text{phd}}$ is understood to be an n -vector with each element equal to p_i^{phd} . By construction, p_i^{phd} is within 50% of the average WTP of shoppers. In similar fashion, we set

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

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 each simulation $i \in \{1, \dots, 5000\}$ and each time period $t \in \{1, \dots, 5000\}$ of each duopoly and oligopoly, we see on average $\lambda_i \cdot \theta_i^{\text{sho}}$ 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 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 quantities, 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.

The share of loyal customers in the market, which is $\frac{1}{3}$ on average, is independent of the number of competitors in the market. This means that in a duopoly, an arriving customer is a loyal customer that is loyal to a specific competitor with probability $\frac{1}{2} \cdot \frac{1}{3}$, whereas in the oligopoly, this probability equals $\frac{1}{8} \cdot \frac{1}{3}$. Therefore, by construction, we anticipate the market to be more competitive in the oligopoly setting than in the duopoly case, as one would expect from economic theory.

3 Results

In Section 3.1, we present a summary of the overall results. In Section 3.2 and 3.3, the results of the oligopoly and duopoly competitions, respectively, are considered.

3.1 Overall Results

In Figure 2, we provide boxplots of the revenue shares for the oligopoly, the duopoly, and the overall competition. The boxplots in the left, middle, and right panel are based on \bar{x}_{ij} , \bar{y}_{ij} ,

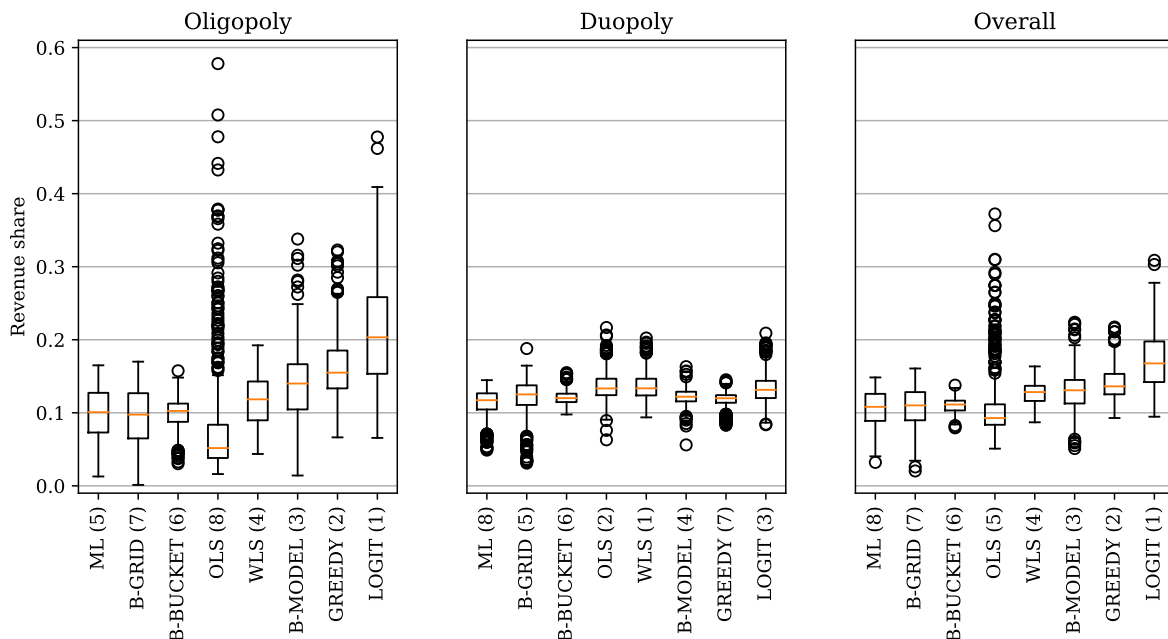


Figure 2: Boxplots of the revenue shares for the oligopoly competition, the duopoly competitions, and the overall competition. In parenthesis after the names of the algorithms are the rankings per competition part (e.g., WLS was the winner of the duopoly part). Only the first 500 of 5000 simulations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

and $\frac{1}{2}(\bar{x}_{ij} + \bar{y}_{ij})$, respectively (all defined in equation (1)). Figure 2 reveals that LOGIT is the winner of the competition and that its success is primarily due to its superior performance in the oligopoly competitions (left panel). Overall, the differences in performance in the oligopoly are substantial (left panel), whereas in the duopoly competitions (middle panel) it proves to be more difficult to outperform one another. Nevertheless, we observe remarkable differences in relative performance across the oligopoly and duopoly competitions. For example, OLS nearly earns the highest mean revenue share in the duopoly competitions (finishing second), while its performance in the oligopoly competition is on average the worst amongst all competitors. Note however that, despite its 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, as they are in the bottom half in both the duopoly and oligopoly 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 of Figure 2. We analyze how the competitors differ in terms of realized sales quantities and prices posted, and analyze how their performance varies as the market composition, i.e., the shares of segments, differs across simulations.

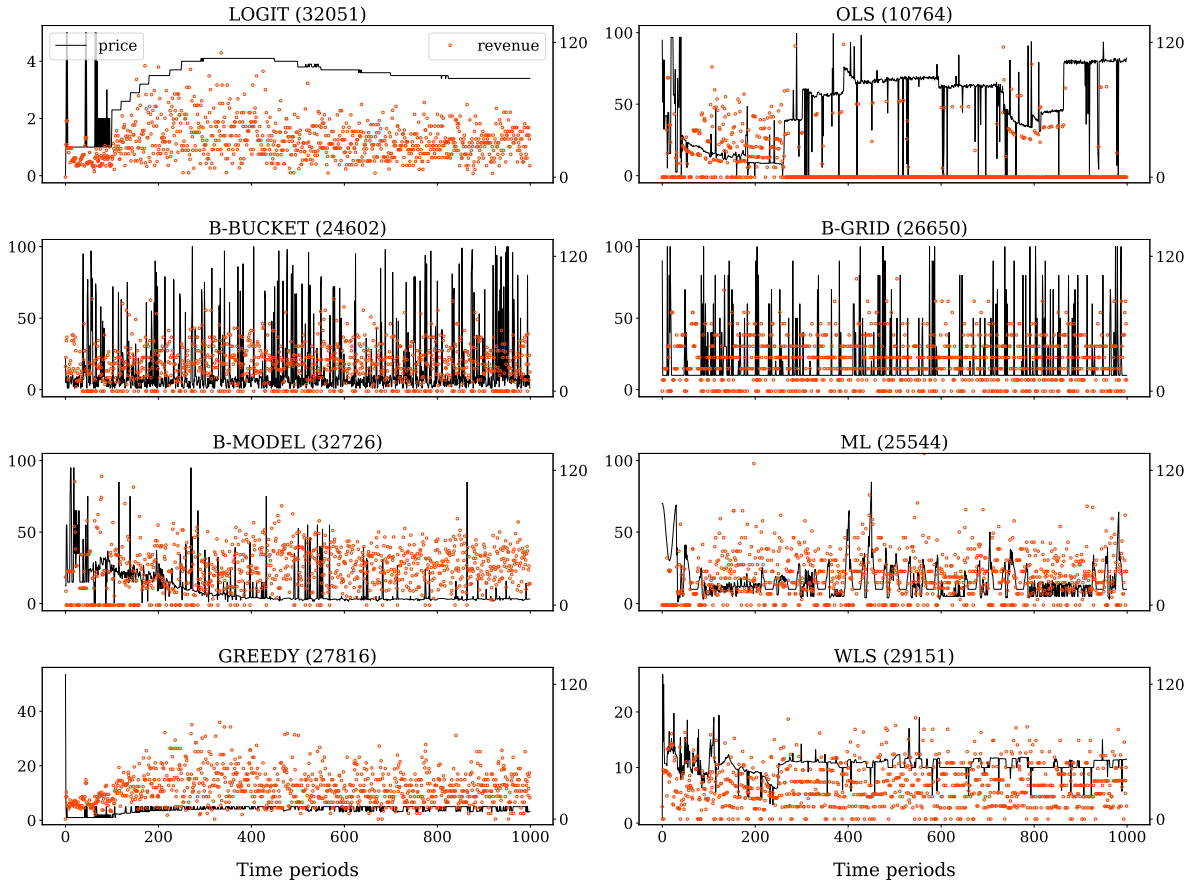


Figure 3: Realizations of prices and revenue for simulation 4955. The solid line represents the prices and pertains to the left axis. The red dots represent the revenue earned and pertain to the right axis. In parenthesis is the total accumulated revenue. The segment shares are equal to $\theta^{\text{sho}}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

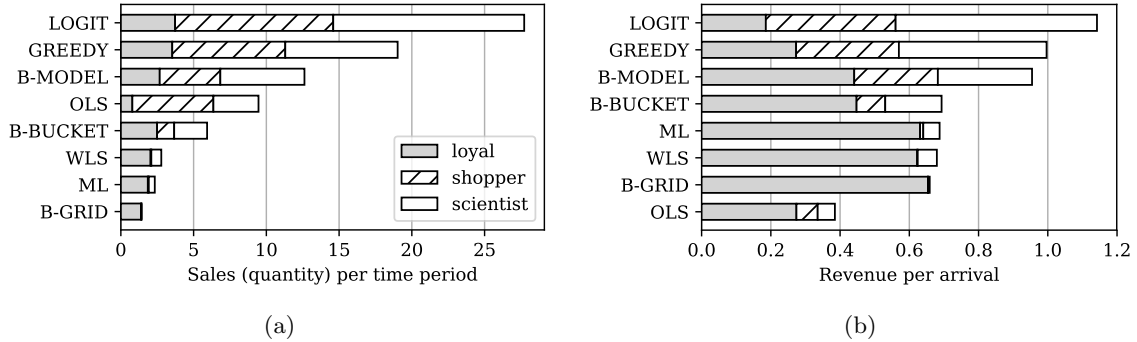


Figure 4: For the oligopoly competition (a) mean sales per time period and (b) mean revenue per arriving customer, split out over the three customer segments.

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 simulation in Figure 3 is demonstrative for most of the scenarios that we visually inspected. In particular, 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, without eventually converging to a competitive price range. This is arguably due to the fact that these algorithms engage in active price exploration with a 20% probability in each period. OLS initially seems to converge to a competitive price, but from around time period 300 onwards it fluctuates around a relatively high price level, earning hardly any revenue. We observe that ML initiates many exploration cycles around a cosine function, which affects its performance negatively in this stationary environment.

To show how the competitors differ in sales per customer segment, Figure 4(a) illustrates the mean sales per time period per segment for each competitor (note that sales is understood to be a quantity here and throughout). From this figure, it follows that the three best-performing algorithms in the oligopoly competitions, namely LOGIT, GREEDY, and B-MODEL (see the left panel of Figure 2), are also the ones that 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 is likely 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 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 attract 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).

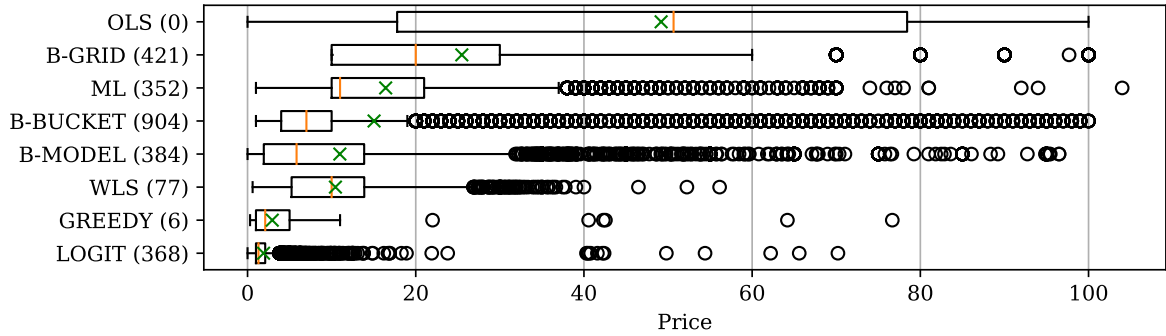


Figure 5: Boxplots of the prices posted in the oligopoly in 5000 randomly chosen time periods. The bars and crosses denote the median and mean, respectively. In brackets are the corresponding number of outliers.

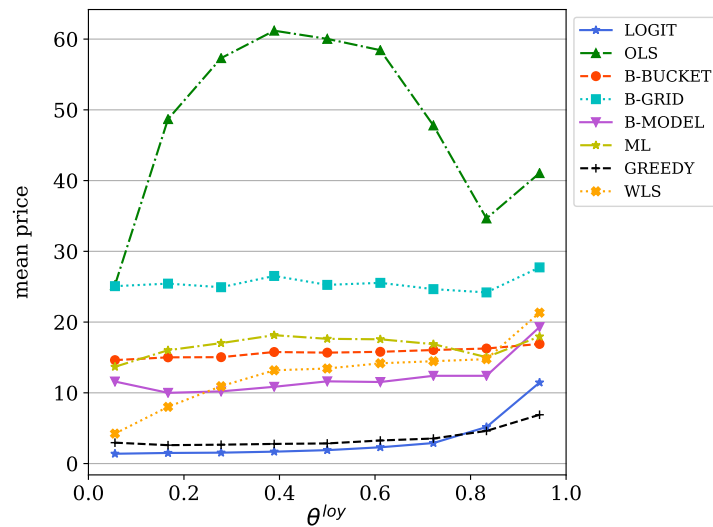


Figure 6: Mean prices for various levels of the share of loyal customers θ^{loy} .

Figure 4(b) illustrates the mean revenue per arriving customer for each competitor (e.g., LOGIT makes on average about 0.20 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: the top performers (LOGIT, GREEDY, and B-MODEL) make the majority of their income from shoppers and scientists, while the other algorithms earn relatively much from the loyal customer segment. These observations concur with Figure 5, which contains boxplots of the prices posted by all competitors.

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. GREEDY does not engage in much price experimentation, as expect from its construction. The other boxplots indicate that the price distributions all have a heavy right tail; these competitors engage in exploration coincidentally, while pricing around a relatively small interval for most of the time.

Figure 6 illustrates the mean prices for different shares of loyal customers. This is insightful, since when the share of loyal customers increases from zero to one, the market moves from

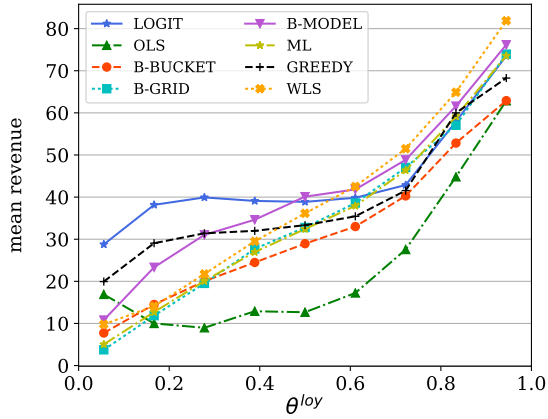


Figure 7: Mean revenue per time period for various levels of θ^{loy} .

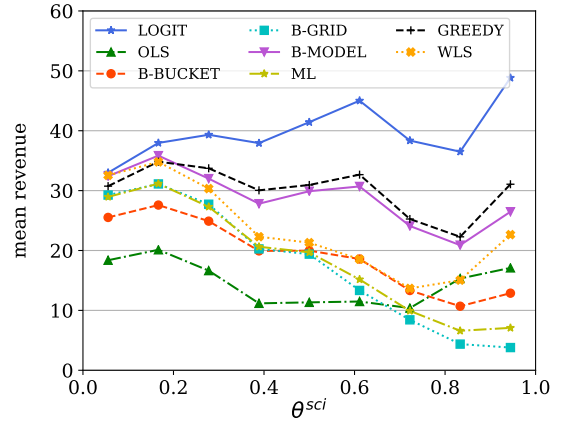


Figure 8: Mean revenue per time period for various levels of θ^{sci} .

a very competitive market to a market in which every competitor is essentially a monopolist. One theoretically expects that competitors post higher prices once their pricing power increases. However, from the figure, we 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 increasing prices when pricing power increases.

Figure 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 we attribute 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).

3.3 Duopoly Competition

From the middle panel of Figure 2 it follows that, in the duopoly competitions, the differences in performance are smaller than in the oligopoly competition (left panel in Figure 2). There are two reasons why this is the case. First of all, the share of loyal customers is relatively large in the duopoly (see the discussion at the end of Section 2.3.4). This means that competitors have in general more pricing power, whilst only having to consider one other competitor, which makes it easier to earn revenue and more difficult 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 duopoly competitions. Nonetheless,

⁵The figure for the shopper segment is omitted to save space since it is very similar to Figure 7, except for that the revenues decrease in the shopper share.

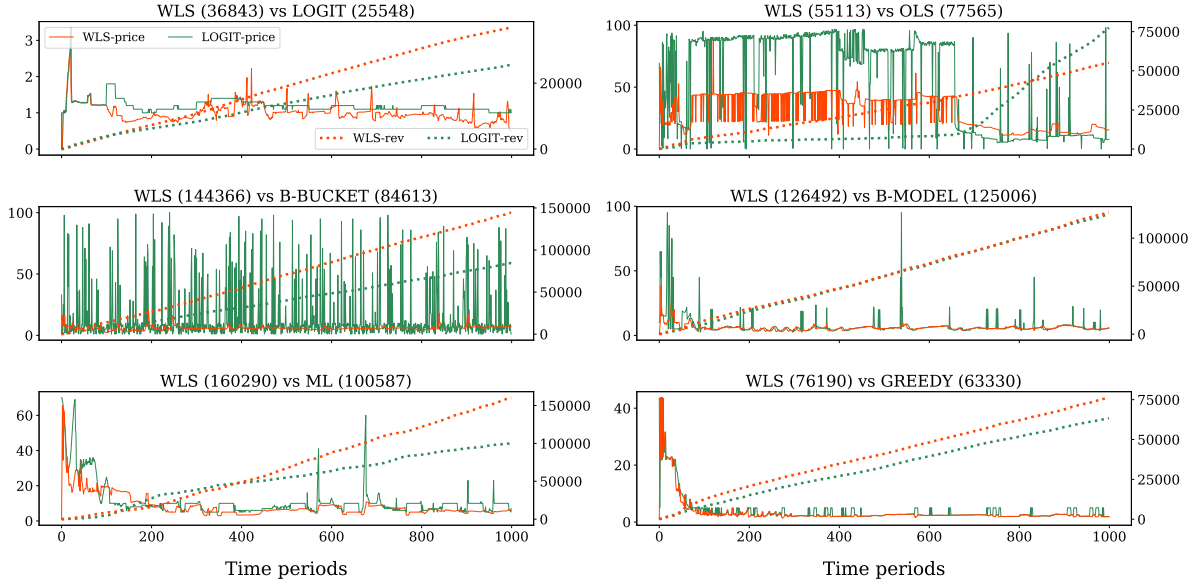


Figure 9: Realizations of prices and revenue for simulation 2285. The red and green solid lines represent the prices of WLS and the corresponding opponent, respectively, and pertain to the left axis. The red and green dotted lines represent the cumulative revenue earned by WLS and the corresponding opponent,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

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 the top performer of the duopoly part, namely WLS, was involved.⁶ The figure reveals that in the duopoly in which WLS and LOGIT compete, prices and revenue are both relatively low (this is not only the case in this example, but is a structural property, which we show 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 (see Table 1); these are the bumps that are visible in the bottom-right panel of Figure 9. As was the case in the oligopoly, OLS is not able to set a competitive price level consistently.

In Table 2, we provide the mean revenue per time period for the duopoly competitions. The last column contains the row-wise averages, which indicate how much revenue each algorithm makes on average. Similarly, the final row contains column-wise averages, which indicate how much revenue other algorithms were able to make against the corresponding algorithm (e.g., LOGIT makes on average 234 per time period, while the other competitors make on average 218 when competing with LOGIT). Table 2 confirms the earlier observation that the market between WLS and LOGIT is on average the smallest in terms of revenue ($153 + 96 = 249$ revenue per time period). In general, 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 revenue with an average revenue per time period of 240. This is remarkable, as WLS performs poorly in the oligopoly.

Table 2 also reveals that there is no competitor that loses against all other competitors.

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

	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 2: 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.

Even the worst performer (based on Figure 2), namely ML, is able to defeat other competitors, namely, LOGIT and GREEDY, which is remarkable as both perform very well overall. Based on Table 2, we observe that GREEDY has average performance—never earning much more (or much less) than its opponents. This is expected, since it simply follows its opponent’s actions, without exploiting any weaknesses. The other top performer in the duopoly part, namely OLS, only beats B-GRID and ML according to Table 2. Nonetheless, we observe that OLS on average earns the same amount of revenue, namely 240, as WLS.

All in all, Table 2 reveals that each competitor’s performance is very much dependent on the opponent’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.

To gain insight in the pricing levels of the competitors in the duopoly competitions, Figure 10 provides a heatmap of the mean prices and the coefficient of variation (the ratio of the standard deviation to the mean). 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 sets the lowest mean price and that OLS, which performed equally well according to Figure 2, set substantially higher prices than WLS. Thus, OLS is able to be 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 revenue. This can 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 revenue. 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 Figure 3 and 9.

Heatmap of mean prices and price dispersion

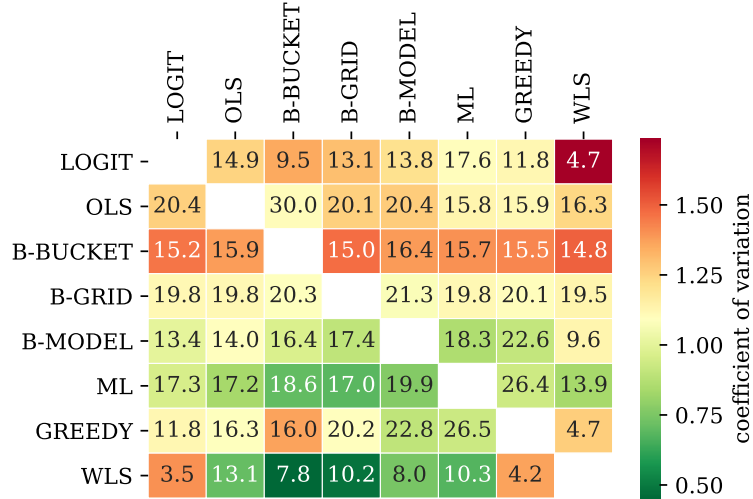


Figure 10: The number in each cell indicates the mean price that the algorithm in the corresponding row posted against the algorithm in the corresponding column. The color of each cell pertains to the coefficient of variation (i.e., the ratio of the standard deviation to the mean) and corresponds to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

4 Discussion

In terms of overall performance, LOGIT was the most effective algorithm in this competition—it earns most revenue in the oligopoly part and is also competitive in the duopoly part, where it finishes in third place, just 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 leads to the highest revenue per arriving scientist (see Figure 4(b) and 8). However, the primary reason for its success is undoubtedly the fact that LOGIT is able to identify and exploit the revenue potential of the shopper segment by setting low prices. This strategy negatively affects performance when pricing power increases (see Figure 7), but proves beneficial overall.

Arguably, if GREEDY would not have had arrangements in place to prevent a “race to the bottom”, LOGIT’s revenue would have been substantially lower. This indicates that LOGIT’s low-pricing strategy does 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 better as it is less reliant on the price-sensitive segments by selling more to loyal customers, whilst earning substantial revenue (Figure 4).

Regarding the two algorithms that rely on linear demand models, namely 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 the price-demand relationship. WLS is designed to maximize own revenue minus its competitor’s revenue (see Table 1), which causes WLS to be very difficult to earn revenue against (see Table 2). On the other hand, OLS ignores competition altogether, which works surprisingly well in the

duopoly competitions, and results in OLS posting much higher prices than WLS according to Figure 10 (but generating roughly the same amount of revenue). In the oligopoly competitions, OLS is the worst-performing algorithm (Figure 2), which indicates that ignoring competition is increasingly harmful as competition increases (the relative performance of B-GRID, which also ignored competition, is also worse in the oligopoly competition than in the duopoly competition). The performance of WLS in the oligopoly is also poor, compared to its performance in the duopoly competitions. This shows that is ineffective to explicitly take into account all competitors' anticipated revenues.

Two of the bandit models, namely B-GRID and B-BUCKET, perform poorly due to a defect in their designs, which only allows prices in very crudely discretized price sets and prevents 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.

The other bandit model, namely B-MODEL, is competitive and finishes third overall. It is 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 relies on machine learning methods to model the demand characteristics, and is 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 persists in engaging in exploration cycles, which hinders its performance in the stationary environment that we designed. This confirms the notion that one should only experiment if the anticipated long-term revenue of doing so outweighs the short-term cost.

Certainly the most simple strategy, namely GREEDY's ad-hoc approach of matching the lowest price in the market, turns out to be very effective in the oligopoly competition. Generally, it follows whoever set the lowest price, and is thereby able to generate substantial revenue—especially from scientists and shoppers. The arrangements that GREEDY put in place to prevent downward price spirals are initiated frequently (Figure 9), which could otherwise have led to even lower prices, and presumably deteriorating revenues. In the duopoly competitions, no competitor outperforms GREEDY significantly, however, GREEDY is also 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, which we analyzed in a simulated market environment with competition. As such, this paper presents 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 allows us to consider market dynamics that are not analytically tractable or can not be empirically analyzed due to practical complications.

Our analysis reveals a number of interesting insights, both from a practical and scientific point of view. First of all, we show that the relative performance of pricing and learning algorithms varies substantially across oligopoly and duopoly markets and across different market dynamics. This 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 other algorithms—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 follows the lowest-priced competitor in a tit-for-tat fashion proves very difficult to outperform. Especially in oligopolistic markets, it is able to generate substantial revenue from price-sensitive customers, whilst 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 other competitors, whilst other competitors earn their 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 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. Another interesting extension is that of a market in which competitors offer assortments of products, and can change both prices and the composition of their assortments over time.

References

- Aghion, P., Espinosa, M. P. and Jullien, B. (1993), ‘Dynamic duopoly with learning through market experimentation’, *Economic Theory* **3**(3), 517–539.
- Akcay, Y., Natarajan, H. P. and Xu, S. H. (2010), ‘Joint dynamic pricing of multiple perishable products under consumer choice’, *Management Science* **56**(8), 1345–1361.
- Alepuz, M. D. and Urbano, A. (1999), ‘Duopoly experimentation: Cournot competition’, *Mathematical Social Sciences* **37**(2), 165–188.
- Anufriev, M., Kopányi, D. and Tuinstra, J. (2013), ‘Learning cycles in Bertrand competition with differentiated commodities and competing learning rules’, *Journal of Economic Dynamics and Control* **37**(12), 2562–2581.
- Araman, V. and Caldentey, R. (2009), ‘Dynamic pricing for nonperishable products with demand learning’, *Operations Research* **57**(5), 1169–1188.
- Belleflamme, P. and Bloch, F. (2001), ‘Price and quantity experimentation: a synthesis’, *International Journal of Industrial Organization* **19**(10), 1563–1582.

- Bergemann, D. and Valimaki, J. (1996), ‘Market experimentation and pricing’, Cowles Foundation Discussion Paper 1122, Cowles Foundation for Research in Economics at Yale University, <https://ideas.repec.org/p/cwl/cwldpp/1122.html>, accessed 5 June 2018.
- Bertsimas, D. and Perakis, G. (2006), Dynamic pricing: a learning approach, in ‘Mathematical and Computational Models for Congestion Charging’, Springer, New York, pp. 45–79.
- Besbes, O. and Zeevi, A. (2009), ‘Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms’, *Operations Research* **57**(6), 1407–1420.
- Bischi, G. I., Chiarella, C. and Kopel, M. (2004), ‘The long run outcomes and global dynamics of a duopoly game with misspecified demand functions’, *International Game Theory Review* **06**(03), 343–379.
- Bischi, G. I., Naimzada, A. K. and Sbragia, L. (2007), ‘Oligopoly games with local monopolistic approximation’, *Journal of Economic Behavior & Organization* **62**(3), 371–388.
- Boissier, M., Schlosser, R., Podlesny, N., Serth, S., Bornstein, M., Latt, J., Lindemann, J., Selke, J. and Uflacker, M. (2017), Data-driven repricing strategies in competitive markets: An interactive simulation platform, in ‘Proceedings of the Eleventh ACM Conference on Recommender Systems’, RecSys ’17, pp. 355–357.
- Broder, J. and Rusmevichientong, P. (2012), ‘Dynamic pricing under a general parametric choice model’, *Operations Research* **60**(4), 965–980.
- Brousseau, V. and Kirman, A. (1992), 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, eds, ‘Game Theory and Economic Applications’, Springer-Verlag, Berlin, pp. 303–331.
- Chen, Y. and Farias, V. (2013), ‘Simple policies for dynamic pricing with imperfect forecasts’, *Operations Research* **61**(3), 612–624.
- Cheung, W. C., Simchi-Levi, D. and Wang, H. (2013), ‘Dynamic pricing and demand learning with limited price experimentation’, Working paper, <http://web.mit.edu/wanghe/www/research.html>, accessed 2 February 2018.
- Chung, B. D., Li, J., Yao, T., Kwon, C. and Friesz, T. L. (2012), ‘Demand learning and dynamic pricing under competition in a state-space framework’, *Engineering Management, IEEE Transactions on* **59**(2), 240–249.
- Cooper, W. L., de Mello, T. H. and Kleywegt, A. J. (2014), ‘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>.
- Cyert, R. M. and DeGroot, M. H. (1970), ‘Bayesian analysis and duopoly theory’, *Journal of Political Economy* **78**(5), 1168–1184.
- Dasgupta, P. and Das, R. (2000), Dynamic pricing with limited competitor information in a multi-agent economy, in P. Scheuermann and O. Etzion, eds, ‘Cooperative Information Systems’, Vol. 1901 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp. 299–310.
- den Boer, A. V. (2015), ‘Dynamic pricing and learning: historical origins, current research, and new directions’, *Surveys in Operations Research and Management Science* **20**(1), 1–18.
- den Boer, A. and Zwart, B. (2014), ‘Simultaneously learning and optimizing using controlled variance pricing’, *Management Science* **60**(3), 770–783.
- den Boer, A. and Zwart, B. (2015), ‘Dynamic pricing and learning with finite inventories’, *Operations Research* **63**(4), 965–978.
- DiMicco, J. M., Maes, P. and Greenwald, A. (2003), ‘Learning curve: a simulation-based approach to dynamic pricing’, *Electronic Commerce Research* **3**(3-4), 245–276.
- Dimitrova, M. and Schlee, E. E. (2003), ‘Monopoly, competition and information acquisition’, *International Journal of Industrial Organization* **21**(10), 1623–1642.
- Farias, V. and van Roy, B. (2010), ‘Dynamic pricing with a prior on market response’, *Operations Research* **58**(1), 16–29.
- Fisher, M., Gallino, S. and Li, J. (2017), ‘Competition-based dynamic pricing in online retailing: A methodology validated with field experiments’, *Management Science* **64**(6), 2496–2514.

- Fishman, A. and Gandal, N. (1994), ‘Experimentation and learning with networks effects’, *Economics Letters* **44**(1-2), 103–108.
- Friesz, T. L., Kwon, C., Kim, T. I., Fan, L. and Yao, T. (2012), ‘Competitive robust dynamic pricing in continuous time with fixed inventories’, arXiv:1208.4374 [math.OC].
- Gallego, A. G. (1998), ‘Oligopoly experimentation of learning with simulated markets’, *Journal of Economic Behavior & Organization* **35**(3), 333–355.
- Greenwald, A. R. and Kephart, J. O. (1999), Shopbots and pricebots, in A. Moukas, C. Sierra and F. Ygge, eds, ‘Agent Mediated Electronic Commerce II’, Springer, pp. 1–23.
- Harrington, J. E. (1995), ‘Experimentation and learning in a differentiated-products duopoly’, *Journal of Economic Theory* **66**(1), 275–288.
- Harrison, J. M., Keskin, N. B. and Zeevi, A. (2012), ‘Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution’, *Management Science* **58**(3), 570–586.
- Isler, K. and Imhof, H. (2008), ‘A game theoretic model for airline revenue management and competitive pricing’, *Journal of Revenue and Pricing Management* **7**(4), 384–396.
- Johnson Ferreira, K., Simchi-Levi, D. and Wang, H. (2016), ‘Online network revenue management using Thompson sampling’, Working paper, <https://ssrn.com/abstract=2588730>, accessed 23 August 2018.
- Jumadinova, J. and Dasgupta, P. (2008), 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’, IEEE, pp. 403–412.
- Jumadinova, J. and Dasgupta, P. (2010), Multi-attribute regret-based dynamic pricing, in W. Ketter, H. La Poutre, N. Sadeh, O. Shehory and W. Walsh, eds, ‘Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis’, Vol. 44 of *Lecture Notes in Business Information Processing*, Springer Berlin Heidelberg, pp. 73–87.
- Keller, G. and Rady, S. (2003), ‘Price dispersion and learning in a dynamic differentiated-goods duopoly’, *The RAND Journal of Economics* **34**(1), 138–165.
- Keskin, N. B. and Zeevi, A. (2014), ‘Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies’, *Operations Research* **62**(5), 1142–1167.
- Kirman, A. (1983), On mistaken beliefs and resultant equilibria, in R. Frydman and E. S. Phelps, eds, ‘Individual forecasting and aggregate outcomes’, Cambridge University Press, New York, pp. 147–166.
- Kirman, A. P. (1975), Learning by firms about demand conditions, in R. H. Day and T. Graves, eds, ‘Adaptive Economics’, Academic Press, New York, pp. 137–156.
- Kirman, A. P. (1995), Learning in oligopoly: Theory, simulation, and experimental evidence, in A. P. Kirman and M. Salmon, eds, ‘Learning and Rationality in Economics’, Basil Blackwell, Cambridge, MA, pp. 127–178.
- Könönen, V. (2006), ‘Dynamic pricing based on asymmetric multiagent reinforcement learning’, *International Journal of Intelligent Systems* **21**(1), 73–98.
- Kutschinski, E., Uthmann, T. and Polani, D. (2003), ‘Learning competitive pricing strategies by multi-agent reinforcement learning’, *Journal of Economic Dynamics and Control* **27**(11-12), 2207–2218.
- Kwon, C., Friesz, T. L., Mookherjee, R., Yao, T. and Feng, B. (2009), ‘Non-cooperative competition among revenue maximizing service providers with demand learning’, *European Journal of Operational Research* **197**(3), 981–996.
- Li, J., Yao, T. and Gao, H. (2010), A revenue maximizing strategy based on Bayesian analysis of demand dynamics, in D. A. Field and T. J. Peters, eds, ‘Proceedings of the 2009 SIAM Conference on Mathematics for Industry’, Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 174–181.
- Mirman, L. J., Samuelson, L. and Urbano, A. (1993), ‘Duopoly signal jamming’, *Economic Theory* **3**(1), 129–149.
- Perakis, G. and Sood, A. (2006), ‘Competitive multi-period pricing for perishable products: a robust optimization approach’, *Mathematical Programming* **107**(1), 295–335.

- Ramezani, S., Bosman, P. A. N. and La Poutre, H. (2011), Adaptive strategies for dynamic pricing agents, in '2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology', IEEE, pp. 323–328.
- Rassenti, S., Reynolds, S. S., Smith, V. L. and Szidarovszky, F. (2000), 'Adaptation and convergence of behavior in repeated experimental Cournot games', *Journal of Economic Behavior & Organization* **41**(2), 117–146.
- Schinkel, M. P., Tuinstra, J. and Vermeulen, D. (2002), 'Convergence of Bayesian learning to general equilibrium in mis-specified models', *Journal of Mathematical Economics* **38**(4), 483–508.
- Sutton, R. S. and Barto, A. G. (1998), *Reinforcement learning: An introduction*, Vol. 1, MIT press Cambridge.
- Tesauro, G. and Kephart, J. O. (2002), 'Pricing in agent economies using multi-agent Q-learning', *Autonomous Agents and Multi-Agent Systems* **5**(1), 289–304.
- Tuinstra, J. (2004), 'A price adjustment process in a model of monopolistic competition', *International Game Theory Review* **6**(3), 417–442.

A Competitor Algorithms

A.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 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.

A.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.

A.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, \dots , 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.

A.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 $(0, 10], (10, 20], \dots, (90, 100]$. 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.

A.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 price 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.

A.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.

A.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).

A.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 revenue compared to the competitors, that is to solve in each time step,

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

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

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.