Adaptive Strategies for Dynamic Pricing Agents

Sara Ramezani, Peter A. N. Bosman, and Han La Poutré
CWI, Dutch National Institute for Mathematics and Computer Science
P.O. Box 94079, NL-1090 GB, Amsterdam, The Netherlands
Email: S.Ramezani, Peter.Bosman, Han.La.Poutre@cwi.nl

***Abstract*—Dynamic Pricing (DyP) is a form of Revenue Management in which the price of a (usually) perishable good is changed over time to increase revenue. It is an effective method that has become even more relevant and useful with the emergence of Internet firms and the possibility of readily and frequently updating prices. In this paper a new approach to DyP is presented. We design adaptive dynamic pricing strategies and optimize their parameters with an Evolutionary Algorithm (EA) offline while the strategies can deal with stochastic market dynamics quickly online. We design two adaptive heuristic dynamic pricing strategies in a duopoly where each firm has a finite inventory of a single type of good. We consider two cases, one in which the average of a customer population's stochastic valuation for each of the goods is constant throughout the selling horizon and one in which the average customer valuation for each good is changed according to a random Brownian motion. We also design an agent-based software framework for simulating various dynamic pricing strategies in agent-based marketplaces with multiple firms in a bounded time horizon. We use an EA to optimize the parameters for each of the pricing strategies in each of the settings and compare the strategies with other strategies from the literature. We also perform sensitivity analysis and show that the optimized strategies work well even when used in settings with varied demand functions.**

I. Introduction

Dynamic Pricing (DyP) is a form of Revenue Management (RM) that involves changing the price of goods or services over time with the aim of increasing revenue. Revenue management is a much broader term that refers to various techniques for increasing revenue of (usually) perishable goods or services. RM particularly became popular within the airline industry after the deregulation of the industry in the United States in the late 1970's [18].

Today, the Internet provides exceptional opportunities for practicing RM and particularly DyP. This is due both to the amount of data available and the restructuring of price posting procedures. Thus, the Internet can facilitate offering different prices for different customers and posting new prices with minimum extra costs. This also allows for the increased use of intelligent autonomous agents in e-commerce, agents designed for automatically buying, selling, price comparison, bargaining, etc..

Most RM methods exploit the differences in different customers' valuations of a good or changes in these valuations in time to boost revenue. This has led to different methods of distinguishing between customers based on their valuation for the goods, such as fare class distinction, capacity control, dynamic pricing, auctions, promotions, coupons, and price discrimination methods such as group discounts [18].

Here we focus on dynamic pricing [7]. By changing prices in time, firms can ask for the price that yields the highest revenue at each moment. This allows them to distinguish between customers in cases where customers with different utilities buy at different times, as well as exploit the changes of valuation of the same customers in time. Many RM methods can be categorized as dynamic pricing, be it the end-of-season markdown of a fashion retailer, or the inflated last-minute price of a business-class flight ticket. The main question is when and how to change the prices in order to obtain the most revenue. This depends on the market structure and dynamics, most importantly, on the customer demand rate and how it changes in time.

In this paper we study dynamic pricing of a limited supply of goods in a competitive finite-horizon market. We design and implement an interactive agent based marketplace where the agents are the firms who wish to increase their revenue using dynamic pricing strategies. We study two cases, one in which the customers' valuation for the products follow the same valuation throughout the selling horizon and one in which the average of their valuations follow a random Brownian motion over time.

We design two adaptive pricing strategies and use learning to optimize them. The execution of the strategies is not computationally intensive ($O(1)$ complexity) and they are understandable from a practical perspective. The strategies use only the observed market response to set new prices at each point in time. The parameters used in each strategy are then optimized using an Evolutionary Algorithm (EA) [3]. A simulation software has been implemented that can generate and simulate dynamic pricing strategies in various oligopolistic markets. The EA uses this simulator as a black box, and optimizes the parameters for a given strategy using the obtained revenue as the fitness criteria. So on one hand, the strategies are capable of very fast adaptive decision making in run-time, and on the other hand their parameters are optimized offline to tune them for a more specific setting.

In many real world applications, some general knowledge of the market dynamics exists beforehand, although it may be different from what will actually happen, both because

of inaccuracies in the estimations and predictions and unexpected changes to the market. Using our proposed approach, this knowledge can be used for offline learning, to optimize the parameters of the strategies before the actual selling starts. Also, because of the adaptiveness of the proposed pricing strategies, any deviations from the expected dynamics of the market will be detected quickly and accounted for by the strategy online, and thus the strategies also work reasonably well in various market settings different from what they have been tuned for.

In the strategies we present, the selling agents do not assume the demand to have an a priori structure, though the EA uses the market simulation with a particular structure to optimize the performance of the strategy over the space of its parameters. Thus, information of the demand structure is passed to the selling agents implicitly through the parameters. The strategies are adaptive in the sense that they detect changes in the market and adjust their prices accordingly. Hence they can effectively deal with market changes.

In order to evaluate our strategies' performance, we compare them to a number of strategies previously studied in the DyP literature. We show that our strategies outperform a fixed price (FP) strategy that is optimized offline. This is significant, because FP strat egies perform very well in many configurations [9]. It should be noted here that in most other models a FP strategy is actually the optimal strategy and DyP is used to find this optimal fixed price, but in our case no fixed price is optimal due to the combination of competition, finite inventory, and a finite time horizon. This can also be shown by the fact that our strategies outperform the offline-optimized FP (that is very close to the actual best possible fixed price). The same features make the analytical computation of the best solution in our model intractable, requiring experimentation to evaluate our strategies. In fact, a benefit of using simulations is that we are able to tackle more complicated models that are too difficult to approach theoretically.

Furthermore, the strategies also perform better than the optimized versions of the derivative follower (DF) learning algorithm that change the price in the same direction (increasing or decreasing) as long as the revenue keeps increasing, and then changes the price change direction. Such algorithms have previously been used successfully in DyP settings [4], [6], [14]. Our proposed strategies are also compared to the Goal Directed (GD) strategy of [6] which is particularly similar to one of our strategies. Both strategies outperform the GD strategy for which the only parameter, the initial price, is optimized for the given setting.

Finally, we show that optimized adaptive strategies still perform reasonably well when various changes are made to the market configuration after the learning phase. For this means, we evaluate the performance of a strategy with parameters optimized for a given configuration, on a stochastically varied configuration. The obtained revenue is then compared with the revenue obtained by using the same strategy with parameters optimized for the varied setting. This can give us the regret of wrongly estimating the market structure. The variations in configuration that we study include altering the demand function by changing the customer/good ratio.

II. Related Work

DyP has been a very active research area in recent years. Many studies try to learn the demand structure, or the parameters for a known demand function, on the fly. They typically use part of the selling horizon for *exploring* the market, trying out the demand rate for different prices in a systematic way, and another portion of the time for *exploiting* the market, using the best price(s) based on their estimates [2], [8]. Others use statistical learning methods and heuristics based on mathematical estimations of the optimal price [1]. While most DyP models are monopolies, there are some that model competitors in the marketplace as well [14], [15], [16]. In this work we deal with a duopoly market, though the firm does not model a competitor explicitly.

Works that are similar to ours in experimenting with heuristic strategies by simulation are fewer. The Information Economics group at IBM has investigated the effect of interacting pricing agents which they call *pricebots* in a number of works (see [14] for a survey). In some, they use use game-theoretic analysis and experiment with heuristics that aim at achieving the optimal equilibrium price [11]. They focus on the market dynamics and pricing patterns that arise when using these strategies against each other. They also study *shopbots* [10], strategic buyer agents, and pricing where agents may be differentiated horizontally or vertically based on their preferences for different attributes of a product. Some of the algorithms discussed in these works rely on more information than can be obtained from the market simulations only, but we have compared our work to the FP and a few versions of the DF strategies, both of which are used in these works.

Multi-attribute DyP is also discussed in [5] and [13]. In [13] a heuristic method for dynamic pricing is presented which consists of a preference elicitation algorithm and a dynamic pricing algorithm. The method is then compared to a DF algorithm and the GD algorithm from [6], their model differs from the one we use in the existence of multiple attributes (we consider a single attribute here, the price) and also because they have a finite number of identifiable buyers (compared to our infinite population of one-time buyers). Their algorithms, although similar to ours in fast online decision making and the use of simulation for the evaluation, were not directly comparable to the strategies in our current model due to the strong dependence on these differences, the simplification of which would significantly undermine the strengths of their strategies.

In [4], a heuristic Model Optimizer (MO) method is designed and compared to a DF pricing strategy. The MO strategy uses information from the previous time intervals for a more detailed model of the demand, and solves a non-linear equation in each time step using a simplex hill-climbing approach. This heuristic strategy differs from ours in its online computational complexity, which is much higher than ours due to the online optimization.

In [6] a DF algorithm and an inventory-based GD algorithm are used in a number of simulations to show how they actually behave in a market and in which scenarios each one is useful. We compare our strategies with both of these strategies, because they are both compatible with our model and comparable with our strategies in their computational intensity and the information they use.

In [17], a few different EA methods are used to solve a dynamic pricing problem. Their approach is not comparable to ours since they use their optimizer to optimize actual prices for a dynamic pricing model for a small number (less than 10) of time steps. A method similar to [17] is not useful in our stochastic model because optimizing prices using an EA would lead to an over-fitted solution that works better than the adaptive strategies only for the instances (see definition 2 in section IV) it is optimized for and considerably worse on average. It is also far more time-consuming when considering a larger number of time steps.

III. Model

We have a market with two competing firms. The revenue (sum of price of items sold) of one of the firms is optimized using DyP. Each firm can change the price of its goods at the start of equi-distant time intervals.

A. Firms

We have a finite number, m, of firms, $\{0, 1, \ldots, m-1\}$. We refer to firm j's good type as g_j. The firm starts off with an initial inventory Y_j of its product and the capacity left of the good at time t is denoted by $y_j(t)$ (the t can be omitted if there is no chance of ambiguity).

The model is a finite horizon model, the goods left at the end of each time step are transferred into the next and all goods are lost at the end of the whole time span. Each firm announces a selling price, $p_j(t)$, for each good type in each time interval t. A cost for each good type, cr_j, serves as a reserve price for goods of that type.

B. Customers

1) Preferences: The customers specify their preferences using non-negative cardinal utilities that are exchangeable with monetary payments. Each customer has a valuation function that determines these utilities. Customers are *unit-demand*, they only have preferences on sets consisting of one item, so their valuation functions are defined as $v : G \to \mathbb{R}^+$. Thus, customers have to specify only a single number for each good type and its valuation for getting more than one item is always zero. Thus, a customer's utility for getting an item is equal to the difference between its valuation for the item and the item's price, which is $u_j(t) = v(g_j) - p_j(t)$ for firm j's good at time t.

2) Population: We model the customer populations as an unbounded population. This means that the distribution of customers does not change after an item is sold. Also, the valuations of all of the customers for each unit of each of the good types follow the same distribution. This distribution, which is denoted by $Pr_{j,t}$ for firm j's good at time t, may or may not change in time.

These distributions are all normal distributions. We consider two settings, in the first setting the normal distribution is the same for each good type and customer segment pair throughout the time horizon (so the t can be omitted from $Pr_{j,t}$). In the second setting, which we refer to as the *Brownian* setting, the mean of the each of the $Pr_{j,t}$ distributions changes over time, following a basic model of Brownian motion: the mean increases by a constant amount (b), decreases by the same constant amount, or does not change, each of these cases happening with equal ($\frac{1}{3}$) probability. This allows for some structured dynamism in the demand pattern in the model.

3) Customer arrival: The number of customers that arrive in each time step follows a Poisson process with a constant intensity a. The firms may be aware of the parameter of this process when making their pricing decisions. In each time interval, the customers arrive consecutively after the firms have set their prices. They may or may not buy a product based on their choice function and, in any case, leave the market afterwards.

4) Choice Model: At any time t that a customer has to make a purchase decision, it will buy one of good g^* offered by firm $f^* \in \arg\max_j \{u_j(t) | u_j(t) > 0\}$, if it exists, i.e. the item for which it has the highest utility if all items are not priced higher than he is willing to pay, with probability $1 - \lambda$ and does not purchase anything with probability λ. The λ factor is to model a general chance for a purchase not occurring, this is close to the natural behavior of customers in many contexts. Note that the effect of the λ can also be achieved by changing the arrival rate when we are using a Poisson arrival process, but it is not so with all arrival models. If g^* does not exist, the customer will not make a purchase. Ties are broken randomly.

As is evident from the model, the customers are myopic (greedy) and purchase only based on current utilities, not any prediction of what will happen next.

C. Modeling Time

In any dynamic pricing model, by definition, the firms should be able to adjust their prices in time. While changing prices at any particular moment may become more plausible, particularly with internet firms, it is still more common in

the literature for the change of prices to occur in fixed time intervals. We suppose that there are T time intervals, numbered from 1 to T successively. At the start of each time interval, all firms set the prices for their goods.

IV. Market Simulation

We have developed software for simulating a marketplace described in the previous section. The software uses an event queue to keep track of two types of events: pricing events, firms setting the price for each of their item types at the beginning of each time period, and customer arrival events.

Some notation that helps describe the experiments in the following section is defined here.

Definition 1 (Configuration): A *configuration* is a model where all parameters are set. These parameters consist of the properties of the firms (costs of goods, initial stock, etc.) and the valuation distributions and arrival rate of the customers.

Definition 2 (Instance): An *instance* of the problem is a specific configuration together with samplings for the stochastic variables (i.e. a fixed random seed for the pseudo random generator in the software).

Definition 3 (Pricing Strategy): A *pricing strategy*, or simply strategy, is a function that given a fixed number of parameters, sets a new price for a unit of the firm's good in each time step. The function can also depend on the previous events that have occurred in the market. We assume here that the firm is aware of the previous customers' behavior and previous prices, and that the firm knows the customer arrival rate, but nothing about the customers' valuation functions. All strategies are deterministic.

Based on the above definitions, an instance of the problem is deterministic given the firms' strategies, i.e. will yield the same results when the firms use the same strategies, while a configuration alone does not contain enough information to determine an outcome.

Definition 4 (Simulation): By a *simulation*, we designate a single execution of a particular instance of the problem with fixed strategies for the firms.

Definition 5 (Batch run): By a *batch run*, or simply batch, consisting of n simulations, we mean the simulation of n different instances of the problem that share the same configuration and use the same strategy for each of the firms throughout the n instances.

V. Adaptive Heuristic Strategies

We present two heuristic pricing strategies in this section.

A. The Inventory Based (IB) Strategy

The first strategy is one that adaptively adjusts the prices for a firm based on the number of goods it has left and the number of goods that it has sold in the previous time interval, we call it the Inventory Based (IB) strategy.

In each time step, the strategy retains the previous price if the rate of items sold in the previous time interval is close to the rate needed to sell all the items by the end of the time horizon (this "closeness" is controlled by the parameters $noChangeThreshUp$ $noChangeThreshDown$). It increases the price if too many items have been sold in the previous interval and decreases it if too little have been sold. The $maxDecPercent$ and $maxIncPercent$ parameters along with the distance that the sales rate has from the expected sales rate control the amount of change in price in each time step. The other parameter used in this strategy is $intialPrice$, the price the firm uses in the first time interval. The details of the algorithm of this strategy can be seen in algorithm 1.

Algorithm 1 InventoryBasedStrategy(initialPrice, noChangeThreshUp, noChangeThreshDown, maxIncPercent, maxDecPercent)

1: **if** $time = 0$ **then**
2: $price \leftarrow initialPrice$
3: **return** $price$
4: **if** $numLeft = 0$ **then**
5: **return** $lastPrice$
6: **if** $pastCustomers = 0$ **then**
7: $pastCustomers \leftarrow 1$
8: $pastSold \leftarrow pastSold \times \frac{aveCustomers}{pastCustomers}$
9: $\alpha \leftarrow \frac{pastSold \times timeLeft}{numGoodsLeft}$
10: **if** $\alpha < 1$ **then**
11: $\Delta = \alpha - 1$
12: **else**
13: $\Delta = 1 - \frac{1}{\alpha}$
14: **if** $|\Delta| < 0$ **then**
15: **if** $\Delta < noChangeThreshDown$ **then**
16: $price \leftarrow lastPrice$
17: **else**
18: $price \leftarrow lastPrice(1 + \Delta \times maxDecPercent)$
19: **else**
20: **if** $\Delta < noChangeThreshUp$ **then**
21: $price \leftarrow lastPrice$
22: **else**
23: $price \leftarrow lastPrice(1 + \Delta \times maxIncPercent)$
24: **return** $price$

In algorithm 1, $pastSold$ is the number of items sold in previous time step, and $numGoodsLeft$ is the number of items left in the inventory. $timeLeft$ is the number of time steps left in the selling horizon, and $pastPrice$ is the price of a unit of the good in the previous time step. Finally, $pastCustomers$ is the total number of customers in the previous time step and $aveCustomers$ is the average number of customers per time step (same as a).

In line 8, the number of items sold in the past time interval is normalized by the average number of customers arriving in each time step and the number of goods left to factor out the stochasticity as much as possible. Note that this is

a dynamic indicator updated in the beginning of each time step, so it will take into account the current state of the agent. In line 9, α is defined as an indicator for determining how fast the inventory would be exhausted if the sales would go on with the current rate. If α is smaller than one, then the sales rate is too slow, and if it is larger than one, the the inventory would be exhausted sooner than the end of the time horizon, so there is an opportunity for increasing the price. The parameter Δ is then (in the if-then statement starting from line 10) defined as a normalized version of α that is negative if the sales rate is too low and positive if it is too high. The if-then statement starting from line 14 is where the final pricing decision is made. If the absolute value of Δ is smaller than the respective threshold for positive or negative threshold parameter, i.e. if the sales rate is close enough to the desired rate, then the price is not changed otherwise it is changed proportional to Δ, and with regards to the maximum allowable change rate.

B. The Revenue Based (RB) Strategy

The Revenue Based (RB) strategy uses an estimation of a desirable price to estimate the price in each time step. It uses the *revenue per customer (RPC)* criterion (which considers all customers, even the ones that did not buy from the firm) to assess the revenue obtained when using a particular price and compares that to the RPC needed to finish the inventory by the end of the selling horizon. The algorithm for this strategy can be seen in algorithm 2. The additional parameters used in this strategy are $expPrice$, the expected price used in the estimation of the expected RPC, and the maximum amount the price can change per time step. These variables are also used: RP, the revenue per customer in the previous time step, and the expected RPC, used as a control parameter, $expRPC$.

Algorithm 2 RevenueBasedStrategy(initialPrice, expPrice, maxDelta)

1: **if** $time = 0$ **then**
2: $\quad price \leftarrow initialPrice$
3: $\quad$ **return** price
4: $expRPC \leftarrow \frac{numGoodsLeft \times expPrice}{timeLeft \times aveCustomers}$
5: **if** $numLeft = 0$ **then**
6: $\quad$ **return** $lastPrice$
7: **if** $pastCustomers = 0$ **then**
8: $\quad pastCustomers \leftarrow 1$
9: $RPC \leftarrow \frac{pastSold \times lastPrice}{pastCustomers}$
10: $\alpha \leftarrow \frac{RPC}{expRPC}$
11: **if** $\alpha \leq 1$ **then**
12: $\quad \Delta \leftarrow \alpha - 1$
13: **else**
14: $\quad \Delta \leftarrow 1 - \frac{1}{\alpha}$
15: $price \leftarrow lastPrice + \Delta \times maxDelta$
16: **return** $price$

In this algorithm, the $expRPC$ (defined in line 4) variable is the expected revenue per customer for the rest of the selling horizon, provided that the firm sells with the expected price, $expPrice$, that is one of the input parameters. Then the RPC parameter is the revenue obtained for each customer in the previous time step (regardless of whether they make a purchase or not). In this algorithm, α is defined as the ratio between the expected RPC and the RPC from the previous time step. Here, also, Δ is a normalization of α, positive if the expected revenue is higher than expected and negative if it is lower (line 11), and the price changes proportional to the magnitude of Δ. Note that here the lower than expected revenue is attributed to a price that is too high, thus prohibiting many customers form making a purchase. This is not always the case though, but this can be a safe assumption when the initial price and expected price are chosen properly, as we can see from the experimental results.

Note that both strategies depend only on information from the sales in one previous time step. Also, the parameters are designed as to sustain a certain amount of stability in the price. This is both a means to control the effects of the stochastic noise causing sudden jumps in the price (specially since the changes depend on one previous time step only), and because too much price fluctuation is not desirable from the customers' perspective. Also, in all strategies reported in this work, the previous price is kept if there are no more goods to be sold. This has no effect on the simulation because customers will not be given the option to buy from firms that have no goods left.

C. Computing the Parameters

We want to have settings for the parameters of the strategies that we have defined such that the strategies perform well. The numerical optimization task associated with this is generally not easy because it is the outcome of a non-trivial simulation that we want to optimize. The problem at hand can thus be seen as a black-box optimization problem with unknown difficulty. We therefore need black-box optimization algorithms that are capable of tackling a large class of problems effectively. The algorithm of our choice is called AMaLGaM. AMaLGaM is essentially an Evolutionary Algorithm (EA) in which a normal distribution is estimated from the better, selected solutions and subsequently adapted to be aligned favorably with the local structure of the search space. New solutions are then constructed by sampling the normal distribution. A parameter-free version of AMaLGaM exists that can easily be applied to solve any optimization problem. This version was recently found to be among the most competent black-box optimization algorithms [3], [12].

In order to tune the experiments for the EA, which is not designed to handle stochasticity on one hand, and not to over-fit a single instance of the problem on the other, we use the following method. The fitness used in the EA is the average revenue obtained from a batch run of 100

0's strategy	0's profit	1's profit
FP(9.895)	80.896	61.921
IB(10.021, 2.245, 1.506, 0.224, 0.210)	90.045	61.041
RB(9.999, 10.195, 0.173, 0.121)	89.323	60.477

Table I
COMPARISON OF PROFITS OBTAINED FROM STRATEGIES WITHOUT BROWNIAN MOTION.

0's strategy	0's profit	1's profit
FP(9.712)	51.178	2.361
DF(9.710, 0.008)	51.224	3.053
GD(9.204)	68.971	5.699
IB(10.067, 3.357, 2.503, 0.018, 0.239)	84.954	17.993
RB(10.001, 10.009, 0.298, 0.206)	83.411	16.99

Table II
COMPARISON OF PROFITS OBTAINED FROM STRATEGIES WITH BROWNIAN MOTION.

different instances. The parameters of the heuristic strategies defined in section V are optimized for firm 0, given that firm 1 follows a fixed price strategy. All evaluations in the EA are executed on the same 100 instances, thus making the optimization problem non-stochastic.

VI. EXPERIMENTAL RESULTS

We ran the EA in this way multiple times for each strategy, both for the case where the customers' valuation distribution does not change with Brownian motion, and for the case in which it does. For each of these cases, the parameters obtained from each of the the experiments were evaluated on an *evaluation set* consisting of 10^4 instances and the parameter set that achieved the highest result on the evaluation set was then selected as the optimized parameter set for the given strategy and configuration and then cross evaluated using a *test set* of 10^4 instances on which no experiments were performed before.

For these experiments (and others in this paper, unless stated otherwise) the parameters of the configuration are as follows: $T = 50$, as previously stated, $m = 2$ and each of the firms has one good type, $cr_0 = 9$ and $cr_1 = 10$, $Y_0 = Y_1 = 100$. Also, $S = 1$ and the $Pr_{s,g_0,t}$ is a normal distribution with mean 10.5 and $Pr_{s,g_j,t}$ is normal with mean 11.5, both distributions with standard deviations of 1. Furthermore $a = 5$ and $\lambda = 0.1$. When using Brownian motion, $b = 0.1$. These parameters are selected so that there is a reasonable competition between firms. It also illustrates a case where firm 1 has a slightly more "expensive" good, its cost, price and the customers' valuation for it are higher than firm 0's good. In any case, the specific choices of the variables are for illustration and are not essential for the overall results.

In addition to the two strategies presented in the previous section and an FP strategy, we have implemented various versions of the DF algorithm and the GD strategy from [6] and compared them to the heuristic strategy in the Brownian case. These algorithms have also been optimized by the EA. The most basic DF has two parameters, the first the initial price and the second the step, or amount of price change in each time step, which is constant in this basic DF algorithm. The range for which the second parameter was optimized is $[0.001, 1]$. The other other DF algorithms, namely the ADF strategy in [4] and the DF strategy in [6], had similar results as the simple DF, all of them that were very close to the FP result. The basic Df had the best results and is reported here.

The results of the experiments with 20 runs of the EA for each strategy for the non-Brownian and Brownian cases can be seen in tables I and II respectively. Note that firm 1's revenue is always higher than firm 0's because its good is generally more expensive, it has a higher cost and the customers value it higher on average. Firm 1's profit is lower than firm 0 in most of these cases.

These results show that the IB and RB strategies consistently outperform the FP and DF strategies. The IB strategy has an 11.3% increase in profit compared to the fixed price strategy, and the RB strategy has an increase of 10.4%. In the Brownian case, the IB strategy has a 66% increase in profit and the RB algorithm has a 63% increase compared to the fixed price strategy, with similar results for the DF strategy. An interesting observation is that although there is not much difference between the change in the competitor firms' profits in the non-Brownian case, in the Brownian case, the IB and RB strategies also result in much better profits for firm 1, so the strategies have resulted in a kind of win-win situation in this case.

As for the GD strategy, its performance falls between the DF and our strategies. The algorithm is similar to our strategies (specially IB) in that both algorithms aim to finish the inventory in the last time step. However, the estimation of the number of items that need to be sold in each time step as made in GD is always based on the initial, complete time interval. Contrary, in our strategies, such estimates are based on the number of time steps left. Another important difference is that in our strategies there are limits on the amount of price change per time step. GD lacks such limits or dampening factors, which ultimately result in larger fluctuations in price, especially in situations where the stochastic behavior of the environment is non-negligible, as is the case here. An illustrative example of this behavior is presented in figure 1. A last difference is that GD uses less information and the only parameter of GD is the initial price. GD still yields an almost %35 profit gain compared to the FP, which is pretty good for an algorithm that is using only one parameter to adjust the price.

Pairwise comparisons of the strategies are also performed and can be seen in table III for the Brownian case (non-Brownian results are similar). We use revenues from the 10^4 instances obtained from the cross evaluation (the average of which is reported in tables I and II) for comparison.

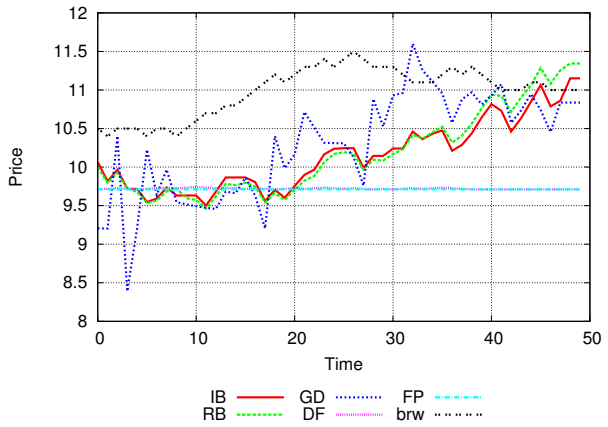

Figure 1. Illustration of the prices set by strategies in a single instance. 'brw' is the average valuation of the customer population for firm 0's good which follows a Brownian motion.

Strategy	FP	DF	GD	IB	RB
FP		33.01 -0.07 -1.02	37.71 -17.82 -11.02	19.99 -33.80 28.54	21.77 -32.26 -27.07
DF	66.99 0.07 1.02		38.42 -17.75 -10.14	20.25 -33.73 -27.81	22.28 -32.19 -26.36
IB	62.30 17.82 11.02	61.58 17.75 10.14		16.03 -14.44 -15.59	13.42 -15.98 -17.00
IB	80.01 33.80 -28.54	79.75 33.73 27.81	83.97 14.44 15.59		62.32 1.54 1.58
RB	78.23 32.26 27.07	77.72 32.19 26.36	86.58 15.98 17.00	37.68 -1.54 -1.58	

Table III
PAIRWISE COMPARISON OF THE STRATEGIES WITH BROWNIAN MOTION.

In each cell of these tables, the first number (from top) shows the percentage of instances in which the first (row) strategy performs better than the other (column strategy). The second number is the mean of the differences (row strategy minus column strategy), and the third is the median of this difference. Note that the parameters used for these strategies in the pairwise comparison are the optimized ones which can be seen in table II. We have used the Sign test to asses the statistical significance of these results. This test checks whether the distribution of the revenue differences in each of these pairwise comparisons significantly differs from zero statistically. Due to the large number of instances, all of results in table III are shown to be statistically significant. Thus, the adaptive strategies are indeed performing much better than the FP, DF, and GD strategies, and so are any other strategy that is shown to outperform another.

VII. Sensitivity Analysis

In this section we study the robustness of the optimized strategies by computing the amount of revenue loss suffered

a	profit opt	profit std	profit loss (%)
4 (80%)	50.873	46.818	7.971
6 (120%)	131.166	122.664	6.482
7 (140%)	158.259	145.018	8.367
8 (160%)	176.953	161.207	8.898

Table IV
COMPARISON OF RESULTS USING THE PARAMETERS OPTIMIZED FOR THE STANDARD CASE AND CASES IN WHICH DEMAND CHANGES BETWEEN 80% AND 140% OF THE STANDARD DEMAND; THE 'PROFIT OPT' COLUMN SHOWS THE AVERAGE PROFIT OBTAINED WHEN USING THE OPTIMIZED PARAMETERS FOR EACH CASE, AND THE 'PROFIT STD' COLUMN SHOWS THE AVERAGE REVENUE OBTAINED USING THE PARAMETERS OPTIMIZED FOR THE STANDARD CASE. THE LAST COLUMN SHOWS THE AMOUNT OF PROFIT LOSS.

a	Profit opt	profit std	profit loss (%)
4 (80%)	42.883	38.939	9.197
6 (120%)	123.447	120.108	2.705
7 (140%)	147.55	143.394	2.817
8 (160%)	166.514	161.012	3.304

Table V
SIMILAR TO TABLE IV, BUT WITH BROWNIAN MOTION.

in case of wrong assumptions about the market configuration. To do this, we run the best performing strategy that we have designed up to now, IB, for some configurations that vary with our default Brownian and non-Brownian configurations (see section VI).

We consider a class of varied configurations where the customer arrival rate, a, is changed compared to the standard non-Brownian configuration discussed above. We have run the EA 10 times for configurations with a taking each of the values in the set $\{4, 6, 7, 8\}$ (corresponding to 80%, 120%, 140%, and 160% of the standard average customer population) and compared the result of using the parameters obtained from optimizing the standard case (in table II) with the results for these varied configurations. The aim is to compute how much revenue will be lost by falsely assuming that the market is of the standard configuration (and learning the parameters accordingly) when it is indeed following one of the varied configurations. Note that cases with a lower customer arrival rate are not considered because the total number of customers will become so low that the firms cannot obtain a positive profit no matter how they price their goods, thus the cases become too extreme to be interesting.

The results of this experiment can be seen in tables IV The results show that even in the most severe cases in our experiments less than 10% of the profit can be lost by incorrectly predicting the model.

VIII. Conclusions and Further Work

We have presented a framework for implementing dynamic pricing in an interactive agent-based marketplace. Additionally, we presented two heuristic pricing strategies for the selling agents and a method for offline optimization of their parameters that can be used within agents or outside

of them. We showed that for the cases we study, the heuristics yield revenues that are consistently better than that of the best offline-optimized fixed price and the results of various derivative follower algorithms.

Our method is applicable in situations where there is some general information about the market configuration beforehand, so that offline optimization may become possible. The strategies are also adaptive and robust to market dynamics. This is particularly evident by observing that the strategies perform even better compared to an optimized fixed price strategy when more (structured) stochasticity is introduced to the model by random fluctuations in the customer demand function throughout time (Brownian motion case). The adaptive strategy can pick up the changes and adjust the prices accordingly in reasonable time.

Another type of robustness is illustrated by using the strategies in cases where the parameters have been optimized with incorrect assumptions about the demand function. In both the Brownian and non-Brownian cases, our IB strategy can still perform well with the same optimized parameters when the demand is increased up to 160% of the original configuration, compared to when the parameters are specifically optimized considering the demand change.

This approach to dynamic pricing is, to the best of our knowledge, a new one. Some possible extensions to the model are having a larger number of firms on the market, firms with multiple good types, customers wishing to buy combinations of goods, etc.. Also, following up on similar research, it would be interesting to study our approach with a multi-attribute model, either using generalized versions of the current strategies or by designing new strategies.

Another important future extension is to use more competitors and competitors that have intelligent and adaptive pricing strategies as well. In the current work, the competitor always has a fixed price strategy, but by giving the option of more varying strategies to the competitor as well, more complex market dynamics can arise.

REFERENCES

[1] Y. Aviv and A. Pazgal. A partially observed markov decision process for dynamic pricing. *Management Science*, 51(9):1400–1416, 2005.

[2] O. Besbes and A. Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Oper. Res.*, 57:1407–1420, November 2009.

[3] P. A. N. Bosman. On empirical memory design, faster selection of bayesian factorizations and parameter-free gaussian edas. In *Proc. 11th Annual Conf. on Genetic and Evol. Comp.*, GECCO '09, pages 389–396, New York, USA, 2009. ACM.

[4] 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.

[5] P. R. Dasgupta and Y. Hashimoto. Multi-attribute dynamic pricing for online markets using intelligent agents. *International Joint Conf. on Autonomous Agents and Multiagent Systems*, 1:277–284, 2004.

[6] J. M. DiMicco, A. Greenwald, and P. Maes. Dynamic pricing strategies under a finite time horizon. In *Proc. 3rd ACM Conf. on Electronic Commerce*, EC '01, pages 95–104, New York, NY, USA, 2001. ACM.

[7] W. Elmaghraby and P. Keskinocak. Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science*, 49(10):1287–1309, 2003.

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

[9] G. Gallego and G. van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8):999–1020, 1994.

[10] A. R. Greenwald and J. O. Kephart. Shopbots and pricebots. In *Proc. 6th International Joint Conf. on Artificial Intelligence*, IJCAI '99, pages 506–511, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[11] A. R. Greenwald and J. O. Kephart. Probabilistic pricebots. In *Proc. 3rd. Workshop on Game Theoretic and Decision Theoretic Agents*, pages 37–44, 2000.

[12] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *Proc. 12th Annual Conf. on Genetic and Evol. Comp.*, GECCO '10, pages 1689–1696, New York, NY, USA, 2010. ACM.

[13] J. Jumadinova and P. Dasgupta. Multi-attribute regret-based dynamic pricing. In W. e. a. Aalst, editor, *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.

[14] J. O. Kephart, J. E. Hanson, and A. R. Greenwald. Dynamic pricing by software agents. *Computer Networks*, 32(6):731 – 752, 2000.

[15] Y. Levin, J. McGill, and M. Nediak. Dynamic pricing in the presence of strategic consumers and oligopolistic competition. *Management Science*, 55(1):32–46, 2009.

[16] K. Y. Lin and S. Y. Sibdari. Dynamic price competition with discrete customer choices. *European Journal of Operational Research*, 197(3):969 – 980, 2009.

[17] S. Shakya, F. Oliveira, and G. Owusu. An application of eda and ga to dynamic pricing. In *Proc. 9th Annual Conf. on Genetic and Evol. Comp.*, GECCO '07, pages 585–592, New York, NY, USA, 2007. ACM.

[18] K. Talluri and G. van Ryzin. *The Theory and Practice of Revenue Management*. Springer US, 2004.