Learning Dynamic Prices in Multi-Seller Electronic Retail Markets with Price Sensitive Customers, Stochastic Demands, and Inventory Replenishments

C.V.L. Raju, Y. Narahari, K. Ravi Kumar

Abstract— **In this paper, we use reinforcement learning (RL) as a tool to study price dynamics in an electronic retail market consisting of two competing sellers and price sensitive and lead time sensitive customers. Sellers, offering identical products, compete on price to satisfy stochastically arriving demands (customers) and follow standard inventory control and replenishment policies to manage their inventories. In such a generalized setting, RL techniques have not been applied before. We consider two representative cases: (1) *no information case*, where none of the sellers has any information about customer queue levels, inventory levels, or prices at the competitors; and (2) *partial information case*, where every seller has information about the customer queue levels and inventory levels of the competitors. Sellers employ automated pricing agents or pricebots, which use RL-based pricing algorithms to reset the prices at random intervals based on factors such as number of back orders, inventory levels, and replenishment lead times, with the objective of maximizing discounted cumulative profit. In the *no information case*, we show that a seller who uses Q-learning outperforms a seller who uses derivative following (DF). In the *partial information case*, we model the problem as a Markovian game and use actor-critic based RL to learn dynamic prices. We believe our approach to solving these problems is a new promising way of setting dynamic prices in multi-seller environments with stochastic demands, price sensitive customers, and inventory replenishments.**

Index Terms— **Retail markets, dynamic pricing, inventory replenishments, price sensitive customers, stochastic demands, pricebots, derivative following, reinforcement learning, multi-agent learning, Q-learning, Markovian game, actor-critic algorithms.**

I. Introduction

Sellers have always faced the problem of setting the right prices for goods and services that would generate the maximum revenue for them. Determining the right prices to charge a customer for a product or a service is a complex task. It requires that a company know not only its own operating costs and availability of supply but also how much the customer values the product and what the future demand would be [10]. A company therefore needs a wealth of information about its customers and also be able to adjust its prices at minimal cost. Advances in Internet technologies and e-commerce have dramatically increased the amount of information the sellers can gather about customers and have provided universal connectivity to customers making it easy to change the prices. This has led to increased adoption of dynamic pricing and to increased interest in dynamic pricing research.

Computer Science and Automation, Indian Institute of Science, Bangalore - 560 012, India. E-mail:raju@csa.iisc.ernet.in

Computer Science and Automation, Indian Institute of Science, Bangalore - 560 012, India, E-mail: hari@csa.iisc.ernet.in (corresponding author)

General Motors India Science Lab, Bangalore (Work initiated while this author was working at IBM India Research Lab, New Delhi)

A. Some Examples of Dynamic Pricing

There is a revolution brewing in pricing that promises to profoundly alter the way goods are marketed and sold. In the future, sellers will offer special deals, tailored just for every customer, just for the moment on everything (right price to the right customer at the right time). Quantity or volume discounts is a simple example of dynamic pricing that is followed by almost all retailers and merchants. Consumer segmentation is another simple example: senior citizens may be allowed discounts, students and academic institutions are allowed discounts on software packages, etc. Sales promotions provide another common example. The airline industry is a common example of deployment of dynamic pricing strategies, called yield management or revenue management [27], [31]. Priceline.com allows travelers to name their price for an airline ticket booked at the last minute and get a ticket from an origin to a destination at a fractional cost of the full fare. Priceline.com uses complex software that enables major airlines to fill unsold seats at marginal revenues. Buy.com [32], [9] uses software agents to search web sites of competitors for competitive prices and in response, Buy.com lowers its price to match these prices. The pricing strategy here is based on the assumption that their customers are extremely price sensitive and will choose to purchase from the seller offering the lowest price. This has resulted in Buy.com register high volumes of trade, however due to the low prices, the profits are low, often times even negative. This example illustrates that overly simplistic or incorrect model of buyer behavior can produce undesirable results. Amazon.com is another example of a e-business company which has experimented with dynamic prices on their products, for example, popular DVDs. Depending on the supply and demand, the prices on a particular DVD varied over a wide range. Yield management methods are currently being practiced in hotel rooms, cruises, rental cars, etc.

B. Contributions and Outline

Today's economy is ready for dynamic pricing, however the prices will have to be adjusted in fairly sophisticated ways to reap the benefits of dynamic pricing. Motivated by this, in this paper, we look into a machine learning based approach for dynamic pricing in typical electronic retail markets. In

particular, we consider a fairly general setting of an electronic market consisting of two competing retail stores (that is, multiple sellers). Price sensitive and lead time sensitive customers arrive into the system in a stochastic fashion. Each retail store stocks an inventory of the same product and replenishes the inventory following a standard policy such as the (q, r) policy. The decision problem is to determine the optimal dynamic prices to be chosen by a particular seller so as to maximize revenue, under the assumed behavior of the customers, the stochastic nature of the customers, inventory considerations, and in the face of competition from the other seller. A problem in such generality has not been solved before, to the best of our knowledge. In this paper, we use a reinforcement learning framework to solve this dynamic pricing problem. The contribution of this paper is in two parts.

- In Part 1, we consider what we call the *no information case* in which we assume that none of the retail stores is aware of the customer queue levels, inventory levels, or prices of the other retail stores. We use two different adaptive strategies, Q-learning and Derivative Following (DF) [1]. We model a two-seller market and analyze the market when one seller uses a Q-learning based pricebot and the other seller uses a DF based pricebot.
- In Part 2, we consider the *partial information case* in which we assume that each retail store has information about the customer queue levels and inventory levels of other stores. We consider a two-seller market and model the problem as a two-person stochastic game, where both the players (that is, sellers) follow RL-based adaptive behavior. Q-learning based multi-agent algorithms have been suggested in [17], [25], [26], but here, we model the problem as a a general-sum Markovian game and use an actor-critic-type of reinforcement learning scheme such as the one proposed in [21], [2], [28]. We show how each seller can learn an equilibrium policy in this non-cooperative, stochastic dynamic pricing game.

The rest of the paper is organized as follows. In Section 2, we review relevant literature. In Section 3, we describe our model in detail and list all the assumptions made. In Section 4, we deal with the *no information case*. In Section 5, we deal with the *partial information case*. Section 6 concludes the paper and discusses directions for future work.

II. A Review of Relevant Work

The work that is relevant to our paper falls into two categories: (1) dynamic pricing in traditional retail market models with inventories (2) recent line of research in using machine learning based models to determine dynamic prices in markets. The second category can be further classified based on whether a single learning agent is used or multiple learning agents are involved.

A. Dynamic Pricing of Traditional Retail Markets

Dynamic pricing in traditional retail markets has been researched quite extensively, since decades. Early works include that of Varian [41] and Salop and Stiglitz [29]. Elmaghraby and Keskinocak [10] and Swann [37] provide a comprehensive review of models of traditional retail markets where inventories are a major consideration.

Gallego and van Ryzin [13] consider optimal dynamic pricing of inventories with stochastic demand over finite horizon. The assumptions made here are: (1) The market is a monopolist market, (2) the selling horizon is finite, (3) the store has a finite stock of items with no replenishment during the selling horizon, (4) demand decreases in price, and (5) unsold items have a salvage value. Gallego and van Ryzin model the demand as a Poisson process with intensity $\lambda(p)$ where $\lambda(p)$ is increasing in p. By charging price p_t at time t, the firm controls the intensity of the demand. Thus the reservation prices of the customers are modeled indirectly. They show under suitable assumptions that: (a) more stock and/or longer remaining time to sell goods leads to higher expected revenues; (b) at a given point in time, the optimal price decreases as the inventory increases - conversely, for a given level of inventory, the optimal price rises if there is more time to sell.

Federgruen and Heching [12] consider the optimal inventory and pricing policy of a seller who faces an uncertain demand where prices are changed periodically over time. In each period, before demand is realized, the seller must decide the quantity to produce, q_t, given his starting inventory position x_t, where t denotes the number of periods remaining. Equivalently, the seller decides how much of inventory y_t to have on hand at the start of the period. It is found that a base stock list price (BSLP) policy is optimal under a wide range of settings.

Elmaghraby and Keskinocak [10] discuss three main characteristics of a market environment that influence the type of dynamic pricing problem a retailer faces:

- Replenishment vs No Replenishment of inventory (R/NR)
- Dependent vs Independent Demand over time(D/I)
- Myopic vs Strategic customers (M/S)

The review focuses on NRIM, NRIS, and RIM categories and provides a comprehensive overview of the literature for the above three categories, summarizing all the important results. Most of the results available are for single seller monopolistic markets.

In our paper here, we consider RIM type of retail markets with multiple competing sellers, stochastic demands, price sensitive customers, and inventory replenishments. Thus we look into a problem, much more general than addressed in the literature.

Obtaining an optimal dynamic pricing policy, without considering price as a decision variable at each period is a well-studied topic. [22], [19]. [12], [40], [44] address the optimal inventory and pricing policy of a monopolist seller, who changes prices periodically over time. There are many other papers that discuss issues of dynamic pricing with inventory considerations. For example, see [19], [5], [6], [27], [29], [30], [34], [35], [36], [40], [42], [16].

B. Models with a Single Learning Agent

In a typical market, the environment constantly changes with demands and supplies fluctuating all the way. In such a scenario, it is impossible to foresee all possible evolutions of

the system. The amount of information available is also limited (for example, a seller does not have complete information about the pricing of the competing sellers). With learning-based models, one can put all available data into perspective and change the pricing strategy to adapt best to the environment.

Gupta, Ravikumar, and Kumar [14] consider a web-based multi-unit Dutch auction where the auctioneer progressively decrements per unit price of the items and model the problem of finding a decrementing sequence of prices so as to maximize total expected revenue, in the presence of uncertainty with regard to arrival pattern of bidders and their individual price-demand curves. The above decision problem is modeled as a single agent RL in an uncertain non-stationary auction environment. Under the assumption of independent bidder valuations, the authors develop a finite horizon Markov decision process model with undiscounted returns and solve it using a Q-learning algorithm.

Carvalho and Puterman [4] consider the problem of a retailer who has to set the price of a good to optimize the total expected revenue over a period of time T. When the demand function is known, the situation reduces to a simple stochastic maximization problem. However, when the demand function is not known, the retailer has to rely on uncertain prior information to guide his pricing decisions. The model is a simple log-linear regression model, where the logarithm of the demand is a linear function of the price. The seller can learn about the parameters of the demand function and use it to set prices so as to maximize the revenues over a given time horizon. Several pricing rules are studied and compared. It is shown that a one-step look-ahead rule performs fairly robustly for a single seller environment studied.

In the paper by Brooks *et al* [3], the performance of two different pricing strategies (both based on machine learning) is compared in the context of single seller markets for electronic goods. The first strategy uses a one parameter pricing model and the second one uses a two parameter pricing model. It is shown that a dynamic pricing strategy based on two parameter learning outperforms the one based on one parameter learning. The paper derives analytical methods to determining optimal prices for a model with partial information.

C. Models with Multiple Learning Agents

Ravikumar, Saluja, and Batra [28] study a service market environment with two sellers who compete to service a stream of buyers who are of two varieties, informed and uninformed. They assume that both the sellers follow an RL-based adaptive behavior and model the system as a general sum Markovian game. They propose an actor-critic type of RL scheme (a variant of the scheme proposed by Konda and Borkar [21]) and provide experimental results on convergence.

Hu [18] studies three different types of pricing algorithms (or pricing agents) in a simulated market. The first agent uses reinforcement learning to determine the prices, by learning an optimal action for one period based on the rewards it receives for that action. The second agent uses a traditional Q-learning method, by learning about Q-values which represent long-term optimal values for the agent's own actions. The third agent uses a Nash Q-learning algorithm, by learning about Q-values which represent long-term Nash equilibrium values for agent's joint actions. The third agent performs better than the second and the second outperforms the first agent in a simulated market where the agents compete with one another.

Greenwald, Kephart, and Tesauro [1] attempt to understand the strategic pricebot dynamics in a multi-seller environment where each seller employs a pricebot that employs a price-setting strategy. They examine four different price-setting strategies: game theoretic pricing, myoptimal pricing, derivative following, and Q-learning, which differ in their informational and computational requirements. In homogeneous settings, when all the pricebots use the same pricing algorithm, derivative following approach is shown to outperform game theoretic pricing and myoptimal pricing. In a market with heterogeneous pricebots, myoptimal and game theoretic pricing outperform derivative following while the Q-learning strategy outperforms all the others.

Kephart and Tesauro [20] study aspects of multi-agent Q-learning in a model market in which two identical, competing pricebots strategically price a commodity. Two fundamentally different solutions are observed: an exact, stationary solution with zero Bellman error consisting of symmetric policies, and a non-stationary, broken-symmetry pseudo-solution, with small but non-zero Bellman error. This pseudo-convergent asymmetric solution has no analog in ordinary Q-learning. The authors compute analytically the form of both solutions, and map out numerically the conditions under which each occurs. It is suggested that this observed behavior will also be found more generally in other studies of multi-agent Q-learning.

Dasgupta and Das [7] study the price dynamics in a multi-agent economy consisting of multiple sellers and multiple buyers. Buyers use shopbots and sellers use pricebots. The authors come up with a learning-based model optimizer algorithm that improves upon a naive derivative following algorithm for dynamic pricing.

Related investigations are reported in the papers by Sridharan and Tesauro [33], Tesauro and Kephart [38], and Tesauro [39], and Lawrence [23].

The paper by DiMicco, Greenwald, and Maes [9] describes a simulator using which different types of markets can be simulated with different types of dynamic pricing algorithms, including machine learning based algorithms. The papers by DiMicco, Greenwald, and Maes [8], [9] compare two dynamic pricing strategies, namely, the goal-directed strategy and the derivative-following strategy in a monopolistic market.

D. Comparison of our Paper with Related Work

Our paper is more general than the papers in the existing literature in the following ways.

- The retail market considered has the following features: (1) There are multiple competing sellers; (2) Customers are price sensitive and are divided into two natural categories (shoppers and captives); (3) Customers arrive randomly into the system; (4) There is a finite inventory at each retail store, which is replenished, following the standard (q, r) policy; and (5) If no inventory is available

at the store, the customers are provided a lead time quote based on which they will choose whether or not to remain in the system. This is the first time the dynamic pricing problem is being studied in such a general setting.
- Single agent and multi-agent learning-based models appearing in the literature address the dynamic pricing problem assuming infinite inventory at each retail store. This is a key assumption that is relaxed in this paper. Inventory replenishments are also modeled in the present paper.
- Multi-agent learning based models for dynamic pricing have so far not captured the competition among the sellers in as much generality as done here.

III. Model Description

In a competitive retail market there are typically several retailers selling an identical product. Any attempt by one of the retailers to sell its product at more than the market price leads consumers to desert the high-priced retailer in favor of its competitors. In a monopolized market there is only one retailer selling a given product and when a monopolist raises its price it loses some, but not all, of its customers. In reality the retail markets are somewhere in between these two extremes. Every retailer, in spite of being part of a competitive market, still enjoys limited monopoly power, coming, for example, from locational advantage of his store or from his customer service strategy.

A. Non-linear Pricing and Customer Segmentation

If a retail store has some degree of monopoly power it has more options open to it than a retail store in a perfectly competitive market. Such a retailer can try to *differentiate* its products from the products sold by his competitors to enhance its market power even further. In this paper we consider a specific form of differentiation, namely the *non-linear pricing*, where price per unit of sale is not constant but depends on how much a consumer buys. Volume discounts for large purchases is an example of such pricing. This form of differentiation appeals for two reasons. It is important to note that retailers in general have limited flexibility with regard to price changes because of lean margins left by manufacturers; quantity discounts offer a simple and feasible implementation of differentiation that can improve the retailer's profit. Secondly, if a retailer were to set the right price for the right customer, he has to know the demand curves of the consumers, that is, the retailer has to know the exact willingness to pay of each person. Even if the retailer knows something about the statistical distribution of willingness to pay, it might be hard to prevent the *gaming*: a high-willingness-to-pay consumer can pretend to be a low-willingness-to-pay person. The retailer may have no effective way to separate them apart. Non-linear pricing will help get around this problem by offering two different price-quantity package, one targeted towards the high-demand consumer and one toward the low-demand consumer. In other words, the retailer constructs price-quantity packages that give the consumers an incentive to *self select*.

B. Motivating Examples

The dynamic pricing model considered in this paper is motivated by the following facts. When customers select a specific retailer, the selection process amounts to a trade-off among three attributes:(1) *the price* (2) the *delivery time promised* (3) all other attributes, such as convenience of shopping. For example, at the online-book stores, "one-click shopping", the ease of "online tracking of order status" belong to the third category of criteria. A retailer uses a balanced trade-off among these criteria to build loyalty among customers. The industries in which the above attributes are used as an explicitly advertised competitive instrument have become numerous.

Many online book stores offer price discount coupons to customers based on their purchase histories. An online book seller may quote a time until shipping that is based on the ability of the firm to withdraw a unit of demand from a warehouse. Customers arriving early in the day may be quoted a longer time until shipping so that inventory may be reserved for customers coming later in the day with higher valued orders. In the service markets, dynamic pricing examples based on delivery time quotes abound. In Pizza delivery business, Domino's has offered a 30-minute delivery guarantee, backed up with a free of charge delivery if the time limit is exceeded. Similarly, banks like Wells Fargo award five dollars to each customer who waits more then five minutes in line.

The above examples can be viewed as different forms of implementation of price-based differentiation mechanisms. These forms of implementations have come to the fore mainly because retailers suspect that consumers may feel exploited by frequent price changes. In this paper we attempt to abstract the basic price-differentiation that lies beneath all the above-presented forms of differentiation.

In particular, we address a dynamic pricing problem a retailer in constrained domains, that is, when the retailer can accommodate only a finite inventory and faces random replenishment lead times. Depending on the availability of items at any instant, and depending on the lead times to supply existing backlogged demands, price changes are used to control the arriving demand or to induce the customer to wait. Strategies such as discount coupons, cash-backs for failing to meet the advertised service standards as described in the earlier examples can be used to implement the price changes modeled here.

C. Motivation for Duopoly Market

To make our dynamic pricing study interesting, we consider a duopoly market with two retailers offering *non-linear pricing* in an electronic market setting. The Internet has given the consumer the *search power* and as a consequence, on-line retailers have lost the potential price advantage that could have resulted from the search costs that consumers incur in their pursuit for best bargains. As a result, electronic retail markets have turned into *monopolistic competition* with competition driving commodity prices to their marginal costs. Since returns from such competitive market cannot substantiate investments, only a few players can remain in the market. Consider the case of

retail market for books where Barnes and Noble and Amazon emerged as two significant players attracting consumers using various forms of differentiation. Hence, duopoly model that we consider here is a good approximation of the online retail world. Motivated by the above considerations, we discuss price dynamics in a two-seller model of retail market when each seller uses non-linear pricing-based differentiation mechanism.

D. Captives and Shoppers

Because of its intrinsic ability to offer a consumer with self-selection, it becomes possible for the retailers to *learn* consumer segmentation from their preferences over price-quantity packages. To make our developments simple, we confine to the case with two types of price-quantity packages offered by each retailer; price for unit quantity and *buy two and get one free*. Price per unit quantity will possibly differ over the two retailers. Such offers can be realistically implemented as follows: consumers who access the web-page of any retailer would initially see unit price offer against the item requested for and also will see an alert depicting *discounts available at higher volumes*. Consumers who select the former option or package can be safely assumed to be not so price-sensitive and hence would be willing to pay high price for the item if any additional service offer also comes as part of the package. We assume that the retailer offers a lead time commitment to such consumers in the event of no stock being available at the time of request. Customers who choose this option will *learn* over time such additional service benefits from the retailer and adhere to the same retailer for future purchases. We call such customers *captives* to the retailer. We assume further that each captive places an order (or purchases, if stock is readily available) only if she derives strictly positive utility from the price quote and lead time quote of the retailer, else leaves the market. On the other hand, the customer who wishes to choose the second offer will have low willingness to pay per item and select the retailer with lower value of the first option (provided this price is affordable to her), with ties broken arbitrarily. These customers would be willing to bear with the inconveniences imposed by the retailer. Dynamic pricing policies of airlines reflect the same phenomenon. In our study, we consider *waiting-based* inconvenience by each seller as detailed below. Captives get preference over the second class of customers with regard to supply of items in the absence of stock. The retailer can implement this priority in the following way. The consumer who clicks on the alert will be next led to a transaction page that exhibits the volume offer in the event of availability of stock after serving the pending *captive* orders. Either when no stock is available or when replenishment quantity to arrive is just sufficient enough to meet the pending orders, then the customer will be requested to place her orders in her shopping cart. The customer will revisit her shopping cart after a random time interval to check for the availability status and will cancel her order if she again observes stock-out, else will purchase the quantity at the offer made provided the price quote is rewarding enough for her and balks from the system otherwise. The retailer serves these customers in the order of the time-stamps recorded at the shopping-carts. We wish to add that these customers upon their first visit would select the retailer with lower unit price offer and discard the offer at the other retailer even though items are available in stock at that seller. We call such customers as *shoppers* in the sequel. This type of behavior models, for example, the consumers who purchase objects for a future resale.

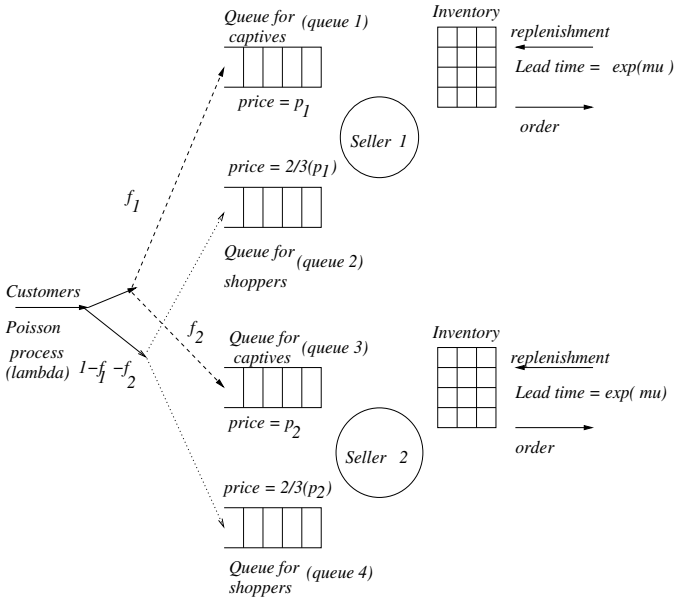

Fig. 1. A Two Seller Retail Market

E. The Model

Figure 1 provides a schematic for our model of a duopoly market. Below we provide the mathematical formalism for this model.

- Customers arrive at the market according to a Poisson process with rate λ. f_l fraction of these customers are captives at seller l, $(l = 1, 2)$ and the remaining fraction constitute the shoppers.
- Seller l posts per unit price p_l as part of his menu to the arriving customers.
- Each seller turns away any arriving request if the total demand backlogged exceeds N at that point in time.
- Each seller has finite inventory capacity $I_{\max}$ and follows a fixed reorder policy for replenishing; when the inventory position, current inventory level plus the the quantity ordered, falls below a level r, would order for replenishment of size $(I_{\max} - r)$. This is the classical (q, r) policy of inventory planning.
- The replenishment lead times at both the sellers are exponentially distributed with mean $\frac{1}{\mu}$.
- Let the captive's utility of a price quote p and lead time quote w (equal to the expected replenishment lead time, $\frac{1}{\mu}$) denoted by $U_c(p, w)$ for some map $U_c(.,.)$. The RL-based approach presented in the paper will work for any type of utility function and this is a strength of the paper. For example, one can use standard, Cobb-Douglas type of utility function which, with its decreasing returns to scale provides an appropriate function for our model.

- The shopper measures her utility of the unit price option $\frac{2p_l}{3}$ on the menu of seller l by:

$$U_s(p) = (p_s - p)\Theta(p_s - p) \quad (1)$$

  where $p_s \sim U(0, p_{\max})$ for a given $p_{\max}$ and p_s is the shopper's maximum willingness to pay per item.
- Every shopper upon placing an order in her shopping cart at a retailer will revisit the retailer again after an interval of time distributed exponentially with rate μ_s.
- Seller l sets his unit price p_l from a finite set A.
- Each seller incurs a holding cost rate of H_I per unit item per unit time and a cost of H_q per each back-logged request. The purchasing price per item is P_c. We assume zero reorder costs.

Each seller dynamically resets the prices at random intervals so as to maximize his expected profits. Under the above Markovian distributional assumptions, the dynamic pricing problem can be reduced to a Semi-Markov Decision Process in the *no-information* case and to a semi-Markovian Game in the partial information case. However, in reality, retailers do not in general have knowledge about the distributions underlying the model and also about buyers' behavior. In such cases, these retailers learn about their most beneficial prices over time using Reinforcement Learning (RL), the right paradigm for learning in Markov decision processes or Markovian games. In the following sections, we develop appropriate RL-based learning schemes in both the described cases and contrast them with other possible non-RL based *learning* schemes.

IV. No Information Case

This case refers to a situation where each seller is oblivious of inventory levels, number of backlogged requests and pricing strategies of the other seller. Further, we also assume that each seller is ignorant about buyers' characteristics and their individual purchase behavior. In the absence of such information, past dynamics will be of some help to decide a future course of action. In this section, we compare two such strategies; the *Q-learning* strategy [43] and *derivative following* strategy [1]. The former is derived from the RL-paradigm while the latter is a simple practical algorithm widely discussed in the dynamic pricing literature.

A. Motivation for Using Reinforcement Learning

Given the parametric nature of the model presented (everything is well defined:arrival rate, replenishment lead time, reorder quantity, reorder point, etc.), it would appear that it is possible to write down explicitly the transition structure of the underlying Markov decision process and solve it through the Bellman's equation or by standard numerical methods. However:

- the computation of transition probabilities is highly non-trivial in this case and in fact, can be extremely challenging for realistic cases;
- the customers' utility function is not known in general to be able to use the standard MDP algorithms and therefore we wish to develop an approach that would work with any utility function form of the customers.

The two reasons above lead naturally to the use of an RL-based approach for the dynamic pricing problem in this paper.

B. Q-learning

The queues queue 1 and queue 2 in Figure 1 at each retailer are virtual queues for captives and shoppers at that retailer (similarly queue 3 and queue 4 are at the other retailer). Note that queue 2 is an orbit queue with each shopper who has registered her orders in the shopping cart will make a retrial after an interval of time that is assumed to be exponentially distributed with mean $\frac{1}{\mu_s}$. Since the two retailers act independently under our no-information assumption, we isolate, for notational convenience, one retailer and develop a model for price dynamics at that retailer.

Let $\mathbf{X}(t) := (X_1(t), X_2(t), I(t))$ be the state of the system at the retailer with $X_i(.)$'s representing the number of backlogged requests in queue i at the retailer and $I(.)$, the inventory level at the retailer at time t. The retailer posts unit price quote and the volume discount alert on his web-page and will reset the prices only at transition epochs, that is, whenever a purchase happens (and hence the inventory drops) or when a request is backlogged in either of the queues. Recall that the captive (shopper) will purchase or make a backlog request only when her utility is positive. It is easy to see that price dynamics can be modeled as a continuous time Markov Decision Process model. Below we give the state dynamics:

At time 0, the process $\mathbf{X}(t)$ is observed and classified into one of the states in the possible set of states (denoted by S). After identification of the state, the retailer chooses a pricing action from A. If the process is in state i and the retailer chooses $p \in A$, then

(i) the process transitions into state $j \in S$ with probability $P_{ij}(p)$
(ii) and further, conditional on the event that the next state is j, the time until next transition is a random variable with probability distribution $F_{ij}(.|p)$.

After the transition occurs, pricing action is chosen again by the retailer and (i) and (ii) are repeated. Further, in state i, for the action chosen p, the resulting reward, $S_p(.)$, the inventory cost, $H(i)$ and the backorder cost $C(i,j)$ costs are as follows: Let $i = [x_1, x_2, i_1]$ and $j = [x_1', x_2', i_1']$.

$$\begin{aligned} S_p(i,p,j) &= p \text{ if } x_1' = x_1 + 1 \\ &= p \text{ if } i_1' = i_1 - 1 \\ &= 2p \text{ if } i_1' = i_1 - 3 \\ &= 0, \text{ otherwise} \\ C(i,j) &= [i_1 - i_1']^+ P_c \\ H(i) &= x_1 H_q + i_1 H_I \end{aligned}$$

The following remarks are in order.

Remark 1: The *complementarity condition* $I(t)X_1(t) = 0 \ \forall t$ for $l = 1, 2$ holds good.

Remark 2: Let p represent the seller's price in the observed states and p' be the price posted by the second retailer. Then the following transitions occur:

- $[0, x_2, i_1] \to [0, x_2, i_1 - 1]$ with rate $f_1 \lambda P(U_c(p, \frac{1}{\mu}) > 0) \ \forall x_2, i_1$

- $[0, x_2, i_1] \rightarrow [0, x_2 - 1, i_1 - 3]$ with rate $(1 - f_1 - f_2)\mu_s P(U_b(p) > 0) \ \forall x_2, i_1$
- $[0, x_2, i_1] \rightarrow [0, x_2, i_1 - 3]$ with rate $(1 - f_1 - f_2)\lambda P(U_b(p) > U_b(p') \geq 0) \ \forall i_1$.
- $[x_1, x_2, 0] \rightarrow [x_1 + 1, x_2, 0]$ with rate $f_1 \lambda P(U_c(p, \frac{1}{\mu}) > 0) \ \forall x_2$
- $[x_1, x_2, 0] \rightarrow [x_1, x_2 + 1, 0]$ with rate $(1 - f_1 - f_2)\lambda P(U_b(p) > U_b(p') \geq 0) \ \forall x_1$
- $[x_1, x_2, 0] \rightarrow [(x_1 - r)^+, x_2, (r - x_1)^+]$ with rate $\mu \ \forall x_2$

Discounted Optimality

Let $g : S \rightarrow A$ denote a stationary deterministic pricing policy, followed by the retailer, that selects an action only based on the state information. Let $t_0 = 0$ and let $\{t_n\}_{n \geq 1}$ be the sequence of successive transition epochs under policy g and $X(t_n-)$ denote the state of the system just before t_n. For any $0 < \alpha < 1$, let the long-term discounted expected profit for the policy g be

$$V_g(i) = E_g[\sum_{n=1}^{\infty} e^{-\alpha t_{n-1}}(S_p(X(t_n-), g(X(t_n-)), X(t_n))) - C(X(t_n-), X(t_n)) - \int_{t_{n-1}}^{t_n} H(X(t_n-))e^{-\alpha t}dt | X_1 = i] \quad (2)$$

α above discounts the rewards to the initial time epoch.

Let $V_\alpha^*(i) = \max_g V_g(i)$. The retailer's problem is to find a $g^* : S \rightarrow A$ such that $V_{g^*}(i) = V_\alpha(i)$.

The domain of optimization above can be in general the set of all non-anticipative policies (randomized, non-randomized or history dependent) and but from the Markovian assumptions it follows that there exists always a stationary deterministic optimal policy and hence, we have restricted the domain to stationary policies only (and they are finite in our finite state model).

From Bellman's optimality condition, it follows that:

$$V_\alpha^*(i) = \max_p \left\{ \overline{R}_\alpha(i,p) + \sum_{j \in S} P_{ij}(p) \int_0^\infty e^{-\alpha t} V_\alpha^*(j) dF_{ij}(t|p) \right\} \quad (3)$$

where

$$\overline{R}_\alpha(i,p) = \sum_{j \in S} P_{ij}(p) \left[R(i,p,j) + \int_0^\infty \int_0^t e^{-\alpha s} H(i) ds dF_{ij}(t|p) \right] \quad (4)$$

(3) above can be solved algorithmically using any fixed point iteration scheme, such as the *value iteration* for the typical Markov decision processes. Such schemes are assured of convergence because of the contraction property coming from the discount factor, α. Also, one can construct the optimal policy g^* by assigning $g^*(i)$ to equal the maximizer on the RHS of (3). However, as we assumed earlier, the retailers do not have any information about underlying distributions involved at various stages and hence the conditional averaging that appears in (3) cannot be performed to derive the optimal value, and hence the optimal pricing policy. This motivates the retailers to use online learning to converge to optimal policy g^* eventually. One can think of devising a *learning* scheme based on any fixed point iteration methods on V^*. Even if we assume that this can be done, we will still need to know about P_{ij}'s above to construct g^*. To obviate this difficulty, the method of *Q-learning* has been proposed by Watkins and Dayan [43] for Markov Decision Processes. Below we proceed to modify it to suit to our continuous time case. To motivate the algorithm, consider the following *Q-value* associated with an action p in state i:

$$Q(i,p) = \overline{R}_\alpha(i,p) + \sum_{j \in S} P_{ij}(p) \int_0^\infty e^{-\alpha t} V_\alpha^*(j) dF_{ij}(t|a) \quad (5)$$

In other words, $Q(i,p)$ represents the long-term expected profit starting from state i when the first action to be followed in state i is p, which is possibly different from the optimal action. It is easy to see that $V_\alpha^* = \max_p Q(i,p)$, and the maximizer is the optimal action to perform in state i. Thus, if one gets somehow the Q-values, it easy to construct an optimal policy. In online learning, these Q-values are obtained from learning through actual execution of various actions in state i and measuring their relative merits. For details, refer to Watkins [43].

Below we give the actual learning update rule involved in Q-learning for our continuous time MDP. Let $t_0 = 0$ and start with an initial arbitrary guess, $Q_0(.)$ of $Q(i,p)$ above for all i and p.

- *Step 1*: At any n-th transition epoch at time t_n, observe the state i and select the price action $p_0 \in argmax_p Q(i,p)$ with probability $1 - \epsilon$ and any other price in A with probability ϵ for some $\epsilon > 0$.
- *Step 2*: If $X(t_n) = i$ and the price action chosen is p, then update its Q-value as follows:

$$Q_{n+1}(i,p) = Q_n(i,p) + \gamma_n \left[S_p(i,p,.) - H(i) \left(\frac{1 - e^{-\alpha T_{ij}}}{\alpha} \right) + e^{-\alpha T_{ij}} max_b Q_n(j,b) - Q_n(i,p) \right] \quad (6)$$

where j above is the state resulting from the action p in i and $S_p(.)$ is the reward collected from such action. T_{ij} is the average of sampled transition times between states i and j. γ_n above is called *learning parameter* and should be such that $\sum_n \gamma_n = \infty$ and $\sum_n \gamma_n^2 < \infty$.

Repeat steps (1)-(2) infinitely. Convergence is slow as is typical of any RL-algorithm. The following remarks are in order:

Remark 3: Because only a finite number of stationary policies exist in the above MDP model, convergence to an optimal policy will happen much before the Q-values stabilize in the above learning procedure. Hence, the number of price experiments required before the retailer actually starts following an optimal policy is comparatively smaller.

Remark 4: The above learning procedure can be executed in a simulated environment with experiments over different models of customer purchase behavior. Later, patter recognition techniques can be applied to quickly identify the model that closely approximates the observed behavior and then, the optimal policy that corresponds to the identified model may be used for online implementation.

Remark 5: The speed of convergence of the learning algorithm can be drastically improved using some function approximations to Q-values based on some observed features. We do not, however, take up that part in this paper.

For a later reference, we will make the following remark:

Remark 6: From the learning procedure above it follows that Q-learning will eventually find *only* a stationary deterministic optimal policy.

C. Derivative Following

This is a simple dynamic pricing strategy suggested in many papers (for example, see [1], [8], [9]) that experiments with incremental increases (decreases) in price, continuing to move its price in the same direction until the observed profitability level falls, at which point the direction of movement is reversed. It requires keeping track of past average profits for each state and increases the prices till the profitability level falls. In our setting, such a strategy can be constructed as follows. At any decision epoch t, set the price according to:

$$p_{t+1}(x_1, x_2, I) = p_t(x_1, x_2, I) + \delta_t sign(\Pi_t(x_1, x_2, I) - \Pi_{t-1}(x_1, x_2, I)) sign(p_t(x_1, x_2, I) - p_{t-1}(x_1, x_2, I)) \quad (7)$$

where $\Pi_t(x_1, x_2, I)$ is the average profit made by the seller during time epoch t, when the state is (x_1, x_2, I) and the step-size δ_t is distributed uniformly between $[a, b]$, where $a, b > 0$.

D. Simulation Experiments

1) Description of the System: We simulate and study the retail store model shown in Figure 1, by considering an action set (that is, set of possible prices) $A = \{$ 8.0, 8.1, 8.2, ..., 13.4, 13.5$\}$. The maximum queue capacities are assumed to be 10 each for queue 1 and queue 2 (this means we do not allow more than 10 waiting captives or more than 10 waiting shoppers in the retail store). The maximum inventory level $I_{\max}$ is assumed to be 20 with a reorder point at $r = 10$. We assume that $f_1 = f_2 = 0.2$, that is, 40 percent of the incoming customers are captives. We consider customers as arriving in Poisson fashion with mean inter-arrival time 15 minutes. The upper and lower limits for the uniform distribution that describes the acceptable price range for captives are assumed to be 8 and 14, respectively. These limits are assumed to be 5 and 9 for shoppers. The upper and lower limits for the uniform distribution that describes the acceptable lead time range for captives are assumed to be 0 hours and 12 hours, respectively. We consider exponential replenishment lead time for reorders with a mean of 3 hours. An impatient shopper drops out of the system after an exponential waiting time having a mean of 1.5 hours. The inventory holding cost (H_I) is chosen as 0.5 per unit per day and the backorder cost (H_q) is chosen as 0.5 per back order per day. We assume that the seller purchases the items at a unit price of 4.

2) Comparison of Q-learning and Derivative Following : We consider that one seller uses Q-learning algorithm and the other seller uses derivative following in order to take pricing decisions. We train the Q-learning based agent (seller) in the environment where the other agent (the other seller) uses derivative follower. Once Q-learning converges, the seller uses the policy derived from the Q-function. We simulated the dynamics while the optimal pricing policy derived from Q-learning is applied against derivative follower. We ran the Q-policy against the derivative follower from different states for 50,000 time steps. Figures 2 and 3 show the results. Figures 2 and 3 provide different insights. Figure 2 shows that the infinite horizon discounted profit, starting from any state is greater for the Q-learning based pricebot in all the states of the system. Figure 3 depicts the profit (average profit per time step) dynamics, with the state (5, 5, 0) as the starting state. This figure also shows the superiority of the Q-learning pricebot over the DF pricebot. The results are consistent with the findings of [1]. Note that the system considered by us has many real-world features (such as finite inventories and inventory replenishment) that not captured in the model of [1].

TABLE I

INFINITE HORIZON DISCOUNTED PROFITS FOR THREE DIFFERENT CASES

Strategy of Seller 1	Strategy of Seller 2	Profit of Seller 1	Profit of Seller 2
Q-learning	Q-learning	38524	37879
Q-learning	DF	43288	33406
DF	Q-learning	33406	43288
DF	DF	31372	30876

We also experimented with two other possibilities, namely both sellers using identical Q-learners and both sellers using identical derivative follower algorithms. We found in either case that the dynamics of the infinite horizon discounted profit of one seller closely matches with that of the other seller. The values of the infinite horizon discounted profits for the Q-learner vs. Q-learner case were higher compared to the values for the DF vs. DF case, which was on expected lines. Table I provides the numerical results for a sample experiment. The same input parameter values as described in the previous experiment were assumed for this sample experiment. Note that in the Q-learning vs. Q-learning case and also the DF vs. DF case, the profits of the two sellers are slightly different though both of them use identical algorithms. Such small differences are not uncommon to expect in stochastic simulations.

3) Optimal Values for q and r: We carried out an experiment to study the effect of q and r values on the performance of the market. Here we assume input parameter values as before and assume that one seller uses Q-learning whereas the other seller uses DF. For the retailer who uses Q-learning, we computed the fraction of customers served by him as a function of different combinations of q and r values. See Table II. The combination $q = 11$ and $r = 9$ provides the best performance. One can use such simulation experiments to aid *tactical decision making*, such as determining optimal inventory levels, optimal reorder point, etc.

V. PARTIAL INFORMATION CASE

Typically, a seller may not be able to know the prices of the other seller and hence he may not know the pay-offs at the other seller. However, a seller may be able to observe changes in the states of the other seller. Here we assume

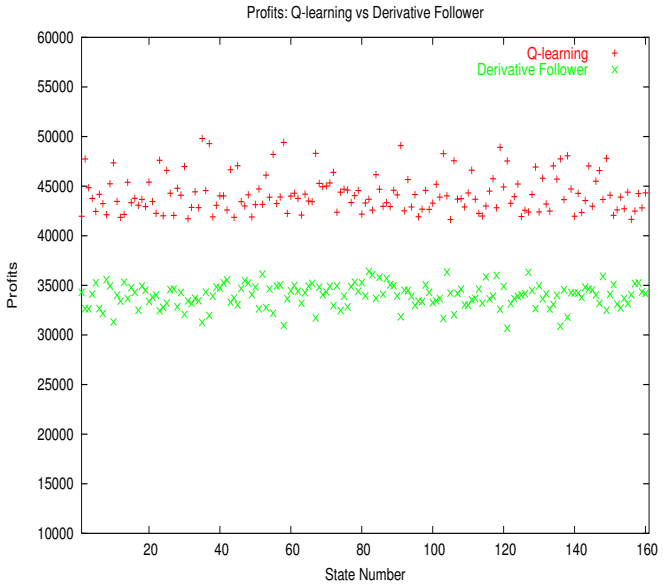

Fig. 2. Infinite horizon discounted profits for different start states

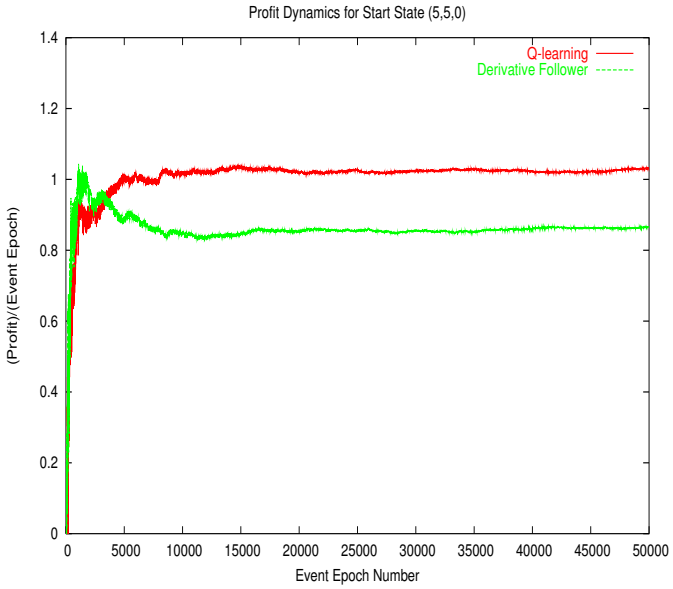

Fig. 3. Dynamics of average profit per unit time with (5,5,0) as the starting state

that each retailer has knowledge of the level of backlogged requests, and inventory levels at the other retailer. Also, we assume each retailer is equipped with a mechanism to observe *changes* in these levels. Retailers can have their own agents to monitor these changes. We make these assumptions for a technical reason to be detailed shortly. However, it is not hard to find examples that satisfy these assumptions. For instance, consider the case when both the retailers get supplies from a single distribution center of their common manufacturer. Either the distribution center or the manufacturer would like to have real-time information with regard to inventory levels and backlogs at each retailer to coordinate their logistics activities. Further, if the retailers follow Vendor Managed Inventory (VMI) policy and if a common third party manages inventories

TABLE II

FRACTION OF CUSTOMERS SERVED FOR DIFFERENT VALUES OF q AND r

(q, r) values	Fraction of customers served by the seller using Q-learning
(1,19)	11.623
(2,18)	12.104
(3,17)	14.020
(4,16)	13.416
(5,15)	16.210
(6,14)	17.738
(7,13)	19.012
(8,12)	19.641
(9,11)	21.129
(10,10)	21.457
(11,9)	24.216
(12,8)	22.187
(13,7)	20.967
(14,6)	19.046
(15,5)	17.275
(16,4)	17.341
(17,3)	14.100
(18,2)	13.912
(19,1)	12.403

at both the retailers, then these retailers have incentive to share information pertaining to their inventory levels with each other. It would be interesting to analyze the coordination game in such situations. We shun to address coordination issues here using cooperative game model but will continue with the competitive game setting of the previous section. Also, it is important to note that learnings from competitive dynamics can possibly result in a coordinating equilibrium. In the following we do not make any assumptions on information sharing with regard to dynamic prices since, interestingly, retailers will not have any incentive to reveal this information as no purchase deal materializes through such revelation. We would like to point out that the above assumption is only on observability of *changes* in levels and hence, information with regard to lost arrivals and type of customers is not explicitly available to any seller.

With the above modification, we can model the dynamic pricing game as a simultaneous-move *stochastic game*. A useful concept in such games is *Nash equilibrium*, a point in the joint policy space of the players where no player (or seller) has incentive to deviate unilaterally. If both the players follow the *same rational learning* algorithm that attempts to learn a seller's best response to the opponent's actions, and *it converges*, then both the players will be *locked in* such Nash equilibrium. However, as noted by Hart and Mas-Colell [15], it is notoriously difficult to formulate sensible adaptive dynamics that guarantee convergence to a Nash equilibrium. In fact, short of variants of exhaustive search (deterministic or stochastic), there are no general results. They reason out that it is an intrinsic consequence of the natural requirement that these dynamics be *uncoupled* and in fact, they show that completely uncoupled dynamics, such as the one developed in the previous section, do not converge to a Nash equilibrium. Given the generality of the result, we attempt here to *couple* the dynamics of the players using the minimal information requirement that we posed above. It is important to note that increased *coupledness* improves the chances of convergence

but demands too much sharing of information which in a sense is not possible in reality. In this paper, we try to investigate if with the minimal possible information assumption, such as our assumption knowledge of *state* of the game, whether it would be feasible for rational adaptive dynamics to converge. To this end, we report in this section, our experimental results on convergence when both the players are RL-driven. In the following we formally develop the game model and detail how the RL-algorithm used here is devised. As opposed to Markov decision models, the optimal policy for a player, that is, a Nash equilibrium policy is ensured only in the space of mixed strategies or in other words, in the space of randomized policies. Hence, from *Remark 3* in the previous section, Q-learning cannot be of use in our dynamic pricing game. We use the RL-algorithm derived from the policy iteration scheme of MDPs to work on the randomized policy domain. Define $S' = S \times S$, where S is the state space of the previous section. Let its elements be enumerated as $1, 2, \ldots, M$.

Consider the process $\{\mathbf{X(t)}\} = [X^1(t), X^2(t)], t \geq 0$ where X^l is $X(t)$ of the previous section for player l and be controlled by the pricing strategies of the two players as follows: At time 0, the process is observed and classified into one of the states in S'. After identification of the state, the players choose pricing actions from a finite set of pricing actions which for notational convenience is assumed to be common for the two players and is denoted by A. If the process is in state i and Players 1 and Player 2 choose p_1 and p_2 respectively, where $p_l \in A$, then

(i) the process transitions into state $j \in S$ with probability $P_{ij}([p_1, p_2])$

(ii) and further, conditional on the event that the next state is j, the time until transition is a random variable with probability distribution $F_{ij}(.|[p_1, p_2])$.

After the transition occurs, pricing actions are chosen again by the players and (i) and (ii) are repeated.

Proceeding along the lines of continuous time MDP, we call the above process a continuous time *Markov game*.

Since we assume a symmetric setting, by replacing p in the model of the previous section by the pricing *strategy profile* $\mathbf{p} := [p_1, p_2] \in A \times A$, the transition structure and rewards can be rewritten appropriately. Further, the costs and rewards there will be indexed below according to the index of the player. Consider a stationary policy $\pi^l : S' \to \mathcal{P}(A)$ for player $l, l = 1, 2$, where $\mathcal{P}(A)$ is the class of probability distributions over the action set A and $\pi^l(i, p)$ be the mass on the action p in state i. We call $\pi = [\pi^1, \pi^2]$, the ordered pair, *stationary policy profile*.

Let $\{\tau_n\}_{n \geq 0}$ denote the sequence of successive transition epochs. We use $\mathbf{X}_k$ to denote $\mathbf{X}(\tau_k)$ and $\mathbf{p}_k$ to denote $\mathbf{p}_{\tau_k}$. Given at a decision epoch τ_n, the state of the game and the strategy profile $\mathbf{p}_n$, the joint distribution, $\mathcal{F}_{ij}(t, \mathbf{p})$, of the transition interval and the next state, is by time homogeneity:

$$\begin{aligned} \mathcal{F}_{ij}(t, \mathbf{p}) &= P\{\tau_{n+1} - \tau_n \leq t, \mathbf{X}_{k+1} = j | \mathbf{X}_k = i, \mathbf{p}_k = p\} \\ &= P_{ij}(\mathbf{p}) F_{ij}(t|\mathbf{p}). \end{aligned} \quad (8)$$

Player l seeks to maximize his total expected discounted reward over the infinite horizon of the game:

$$V_\pi^l(i) = E_\pi[\sum_{n=1}^{\infty} e^{-\alpha \tau_{n-1}} (S_p^l(\mathbf{X}(\tau_n-), \pi(\mathbf{X}(\tau_n-)), \mathbf{X}(\tau_n))$$
$$-C^l(\mathbf{X}(\tau_n-), \mathbf{X}(\tau_n)) - \int_{\tau_{n-1}}^{\tau_n} H^l(\mathbf{X}(t_n-)) e^{-\alpha t} dt | X_1 = i] \quad (9)$$

$\alpha \in (0, 1)$ above is a discount factor. In order to help devise a recursive scheme to evaluate the total expected reward for a given π and thus to motivate the learning scheme to be described later, we define the following single stage terms:

$$m_{ij}(\pi) := \sum_{p_1, p_2} \int_0^\infty e^{-\alpha \tau} \pi^1(i, p_1) \pi^2(i, p_2) d\mathcal{F}_{ij}(t, \mathbf{p})$$

$$\overline{S}_p^l(i, \pi) := \sum_{j=1}^{M} \sum_{p_1, p_2} P_{ij}(\mathbf{p}) S_p^l(i, j, \mathbf{p}) \pi^1(i, p_1) \pi^2(i, p_2)$$

$$\overline{G}^l(i, \pi) := \sum_{j=1}^{M} P_{ij}(\mathbf{p}) C^l(i, j) \pi^1(i, p_1) \pi^2(i, p_2)$$

$$+H^l(i) \sum_{p_1, p_2} \sum_{j=1}^{M} \int_0^\infty \frac{1 - e^{-\alpha \tau}}{\alpha} \pi^1(i, p_1) \pi^2(i, p_2) d\mathcal{F}_{ij}(t, \mathbf{p})$$

Note that transitions to states and transition intervals are governed by both the players and depend on the randomized actions $\pi^l(.,.)$ taken by both the players and also on buyer's utilities. $m_{ij}(\pi)$ above is the expected discount factor if the state of the game next transitions to j when in state i players follow π. $\overline{S_p}^l(i, \pi)$ and $\overline{G}^l(i, \pi)$ are the expected reward and the expected costs for a single transition if the game starts in state i and the pricing action selected by the players is according to π. When π is a profile of pure (deterministic) strategies with unit mass on $[p_1, p_2]$, we denote $m_{ij}(\pi)$ by $m_{ij}([p_1, p_2])$ and $\overline{G}^l(i, .)$ above by $G^l(i, [p_1, p_2])$.

For a stationary policy profile π, denote the constant policy evaluation function $\overline{V}^l_\pi$ for player l for a fixed policy of the opponent by $\overline{V}^l_\pi(i)$. Then the player l has to solve an MDP to get his best response. Hence it follows from standard semi-Markov decision theoretic arguments that $\overline{V}^l_\pi(.), l = 1, 2$ is the unique solution to the fixed point equation:

$$\overline{V}^l_\pi(i) = \overline{r}^l(i, \pi) - \overline{G}^l(i, \pi) + \sum_j m_{ij}(\pi) \overline{V}^l_\pi(j) \forall i \in S \quad (10)$$

We call the policy profile $\pi(.,.)$ a *Nash equilibrium* if for every l, $\overline{V}^l_\pi(i) \geq \overline{V}^l_{\bar{\pi}}(i) \quad \forall i$ whenever, $\bar{\pi}^k(.,.) = \pi^k(.,.)$, for $k \neq l$.

Such a Nash equilibrium can be shown to exist following the arguments in Federgruen [11] for discrete games. Let $\pi = [\pi^1, \pi^2]$ be a Nash equilibrium in the above sense. Each player's policy in a Nash equilibrium is a point in his best response correspondence against the opponent's policy and moreover, if we freeze policies for one agent, it becomes a semi-Markov Decision Process for the other agent. Now assume that player 1 *knows* the Nash equilibrium policy of his/her opponent. To avoid confusion, in the case when player k's equilibrium policy is *known*, we use $\overline{V}^l_{\pi^k}$ to denote $\overline{V}^l_\pi$ for $l \neq k$. Now, it follows that at such Nash equilibrium π,

$\overline{V}^1_{\pi^2}(i)$ satisfies the following dynamic programming equation: $\forall x \in S$,

$$\overline{V}^1_{\pi^2}(i) = \max_p \sum_{p_2} \pi^2(i,p_2)[P_{ij}([p,p_2])S^1_p(i,[p,p_2],j) - G(i,[p,p_2]) + \sum_{j\in S} m_{ij}([p,p_2])\overline{V}^1_{\pi^2}(j)] \quad (11)$$

A similar relation holds for $\overline{V}^2_{\pi^1}(.)$. In particular, it also follows that

Remark (I): π^l itself is supported on the *argmax* of the r.h.s above.

Remark (II): Player l cannot do any better by using any other general non-anticipative policy.

A. Reinforcement Learning in the Pricing Game

In this section, we will provide motivation underlying the development of the RL algorithm used for the pricing game. The description here follows the treatise given in Ravikumar et al [28].

From the remarks below equation (11), it is enough for the players to concentrate on stationary policies to play the game. Now assume that a player, say Player 2, follows a fixed stationary policy and further that, the policy is known to Player 1. For the sake of argument, let us also assume that buyers' behavior and their utilities are also common knowledge. In other words, all the parameters of the game are known. Then, Player 1 has to solve (11) to find his best response to Payer 2's strategy.

Consider the celebrated policy iteration scheme for solution of (11) which is detailed below: Player 1 starts with a guess for optimal stationary deterministic policy $\hat{\pi}^0(.)$ and at iteration $n \geq 0$ does the following:

(**a**) Find $\overline{V}^n_{\pi^2} : S \to A$ by solving

$$\overline{V}^n_{\pi^2}(i) = \sum_{p_2} \pi^2(i,p_2)[P_{ij}([\hat{\pi}^0(i),p_2])S^1_p(i,[\hat{\pi}^0(i),p_2],j) - G(i,[\hat{\pi}^0(i),p_2]) + \sum_{j\in S} m_{ij}([\hat{\pi}^0(i),p_2])\overline{V}^n_{\pi^2}(j)]$$

(**b**) Set $\hat{\pi}^{n+1}(i)$ as an element in

$$Argmax(\sum_{p_2} \pi^2(i,p_2)[P_{ij}([.,p_2])r^1(i,[.,p_2],j) - G(i,[.,p_2]) + \sum_{j\in S} m_{ij}([.,p_2])\overline{V}^n_{\pi^2}(j)]$$

Now let us relax the earlier assumption on common knowledge about buyers' behavior. In this case, the transition structure above is not known and one has to use adaptive mechanisms based on reinforcement learning. The reinforcement learning to be described here for the above game is similar in spirit to the one in Ravikumar et al [28]. To motivate the algorithm, replace the step (**a**) above, by the following iterative scheme to solve the underlying linear system of equations.

(**a'**) $\overline{V}^n_{m+1}(i) =$

$$\sum_{p_2} \pi^2(i,p_2)[P_{ij}([\hat{\pi}^0(i),p_2])S^1_p(i,[\hat{\pi}^0(i),p_2],j) - G(i,[\hat{\pi}^0(i),p_2]) + \sum_{j\in S} m_{ij}([\hat{\pi}^0(i),p_2])\overline{V}^n_m(j)] \quad (12)$$

Note that this can be considered as a subroutine to perform the task of step (**a**). If the transition structure is not known, then the conditional averaging in (12) cannot be performed. One might then consider replacing the conditional average in (12) by an actual evaluation at states and transition intervals obtained from on-line learning. In other words,

$$V^n_{m+1}(i) = V^n_{m+1}(i) + b^l(v(i,m))I_{\{X_m=i\}}[(S^l_p(i,Z_m,j) - C^l(i,j)\frac{1-e^{\alpha\tau}}{\alpha}H^l(i) + e^{-\alpha\tau}V^n_{m+1}(X_{m+1}) - V^n_{m+1}(i)]$$

where Z_m is the player's policy process at step m and j is the resulting state. Now consider (**b**), the policy improvement step, of the policy iteration, which entails solving an optimization problem for each iteration n. In order to do so, it needs to wait for the policy evaluation step to converge and then reevaluate the new policy again following the above learning procedure. To obviate this difficulty, one may try to execute both the steps together for on-line learning and update policies and values in a *coupled* fashion : the policy is updated using an approximate gradient scheme (to be detailed shortly). The gradient estimate is derived from the available estimates of the value function obtained from the above learning step. But importantly, to underscore the fact that policy update should wait till convergence of step (**a'**), the policy update is run at a slower time scale than the value update, that is, step (**a'**) to get the same effect albeit asymptotically. This notion is formalized later.

Now the above argument assumes that Player 2 follows a fixed stationary strategy. However, if Player 2 were also to learn his best strategy, and hence both hope to head to a Nash equilibrium, then simultaneous adaptation of both the players creates convergence problem. In this case, where both the agents try to learn their Nash equilibrium strategies following best response dynamics, it can be hoped that both will converge to an equilibrium (more so, if it is unique) if Player 1 *sees* Player 2 as quasi-static and Player 2 *sees* Player 1 as playing equilibrium strategy in their pursuit for mutual best responses. With this intuition, we devise two similar actor-critic learners that operate on different time scales for updates. This notion will be made precise shortly. As it is known that existence of equilibrium for the above dynamic pricing game can be ascertained only in the randomized policies (perhaps not in pure strategies), let us extend the domain of optimization in (**b**) to the space of probability measures on action space. Advantageously, this space is convex and hence one can use gradient based numerical schemes to solve the underlying optimization problem at each step of learning. Approximate gradient estimate is provided through step (**a'**). Whenever this updated policy falls out of the boundary of the convex policy space, it is projected back to the convex domain. This procedure is formalized below.

Consider the simplex of probability vectors over the action space A, $\mathbf{P}(A)$. Any stationary randomized policy for Player l is a map $\pi^l : S \to \mathbf{P}(A)$. For $i \in S$, $\pi^l(i)$ is an $|A|$- vector whose components are $\pi^l(i,p), p \in A$. We search for optimal $[\pi^l(i,p)]_{i\in s, p\in A}$ in $(\mathbf{P}(A))^M$. For convenience, we enumerate $p \in A$ as $\mathcal{B} = \{1,2,\ldots,|A|\}$ and index the elements in $\mathcal{B}$ by p in the following portion.

The actor-critic algorithm for player, l, $l = 1, 2$, is defined as follows. Equations (10) and (11) suggest the following update procedures for the critic (*policy evaluator*) and the actor (*policy*) respectively.

For any $i \in S$,

$$V_{n+1}^l(i) = V_l^n(i) + b^l(\upsilon(i,n)) I_{\{X_n=i\}}[(S_p^l(i,Z_n,j) - C^l(i,j) \frac{1-e^{\alpha\tau}}{\alpha} H^l(i) + e^{-\alpha\tau} V_l^n(X_{n+1}) - V_l^n(i)]$$

$$\hat{\pi}_{n+1}^l(i.,) =$$

$$\Gamma(\hat{\pi}_n^l(i,.) + \sum_{p \neq p_0} a^l(\nu(i,p,n)) I_{\{X_n=i, Z_n^l=p\}}[(S_p^l(i,Z_n,j) - C^l(i,j)\frac{1-e^{\alpha\tau}}{\alpha} H^l(i) + e^{-\alpha\tau} V_l^n(X_{n+1}) - V_l^n(i)] e_p)$$

where e_p is the unit vector with value 1 in the $p-th$ position,$\{a^l(n)\}$ and $\{b^l(n)\}$ are the step size parameter sequences satisfying the standard stochastic approximation conditions and $\nu(i,p,n)$ is the number of times (i,p) is encountered in the chain $\{(X_n, Z_n)\}$ and and $\upsilon(i,n)$ is the number of times state i is visited by time n. $\Gamma(.)$ is the projection on to the probability simplex $P_0(A) := \{x : \sum_i x_i = 1, x_i \geq 0, \forall i\}$. Finally, let $\varepsilon \in (0,1)$ be a small positive number. Then, player l picks Z_n^l according to the distribution $\pi_n^{l\,\varepsilon}(X_n,.)$ defined for any $\phi \in (\mathbf{P}(A))^M$, by $\phi^\varepsilon(i,.) := \varepsilon\varsigma + (1-\varepsilon)\phi(i,.)$ where ς is the uniform distribution over A to ensure sufficient exploration. τ above is an appropriately averaged sample transition time.

In addition to the standard conditions on $\{a^l(n)\}$ and $\{b^l(n)\}$ for stochastic approximation schemes, we also require that the sequences $\{a^l(n)\}$ and $\{b^l(n)\}$ satisfy:

$$a^l(n) \quad = o(b^l(n)), l = 1, 2 \quad \text{and} \quad a^1(n) = o(a^2(n)) \quad (13)$$

If one interprets $\{a^l(n)\}$ and $\{b^l(n)\}$ as time scales, then (13) defines three time scales for operation of the two actor-critics; while the two actors operate on different time scales, their respective critics operate on the same time scale faster than their respective actors.

B. Projection Algorithm

Given any vector $\overline{\mathbf{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$, for finding its projection, $\Gamma\overline{\mathbf{a}}$ on to the probability simplex $\mathcal{P}$, we repeat the following two steps until

Condition 1: $\overline{a}_i \geq 0$ and

Condition 2: $\overline{a}_1 + \ldots + \overline{a}_n = 1$ are satisfied.

Step1: Truncate all negative components to zero

Step2: If $\overline{a}_1 + \ldots + \overline{a}_n > 1$, then $\overline{a}_i = \overline{a}_i - (\overline{a}_1 + \ldots + \overline{a}_n - 1)/N^+$, where N^+ is the number of positive components in $(\overline{a}_1 + \ldots + \overline{a}_n)$, now verify the two conditions.

Lemma 1: The above algorithm gives a feasible solution within n steps.

Proof: See appendix.

C. Simulation Experiment for a Two Seller Market with Partial Information

We assume the action (price) set A = {8.0, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5}, maximum queue lengths are 5 at each queue, and maximum inventory is 10 at each seller, with fixed reorder point $r = 5$ at each seller. The rest of the system parameters are identical to that of the previous model in Section IV.C. We use two actor-critic learners with learning rate parameters $a^1(n) = 1/n, a^2(n) = 1/n^{1.5}, b^1(n) = 1/n^{0.6}$ and $b^2(n) = 1/(n^{0.6} + 10)$. The discount factor α is set at 0.0001. The system is simulated over one million iterations. Figure 5 shows the convergence and stability of learning of randomized policies at state ([5 5 0] [0 0 10]) for seller 1. The trends for seller 2 are similar. The results provide empirical evidence to our convergence argument in Section V.A. The convergence and stability of learning of randomized policies support our hope that the pricing strategies of both the sellers converge to an equilibrium dynamic pricing strategy in their pursuit for mutual best responses.

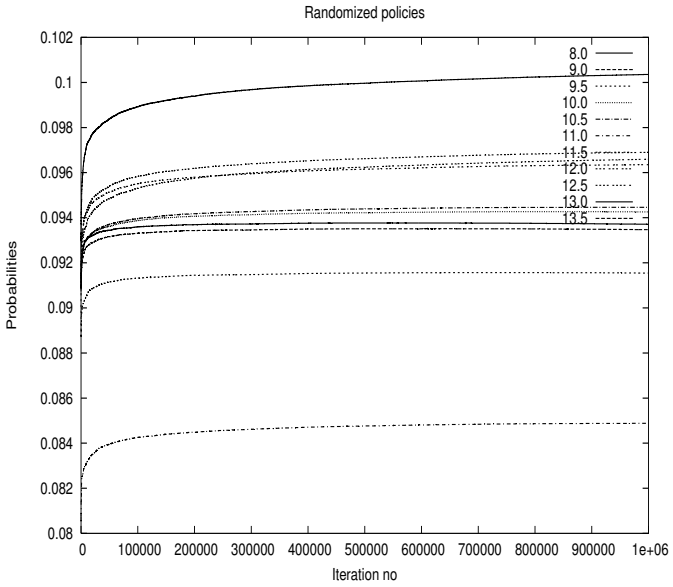

Fig. 4. Convergence of randomized policies of seller 1 at state ([5 5 0] [0 0 10])

Figure 6 shows the convergence of the value functions at state ([5 5 0] [0 0 10]) for seller 1 and seller 2. Since the two sellers are symmetric in all aspects, the value functions converge to almost the same value.

Table III shows the converged strategies of the two sellers for two randomly picked states ([5 5 0][0 0 10]) and ([0 0 10][5 5 0]). Each entry in the table represents the probability of the designated seller selecting the designated price (action) in the designated state. It is interesting to note that at state ([5 5 0][0 0 10]), seller 2 randomizes his prices in high price domain and seller 1 randomizes his prices in low price domain. This result can be explained as follows: since the store of seller 1 is over-crowded, he tries to encourage incoming buyers by displaying a lower price than the opponent. On the other hand, seller 2 does not have enough customers, and he can supply the items with zero lead time, so he can display a high price for incoming buyers. A reverse trend can be observed in state ([0 0 10][5 5 0]).

TABLE III

PROBABILITIES OF THE SELLERS SELECTING INDIVIDUAL PRICES (ACTIONS) IN DESIGNATED STATES

Price (Action)	Policy of Seller 1 in state ([0 0 10] [5 5 0])	Policy of Seller 2 in state ([0 0 10] [5 5 0])	Policy of Seller 1 in state ([5 5 0] [0 0 10])	Policy of Seller 2 in state ([5 5 0] [0 0 10])
8.0	0.0	0.1402	0.0574	0.0000000596
9.0	0.0785	0.0000026226	0.0937	0.0
9.5	0.0834	0.1171	0.0934	0.0
10.0	0.0841	0.2966	0.0915	0.0
10.5	0.0914	0.0000048212	0.0942	0.0
11.0	0.0797	0.0464	0.09446	0.0000046158
11.5	0.0808	0.0749	0.0848	0.0
12.0	0.0804	0.0951	0.0963	0.0000025397
12.5	0.0804	0.0983	0.0969	0.5041
13.0	0.2258	0.08771	0.0965	0.3239
13.5	0.1198	0.0432	0.1003	0.1719

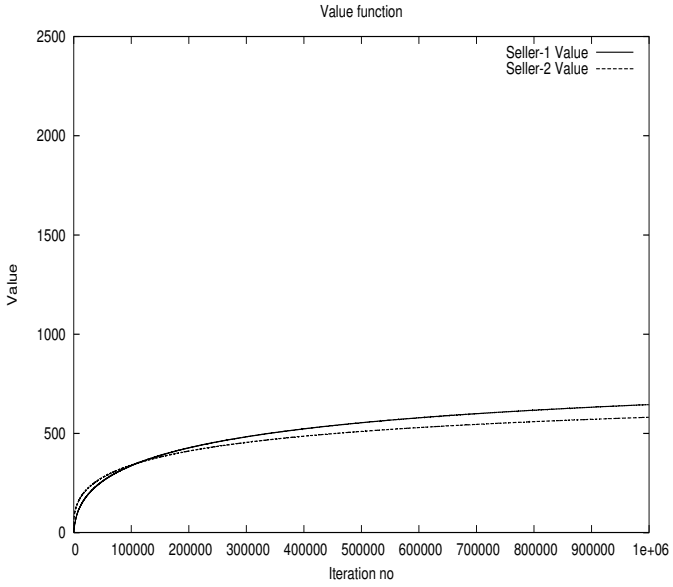

Fig. 5. Convergence of value functions of the sellers at state ([5 5 0] [0 0 10])

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we used reinforcement learning (RL) as a tool to study price dynamics in an electronic retail market consisting of multiple competing sellers stochastic demands, price sensitive and lead time sensitive customers, and inventory replenishments. In such a generalized setting, RL techniques have not been applied before. We considered two representative cases: (1) *no information case*, where none of the sellers has any information about states and prices of the competitors; and (2) *partial information case*, where every seller has information about the customer queue levels and inventory levels of all the competitors. In the *no information case*, we used the Q-learning algorithm for a distinguished pricebot and compared the performance with that of other pricebots running other adaptive techniques such as derivative following (DF). In the *partial information case*, we considered a two seller problem and modeled it as a Markovian game and formulated the problem in the RL framework. We used actor critic algorithms as the solution approach. We believe our approach to solving these problems is a new promising way The models can be generalized to the case of more than two sellers.

There are several directions for future work. First of all, some of the assumptions made by us in the retail market model need to be relaxed: (1) nature of volume discounts (2) nature of inventory policy (3) assumptions regarding shoppers and captives. Secondly, safety stock to be maintained by sellers can be introduced as a decision variable and the model will then become much more interesting and complex. Convergence of the learning algorithms used is another important area of investigation which has engaged researchers in machine learning for quite sometime now. Connections of our work in Section V to the work by Leslie and Collins [24] also need to be investigated.

REFERENCES

[1] G. A, J. O.Kephart, and G. J. Tesauro, "Strategic pricebot dynamics," in *Proceedings of the First ACM Conference on Electronic Commerce (EC-99)*, 1999.

[2] V. S. Borkar, "Reinforcement learning in markovian evolutionary games," *Advances in Complex Systems*, vol. 5, pp. 55–72, 2002.

[3] C. Brooks, R. Fay, R. Das, J. K. MacKie-Mason, J. Kephart, and E. Durfee, "Automated strategy searches in an electronic goods market: Learning and complex price schedules," in *Proceedings of the First ACM Conference on Electronic Commerce (EC-99)*, 1999, pp. 31–40.

[4] A. Carvalho and M. Puterman, "Dynamic pricing and reinforcement learning," Department of Statistics, University of British Columbia, Vancouver, Canada, Tech. Rep., 2003.

[5] L. Chan, Z. J. M. Shen, D. Simchi-Levi, and J. Swann, "Review of dynamic and online pricing research to improve supply chain performance," in *Handbook on Supply Chain Analysis in the eBusiness Era*. Kluwer Academic Publishers, 2001.

[6] L. Chan, D. Simchi-Levi, and J. Swann, "Dynamic pricing strategies for manufacturing with stochastic demand and discretionary sales," Industrial Engineering and Operations Research, Northwestern University, USA, Tech. Rep., 2002.

[7] P. Dasgupta and R. Das, "Dynamic pricing with limited competitor information in a multi-agent economy," in *Conference on Cooperative Information Systems*, 2000, pp. 299–310. [Online]. Available: citeseer.nj.nec.com/dasgupta01dynamic.html

[8] J. DiMicco, A. Greenwald, and P. Maes., "Dynamic pricing strategies under a finite time horizon," in *Proceedings of the Third ACM Conference on Electronic Commerce (EC-01)*, 2001, pp. 51–60.

[9] J. M. DiMicco, A. Greenwald, and P. Maes, "Learning curve:

A simulation-based approach to dynamic pricing," 2002. [Online]. Available: citeseer.nj.nec.com/563289.html

[10] W. Elmaghraby and P. Keskinocak, "Dynamic pricing: Research overview, current practices and future directions," Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA, Tech. Rep., 2002.

[11] A. Federgruen, "On n-person stochastic games with denumerable state space," *Advances in Applied Probability*, vol. 10, pp. 452–471, 1978.

[12] A. Federgruen and A. Heching, "Combined pricing and inventory control under uncertainty," *Operations Research*, vol. 47, pp. 454–475, 1999.

[13] G. Gallego and G. van Ryzin, "Optimal dynamic pricing of inventories with stochastic demand over finite horizons," *Management Science*, vol. 40, no. 8, pp. 999–1020, 1994.

[14] M. Gupta, K. Ravikumar, and M. Kumar, "Adaptive strategies for price markdown in a multi-unit descending price auction: A comparative study," in *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, 2002, pp. 373–378.

[15] S. Hart and A. Mas-colell, "Uncoupled dynamics do not lead to nash equilibrium," *American Economic Review*, vol. 93, pp. 1830–1836, 2003.

[16] P. Hong, R. P. McAfee, and A. Nayyar, "Equilibrium price dispersion with consumer inventories," *Journal of Economic Theory*, vol. 22, pp. 1–15, 2001.

[17] J. Hu and M. P. Wellman, "Multiagent reinforcement learning: theoretical framework and an algorithm," in *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1998, pp. 242–250. [Online]. Available: citeseer.nj.nec.com/hu98multiagent.html

[18] J. Hu and Y. Zhang, "Online reinformcenet learning in multiagent systems," William E. Simon Graduate School of Business Administration, University of Rochester, Rochester, New York, USA, Tech. Rep., 2002.

[19] S. Karlin and C. Carr, "Prices and optimal inventory policies," *Studies in Applied Probability and Management Science*, 1962.

[20] J. O. Kephart and G. J. Tesauro, "Pseudo-convergent Q-learning by competitive pricebots," in *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000, pp. 463–470. [Online]. Available: citeseer.nj.nec.com/306829.html

[21] V. R. Konda and V. S. Borkar, "Actor-critic type learning algorithms for markov decision processes," *SIAM Journal on Control and Optimization*, vol. 38, pp. 94–123, 1999.

[22] A. Lau and H. Lau, "The newsboy problem with price-dependent demand distribution," *IIE Transactions*, vol. 20, pp. 168–175, 1988.

[23] R. Lawrence, "A machine-learning approach to optimal bid pricing," IBM," Research Report, July 2002.

[24] D. Leslie and E. Collins, "Convergent multiple-timescales reinforcement learning algorithms in normal form games," *Annals of Applied Probability*, vol. 13, pp. 1231–1251, 2003.

[25] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, pp. 157–163.

[26] ——, "Friend-or-foe q-learning in general-sum games," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 322–328.

[27] J. Mcgill and G. van Ryzin, "Revenue management: Research overview and prospects," *Transportation Science*, vol. 33, no. 2, pp. 233–256, 1999.

[28] K. Ravikumar, G. Batra, and R. Saluja, "Multi-agent learning for dynamic pricing games of service markets," *Communicated*, 2002.

[29] S. Salop and J. Stiglitz, "The theory of sales: A simple model of equilibrium price dispersion with identical agents," *The American Economic Review*, vol. 72, no. 5, pp. 1121–1130, 1982.

[30] C. Shapiro and H. Varian, *Information Rules*. Cambridge, MA, USA: HBR Press, 1998.

[31] B. Smith, D. Gunther, B. Rao, and R. Ratliff, "E-commerce and operations research in airline planning, marketing, and distribution," *Interfaces*, vol. 31, no. 2, 2001.

[32] M. Smith, J. Bailey, and E. Brynjolfsson, *Understanding Digital Markets: Review and Assessment*. Cambridge, MA: MIT Press, 2000.

[33] M. Sridharan and G. J. Tesauro, "Multi-agent q-learning and regression trees for automated pricing decisions," in *In Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000.

[34] G. Stigler, "The economics of information," *Journal of Political Economics*, vol. 69, pp. 213–225, 1961.

[35] J. Stiglitz, "Equilibrium in product markets with imperfect information," *American Economic Review Proceedings*, vol. 69, pp. 339–345, 1979.

[36] J. Swann, "Dynamic pricing models to improve supply chain performance," Doctoral Dissertation, Northwestern University, Tech. Rep., 2001.

[37] ——, "Flexible pricing policies: Introduction and a survey of implementation in various industries," General Motors Corporation, Tech. Rep. Contract Report # CR-99/04/ESL, October 1999.

[38] G. Tesauro and J. O. Kephart, "Pricing in agent economies using multi-agent q-learning," in *Proceedings of Workshop on Decision Theoretic and Game Theoretic Agents*, London, England, 1999.

[39] ——, "Pricing in agent economies using neural networks and multi-agent q-learning." in *In Proceedings of Workshop ABS-3: Learning About, From and With other Agents (held in conjunction with IJCAI'99)*, Stockholm, Sweden, 1999.

[40] G. Thowsen, "A dynamic nonstationary inventory problem for a price/quantity setting firm," *Navel Research Logistics Quarterly*, vol. 22, pp. 461–476, 1975.

[41] H. R. Varian, "A model of sales," *The American Economic Review*, pp. 651–659, 1980.

[42] ——, "Differential pricing and efficiency," *First Monday*, vol. 1, 1996. [Online]. Available: www.firstmonday.dk

[43] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[44] E. Zabel, "Monopoly and uncertainty," *The Review of Economic Studies*, vol. 37, pp. 205–219, 1970.

APPENDIX

Proof of Lemma 1: The optimization problem is a constrained nonlinear optimization problem. It is in fact optimization of a convex function on a convex set, and therefore has a unique minimum. That is *Kuhn-Tucker* conditions are both necessary and sufficient to characterize the solution. One can write down these conditions and observe that the following procedure gives the solution. There are two ways in which the constraints of the probability simplex can be violated.

1) Some of the components can be negative.
2) The sum of all components is greater than one.

It is easy to repair the first violation by truncating all negative components to zero. But doing this can make the resulting vector violate the second constraint even though the original vector did not violate the second constraint. Therefore we do the following step. Assuming only the second constraint is violated, it can be repaired by dividing the excess sum $(i.e., \overline{a}_1 + \ldots + \overline{a}_n - 1)$ by the number of positive components in $(\overline{a}_1, \ldots, \overline{a}_n)$ and subtracting this from all the positive components. Even though this step does not touch the components that are already zero, it can introduce new negative components (for example, if there are positive components that are very small compared to the rest of the positive components). In this case, we go back to the first step and repeat these two steps alternately until we have a feasible solution. So, we are guaranteed to have a feasible solution within n steps as at least one component will become zero in an iteration and once a component becomes zero it remains zero throughout the procedure. ■