IBM Research Report

Reinforcement Learning for Dynamic Pricing in Service Markets

K Ravikumar
IBM Research Division
IBM India Research Lab
Block I, I.I.T. Campus, Hauz Khas
New Delhi - 110016. India.

Gaurav Batra
Mechanical Engineering Department
Indian Insitute of Technology
New Delhi - 110 016, India

Rohin Saluja
Mechanical Engineering Department
Indian Insitute of Technology
New Delhi- 110 016, India

IBM Research Division
Almaden - Austin - Beijing - Delhi - Haifa - T.J. Watson - Tokyo - Zurich

Reinforcement Learning for Dynamic Pricing in Service Markets

K Ravikumar RKARUMAN@IN.IBM.COM
IBM India Research Laboratory, Indian Institute of Technology, New Delhi, India 110 016

Gaurav Batra GAURAV.BATRA@SOFTHOME.NET
Mechanical Engineering Department, Indian Institute of Technology, New Delhi, India 110 016

Rohin Saluja ROHINSALUJA@YAHOO.COM
Mechanical Engineering Department, Indian Institute of Technology, New Delhi, India 110 016

Abstract

We study price dynamics in an electronic service market environment consisting of buyers and competing service providers. In this market, each service provider has limited capacity to serve the buyers. We present price dynamics in a two-seller market when buyers use comparison shopping agents to know about price and expected delay at each service provider. Each seller uses an automated pricing agent to reset the price at random intervals in order to maximize his expected profits. A Q-learning algorithm for pricing agent is developed and comparative experimental study on various other adaptive strategies is presented.

Further, we present a new multi-time scale actor-critic-type algorithm for multi-agent learning in the underlying stochastic games. Preliminary experimental results on convergence of the proposed algorithm in a degenerate version of the dynamic pricing game and also on convergence of the algorithm in iterated general-sum bi-matrix games are presented.

1. Introduction

E-commerce has undoubtedly changed how the business is done. On the Internet, competition is just a click away. This fact has potentially lead to intense price competition for commodity products. Search engines like ASCES, and Web-based comparison shopping agents (also known as *shopbots*), like Dealpilot.com allow consumers easy access to all competing firms' prices. In order to attract consumers, sellers use automated pricing agents, (also called *pricebots*) for constant resetting of prices. Kephart and Greenwald [1999] have investigated "economics of shopbots" and pricebot dynamics. In their models, some consumers have access to shopbots while other consumers do not. These models generate equilibria: firms randomize their prices in order to price discriminate between the searchers and non-searchers.

For studies on such temporal price dispersion model of "holding" sales in the presence of such a mix of "informed" and "uninformed" consumers, see for instance, Varian [1980] in a physical retailer market setting and Greenwald and Kephart [1999], Greenwald, Kephart and Tesauro [1999], Dasgupta and Das [2000], for its online counterpart. In the same vein, in this paper, we study price dynamics in an electronic service market with sellers of identical service. This models a situation where online utility services or digital goods or videos are offered on rent.

Previous studies particularize to a situation in commodity markets where it is assumed that "supply" ("or capacity) is infinite and hence can "hold" sales. In contrast, seller of a service can process requests of consumers only at a finite rate. As a result, a buyer approaching for service will incur waiting cost before his request is initiated. In our model, we assume that shopbots not only will collate *posted prices* of all the sellers but also provide information pertaining to *posted* expected delay at each such service provider. Further, each service provider uses automated pricing agents (or pricebots) to reset prices whenever an arrival or departure happens.

In this paper, we aim to identify those pricebot algorithms that are most likely to be profitable in a competitive scenario. To this end, we experiment with various adaptive strategies that differ in their information requirements. Further, we consider the market with two sellers and we pose the dynamic pricing problem as multi-agent reinforcement learning problem in stochastic games. We develop a Q-learner that is oblivious of opponent's prices and delays and present exper-

imental results when such a Q-learner is pitted against other adaptive strategies. Also, in the Multi-Agent Reinforcement Learning case, we develop new actor-critic-type of learners, a variant of the type discussed in Konda and Borkar [2001]and Borkar [2002]. We model two players as two actor-critic learners, but the actors (policies) are updated on different time scales. Intuition behind such update is as follows: If two actors run on different time scales, the slower player sees the other player as "equilibrated" and the faster player sees the other player as quasi-static.

Reinforcement Learning as a paradigm for multi-agent learning in stochastic games has been studied by Littman [1994] in zero-sum games and Patek and Bertsekas [1999] in zero-sum stochastic- path games using minmax-Q learning that is shown to converge. Nash -Q learning for general-sum games of Hu and Wellman[1998] imposes many restrictive assumptions for convergence whereas more general and convergent Friend or Foe Q-learning of Littman [2001] requires information with regard to opponent: friend or foe and uses Nash-Q or minimax-Q accordingly. Even in the iterated game cases, no algorithms with guaranteed convergence are known to exist. In the complete information iterated two-action bi-matrix games , Singh, Kearns and Mansour (1999), develop a gradient ascent algorithm with constant steps, and show that either the agents converge to a Nash equilibrium or their average pay-offs will converge to the pay-offs correspoding to a Nash equilibrium. Bowling and Veloso (2001) modify the above algorithm to include steps that vary with time to show convergence.

Our proposed algorithm differs from the above works in the following ways: Firstly, all the above algorithms follow the philosophy of value iteration scheme of-Markov decision processes (or more generally, Markovian games). At every step of learning such schemes involve solving Linear Program (in the case of Zero sum games or Foe learning) or a quadratic program (in Nash Q learning) to identify the policy for next step of learning. Further, in Nash-Q learning one needs to maintain estimates of Q-values of the opponent. In this paper we give an actor-critic type of learner (Barto, Sutton and Anderson [1983], Konda and Borkar [2000]), a derivative of policy-iteration scheme, that maintains values as well as policy and the updates move in a coupled fashion albeit on different time scales. Further, it does not entail maintaining estimates of opponent's pay-offs as in Nash-Q learning. In the iterated bi-matrix game scenario, our algorithm is general enough to handle multiple action incomplete information general-sum two-player games. In such iterated game cases, the value update procedure (critic update) degenerates to a simple stochastic gradient based scalar update.

In our computational experiments, we report results on performance of Q-learner against other adaptive agents. Average profits from Q-learning are far above those from other strategies. In multi-agent reinforcement learning with the proposed actor-critic learners, we only report preliminary results obtained over a degenerated case of dynamic pricing problem, wherein it is assumed that in each state of the pricing game, the pay-off matrix is the same. However, the pay-off matrix considered for experimentation is a fairly complex six-action general-sum bi-matrix game (an example from Mangasarian and Stone [1964]) with no apparent special structure. The algorithm converges exactly to the unique Nash equilibrium mentioned therein.

In iterated game cases, though our experimentation is extensive, for space reasons, we report only results on convergence of the proposed algorithm on a constant-sum game and iterated bi-matrix game an example case presented in Bowling and Veloso [2001] (that exposes some difficulties involved with arbitrary starting strategies in identifying "winning" position in their WoLF algorithm).

The paper is organized as follows. In the next section we introduce the dynamic pricing model in service markets. Section 3 develops various adaptive strategies including the opponent-oblivious Q-learning algorithm for the underlying semi-Markov decision model. In Section 4 we give a formal description of a two-player stochastic game and present the multi-time scale actor-critic algorithm. Section 5 gives results of our experimentation.

2. Description of the Model

We consider a simple model of a service market with two service providers. A Poisson stream of buyers with rate λ approaches the market with i.i.d service time requirements sampled from a distribution $F(.)$ with finite support having mean $\frac{1}{\mu}(\lambda < \mu)$. Buyers are classified into two categories: A Type 1 buyer randomly chooses a service provider and requests for a quote on price per unit service and the expected delay to be incurred to initiate processing his request. In contrast, Type 2 buyers, use a *shopbot*, to know *posted price* quotes of all the sellers and also the posted expected delay at each such individual service provider. In both the cases, a seller with n requests queued up, will quote a delay of $n\mu$. Associated with each buyer is a utility function that combines price and delay in a form to be described shortly. Each buyer has his own upper-

limits, p_b, w_b, on price and waiting time respectively. These are assumed to be *i.i.d* and uniform and are sampled respectively from $U(0, p_{max}]$ and $U(0, w_{max}]$ for known p_{max} and w_{max}.

A Type 1 buyer joins the queue of the selected seller only when his utility is positive and leaves the market otherwise. But a Type 2 buyer joins a queue (seller) using a greedy policy that maximizes his utility computed from posted-price and posted-delay quotes contingent on the utility being positive or else leaves the system. We assume that the probability that an arriving buyer is of Type i is ω_i, $i = 1, 2$ and $\omega_1 + \omega_2 = 1$. For simplicity, we assume that each seller can process only one request at a time and further that buffer where the requests queue up has finite capacity. Each seller uses his own automated price-setting agent, a *pricebot*, to price the requested service dynamically based on competitive factors, current queuelength and also, based on relative proportion of the informed buyers (Type 2 above) approaching the market.

A typical buyer's utility is a composite function that encompasses the buyer's individual preferences for price and waiting time and is assumed to be of the following form:

$$U(p, w) = [\alpha(p_b - p) + (1-\alpha)(w_b - w)]\Theta(p_b - p)\Theta(w_b - w) \quad (1)$$

where $\Theta(x) = 1$ if $x \geq 0$ and $\Theta(x) = 0$ otherwise. for any quoted price p and waiting time w.

Further, assume that all sellers have identical service cost per unit of service.

The pricing strategies are developed for two different cases depending on the type of market information available to sellers.

Case 1: Complete Information : This models a situations where each service provider has complete information (or perhaps uses a shopbot!) about other service provider's prices and queue lengths and also, about buyer population and there preferences.

Case 2: No Information: For this case, we assume that each service provider is oblivious of other sellers' prices and queue lengths and is also ignorant about buyer characteristics.

For reinforcement learning, we develop a Q-learning algorithm for the opponent oblivious case (Case 2 above) using semi-Markov decision model. Also, taking a cue from recent works of Kephart et al [1999,2000], we develop various other fairly robust adaptive strategies that differ in their informational requirements and analyze dynamics when a Q-learner is pitted against an opponent that uses such adaptive strategies. These strategies are described in next section.

For two-agent reinforcement learning, we direct the reader to Section 4.

3. Adaptive Strategies

3.1 Myopic or Myopically Optimal (MY) Strategy

This strategy is applicable in Case 1 discussed above. This strategy performs an exhaustive search over a discrete price space for a price that maximizes its immediate expected profit and hence is myopic. An agent following this strategy is equipped with complete information about competitor and buyers but assumes that competitors prices are static until his next decision , and hence a change in its prices will elicit no response from the competitor.

This strategy works as follows:

Step 1 Compute the expected profit, π_1, by setting its price at a level, say p_1, which just exceeds the utility to the buyers offered by the competitor.

Step 2 Compute the price p_{1s} at which maximum expected profit, say, π_2, will be achieved serving only buyers who use the random strategy (Type 1).

If the price found in Step 2 is lesser than the competitor's price, then the current price is set at this value. This will ensure maximum profit from all types of buyers. If the price found in Step 2 is more than the competitor's price, set the current price value at the price that gives maximum expected profit as the current price.

We provide below the computation details involved in the above procedure.

Let p_2 and $w_2$be the competitor's price and waiting time quote respectively. If the current delay at the seller in initiating processing of a newly arriving customer is w_1, then its price quote p_1 for the case in Step 1 can be computed as follows:

The utility values of an arriving buyer at the seller and the competitor are respectively :

$U_1(p_1, w_1) = \alpha(p_b - p_1) + (1-\alpha)(w_b - w_1)$, $U_2(p_2, w_2) = \alpha(p_b - p_2) + (1-\alpha)(w_b - w_2)$

if both p_1, p_2, w_1, w_2 are less than their respective upperlimits.

Now if the seller wants to set a price at a value that offers a marginal increase in utility value of the up-

coming buyer over his utility value corresponding to the competitor, then such a price can be computed as follows. Let $\varepsilon > 0$. We need to find a price p for the seller such that

$$U_1(p) = U_2(p_2, w_2) + \varepsilon \quad (2)$$

In other words,

$$p_1 = [\frac{\alpha}{1-\alpha}(w_2 - w_1) + p_2] - \varepsilon$$

In our experiments we randomize over ε. The expected profit corresponding to the above price, $\pi_1(p_1, w_1)$ can be computed from our distributional assumptions. For space reasons we omit the exact expression for $\pi(p_1, w_1)$.

Now we briefly give details of the procedure underlying the computation of price p_{1s} mentioned in Step 2. Note that a randomly arriving buyer will join the queue of the seller only when his computed utility corresponding to the seller's price quote and delay is positive. That is, for a quoted price p when the delay is w, a type 1 buyer with price limit x and waiting time limit w_b will join the queue only when

$$U(p, w) > 0, \;\; w_b > w \;\; \& \;\; p_b > p$$

Note that, $U(p, w) > 0$ hold true when,

$$w_b > w + \frac{(x-p)\alpha}{(1-\alpha)}$$

Noting that p_b and w_b are uniform random variables on $(0, p_{max}]$ and $(0, w_{max}]$, respectively, it is easy to see that for a price quote and delay pair (p, w), probability that the utility of a buyer is positive is given by:

$P(U(p, w) > 0) =$

$$\int_0^{p_{\max}} \frac{1}{p_{\max} w_{\max}} (w_{\max} - w + \frac{(x-p)\alpha}{(1-\alpha) w_{\max}}) dx$$

Let $E[\pi(p, w)]$ denote the expected profit obtained by the seller for a random strategy buyer at the quoted price p and when delay at the seller is w. Then,

$\mathrm{E}\pi(\mathrm{p,w}) = P(U(p, w) > 0)(p - c)(1 - \frac{w}{w_{\max}})(1 - \frac{p}{p_{\max}})$.

First order conditions for optimal p entail solving a quadratic expression of the following form:

$$a'p^2 + b'p + c' = 0 \quad (3)$$

where, $a' = \frac{-3a}{p_{\max}}$, $b' = 2(\frac{ac}{p_{\max}} + a - \frac{b}{p_{\max}})$, $c' = -ac + \frac{bc}{p_{\max}} + b$ with a and b as given below:

$$a = \frac{-\alpha}{(1-\alpha) w_{\max}} \quad , \quad b = (1 - \frac{w}{w_{\max}}) - \frac{a p_{\max}}{2}$$

Now using the value of p obtained from (3), one can derive optimal expected profit for a random strategy buyer π_l and also π_2 following similar arguments.

Now the Myopic policy is to set the ongoing price at a new level p_0 defined by:

$p_0 = p_{1s}$ if $p_{1s} < p_2$ or $\pi_1 < \pi_2$

$\quad = p_1$ if $\pi_1 > \pi_2$

The Myopical optimal strategy described above requires knowledge of buyers utility functions, competitor's price and delay at the competitor. In the forthcoming sections, we develop adaptive strategies for no-information case.

3.2 Adaptive Strategies in No-Information Case

In the absence of information about competitors' prices and delays and any means to measure buyers' preferences, past dynamics will help decide future course of action. In this section, we devise few such strategies which differ in their fore-sightedness and also in their computational requirements.

3.3 Derivative Follower

This is the simplest practicable dynamic pricing strategy and is least computationally intensive. This experiments with experimental increases/decreases in prices till observed profit falls, after which the direction of movement is reversed. It requires keeping track of past average profits for each value of queue length and increases the prices till the profitability level falls. Or more explicitly, the price setting is according to:

$p_{t+1}(w) =$

$p_t(w) + \delta_t sign(\pi_t(w) - \pi_{t-1}(w)) sign(p_t(w) - p_{t-1}(w))$

where $\pi_t(w)$ is the profit made by the seller during time t when the expected waiting time was w. δ_t is the step size parameter and is distributed uniformly between $[a, b]$ for a judicious choice of parameters a,$b > 0$.

3.4 Model Optimizer with Exploration

This strategy attempts to utilize the statistical data available effectively. Instead of single step history, this uses a multi-step history to decide right price. This is implemented using a polynomial regression of average profits corresponding to each state over a fixed number of previous prices relationship as in Das et.al[2000].

The model is built to minimize the least square error and then the constructed model is used to determine

optimal price to quote corresponding to that state. The model is refined periodically when enough new information arrives. The number of steps to look or the degree of the polynomial is based on trade-offs on increased profit and increased computational and memory requirements.

In brief, the seller uses his price(p), queue length (q) and the measured profits(π). Later, a (polynomial) regression of average profit $\pi_{t,q}$ on price $p_{t,q}$ for an observed queue length q is performed at time t as given below:

$$\pi_{t,q} = c_0 + c_1 p_{t,q} + c_2 p_{t,q}^2 + ...c_r p_{t,q}^r$$

The deviation, ε_t, at any decision epoch is

$$\varepsilon_t = c_0 + c_1 p_{t,q} + c_2 p_{t,q}^2 + ...c_r p_{t,q}^r - \pi_{t,q}$$

The coefficients are chosen so as to minimize the the least squared error. Historical data corresponding to the same queue length as the current observed queue-length is used for regression.

In our experiments we supplement the regression equation with an exploration phase to find optimal price. Exploration is initiated when the model optimizer fails to find a price in the immediate vicinity of the ongoing price in any state. Exploration selects a price from a uniform distribution over (a, b), an interval around the ongoing price.

3.5 Q-Learning

Observe that a seller's learning problem when his opponent follows a stationary (perhaps randomized) strategy is a learning problem in semi-Markovian decision processes. Since we assume no information about opponent's strategies and buyers preferences, we develop a Q-learning algorithm on "states" of the seller only.Opponent's strategies and buyers' behaviour get reflected in the reward obtained and transitions occurred. Let $r(i, u)$ denote the rate at which reward accumulates when action u is used.

We use queue-length at the seller as state in the following and actions refer to price to be set. Probability of transitions $P_{ij}(u)$ and and time to transition, from i to j are function of the seller's action u, the latter having distribution $F_{ij}(u)$. Opponent's strategy has indirect influence on these distributions which is not explicitly accounted in the Q-learning procedure stated below.

For any stationary policy, π, its value, the discounted expected reward, as:

$V_\pi(i) =$

$$\sum_{j\in S} P_{ij}(\pi(i)) \int_0^\infty \int_0^i e^{-\beta s} r(i, \pi(i)) ds dF_{ij}(t|\pi(i)) +$$

$$\sum_{j\in S} P_{ij}(\pi(i)) \int_0^\infty e^{-\beta t} V_\pi(j) dF_{ij}(t|\pi(i))$$

If the expected reward associated with an action u during transition from i to j is denoted by $R(i, j, u)$ and the expected discounted factor $\int_0^\infty e^{-\beta t} dF_{ij}(t|u)$ is denoted by $\gamma(i, j, u)$, then the above expression can be rewritten as

$$V_\pi(i) = \sum_{j\in S} P_{ij}(\pi(i)) R(i, j, \pi(i)) + \sum_{j\in S} P_{ij}(\pi(i)) V_\pi(j) \gamma(i, j, \pi(i)) \quad (4)$$

where $R(i, j, u)$ is the expected reward received on transition from i to j under action u.

Now define $V^*(i) = \sup_\pi V_\pi(i)$ and the Bellman's optimality condition is

$V^*(i) =$

$$\max_u (\sum_{j\in S} P_{ij}(u) \int_0^\infty \int_0^i e^{-\beta s} r(i, u) ds dF_{ij}(t|u) +$$

$$\sum_{j\in S} P_{ij}(u) \int_0^\infty e^{-\beta t} V^*(j) dF_{ij}(t|u))$$

Define term inside the braces as $Q^*(i, u)$. Proceeding along the lines of Q-learning for MDPs, Q-value update procedure can be written as follows:

$Q^{n+1}(i, u) =$

$$Q^n(i, u) + \alpha_n(i, u) \left(\frac{1-e^{-\beta\tau}}{\beta} \psi(i, j, u) + e^{-\beta\tau} \max_b Q^n(i, b) - Q^n(i, u) \right) \quad (5)$$

where the sampled transition time from state i to state j is units and the term $\frac{1-e^{-\beta\tau}}{\beta}\psi(i, j, u)$ is the sample reward during the time τ units and the term $e^{-\beta\tau}$ is the sample discount on the *value* of the next state given a transition time of τ units.

Prices are discretized in multiples of $\delta > 0$ so that the number of actions are finite. The above Q-learning is implemented using Gibbs functions for exploration.

4. Multi-agent Reinforcement Learning: An Algorithm

The dynamic pricing problem with two-agent learning is a learning problem in semi-Markovian game. In this section, we depart from semi-Markovian treatment and consider only the simple discrete version.

Consider a stochastic game with two players (agents). Let their control processes be $\{Z_n^i\}, i = 1, 2$ taking

values in A their common action space. Let the finite state space of the game be denoted by S. The transition probability of the underlying state process is according to the following conditional law:

$$P(X_{n+1} = j | X_m, Z_m, m \leq n) = p(X_n, Z_n, j) \; j \in S.$$

where $p : S \times A^2 \times S \rightarrow [0,1]$ such that $\sum_j p(i, \bar{u}, j) = 1 \quad \forall \; \bar{u} \in A^2$.

Agent $i, i = 1, 2$ seeks to minimize his costs or maximize his pay-offs:

$$E[\sum_{n=0}^{n=\infty} \beta^n c^i(X_n, Z_n)$$

for his prescribed pay-off function $c^i : S \times A^2 \rightarrow \mathbf{R}$.

For any $\pi(.,.) = [\pi^1(.,.), \pi^2(.,.)] \in (\mathbf{P}(A))^{2|S|}$, define the transition probabilities and pay-offs for a policy π as

$$\bar{p}(x, \pi, y) = \sum_{a_1, a_2} p(x, [a_1, a_2], y) \pi^1(x, a_1) \pi^2(x, a_2) \; x, y \in S$$
$$\bar{c}^l(x, \pi) = \sum_{a_1, a_2} c^l(x, [a_1, a_2]) \pi^l(x, a_1) \pi^2(x, a_2), \; l = 1, 2$$

Correspondingly, define the policy value function for player l is :

$$V_\pi^l(x) = E[\sum_{m=0}^{m=\infty} \beta^m \bar{c}^l(X_m, \pi) | X_0 = x], \quad x \in S.$$

where Z_n are being chosen according to the stationary randomized policy π. Then $V_\pi^l(.)$ is the unique solution to the fixed point equation:

$$V_\pi^l(x) = c(x, \pi) + \sum_y \bar{p}(x, \pi, y) V_\pi^l(y) \forall x \in S$$

Following Federgruen (1978), we call the policy profile $\pi(.,.)$ a Nash equilibrium of for every i,

$$V_\pi^l(x) \leq V_{\bar{\pi}}^l(x) \quad \forall x$$

whenever, $\bar{\pi}^k(.,.) = \pi^k(.,.)$, for all $k \neq l$.

Such a Nash equilibrium is known to exist. See Federgruen (1978). Moreover, if we freeze policies for one agent, it becomes a Markov Decision Process for the other agent whence it follows that $V_\pi^1(x)$ satisfies the following dynamic programming equation: $\forall x \in S$,

$$V_\pi^1(x) =$$
$$\min_a \sum_{a_2} \pi^2(x, a_2)[c^1(x, [a, a_2]) + \beta \sum_{y \in S} p(x, [a_1, a_2], y) V_\pi^l(y) \tag{6}$$

Similar relation holds for $V_\pi^2(.)$.

Since, existence of a Nash equilibrium is ensured in the mixed strategy domain perhaps not in the pure strategy space, and also since if a player plays a Nash equilibrium strategy, the other player needs to solve an MDP, actor-critic type of learning paradigm is a natural choice for stochastic games to learn such strategies.

Now in the case where both the agents try to learn their Nash equilibrium strategies in a similar fashion, that is, follow the same learning behavior, one can hope that both will converge to the Nash equilibrium (provided it is is unique) if Player 1 *sees* Player 2 as quasi-static and Player 2 *sees* Payer 1 as playing equilibrium strategy in their pursuit for mutual best responses. With this intuition, we devise two similar actor-critic learners where one learner updates its actor on a slower time scale than the other. Further, the critics, that perform their respective actor's policy evaluation run at the same time scale but faster than their respective actors. Formally, we define the actor-critic learners as follows:

Consider the simplex of probability vectors over the action space A, $\mathbf{P}(A)$. Any stationary randomized policy is a map $\phi : S \rightarrow \mathbf{P}(A)$. For $i \in S$, $\phi(i)$ is an $|A|$- vector whose components may be denoted by $\phi(i, a) a \in A$.. We search for optimal $[\phi(i, a)]_{i \in s, a \in A}$ in $(\mathbf{P}(A))^{|S|}$. These being probability vectors it suffices for us to search for optimal $\hat{\phi} = [\phi(i, a)]_{i \in S, a \in A, a \neq a_0}$ for a fixed a_0.

$$V_{n+1}^l(x) = V_n^l(x) + a^l(\upsilon(x, n) I\{X_n = x\}$$
$$(V_n^l(x) - c^l(x, Z_n) - \beta V_n^l(X_{n+1}))$$

$$\hat{\phi}_{n+1}^l(x, .) =$$
$$\Gamma(\hat{\phi}_n^1(x, .) + \sum_{a \neq a_0} b^l(\upsilon(x, a, n)) I\{X_n = i, Z_n^l = a)$$
$$(V_n^l(x) - c^l(x, a) - \beta V_{n+1}^l(X_{n+1})))e_a)$$

where e_a is the unit vector with value 1 in the $a - th$ position,$\{a^l(n)\} \& \{b^l(n)\}$ are the step size parameter sequences satisfying the standard stochastic approximation conditions and $\upsilon(x, a, n)$ is the number of times (x, a) is encountered in the chain $\{(X_n, Z_n)\}$and $\Gamma(.)$ is the projection on to the sub-probability simplex $P_0(A) := \{x : \sum_i x_i \leq 1, x_i \geq 0, \forall i\}$. Finally, let $\varepsilon \in (0, 1)$ be a small positive number. Then, pick Z_n^l

according to the distribution $\phi_n^{l\ \varepsilon}(X_n, .)$ defined for any $\phi^l \in (\mathbf{P}(A))^{|S|}$, by $\pi^\varepsilon(x, .) = \varepsilon\varsigma + (1-\varepsilon)\phi(x, .)$ where ς is the uniform distribution over A to ensure sufficient exploration.

In addition to all the above, we require that the sequences $\{a^i(n)\}$ &$\{b^i(n)\}$ satisfy:

$$\begin{aligned} a^i(n) &= o(b^i(n), i = 1, 2 \\ b^1(n) &= o(b^2(n)) \end{aligned} \tag{7}$$

If one interprets $\{a^i(n)\}$ &$\{b^i(n)\}$ as time scales, then (7) 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.

In the next section, we present our simulation studies over a set of general-sum games that includes a degenerate version of the dynamic pricing problem in which both the players encounter the same pay-off matrix in each state.

5. Simulation Study

The market was simulated for 2 sellers. Buyers arrive in Poisson fashion with rate .5. Service times are deterministic with value equal to 0.9. The production cost of each seller was set at 10 and Upperlimits on price and waiting time for a buyer are uniformly distributed over (0,50]. The fraction of buyers that use random selection of seller is 0.2 In the case of all other adaptive strategies different from Q-learners, the prices are revised after fixed time interval that equals twice the mean inter-arrival time. In the case of Q-learner, the prices are revised whenever state changes: whenever an arrival or departure occurs.

All the developed strategies were pitted against each other to analyze price dynamics. However, we report only the experiments with Q-learner against an opponent with all other strategies. Figure 1 shows the average profit curves. It can be seen from the profits obtained that Q >> MY MOE >DF. That is, Q-learners always yield higher profits as one would expect. The Q-learning based pricing agent will require high memory compared to other strategies. But the computational requirements are minimal compared to other strategies. Further, the profit margins observed may as well substantiate use of such algorithms for pricing.

5.1 Actor-Critic Learner

As mentioned earlier we experimented with the three time scale algorithm on a degenerate dynamic pricing game where in each state (the vector of queulengths), the pay-offs corresponding to different price actions are set equal to corresponding components of the matrix given below, which is taken from Mangasarian and Stone (1964). The price interval $[0, 50]$ is divided into six equally spaced price actions and the pay-off for each price profile is set equal to corresponding value in the matrix. The learning rates are $a^1(n) = \frac{1}{n}, a^2(n) = \frac{1}{n}$ whereas $b^1(n) = \frac{1}{n^{0.6}}$ and $b^2(n) = \frac{1}{n^{0.85}}$ in all the experiments and the discount parameter is set at a value of 0.9 in all our learning experiments.

The learners converge to the unique equilibrium which for the first player is to play action 1 and action 6 with probability 0.5 and for Player 2, to play action 3 and action 4 with probability 0.5. The state transitions depend on the buyers' arrival pattern. The strategies converge to the equilibrium of the stage game as expected in such unique equilibrium games. See Figure 2.

We also experimented with a iterated constant sum game shown in the matrix below which has again unique equilibrium [0.33, 0.67] for both the players. Figure 3 shows convergence behaviour.

In another experiment of general-sum bi-matrix game described in Bowling and Veloso that exposes some difficulties in determining winning position of (their WoLF algorithm, a feature needed for convergence) with arbitrary start states. Here again, the actor-critic shows convergence to the Nash equilibrium. See Figure 4.

We still need to conduct experiments on a truly dynamic game and games with multiple equlibria.

$$\mathbf{A} = \begin{pmatrix} 0 & 0.2 & 0.4 & 0.6 & 0.8 & 1.0 \\ 0.2 & 0 & 0.2 & 0.4 & 0.6 & 0.8 \\ 0.4 & 0.2 & 0 & 0.2 & 0.4 & 0.6 \\ 0.6 & 0.4 & 0.2 & 0 & 0.2 & 0.4 \\ 0.8 & 0.6 & 0.4 & 0.2 & 0 & 0.2 \\ 1.0 & 0.8 & 0.6 & 0.4 & 0.2 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 0 & -0.02 & -0.08 & -0.18 & -0.32 & -0.50 \\ 0 & 0.02 & 0 & -0.06 & -0.16 & -0.30 \\ 0 & 0.06 & 0.08 & 0.06 & 0 & -0.10 \\ 0 & 0.10 & 0.16 & 0.18 & 0.16 & 1.0 \\ 0 & 0.14 & 0.24 & 0.30 & 0.32 & 0.30 \\ 0 & 0.16 & 0.32 & 0.42 & 0.48 & 0.50 \end{pmatrix}$$

General-Sum Bi-Matrix Game (Mangasarian and Stone (1964))

$$\begin{pmatrix} 3,2 & 1,4 \\ 1,4 & 2,3 \end{pmatrix}$$

Constant Sum Bi-Matrix Game

$$\begin{pmatrix} 0,3 & 3,2 \\ 1,0 & 2,1 \end{pmatrix}$$

General-Sum Game (Bowling and Veloso(2001))

6. Conclusions and Future Research

In this paper, we have developed a model for dynamic pricing in service markets and analysed performance of various adaptive strategies. An opponent-oblivious Q-learning strategy has been observed to yield very high profits compared to other adaptive strategies considered in literature. High profit margins obtained will substantiate use of Q-learning based pricing agents in agent-mediated electronic service market domains in future.

For two-agent reinforcement learning, we proposed a multi-time scale actor-critic algorithm. Computational results on convergence in iterated games framework have been very encouraging. Even in the iterated game cases, no general convergence algorithm is known to exist. Multi-time scale actor-critic algorithms of the type presented seem to offer some promise in this direction. Currently, we are experimenting on multiple-equilibria cases. We are yet to develop similar actor-critics that can address semi-Markovian games of our dynamic pricing problem. Reinforcement learning in dynamic games is of great help in e-commerce domain, particularly in dynamic auction/negotiation games. We address these topics in our future work.

References

[1] Barto, A. & Sutton, R. & Anderson, C (1983) Neuron-like Elements That Can Solve Difficult Learning Control Problems, *IEEE Transactions on Systems, Man and Cybernetics*, Vol.13, pp 835-857.

[2] Borkar, V. S.(2002), Reinforcement Learning in Markovian Evolutionary Games, *Advances in Complex Systems*, To appear

[3] Bowling, M & Veloso, M. (2001). Variable Learning Rate and Convergence of Gradient Dynamics, in *Proceedings of the Eighteenth International Conference on Machine Learning*, Williams College

[4] Dasgupta, P & Das, R. (2001). Dynamic Pricing with Limited Competitor Information in a Multi-Agent Economy, *Technical Report, IBM Institute of Advanced Commerce*, IBM Research, Hawthorne, NY, USA

[5] Federgruen,A (1978) *On N-person Stochastic Games with Denumerable State Space*, Advances in Applied Probability, pp 452-471.

[6] Greenwald A. R.,Kephart J. O. & Tesauro G. J. (1999). Strategic Pricebot Dynamics. In *Proceedings of 1st ACM Conference on E-Commerce.*

[7] Hu, J.C & Wellman, M. (1998) Multi-agent Reinforcement Learning:Theoretical Framework and an Algorithm, *The Fifteenth International Conference on Machine Learning*, pp 242-250.

[8] Kephart J. O. & Greenwald A. R. (1999) Shopbot Economics. In *Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty.*

[9] Konda V.R and Borkar V.S., (1999) Actor-Critic-Type Learning Algorithms for Markov Decision Processes, *SIAM Journal on Control and Optimization*, Vol.38, No.1, pp 94-123.

[10] Littman, M.L. (1994) Markov Games as a Framework for Multi-Agent Reinforcement Learning. In: Proceedings of *the Eleventh International Conference on Machine Learning, San Fransisco,CA*,

[11] Littman,L.M. (2001) Friend-or-Foe Q-Learning in General-Sum Games , *Eighteenth International Conference on Machine Learning*, Wilson College.

[12] Mangasarian, O and Stone, H. (1964) Two-Person Non-Zero-Sum Games and Quadratic Programming, *Journal of Mathematical Analysis and Applications*, 9: 348-355.

[13] Singh, S. Kearns, M & Mansour, Y. (2000). Nash Convergence of Gradient Dynamics in General-Sum Games, *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kauffman, pp: 541-548

[14] Patek,S.D & Bertsekas, D P, (1999) Stochastic Shortest Path Games, *SIAM Journal on Control and Optimization*, Vol.37, No.3, pp 804-824.

[15] Varian, H. R. (1980) A Model of Sales, *American Economic Review*, 70: 651-659.

	MY	DF	MOE	QL
MY	0.742	2.519	2.258	0.742
DF	0.201	2.050	0.078	0.074
MOE	0.501	1.776	0.493	0.727
QL	3.181	7.585	2.536	4.780

Table 1. Average profits obtained by seller 1 for different strategies

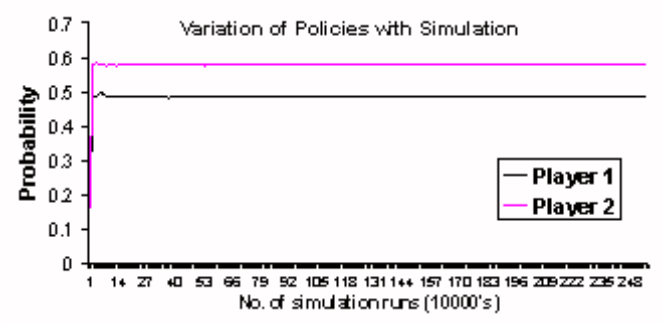

Figure 1. Actor Convergence for the Repeated General Sum Bi-Matrix Game

Figure 2. Actor Convergence for the Repeated Constant Sum Bi-Matrix Game

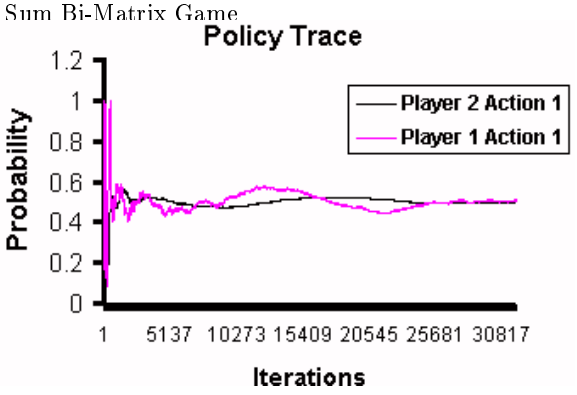

Figure 3. Actor-Critic in Repeated General-Sum Game(of WoLF)

Figure 4. Average Profits from Q-learner