Dynamic Pricing with Limited Competitor Information in a Multi-Agent Economy

Prithviraj Dasgupta[1] and Rajarshi Das[2]

[1] Dept. of Electrical and Computer Engineering, University of California
Santa Barbara, CA 93106, USA,
pdg@alpha.ece.ucsb.edu,
[2] Institute for Advanced Commerce, IBM Research,
Hawthorne, NY 10532, USA,
rajarshi@us.ibm.com

Abstract. We study the price dynamics in a multi-agent economy consisting of buyers and competing sellers, where each seller has limited information about its competitors' prices. In this economy, buyers use *shopbots* while the sellers employ automated pricing agents or *pricebots*. A pricebot resets its seller's price at regular intervals with the objective of maximizing revenue in each time period. Derivative following provides a simple, albeit naive, strategy for dynamic pricing in such a scenario. In this paper, we refine the derivative following algorithm and introduce a model-optimizer algorithm that re-estimates the price-profit relationship for a seller in each period more efficiently. Simulations using the model-optimizer algorithm indicate that it outperforms derivative following even though it does not have any additional information about the market. Our results underscore the role machine learning and optimization can play in fostering competition (or cooperation) in a multi-agent economy where the agents have limited information about their environment.

Keywords: Electronic commerce, dynamic pricing, multi-agent systems, shopbots and pricebots.

1 Introduction

With the rapid growth of agent-mediated electronic commerce, it is becoming increasingly evident that in a few years the Internet will host large numbers of interacting software agents. A vast number of these agents will be economically motivated, and will exchange a variety of information goods and services. In this information economy, software agents will be economic decision makers and will play a fundamental role in many different aspects of electronic commerce, including negotiations, sales, and purchase. Compared to human agents, such software agents will have limited intelligence and rationality, and will need to take decisive actions without having complete information about the market. On the other hand, software agents will be adaptive, and will be able to think and act much faster than human agents. How will the dynamics of the collective behavior of such agents affect our economy? We believe that there is the need to understand and anticipate the collective behavior of economically motivated software agents before employing them in the real world. With these issues as the underlying motivation for our work, this paper focuses on a particular model of an information economy involving groups of autonomous software agents employing *dynamic posted pricing*, i.e., take-it-or-leave-it pricing in which the seller may change the price at any time. Our particular emphasis is on the price dynamics engendered by small groups of myopic and selfish software agents trying to maximize their payoff from the market.

Typically, in determining its own pricing strategy, a seller uses available information about the market, such as the distribution of buyer preferences, or its competitor's prices. There has been recent work in the literature which attempt to address the question of automated dynamic pricing assuming more or less complete information about the market [6, 8, 12]. But what if the seller has only limited information about its environment? In our earlier work, we have explored how a monopolistic seller might dynamically set its price schedule to maximize profit in a market where it has to learn the buyer preferences [2, 7]. In this work, we study markets with multiple sellers competing for the largest market share, where each seller has no information about the buyer preferences or its competitors' prices.

In the traditional economy, obtaining a competitor's pricing information often involves considerable effort, and in certain situations such information may be unavailable (e.g., sealed-bid auctions) or it may be unethical and illegal to gather such data. In contrast, price-checks from competitors in electronic markets is fairly trivial and, indeed, is a common practice. However, there is no guarantee that online competitors will continue to maintain the price they have revealed. An intelligent seller might reset its price slightly at irregular intervals to leave an inquisitive competitor with outdated price information. Moreover, with a huge population of online sellers, price comparison with competitors can become an arduous burden in comparison to the actual task of selling goods. Our objective in this paper is to investigate strategies that enable profit maximizing sellers to identify price settings without having direct knowledge of the price charged by competitors.

Derivative following, which neither uses information about competitors' prices nor any information about buyers, offers one possible dynamic pricing strategy. We show that a naive derivative following approach is inefficient and in certain real world situations, it fails miserably. In this paper, we extend our earlier work in this area [4] by refining the derivative following algorithm and by introducing a model-optimizer algorithm which enables sellers to be more competitive in information-limited environments.

The paper is organized as follows. In Section 2, we describe the shopbot model of the multi-agent economy considered in this work, and outline previous work related to this model. Section 3 delineates the different non-negotiable dynamic price-setting algorithms, while Section 4 contrasts the interacting dynamics and the performance of agents employing the different pricing strategies through simulations. We conclude with an outline of the directions for future research.

2 Model

We study the price dynamics in a simple model of the shopbot economy proposed by Kephart and Greenwald [6, 8, 9]. In this model, the market consists of S sellers who compete to provide B buyers ($B \gg S$) with a single indivisible commodity, such as a specific book.

Each buyer has a valuation p_m corresponding to the maximum unit price it is willing to pay. Prior to purchase, each buyer samples the market price of the item by querying the sellers at a rate ρ_b. Buyers are of two types depending on their selection criterion of a seller:

- A *bargain-hunting buyer* employs a shopbot to select the seller that offers the lowest price and purchases the good if the price offered by the seller is below its valuation p_m. Price ties between multiple sellers are broken randomly.
- A *randomly-selecting buyer* chooses a seller at random and purchases the good if the price offered by the seller is below the valuation price p_m.

The model further assumes that a buyer's strategy in selecting a seller is uncorrelated with its valuation p_m, and that it does not change with time.

Upon entering the market with an initial price p_0, sellers are allowed to reset their price at time intervals $\tau_s = 1/\rho_s$. At any time interval t, a seller's goal is to maximize its immediate profit by setting a price p_t for a single unit of the good, given its production cost p_{co}.

If B_{BH} and B_{RS} respectively represent the number of bargain-hunting buyers and the number of randomly-selecting buyers in the buyer population, then the profit of a seller at time t is given by

$$\pi_t = p_t(B_{BH} + \frac{B_{RS}}{S})\frac{\rho_b}{\rho_s}$$

if the seller is charging the minimum price in the market at time t, and,

$$\pi_t = p_t(\frac{B_{RS}}{S})\frac{\rho_b}{\rho_s}$$

if the seller is not charging the minimum price in the market. The expression in parentheses in the above equations represent the expected number of times a seller is selected by the buyers during the time interval t.

In [5, 6], Greenwald and Kephart view the price setting problem as a one-shot game, and provide a detailed game theoretic analysis of the shopbot economy showing that although there is no pure strategy Nash equilibrium, there exists a symmetric mixed strategy Nash equilibrium. Greenwald and Kephart also show through simulations that pricebots using a Myoptimal or a No-regret pricing strategy give rise to cyclic price-wars. However, the Myoptimal and No regret pricing strategies studied in their work are informationally intensive and require the details knowledge of the buyer demand function, competitors' prices and payoffs. Here, we adopt the opposite scenario by assuming that the pricebots are relatively uninformed about the buyer's demand function or their competitors' pricing strategies.

In our previous work on dynamic pricing strategies [4] we have viewed the price-setting problem for sellers as a Bertrand game where sellers choose prices and sell as much as they can. Simulations using our experimental e-market revealed that a predominantly bargain-hunting buyer population caused sellers to sell at the production cost of the good, while an increase in the number of randomly-selecting buyers resulted in cyclic price wars.

3 Dynamic pricing algorithms

In this section we detail the different dynamic posted pricing algorithms considered in this work, and highlight the motivation behind each approach.

3.1 Derivative Follower Pricing Strategy (DF)

This is the simplest possible dynamic pricing strategy and is the least computationally intensive. It simply experiments with incremental increases (decreases) in price, and as long as the observed level of profitability increases it continues to move its price in the same direction. If the profit level decreases, it changes the direction of the price movement. Thus

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

where π_t is the profit made by the seller during time t and the step-size δ_t is distributed uniformly between $[a, b]$, where $a, b > 0$.

3.2 Adaptive Step-size Derivative Follower Strategy (ADF)

In our earlier work [4], we showed that the simple derivative following approach results in large oscillations in prices when the seller's price reaches a magnitude close its asymptotic value. The large fluctuations in the price dynamics occur because the step size is chosen from a uniform distribution whose range remains constant with respect to time. Also, due to the large oscillations, a seller can lose a significant amount of potential revenue.

We have refined the derivative follower algorithm to dynamically shrink or expand the step size at the end of each time step according to the following equation

$$\delta_t = \epsilon^{\mathrm{sign}(\pi_t - \pi_{t-1})} \times \delta_{t-1}$$

where $\delta_{min} \leq \delta_t \leq p_{t-1}$ and $\epsilon > 1$. δ_{min} is the lower bound on the step-size for the adaptive derivative follower. The initial value of the step-size is given by δ_0 which is chosen uniformly between $[a, b]$, where $a, b > 0$. Dynamic step adjustment enables a seller to reduce the step size when the price is in the vicinity of the optimum and increase it otherwise.

3.3 Dynamic Pricing using the Model-Optimizer Strategy (MO)

A significant drawback of the derivative following algorithm is that the size of the history window of a seller spans only one time interval. One straightforward approach to extend the derivative following algorithm is to increase the time window beyond one time-step and fit a polynomial to the historical price-profit relationship. Once this relationship is modeled, a new price that provides the maximum profit in the model can be determined, and this new price can be posted to the market. As new price-profit information arrives, the model can be continuously updated to reset the posted price.

A major handicap of the above approach is that Internet markets consist of more than one independent seller, and thus, the dynamics of the entire system is non-stationary. In such a situation, it can be argued that a seller has less confidence in applicability of past price-profit relationships in determining its future pricing strategy. This suggests an approach which can assign weights to historical price-profit information in terms of some criterion such as relevance. For this purpose we have used a nonlinear regression approach using least squares. It should be emphasized that the underlying non-stationarity of the system would allow this approach to be applicable only for short-term decision making. Additional information such as consumer preferences and competitors' choices can then be used for more long-term pricing strategies.

Assuming that pricebots have exact information about prices (p) and the measured profits (π) contain all the error or noise, a regression of profit π on price p can be performed. Since π is expected to be non-linearly related to p, we use a high-degree polynomial to model the price-profit data. Let $\pi = c_0 + c_1 p + c_2 p^2 + \ldots + c_r p^r$, where c_i are the coefficients of the polynomial we want to determine. We assume there are m data points (p_t, π_t) each associated with weight w_t. The weights express our confidence in the accuracy or relevance of the points. (Note that m must be greater than r.) The deviation at each point is

$$e_t = c_0 + c_1 p_t + c_2 p_t^2 + \ldots + c_r p_t^r - \pi_t.$$

We now form the weighted sum of squares of the deviations for points (p_t, π_t) with $\tau \leq t \leq \tau + m$

$$F(c_0, c_1, \ldots, c_r) = \sum_{t=\tau}^{\tau+m} (c_0 + c_1 p_t + c_2 p_t^2 + \ldots + c_r p_t^r - \pi_t)^2 w(t)$$

By setting the partial derivatives of F with respect to the coefficients equal to zero we find the normal equations which can be expressed in the matrix form as

$$\begin{bmatrix} \sum w_t & \sum w_t p_t & \cdots & \sum w_t p_t^r \\ \sum w_t p_t & \sum w_t p_t^2 & \cdots & \sum w_t p_t^{r+1} \\ & & \vdots & \\ \sum w_t p_t^r & \sum w_t p_t^{r+1} & \cdots & \sum w_t p_t^{2r} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_r \end{bmatrix} = \begin{bmatrix} \sum w_t \pi_t \\ \sum w_t p_t \pi_t \\ \vdots \\ \sum w_t p_t^r \pi_t \end{bmatrix}$$

By solving the matrix expression we can find the set of coefficients which best fits a given set of m data points (p_t, π_t) along with their corresponding weights w_t.

After obtaining the polynomial fit on the price-profit relationship, a non-linear optimization scheme must be used to locate the price that corresponds to the maximum profit. We have selected the Nelder-Mead algorithm [10, 11, 13] to identify the price corresponding to the maximum profit in the modeled price-profit relationship. The Nelder-Mead algorithm employs a simplex hill-climbing approach to solve unconstrained maximization problems. Our previous work has shown that this approach well suited for price-setting problems in information limited environments [2, 7]

4 Simulations

We have designed and implemented an electronic market in software to study the performance of the different price-setting algorithms. To simulate asynchronous buyer requests in our system we have implemented the buyers on independent threads using Java. Multi-threading also enables a seller to perform price setting calculations simultaneously while it is busy receiving quotes from buyers.

The parameters used for our simulations are the buyer's valuation $p_m = 1.0$, the seller's cut-off price $p_{co} = 0.1$, the lower bound on the step size for the adaptive step-size derivative follower $\delta_{min} = 0.01$, and $\epsilon = 2$ for the adaptive step-size derivative follower. The step size δ, for the the fixed step size derivative follower is drawn from the uniform distribution $[0.01, 0.02]$. The price adjustment interval for sellers is taken as $\tau_s = 20$ quote requests from buyers. We have selected number of quote-requests received from buyers as the unit for measuring time to equalize differences between sellers with different response times. For all the simulations, as a crude approximation to an e-commerce survey presented in [3], we used $B_{BH} = 750$ and $B_{RS} = 250$.

For our simulations we have successively compared the different dynamic pricing strategies. We begin with derivative following, identify its drawbacks and then verify the performance of the model optimizer. To contrast the performance of different pricing strategies, we have also used a seller agent employing the simplest pricing strategy; i.e., maintain a constant price $p = 0.5$.

4.1 Derivative Follower Strategy

Figure 1a illustrates the pricing pattern that results from competition between two fixed step-size derivative followers DF$_1$ (with $p_0 = 0.3$) and DF$_2$ (with $p_0 = 0.9$), and one adaptive step-size derivative follower ADF (with $p_0 = 0.4$). From the beginning of the simulation, DF$_1$ and DF$_2$ enter into a price war lowering their price until they reach $p_{co} = 0.1$ and continue to maintain their price in the vicinity of p_{co}. ADF however fails to discover that there is an ongoing price war to attract the bargain-hunting buyers. It raises its price close to p_m to extract maximum profit from the randomly-selecting buyers and then reduces its step size to keep the price near p_m. This illustrates that a derivative follower strategy can be incapable of discovering a better price that is not in the vicinity of the seller's price.

Derivative following also suffers from an inability to track prices over long time intervals. This is shown in Figure 1b where a fixed step-size derivative follower DF, an adaptive step-size derivative follower ADF, and a fixed-price seller FP compete against

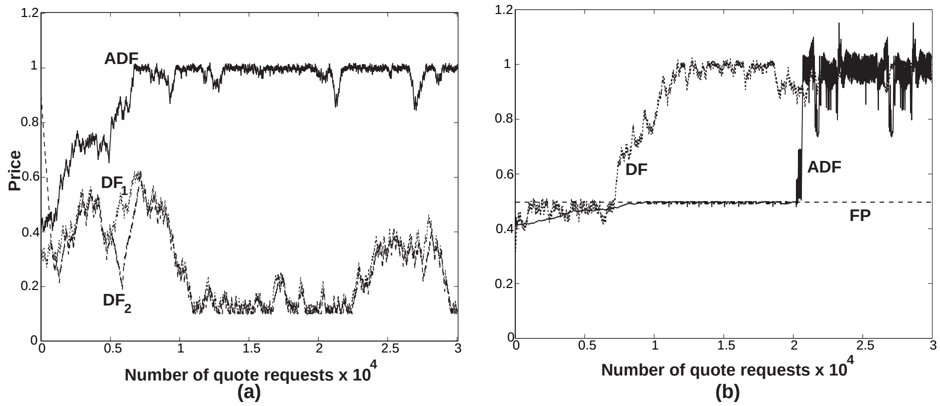

Fig. 1. (a) Price profile of two fixed step-size derivative followers (DF_1, DF_2) and one adaptive step-size derivative follower (ADF) competing against each other. (b) Price profile of an adaptive step size (ADF) and a fixed step-size derivative follower (DF) competing with a fixed-price seller (FP).

each other. Initially, DF succeeds in just under-cutting FP. However, around 7500 quote-requests, DF overshoots the price of FP by a small margin x. As a result, its profit decreases and in the next time interval it reverses the direction of the price change, and DF reduces its price by a step size of y. Note that the step size is drawn from a uniform distribution in $[0.01, 0.02]$, and since in this particular case the step size y was less than x, DF was unable to revert to under-cutting FP. Thus, DF's profit reduced further after reversing the direction of the price change. Since the history window of a derivative follower spans only one time step, DF became oblivious of its incorrect decision to increase its price beyond that of FP and continued to increase its price till it reached a value close to p_m. On the other hand, the adaptive step-size derivative follower ADF reduced its step-size to just under-cut the fixed price seller FP and it succeeded in avoiding the same mistake of crossing the price of the fixed price seller until 20000 quote-requests. However, once its price was more than that of FP it received less profit. In response, it increased its step size and rapidly reached p_m. Unlike in Figure 1a ADF exhibits large oscillations near p_m due to a price war with DF which also has a price close to p_m.

The fluctuations in the price around p_{co} in Figure 1a and around p_m in Figure 1b are due to the finite size of the buyer population and the stochastic nature of choosing sellers by the randomly-selecting buyers. This results in variation of the fraction of quote-requests from randomly-selecting buyers among the $\tau_s = 20$ quote-requests. Figure 2 illustrates this through a plot of the price-profit relationship when an adaptive step-size derivative follower that starts with $p_0 = 0.1$ competes against a fixed price seller with $p = 0.5$. Initially the derivative follower's profit increases as it increments its price towards the fixed price seller's price. In this interval the derivative follower attracts the entire bargain-hunting buyer population since it has the lowest price in the market. In this scenario, $p = 0.5$ corresponds to the optimum price in the system, since

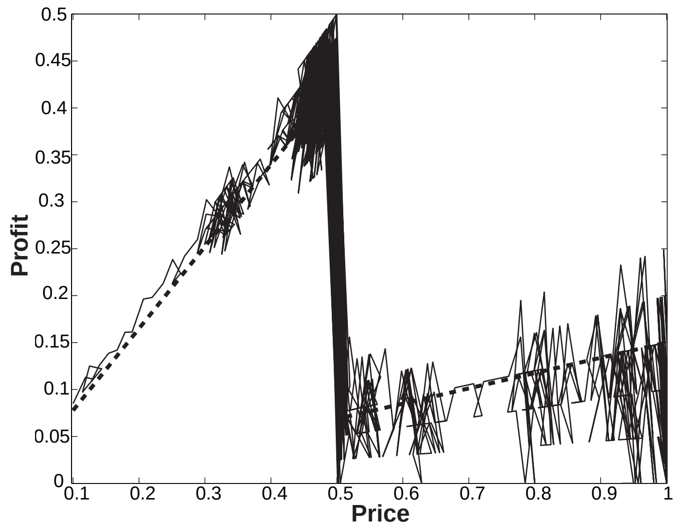

Fig. 2. Price-profit profile for an adaptive step-size seller competing against a fixed price seller with price set to $p = 0.5$. The size of the buyer population is 750 bargain-hunting buyers and 250 randomly-selecting buyers. The dotted line represents the theoretical price-profit relationship for an infinite size buyer population.

it guarantees the highest profit. The presence of large number of points around $p = 0.5$ indicates that the adaptive step-size derivative follower discovers and frequently sets its price at this optimum. However, due to the inability of derivative followers to track prices, the adaptive step-size derivative follower makes a wrong decision to increase its price beyond that of the fixed price seller and fails to revert its price back to the optimum. Thereafter, its profit is derived from solely from randomly-selecting buyers as it raises its price close to p_m. The broken line in Figure 2 shows the price vs. profit profile for the theoretical case when the size of the buyer population is infinite.

4.2 Model-Optimizer Strategy

The derivative follower algorithm performed poorly because its history window spans only one time interval. In the model optimizer algorithm we have addressed this problem by increasing the size of the history window to cover the last h time intervals. We ran our simulations for $h = 3$ and $h = 5$ and found, as expected, that with a larger window size the model optimizer estimates a better price to set during the next time interval. For all the simulation results of the model optimizer reported here, the history window was $h = 5$. In the beginning of the simulation, before the history window gets

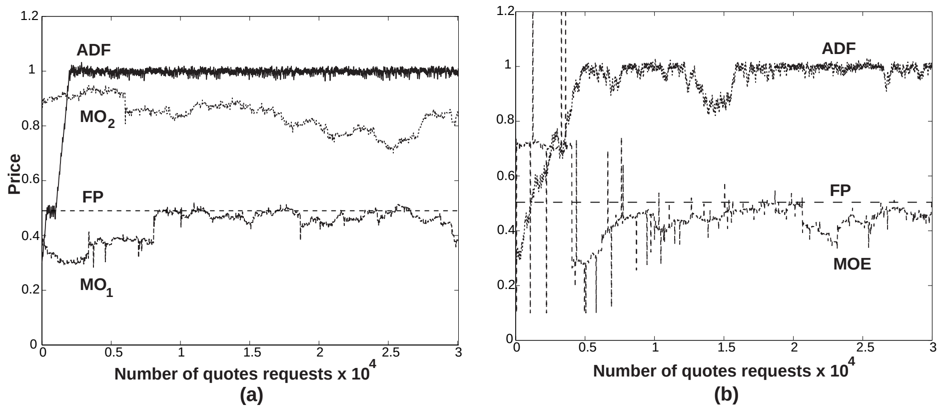

Fig. 3. (a)Price profile of a competition between a fixed-price seller (FP), an adaptive step-size derivative follower (ADF), and a model optimizer (MO). (b) Price profile of a fixed-price seller (FP), an adaptive step-size derivative follower (ADF), and a model optimizer with exploration (MOE) competing against each other.

initialized, the sellers move the price in a random direction with a step-size distributed uniformly in $[0.01 - 0.02]$. The weights w_t, denoting the relevance of the previously seen price-profit values, were assigned linearly decreasing values with $w_1 = 0.9$ and $w_5 = 0.5$. Also, for simplicity the historical price-profit relationship was modeled with a second degree polynomial.

Figure 3a compares the performance of two model optimizers MO_1 and MO_2 competing with an adaptive step-size derivative follower ADF and a fixed price seller FP. MO_2 (with $p_0 = 0.3$) is initially under-cut by the derivative follower ADF (with $p_0 = 0.4$). However, at almost 1000 quote-requests ADF makes a wrong decision and increases its price beyond that of FP, while MO_2 discovers FP at $p = 0.5$ and thereafter resorts to under-cutting it. MO_1, starting at $p_0 = 0.9$, however, fails to discover the sellers charging lower prices than itself and maintains almost the same price all along. This indicates that although model optimizers are more efficient than derivative followers, they are not capable enough to discover a price that is not in the vicinity of their current price. One solution to this problem is to allow the model optimizer to occasionally explore the price space and search for better prices.

4.3 Model-Optimizer Strategy with Exploration

We extended the model optimizer algorithm to explore the price space intermittently for better prices. The interval between two successive explorations, measured in number of price resets, is distributed uniformly between $[40 - 60]$ quote-requests [1]. In the beginning, the range of prices chosen during exploration is drawn from a uniform distribution in $[p_{co}, 5p_0]$. We assume that the maximum buyer valuation lies below the upper limit of

[1] Preliminary experiments showed this to be a suitable distribution that balances exploration against exploitation.

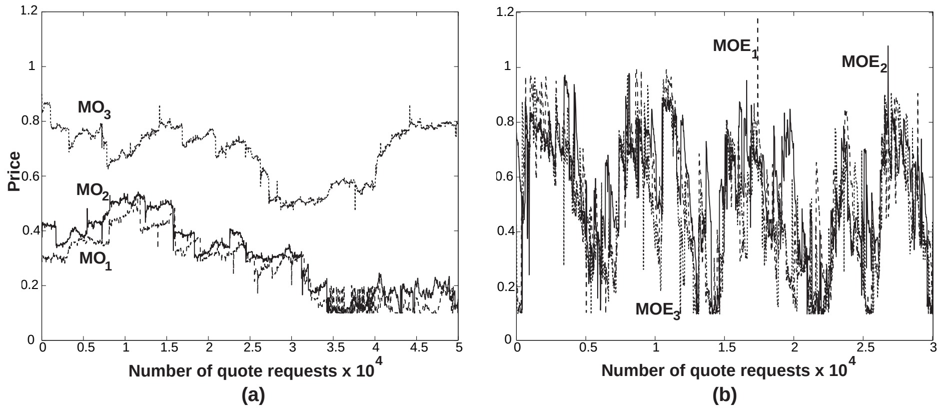

Fig. 4. (a) Price profile of three competing model optimizers without exploration (MO_1, MO_2, MO_3). (b) Price profile of three model optimizers with exploration (MOE_1, MOE_2, MOE_3) competing against each other.

this distribution. The upper limit of the distribution is adaptively adjusted at successive explorations to a price that returns positive profits.

The results of our simulation using the model optimizer strategy with exploration is shown in Figure 3b. All other parameter settings are retained from the simulation shown in Figure 3a. Although MOE starts with $p_0 = 0.9$, it explores the price space and discovers that profits increase when it reduces its price. It thus resets its price to $p = 0.3$ around 4000 quotes. Thereafter, it gradually increases its price and discovers the fixed price seller FP at $p = 0.5$, and resorts to under-cutting FP. Exploring the price space thus enabled the model optimizer to find new prices that were not in the vicinity of its current price.

Thus far we have considered competition among a heterogeneous population of sellers employing different price setting algorithms. Now we focus our attention on a homogeneous population of sellers. Figure 4a shows the price dynamics resulting from a competition between three model optimizers with three different starting prices: MO_1 starts with $p_0 = 0.3$, MO_2 begins with $p_0 = 0.4$, and MO_3 has $p_0 = 0.9$. Initially, sellers MO_1 and MO_2 engage in a price war with each other until they reach their cut-off price $p_{co} = 0.1$. Thereafter, their prices hover near p_{co}. However, as exploration was not enabled, seller MO_3, fails to discover the sellers charging a price lower than itself and keeps on maintaining a high price.

Figure 4a can be contrasted with Figure 4b which provides the results of similar experiment with three competing model optimizers with exploration. The sellers start at the same initial price as those from the last simulation. However, exploration of the price space enables them to discover each other's prices and they start a price war even before receiving 1000 quotes. The price war causes each seller to reduce its price till it reaches the cut-off price $p_{co} = 0.1$. However, unlike Figure 4a, the sellers' prices no longer fluctuate around p_{co}. Exploration of the price space now enables sellers to discover that they are better off by charging a price in the vicinity of p_m and attracting only the randomly-selecting buyers. This results in repeated cycles of price wars at intervals of approximately 5000 quotes as illustrated in Figure 4b. Our results also show that exploration of the price space can increase the income of the model optimizers since a successful exploration yields more profits. The income of model optimizers with exploration in Figure 4b were approximately 1.5 times the income obtained by model optimizers without exploration shown in Figure 4a.

The behavior observed in Figure 4b is similar to the cyclic price wars obtained for the shopbot economy [8, 9] where the sellers had detailed knowledge about buyer preferences and their competitors' prices. Our results suggest that similar cyclic price wars also occur in agent economies even when seller agents are ignorant of competitors' prices or profits.

5 Conclusion

This work addresses the problem of dynamic posted pricing for sellers in an environment where they have limited market information. We studied derivative following as a possible strategy in such an information limited situation and found it to be handicapped due to the limited size of its history window. We developed a model-optimizer algorithm which sets price more efficiently. by making use of historical price-profit relationship.

The objective of a seller agent in our system was to maximize its immediate profit by reseting a single attribute; viz., its price at regular intervals. In the real world, electronic goods and services can have multiple attributes, such as the quality of the good, its price per unit, or its bundle price. In such situations, a pricebot will need to set a number of attributes in response to sparse feedback from the market. Work is in progress to extend the dynamic pricing algorithms developed in this paper for electronic goods and services with multiple attributes.

In this work, we have used a relatively simple model for studying an information economy using pricebots that are selfish and myopic and aim to maximize their immediate profits. Competing pricebots engage in price wars under-cutting each other even when they are ignorant of the prices and profits of competitors, and perhaps, of each others existence. Therefore, we envisage that in the absence of cooperation there exists little friction in the economy and the market price will be repeatedly driven down to the production cost of the sellers giving rise to cyclic price wars. The work reported in this paper indicates that repeated cycles of price wars is a common phenomenon in agent-mediated electronic commerce even when the sellers possess limited information about the market.

References

1. C. H. Brooks, E. H. Durfee, and R. Das. Price wars and niche discovery in an information economy. In preparation, 2000.
2. C. H. Brooks, S. Fay, R. Das, J. K. Mackie-Mason, J. O. Kephart, and E. H. Durfee. Automated search strategies in an electronic goods market: Learning and complex price scheduling. In *Proceedings of the ACM Conference on Electronic Commerce*, 1999.

3. D. Clark. Shopbots become agents for business change. *IEEE Computer*, 33:18–21, February 2000.
4. P. Dasgupta and R. Das. Dynamic service pricing for brokers in a multi-agent economy. In *Proceedings of the Fourth International Conference on Multi-agent systems*. Springer-Verlag, July 2000.
5. A. R. Greenwald and J. O. Kephart. Shopbots and pricebots. In *Lecture Notes on Artificial Intelligence: Proceedings of the IJCAI Workshop on Agent-mediated Electronic Commerce*. Springer-Verlag, 2000.
6. A. R. Greenwald, J. O. Kephart, and G. J. Tesauro. Strategic pricebot dynamics. In *Proceedings of 1st ACM Conference on E-Commerce*, 1999.
7. J. O. Kephart, R. Das, and J. K. Mackie-Mason. Two-sided learning in an agent economy for information bundles. In *Lecture Notes on Artificial Intelligence: Proceedings of the IJCAI Workshop on Agent-mediated Electronic Commerce*. Springer-Verlag, 2000.
8. J. O. Kephart and A. R. Greenwald. Shopbot economics. In *Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 1999.
9. J. O. Kephart, J. E. Hanson, and A. R. Greenwald. Dynamic pricing by software agents. To appear in Computer Networks, 2000.
10. J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.
11. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1988.
12. G. J. Tesauro and J. O. Kephart. Pricing in agent economies using multi-agent q-learning. In *Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 1999.
13. P. Tseng. Fortified-descent simplical search method: a general approach. Department of Mathematics, University of Washington, Seattle, 1998.
14. H. R. Varian. Market structure in the network age. In proceedings, Understanding the Digital Economy Conference, Department of Commerce, Washington, DC, USA, May 1999.