Proceedings of the IEEE
International Conference on Automation and Logistics
Shenyang, China August 2009

Application of Reinforcement Learning in Dynamic Pricing Algorithms*

Wang Jintian
Department of Computer Science and Technology
Hefei University of Technology
Hefei, Anhui Province, China
neo81@163.com

Zhou Lei
Department of Computer Science and Technology
Hefei University of Technology
Hefei, Anhui Province, China
zhouleizhl@163.com

***Abstract* - This paper is concerned with the dynamic pricing problems of a duopoly case in electronic retail markets. Combined with the concept of performance potential, the simulated annealing Q-learning (SA-Q) and the win-or-learn-fast policy hill climbing algorithm (WoLF-PHC) are used to solve the learning problems of multi-agent systems with either average- or discounted-reward criteria, under the case that only partial information about the opponent is known. The simulation results show that the WoLF-PHC algorithm performs well in adapting environment's change and in deriving better learning values than the SA-Q algorithm.**

Index Terms* - *multi-agent, performance potential, WoLF-PHC, simulated annealing Q-learning.

I. INTRODUCTION

With the development of internet and electronic commerce, more and more customers can obtain easily the seller's quoted price, preferential measures, and their inventory, etc. Based on the above information, customers will make their choice. Because of demand volatility and fierce competition, the retailers need to adjust dynamically quoted price according to inventory and customers' demand information for revenue maximization [1].

Dynamic pricing problem of a single-seller has been investigated several years ago [2], and multi-seller cases have been focused on recently [3], [4]. Among these literatures, Brooks *et al.* [2] and Greenwalrd, Kephart, and Tesuaro [3] only consider the case that all customers are sensitive to price, and inventory is infinite, whereas, Chinthalapati, Yadati, and Karumanchi [4] has considered a complicated model with multi-seller, stochastic demands, different types of customers, and finite inventory which need to be replenished according to a fixed reorder policy, and model it as a Markov game. Chinthalapati supposes that two sellers share minimal information, i.e., a seller knows the other seller's state, but not the action (price) and payoff of the other seller. Some multi-agent algorithms, such as Nash-Q [5], FF-Q [6] and CE-Q [7], can not apply to this case since they all need to observe the other agents' actions and rewards. So, under the case of sharing minimal information, the other agents can be viewed as a part of environment, and Q-learning can be applied. However, there is no adaptive scheme in Q-learning to consider the variations of actions that the other agents performed. On the contrary, the WoLF-PHC algorithm uses variable learning rates to adapt to the other agents' policies [8]. As a result, it adapts to environmental changes better than Q-learning. In this paper, we apply the SA-Q algorithm [9] and the WoLF-PHC algorithm to solving the dynamic pricing problems in multi-seller electronic retail markets by using the concept of performance potential, and compare their optimization performance.

II. MODEL STATEMENT

Assume that there are two homogeneous sellers monopolizing a commodity in an electronic retail market [4]. Each seller has finite inventory capacity $I_{\max}$ and follows a fixed reorder policy for replenishing, i.e., the classic (q,r) policy. The replenishment lead time is exponentially distribution with mean $1/\mu_r$. According to the sensitivity to price, customers are classified into two categories: shoppers and captives [10]. Shoppers are influenced by price and volume discounts (assumed to be buy-two-get-three), and always visit a seller with lower price at the same volume discounts. If these two sellers' prices and volume discounts are equal, they are chosen by shoppers with equal probability. Contrarily, a captive is only loyal to a specific seller and buys a unit item every time. Suppose that customers arrive at the market with a Poisson flow. A fraction f_i of these customers are captives of seller i, ($i=1,2$), and the others are shoppers. A captive will purchase a unit item if the retailer's price is acceptable and the stock is available, or place an order if the price quote and lead time quote of the retailer is acceptable; otherwise, this captive will leave the market. On the contrary, shoppers wish to choose a retailer who offers lower price quote even if under the case of out of stock. That is to say if the price is acceptable, she will place an order or purchase; otherwise she will leave the market. And sellers provide two different service types for captives and shoppers when out of stock. Captives will get a price quote, a lead time quote and priority to obtain items when replenishment arrived, but shoppers do not have any quote. A shopper places an order at a seller, and will revisit him after a random time interval

* This work is partially supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, Nature Science Foundation of Anhui Province (070416242, 090412046), The Nature Science Foundation of Education Department of Anhui Province (KJ2008A058).

978-1-4244-4795-4/09/$25.00 © 2009 IEEE

following exponentially distribution with mean time $1/\mu_s$, and then check whether the inventory is available or not. If the seller is stock out again, she will leave the market; otherwise she will purchase the items when the price quote is acceptable, or still stay at the system as it is not acceptable. The detailed description of this model is referred to [4].

Assume each seller to be an agent. Let $s(t)=(s_1(t),s_2(t))$ be the system state at time t, and $s_i(t)$ be the state of agent i. Here, $s_i(t)=(x_i(t),y_i(t),I_i(t))$ with $x_i(t)$ and $y_i(t)$ being the queue lengths of captives and shoppers (bounded by N_c and N_s, respectively) representing the number of backlogged requests, and $I_i(t)$ being the inventory level of agent i. So, the system state space is $S=\{0,1,\cdots,|S|-1\}$ with $|S|=((N_c+1)\cdot(N_s+1)+I_{\max}\cdot(N_s+1))^2$ denoting the number of system states. The system action is jointed by the two agents' action. At nth decision epoch T_n, T_0 =0, suppose $a_i(s(T_n))\in A$, with A being the action set of each agent, and $a_i(s(T_n))$ represents the action chosen by agent i at state $s(T_n)$, denoting the price quote per unit item. Under the joint action, the system transits to $s(T_{n+1})$ at T_{n+1}. During such transition, each agent i obtains or pays the immediate payoff $R_i(s(T_n),a_i(s(T_n)),s(T_{n+1}))$, the backorder cost $C_i(s(T_n),s(T_{n+1}))$, and the inventory cost $H_i(s(T_n))$, and their detailed expressions are referred to [4].

Write agent i's policy as $v_i=(a_i(0),a_i(1),\cdots,a_i(|S|-1))$. According to [11], we have the discounted-reward criterion $\eta_\alpha^{v_i}(\cdot)$ for agent i under discount factor α, and the average-reward criterion η^{v_i}. The goal of agent i is to find an optimal pricing policy to maximize the value of a chosen reward criterion.

III. Algorithms

In this section, we introduce two reinforcement learning algorithms, i.e., the SA-Q algorithm and the WoLF-PHC algorithm, to solve the dynamic pricing problems under the case that only partial information of the opponent is acquired excluding the opponent's actions and rewards.

A. SA-Q algorithm based on performance potentials

Q-learning is a reinforcement learning method proposed by Watkins in 1989 [12], which is an effective method to solve Markov decision process (MDP) problems with incomplete information. However, the tradeoff between exploration and exploitation in Q-learning is difficult. Excessive exploration will drastically decrease the learning performance, but excessive exploitation will lead to local optimal solutions [13]. In order to acquire balance between exploration and exploitation, the Metropolis criterion from simulated annealing algorithm can be applied to the Q-learning [9].

Suppose the system is at state $s(T_n)$ at decision epoch T_n, and transits to $s(T_{n+1})$ at next decision epoch T_{n+1} under the system action. Then the state sojourn time $\omega_n=T_{n+1}-T_n$, and agent i's practical accumulated reward is as follows

$$\begin{aligned} f_i(s(T_n),a_i(s(T_n)),s(T_{n+1})) &= R_i(s(T_n),a_i(s(T_n)),s(T_{n+1})) \\ &\quad - C_i(s(T_n),s(T_{n+1}))-H_i(s(T_n))\cdot T_\alpha(\omega_n) \end{aligned} \tag{1}$$

here

$$T_\alpha(\omega_n)=\int_0^{\omega_n} e^{-\alpha t}dt$$

denoting the discount sojourn time at state $s(T_n)$, and write $T_{\alpha=0}(\omega_n)=\omega_n$.

Combined with the concept of performance potential [14], a unified temporal difference formula of agent i for both average- and discounted-reward criteria is obtained as follows [15], [16]

$$\begin{aligned} d_\alpha^i(T_n) &= f_i(s(T_n),a_i(s(T_n)),s(T_{n+1}))-T_\alpha(\omega_n)\cdot\bar{\eta}_i \\ &\quad +e^{-\alpha\omega_n}\max_{a'\in A}Q(s(T_{n+1}),a')-Q(s(T_n),a_i(s(T_n))) \end{aligned} \tag{2}$$

Here, $\bar{\eta}_i$ is an estimate of average reward η^{v_i}, satisfying

$$\bar{\eta}_i=\frac{S_f^i}{S_\omega^i} \tag{3}$$

with S_f^i and S_ω^i being learned, respectively, as follows [16]

$$S_f^i := S_f^i+\zeta_i\left[f_i(s(T_n),a_i(s(T_n)),s(T_{n+1}))-S_f^i\right] \tag{4}$$

$$S_\omega^i := S_\omega^i+\zeta_i(\omega_n-S_\omega^i) \tag{5}$$

where ζ_i is a stepsize. Then we have

$$Q_i(s(T_n),a_i(s(T_n))) := Q_i(s(T_n),a_i(s(T_n)))+\gamma_i\cdot d_\alpha^i(T_n) \tag{6}$$

Here, γ_i is a learning stepsize that usually degrades to zero slower than ζ_i. Equation (6) denotes the iteration formula of Q-value for discounted-reward criterion as $\alpha>0$, otherwise, it denotes the discounted case.

So we can establish a potential-based SA-Q algorithm as depicted by Algorithm 1.

Algorithm 1: Potential-based SA-Q algorithm

step1. Set discount factor α, temperature decay factor λ_T, the number of learning episode Z, and learning steps of each episode N. Initialize system state s, Q-table, temperature T. Let z =0 and n =0.

step2. Select an action a_r arbitrarily, and an action a_p greedy on the Q-value at state s. Generating a random number $\xi\in(0,1)$, if $\xi<e^{(Q(s,a_r)-Q(s,a_p))/T}$, then let $a:=a_r$, otherwise $a:=a_p$.

step3. Execute the action a, observe the new state s', record the state sojourn time ω, and calculate the reward by (1).

step4. Calculate the estimate of average reward $\bar{\eta}$ and

temporal difference d_α, and update Q-value according to (2)-(6).

step5. Let $s := s'$, $n := n+1$. If $n < N$, turn to step3; otherwise, let $T := \lambda_T \cdot T$, $z := z+1$.

step6. If $z = Z$, exit; otherwise, set n=0, and turn to step2.

B. WoLF-PHC algorithm based on performance potentials

Due to simplicity, Q-learning has been introduced to multi-agent learning. However, it does not consider actions the other agents' performed. WoLF-PHC is an extension of Q-learning using policy hill-climbing (PHC) with variable learning rates [8], which need not assume that an agent can observe the other agents' actions and rewards. It adapts to environmental changes better than Q-learning because it uses variable learning rate according to the opponents' policies, and can solve the case of sharing minimal information.

PHC uses mixed strategies, and improves the current mixed policy by increasing the probability of the utilized action at current state if it is greedy on the current Q-value, otherwise decreasing. Specifically,

$$\pi(s,a) := \pi(s,a) + \begin{cases} -\delta_{sa} & \text{if } a \neq \arg\max_{a'} Q(s,a') \\ \sum_{a' \neq a} \delta_{sa'} & \text{otherwise} \end{cases} \quad (7)$$

$$\delta_{sa} = \min(\pi(s,a), \delta/(|A|-1)) \quad (8)$$

where $\pi(s,a)$ is the chosen probability of action a at state s, and δ is a fixed learning rate. All the probabilities $\pi(s,a), a \in A$ at state s should be normalized to satisfy $\sum_{a \in A} \pi(s,a) = 1$.

WoLF principle is introduced to improving the adaptability by using variable learning rates [8]. The underlying principle is to learn quickly while losing and slowly while winning. In WoLF-PHC algorithm, two different learning rates, δ_w and δ_l, are used. If the expected value of the current policy is higher than the expected value of the average policy, the smaller learning rate δ_w is used, otherwise, δ_l is used. That is

$$\delta = \begin{cases} \delta_w & \text{if } \sum_{a' \in A} \pi(s,a') Q(s,a') > \sum_{a' \in A} \bar{\pi}(s,a') Q(s,a') \\ \delta_l & \text{otherwise} \end{cases} \quad (9)$$

Here, $\delta_l, \delta_w \in (0,1]$, both degrading as time goes on, and $\bar{\pi}(s,a)$ is the following average policy used to replace the unknown equilibrium policy for judging winning or losing

$$\bar{\pi}(s,a) := \bar{\pi}(s,a) + \frac{1}{C(s)}(\pi(s,a) - \bar{\pi}(s,a)) \quad \forall a \in A \quad (10)$$

Where $C(s)$ denotes the number of visiting state s.

The potential-based WoLF-PHC algorithm is depicted in Algorithm 2.

Algorithm 2: Potential-based WoLF-PHC algorithm

step1. Set discount factor α, learning steps N, the value of δ_l / δ_w. Initialize system state s, Q-table, learning rate δ_w, mixed policy π. Let n=0.

step2. From the state s select an action a according to the mixed strategy $\pi(s)$ with suitable exploration.

step3. Execute the action a, observe the next state s', record the state sojourn time ω, and calculate the reward by (1).

step4. Calculate the estimate value of average reward $\bar{\eta}$, temporal difference d_α, and update Q-value according to (2)-(6).

step5. Update the estimate of average policy $\bar{\pi}$ by (10), and $\pi(s,a)$ according to (7)-(9).

step6. $s := s'$, $n := n+1$ .If $n = N$, stopping criterion is satisfied, exit; otherwise, turn to step2.

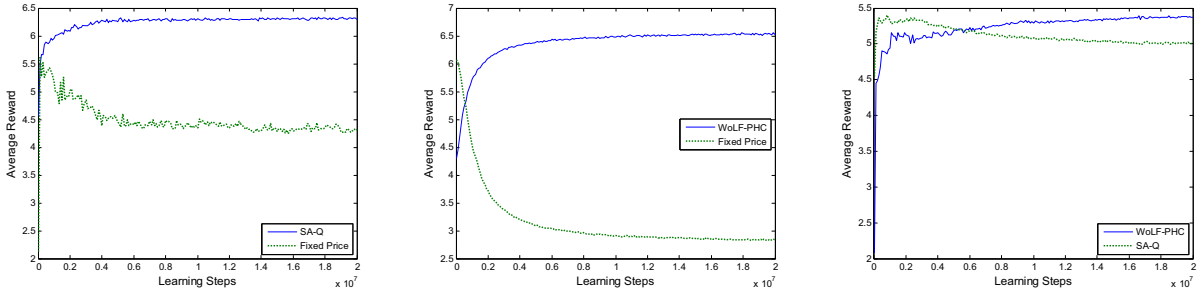
IV. NUMERICAL RESULTS

Suppose that the arrival rate of customers is λ =4. $f_1 = f_2$ =0.2, $N_c = N_s$ =5, $I_{\max}$ =10, r =5, μ_r =1/3, μ_s =1/1.5. The purchasing cost per unit item is 4, the back-logged cost for each back-logged demand per unit time is 0.5 and the holding cost per unit item per unit time is 0.5. The acceptable price of captives follows uniform distribution from 8 to 14, and the acceptable time follows uniform distribution from 0 to 12. The acceptable price of shoppers follows uniform distribution from 5 to 9. The action set A= {8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5}.

Set discount factor $\alpha = 0.01$, initial temperature T =100, temperature decay factor λ_T =0.95, and $\delta_l / \delta_w = 4$. The fixed price policy is regarded as a benchmark, which provides the same price per unit item to all customers. In the numerical examples, we define the price as 10.5 because two sellers gain the maximum revenue at 10.5 when they both use the fixed price policy.

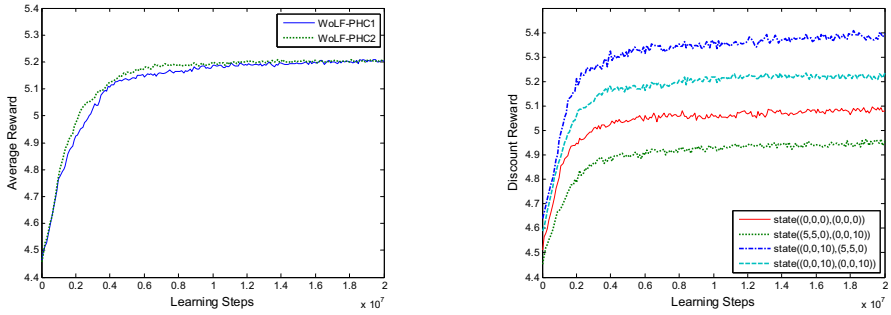
Fig. 1 denotes the optimization curves of average reward per unit time when the two agents use different algorithms. It can be concluded that both the SA-Q algorithm and the WoLF-PHC algorithm obtain higher reward than the opponent who uses the fixed price policy described by Fig. 1(a) and 1(b). We can also see that the seller who using WoLF-PHC algorithm obtains higher reward than the SA-Q algorithm when their opponents both use the fixed price policy. The same result is shown by Fig. 1(c), which describes the two sellers use the WoLF-PHC algorithm and the SA-Q algorithm respectively. Because the WoLF-PHC algorithm uses mixed strategies, the probability of every action be chosen is equal at any state at the beginning of learning, and then it convergences to an optimal policy slowly. While the SA-Q algorithm uses a deterministic policy, which does not consider actions the other seller adopted, and may be changed greatly. So, the SA-Q algorithm does well at the beginning of learning, while the WoLF-PHC algorithm shows better adaptability after learning some steps.

Fig. 2 shows average reward per unit time and discounted



(a) SA-Q vs. fixed price policy (b) WoLF-PHC vs. fixed price policy (c) WoLF-PHC vs. SA-Q

Fig. 1: Average reward per unit time optimization curves when two sellers use different algorithms



(a) average reward per unit time (b) discounted reward optimization curves at several state of seller1

Fig. 2: Average reward per unit time and discounted reward optimization curves when two sellers both use the WoLF-PHC algorithm

reward optimization curves when two sellers both use the WoLF-PHC algorithm. It is shown in Fig. 2(a) that two optimization curves are very close, and a little difference is caused by randomness of the system. Comparing Fig. 1 with Fig. 2(a), we can see that a seller using a better algorithm obtains higher reward than the other seller because of competition. Fig. 2(b) depicts the discounted reward optimization curves of seller1 at several representative states when discounted factor α is 0.01. At state ((0,0,10),(5,5,0)), seller1's inventory is full, and he can offer items to incoming customers, while seller2's inventory is not available and he is so overcrowded that can not accept any request for ordering. Therefore, at this state, seller1 obtains higher reward. A reverse conclusion can be reached at state ((5,5,0), (0,0,10)). Fig. 2(a) and 2(b) show that the unified WoLF-PHC algorithm is effective for both average- and discounted-reward criteria.

V. Conclusion

In this paper, we consider a dynamic pricing problem between two sellers, where each of them knows the other seller's state, but not the price and payoff. It can be solved by SA-Q algorithm and WoLF-PHC algorithm effectively. Simulation results show that the latter performs better than former. In the real electronic retail market, there are usually more retailers and each one may sell more product categories rather than one. Besides, each seller may try to acquire more information to maximize his revenue. Therefore the dynamic pricing problems under these circumstances are deserved further researches.

References

[1] W. Elmaghraby and P. Keskinocak, "Dynamic Pricing in the Presence of Inventory Considerations: Research Overview, Current Practices and Future Directions," *Management Science*, vol. 49, no.10, pp. 1287-1309.

[2] 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 *Proc. 1st ACM Conf. Electronic Commerce EC-99*, 1999, pp. 31–40.

[3] A. R. Greenwald, J. O. Kephart, and G. J. Tesauro, "Strategic pricebot dynamics," in *Proc. 1st ACM Conf. Electronic Commerce EC-99*, 1999, pp. 58-67.

[4] V. L. R. Chinthalapati, N. Yadati, and R. Karumanchi, "Learning dynamic prices in multiseller electronic retail markets with price sensitive customers, stochastic demands, and inventory replenishments," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 1, pp. 92–106, Jan. 2006.

[5] J. Hu and M. P. Wellman, "Multiagent reinforcement learning: Theoretical framework and an algorithm," in *Proc. 15th Int. Conf. Mach. Learn. (ICML-98)*, Madison, WI, Jul. 24–27, pp. 242–250.

[6] M. Littman, "Friend-or-Foe Q-learning in general-sum games," in *Proc. 18th Int. Conf. Mach. Learn. (ICML-01)*, San Francisco, CA, Jun. 28–July. 1, pp. 322–328.

[7] A. Greenwald and K. Hall, "Correlated-Q learning," in *Proc. 20th Int. Conf. Mach. Learn. (ICML-03)*,Washington, DC, Aug. 21–24, pp. 242–249.

[8] M. Bowling and M. Veloso. "Multiagent learning using a variable learning rate," *Artificial Intelligence*, 136:215–250, 2002.

[9] M. Guo, Y. Liu, and J. Malec, "A new Q-learning algorithm based on the metropolis criterion," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 34, no. 5, pp. 2140–2143, Oct. 2004.

[10] H. R. Varian, "A model of sales," *Amer. Econ. Rev.*, pp. 651–659, 1980.

[11] X. R. Cao, "Semi-Markov decision problems and performance sensitivity analysis," *IEEE Trans. Automat. Contr.*, vol. 48, no.5, pp. 758-769, May 2003.

[12] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. thesis, King's College, Cambridge, U.K.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
[14] X. R. Cao, *Stochastic Learning and Optimization: A Sensitivity-Based View*. New York, NY: Springer, 2007.
[15] H. Tang, L. Zhou, and J. B. Yuan, "Unified NDP method based on TD(0) learning for both average and discounted Markov decision processes," *Control Theory & Applications*, 2006, 23(2): 292-296.
[16] H. Tang and Arai Tamio, "Potential-based online policy iteration algorithm for look-ahead control of CSPS by stochastic directed exploration," *International Journal of Control*, in press.