Keywords: reinforcement learning, neural networks, adaptive multi-agent systems, agent economies

Pricing in agent economies using neural networks and multi-agent Q-learning

Gerald Tesauro
IBM T. J. Watson Research Center
30 Saw Mill River Rd., Hawthorne NY, 10532
e-mail: tesauro@watson.ibm.com

Abstract

This paper investigates how adaptive software agents may utilize reinforcement learning algorithms such as Q-learning to make economic decisions such as setting prices in a competitive marketplace. For a single adaptive agent facing fixed-strategy opponents, ordinary Q-learning is guaranteed to find the optimal policy. However, for a population of agents each trying to adapt in the presence of other adaptive agents, the problem becomes non-stationary and history dependent, and it is not known whether any global convergence will be obtained, and if so, whether such solutions will be optimal. This paper studies simultaneous Q-learning by two competing seller agents in three moderately realistic economic models. This is the simplest case in which interesting multi-agent phenomena can occur, and the state space is small enough so that lookup tables can be used to represent the Q-functions. Despite the lack of theoretical guarantees, simultaneous convergence to self-consistent optimal solutions is obtained in each model, at least for small values of the discount parameter. In some cases, such convergence is also found even at large discount parameters. Furthermore, the Q-derived policies increase profitability and damp out or eliminate cyclic price "wars" compared to simpler policies based on zero lookahed or short-term lookahead. The use of function approximators (neural nets) instead of lookup tables is also investigated; preliminary findings indicate that reasonably good policies can be obtained even though the absolute accuracy of the function approximation may be poor.

1 Introduction

Reinforcement Learning (RL) procedures have been established as powerful and practical methods for solving Markov Decision Problems. One of the most significant and actively investigated RL algorithms is Q-learning (Watkins, 1989). Q-learning is an algorithm for learning to estimate the long-term expected reward for a given state-action pair. It has the nice property that it does not need a model of the environment, and it can be used for on-line learning. A number of powerful convergence proofs have been given showing that Q-learning is guaranteed to converge with probability 1, in cases where the state space is small enough so that lookup table representations can be used (Watkins and Dayan, 1992). Furthermore, in large state spaces where lookup table representations are infeasible, RL methods can be combined with function approximators to give good practical performance despite the lack of theoretical guarantees of convergence to optimal policies.

Most real-world problems are not fully Markov in nature – they are often non-stationary, history-dependent and/or not fully observable. In order for RL methods to be more generally useful in solving such problems, they need to be extended to handle these non-Markovian properties. One important application domain where the non-Markovian aspects are paramount is the area of multi-agent systems. This area is expected to be increasingly important in the future, due to the potential rapid emergence of "agent economies" consisting of large populations of interacting software agents engaged in various forms of economic activity. The problem of multiple agents simultaneously adapting is in general non-Markov, because each agent provides an effectively non-stationary environment for the other agents. Hence the existing convergence guarantees do not hold, and in general, it is not known whether any global convergence will be obtained, and if so, whether such solutions are optimal.

Some progress has been made in analyzing certain special case multi-agent problems. For example, the problem of "teams," where all agents share a common utility function, has been studied, for example, in (Crites and Barto, 1996). Likewise, the purely competitive case of zero-sum utility functions has been studied in (Littman, 1994), where an algorithm called "minimax-Q" was proposed for two-player zero-sum games, and shown to converge to the optimal value function and policies for both players. Sandholm and Crites studied simultaneous Q-learning by two players in the Iterated Prisoner's Dilemma game (Sandholm and Crites, 1995), and found that the learning procedure generally converged to stationary solutions. However, the extent to which those solutions were "optimal" was unclear. Recently, Hu and Wellman proposed an algorithm for calculating optimal Q-functions in two-player arbitrary-sum games (Hu and Wellman, 1998). This algorithm is an important first step. However, it does not yet appear to be useable for practical problems, because it assumes that policies followed by both players will be Nash equilibrium policies, and it does not address the "equilibrium coordination" problem, i.e. if there are multiple Nash equilibria, how do the agents decide which equilibrium to choose? This may be a serious problem, since according to the "folk theorem of iterated games" (Kreps, 1990), there can be a proliferation of Nash equilibria when there is sufficiently high emphasis on future rewards, i.e., a large value of the discount parameter γ. Furthermore, there may be inconsistencies between the assumed Nash policies, and the policies implied by the Q-functions calculated by the algorithm.

The present work examines simultaneous Q-learning in an economically motivated two-player game. The players are assumed to be two sellers of similar or identical products, who compete against each other on the basis of price. At each time step, the sellers alternately take turns setting prices, taking into account the other seller's current price. After the price has been set, the consumers then respond instantaneously and deterministically, choosing either seller 1's product or seller 2's product (or no product) based on the current price pair (p_1, p_2), leading to an instantaneous reward or utility (U_1, U_2) given to sellers 1 and 2 respectively. It is assumed that both sellers have full knowledge of the expected consumer response for any given price pair, and in fact have full knowledge of both utility functions.

This work builds on prior research reported in (Tesauro and Kephart, 1998; Tesauro and Kephart, 1999). Those papers examined the effect of including foresight, i.e. an ability to anticipate longer-term consequences of an agent's current action. Two different algorithms for agent foresight were presented: (i) a generalization of the minimax search procedure in two-player zero-sum games; (ii) a generalization of the Policy Iteration method from dynamic programming, in which both players' policies are simultaneously improved, until self-consistent policy pairs are obtained that optimize expected reward over two time steps. It was found that including foresight in the agents' pricing algorithms generally improved overall agent profitability, and usually damped out or eliminated the pathological behavior of unending cyclic "price wars," in which long episodes of repeated undercutting amongst the sellers alternate with large jumps in price. Such price wars were found to be rampant in prior studies of agent economy models (Kephart, Hanson and Sairamesh, 1998; Sairamesh and Kephart, 1998) when the agents use "myopically optimal" or "myoptimal" pricing algorithms that optimize immediate reward, but do not anticipate the longer-term consequences of an agent's current price setting.

There are three primary motivations for studying simultaneous Q-learning in this paper. First, if Q-functions can be learned simultaneously and self-consistently for both players, the policies implied by those Q-functions should be self-consistently optimal. In other words, an agent will be able to correctly anticipate the longer-term consequences of its own actions, the other agents' actions, and will correctly model the other agents as having an equivalent capability. Hence the classic problem of infinite recursion of opponent models will be avoided. In contrast, in other approaches to adaptive multi-agent systems, these issues are more problematic. For example, (Hu and Wellman, 1996) study the situation of a single "strategic" agent, which is able to anticipate the market impact of its pricing actions, in a population of "reactive" agents, which have no such anticipatory capability. Likewise, (Vidal and Durfee, 1998) propose a recursive opponent modeling scheme, in which level-0 agents do no opponent modeling, level-1 agents model the opponents as being level-0, level-2 agents model the opponents as being level-1, etc.. In both of these approaches, there is no effective way for an agent to model other agents as being at an equivalent level of depth or complexity.

The second advantage of Q-learning is that the solutions should correspond to deep lookahead: in principle, the Q-function represents the expected reward looking infinitely far ahead in time, exponentially weighted by a discount parameter $0 < \gamma < 1$. In contrast, the prior work of (Tesauro and Kephart, 1999) was based on shallow finite lookahead. Finally, in comparison to directly modeling agent policies, the Q-function approach seems more extensible to the situation of very large economies with many competing sellers. Approximating Q-functions with nonlinear function approximators such as neural networks seems intuitively more feasible than approximating the corresponding policies. Furthermore, in the Q-function approach, each agent only needs to maintain a single Q-function for itself, whereas in the policy modeling approach, each agent needs to maintain a policy model for every other agent; the latter seems infeasible when the number of sellers is large.

The remainder of this paper is organized as follows. Section 2 describes the structure and dynamics of the model two-seller economy, and presents three economically-based models of seller utility (Price-Quality, Information-Filtering, and Shopbot) which are known to be prone to price wars when agents myopically optimize their short-term payoffs. System parameters are chosen to place each of these systems in a price-war regime. Section 3 describes implementation details of Q-learning in these model economies. As a first step, the simple case of ordinary Q-learning is considered, where one of the two sellers uses Q-learning and the other seller uses a fixed pricing policy (the myopically optimal, or "myoptimal" policy). Section 4 examines the more interesting and novel situation of simultaneous Q-learning by both sellers. Section 5 studies single-agent Q-learning in these models using neural networks, and compares the results to those of section 3 using lookup tables. Finally, section 6 summarizes the main conclusions and discusses promising directions and challenges for future work.

2 Model agent economies

Real agent economies are likely to contain large numbers of agents, with complex details of how the agents behave and interact with each other on multiple time scales. In order to make initial progress, a number of simplifying assumptions are made. The economy is restricted to two competing sellers, offering similar or identical products to a large population of consumer agents. The sellers compete on the basis of price, and it is assumed that prices are discretized and can lie between a minimum and maximum price, such that the number of possible prices is at most a few hundred. This renders the state space small enough that it is feasible to use lookup tables to represent the agents' pricing policies and expected utilities. Time in the simulation is also discretized; at each time step, the consumers compare the current prices of the two sellers, and instantaneously and deterministically choose to purchase from at most one seller. Hence at each time step, for each possible pair of seller prices, there is a deterministic reward or utility given to each seller. The simulation can iterate forever, and there may or may not be a discounting factor for the present value of future rewards.

It is worth noting that the consumers are not regarded as "players" in the model. The consumers have no strategic role: they behave according to an extremely simple, fixed, short-term greedy rule (buy the lowest priced product at each time step), and are regarded as merely providing a stationary environment in which the two sellers can compete in a two-player game. This is clearly a simplifying first step in the study of multi-agent phenomena, and in future work, the models will be extended to include strategic and adaptive behavior on the part of the consumers as well. This will change the notion of "desirable" system behavior. In the present model, desirable behavior would resemble "collusion" between the two sellers in charging very high prices, so that both could obtain high profits. Obviously this is not desirable from the consumers' viewpoint.

Regarding the dynamics of seller price adjustments, it is assumed that the sellers alternately take turns adjusting their prices, rather than simultaneously setting prices (i.e. the game is extensive form rather than normal form). The choice of alternating-turn dynamics is motivated by two considerations: (a) As the number of sellers becomes large and the model becomes more realistic, it seems more reasonable to assume that the sellers will adjust their prices at different times rather than at the same time, although they probably will not take turns in a well-defined order. (b) With alternating-turn dynamics, one can stay within the normal Q-learning framework where the Q-function implies a deterministic optimal policy: it is known that in two-player alternating turn games, there always exists a deterministic policy that is as good as any non-deterministic policy (Littman, 1994). In contrast, in games with simultaneous moves (for example, rock-paper-scissors), it is possible that no deterministic policy is optimal, and that the existing Q-learning formalism for MDPs would have to be modified and extended so that it could yield non-deterministic optimal policies.

Q-learning is studied in three different economic models that have been described in detail elsewhere Sairamesh and Kephart, 1998; Kephart, Hanson and Sairamesh, 1998; Greenwald and Kephart, 1999). The first model, called the "Price-Quality" model (Sairamesh and Kephart, 1998), models the sellers' products as being distinguished by different values of a scalar "quality" parameter, with higher-quality products being perceived as more valuable by the consumers. The consumers are modeled as trying to obtain the lowest-priced product at each time step, subject to threshold-type constraints on both quality and price, i.e., each consumer has a maximum allowable price and a minimum allowable quality. The similarity and substitutability of seller products leads to a potential for direct price competition; however, the "vertical" differentiation due to differening quality values leads to an asymmetry in the sellers' utility functions. It is believed that this asymmetry is responsible for the unending cyclic price wars that emerge when the sellers employ myoptimal pricing strategies.

The second model is an "Information-Filtering" model described in detail in (Kephart, Hanson and Sairamesh, 1998). In this model there are two competing sellers of news articles in somewhat overlapping categories. In contrast to the vertical differentiation of the Price-Quality model, this model contains a horizontal differentiation in the differing article categories. To the extent that the categories overlap, there can be direct price competition, and to the extent that they differ, there are asymmetries introduced that again lead to the potential for cyclic price wars.

The third model is the so-called "Shopbot" model described in (Greenwald and Kephart, 1999), which is intended to model the situation on the Internet in which some consumers may use a Shopbot to compare prices of all sellers offering a given product, and select the seller with the lowest price. In this model, the sellers' products are exactly identical and the utility functions are symmetric. Myoptimal pricing leads the sellers to undercut each other until the minimum price point is reached. At that point, a new price war cycle can be launched, due to buyer asymmetries rather than seller asymmetries. The fact that not all buyers use the Shopbot, and some buyers instead choose a seller at random, means that it can be profitable for a seller to abandon the low-price competition for the bargain hunters, and instead maximally exploit the random buyers by charging the maximum possible price.

An example economic utility function, taken from the price-quality model, is as follows: Let p_1 and p_2 represent the prices charged by seller 1 and seller 2 respectively. Let q_1 and q_2 represent their respective quality parameters, with $q_1 > q_2$. Let $c(q)$ represent the cost to a seller of producing an item of quality q. Then assuming the particular model of consumer behavior described in [9], one can show analytically that in the limit of infinitely many consumers, the instantaneous utilities (profits per consumer) U_1 and U_2 obtained by seller 1 and seller 2 respectively are given by:

$$U_1 = \begin{cases} (q_1 - p_1)(p_1 - c(q_1)) & \text{if } 0 \le p_1 \le p_2 \text{ or } p_1 > q_2 \\ (q_1 - q_2)(p_1 - c(q_1)) & \text{if } p_2 < p_1 < q_2 \end{cases} \tag{1}$$

$$U_2 = \begin{cases} (q_2 - p_2)(p_2 - c(q_2)) & \text{if } 0 \le p_2 < p_1 \\ 0 & \text{if } p_2 \ge p_1 \end{cases} \tag{2}$$

A plot of the utility landscape for seller 1 as a function of prices p_1 and p_2 is given in figure 1, for the following parameter settings: $q_1 = 1.0$, $q_2 = 0.9$, and $c(q) = 0.1(1+q)$. (These specific parameter settings were chosen because they are known to generate harmful price wars when the agents use myopic optimal pricing.) We can see in this figure that the myopic optimal price for seller 1 as a function of seller 2's price, $p_1^*(p_2)$, is obtained for each value of p_2 by sweeping across all values of p_1 and choosing the value that gives the highest utility. We can see that for small values of p_2, the peak utility is obtained at $p_1 = 0.9$, whereas for larger values of p_2, there is eventually a discontinuous shift to the other peak, which follows along the parabolic-shaped ridge in the landscape. An analytic expression for the myopic optimal price for seller 1 as a function of p_2 is as follows (defining $x_1 = q_1 + c(q_1)$ and $x_2 = q_2 + c(q_2)$):

$$p_1^*(p_2) = \begin{cases} q_2 & \text{if } 0 \le p_2 < x_1 - q_2 \\ p_2 & \text{if } x_1 - q_2 \le p_2 \le \frac{1}{2}x_1 \\ \frac{1}{2}x_1 & \text{if } p_2 > \frac{1}{2}x_1 \end{cases} \tag{3}$$

Similarly, the myopic optimal price for seller 2 as a function of the price set by seller 1, $p_2^*(p_1)$, is given by the following formula (assuming that prices are discrete and that ϵ is the price discretization interval):

$$p_2^*(p_1) = \begin{cases} c(q_2) & 0 \le p_1 \le c(q_2) \\ p_1 - \epsilon & \text{if } c(q_2) \le p_1 \le \frac{1}{2}x_2 \\ \frac{1}{2}x_2 & \text{if } p_1 > \frac{1}{2}x_2 \end{cases} \tag{4}$$

There are also similar utility landscapes for each seller in the Information-Filtering model and in the Shopbot model. In all three models, it is the existence of multiple, disconnected peaks in the landscapes, with relative heights that can change depending on the other seller's price, that leads to price wars when the sellers behave myopically.

In these models it is assumed for simplicity that the players have essentially perfect information. They can model the consumer behavior perfectly, and they also have perfect knowledge of each other's costs and utility functions. Hence the model is in essence a two-player perfect-information deterministic game, similar to games

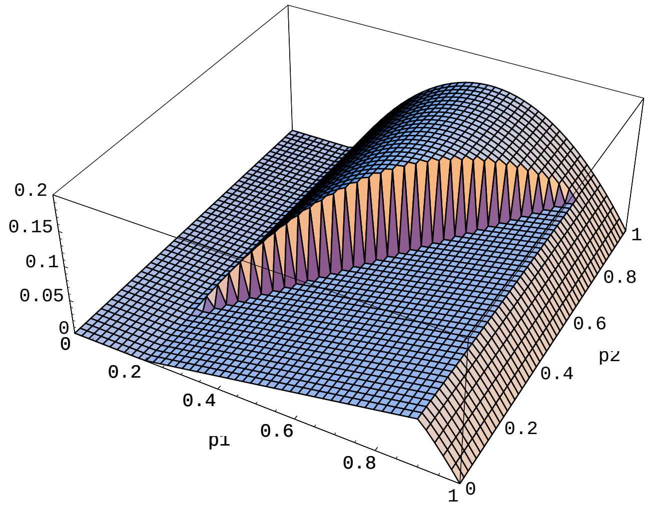

Figure 1: Sample utility landscape for seller 1 in price-quality model, as a function of seller 1 price p_1 and seller 2 price p_2.

like chess. The main differences are that the utilities are not strictly zero-sum, and that there are no terminating or absorbing nodes in the state space. Also, payoffs are given to the players at every time step, whereas in games such as chess, payoffs are only given at the terminating nodes.

As mentioned previously, the possible seller prices are constrained to lie in a range from some minimum to maximum allowable price. The prices are discretized, so that one can create lookup tables for the seller utility functions $U(p_1, p_2)$. Furthermore, the optimal pricing policies for each seller as a function of the other seller's price, $p_1^*(p_2)$ and $p_2^*(p_1)$, can also be represented in the form of table lookups.

3 Single-agent Q-learning

Let us first consider ordinary single-agent Q-learning in the above two-seller economic models: one seller uses Q-learning to learn a Q-function and corresponding policy, while the other seller maintains a fixed pricing policy.

The procedure for Q-learning is as follows. Let $Q(s, a)$ represent the discounted long-term expected reward to an agent for taking action a in state s. The discounting of future rewards is accomplished by a discount parameter γ such that the value of a reward expected at n time steps in the future is discounted by γ^n. Assume that the $Q(s, a)$ function is represented by a lookup table containing a value for every possible state-action pair, and assume that the table entries are initialized to arbitrary values. Then the procedure for solving for $Q(s, a)$ is to infinitely repeat the following two-step loop:

1. Select a particular state s and a particular action

a, observe the immediate reward r for this state-action pair, and observe the resulting state s'.

2. Adjust $Q(s,a)$ according to the following equation:

$$\Delta Q(s,a) = \alpha[r + \gamma \max_b Q(s',b) - Q(s,a)] \qquad (5)$$

where α is the learning rate parameter, and the max operation represents choosing the optimal action b among all possible actions that can be taken in the successor state s' leading to the greatest Q-value. A wide variety of methods may be used to select state-action pairs in step 1, provided that every state-action pair is visited infinitely often. For any stationary Markov Decision Problem, the Q-learning procedure is guaranteed to converge to the correct values, provided that the learning rate parameter is decreased over time with an appropriate schedule.

In the simulations described below the fixed-policy seller uses the myoptimal policy $p*$ represented for example in the Price-Quality model by equations 3 and 4.

In this model, the distinction between states and actions is somewhat blurred. It will be assumed that the "state" for each seller is sufficiently described by the other seller's last price, and that the "action" is the current price decision. This should be a sufficient state description because no other history is needed either for the determination of immediate reward, or for the calculation of the myoptimal price by the fixed-strategy player. The definitions of immediate reward r and next-state s' have also been modified for the two-agent case as follows: let s' be the state that is obtained, starting from s, of one action by the Q-learner and a response action by the fixed-strategy opponent. Likewise, the immediate reward is defined as the sum of the two rewards obtained after those two actions. These modifications were introduced so that the state s' would have the same player to move as state s. (A possible alternative to this, which has not been investigated, is to include the side-to-move as additional information in the state-space description.)

An important issue in Q-learning is "exploration" of the state space. Each state-action pair must be visited sufficiently often for learning to converge. This often necessitates some sort of randomized off-policy trajectory generation, for example, by Boltzmann exploration. In the simulations reported below, the sequence of state-action pairs selected for the Q-table updates were generated by uniform random selection from amongst all possible table entries. This does not correspond to actual on-line training in a real environment, but is appropriate for situations where one has an accurate simulation of the environment, and can envision attempting to solve for optimal strategies for both players via off-line training in the laboratory.

The initial values of the Q-tables were generally set to the immediate reward values. (Consequently the initial Q-derived policies corresponded to myoptimal policies.) The learning rate was varied with time according to:

$$\alpha(t) = \alpha(0)/(1 + \beta t) \qquad (6)$$

where the initial learning rate $\alpha(0)$ was usually set to 0.1, and the constant $\beta \sim .01$ when the simulation time t was measured in units of N^2, the size of the Q-table. (N is the number of possible prices that could be selected by either player.) A number of different values of the discount parameter γ were studied, ranging from $\gamma = 0$ to $\gamma = .9$.

Results for single-agent Q-learning in all three models indicated that Q-learning worked well (as expected) in each case. In each model, for each value of the discount parameter, exact convergence of the Q-table to a stationary optimal solution was found. The convergence times ranged from a few hundred sweeps through each table element, for smaller values of γ, to at most a few thousand updates for the largest values of γ. In addition, once Q-learning converged, the expected cumulative profit or utility of the policy derived from the Q-function was then measured, by running the Q-policy against the other player's myopic policy from 100 random starting states, each for 200 time steps, and averaging the resulting cumulative utility for each player. In each case, it was found that the seller achieved greater profit against a myopic opponent by using a Q-derived policy than by using a myopic policy. (This was true even for $\gamma = 0$, because, due to the redefinition of Q updates summing over two time steps, the case $\gamma = 0$ effectively corresponds to a two-step optimization, rather than the one-step optimization of the myopic policies.) Furthermore, the cumulative utility obtained with the Q-derived policy monotonically increased with the increasing γ (as expected).

It was also interesting to note that in many cases, the expected utility of the myopic opponent also increased when playing against the Q-learner, and also improved monotonically with increasing γ. The explanation is that, rather than better exploiting the myopic opponent, as would be expected in a zero-sum game, the Q-learner instead reduced the region over which it would participate in a mutually undercutting price war. Typically one finds in these models that with myopic vs. myopic play, large-amplitude price wars are generated that start at very high prices and persist all the way down to very low prices. When a Q-learner competes against a myopic opponent, there are still price wars starting at high prices, however, the Q-learner abandons the price war more quickly as the prices decrease. The effect is that the price-war regime is smaller and confined to higher average prices, leading to a closer approximation to cooperative or collusive behavior, with greater expected utilites for both players. It is interesting that this can come about even though both players are entirely selfish.

An illustrative example of the results of single-agent Q-learning is shown below in figures 2 and 3. Figure 2 plots the average utility for both sellers in the Shopbot model, when one of the sellers is myopic and the other is a Q-learner. (As the model is symmetric, it doesn't matter which seller is the Q-learner.)

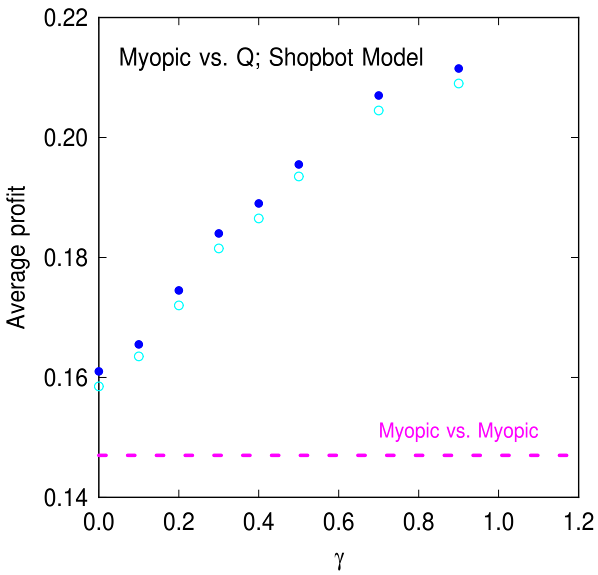

Figure 2: Plot of average utility per time step for seller 1 and seller 2, as a function of discount parameter γ, in the Shopbot model when one seller (seller 2, open circles) uses a myopic policy, and the other seller (seller 1, filled circles) uses Q-learning to learn an optimal policy against the myopic player. The dashed line indicates baseline expected utility when both sellers are myopic.

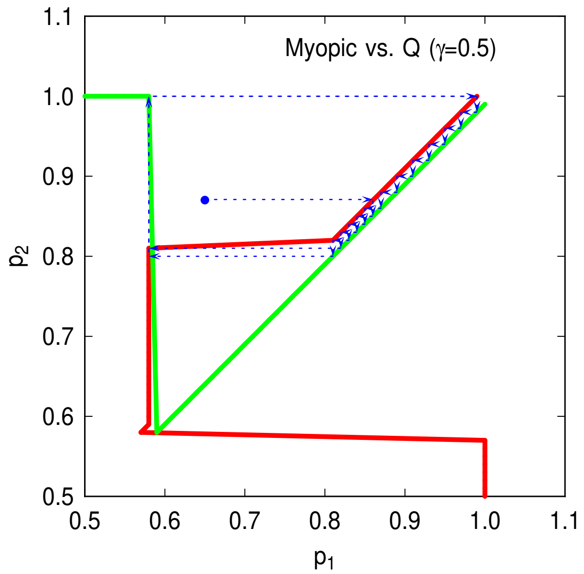

Figure 3: Cross-plot of myopic price curve vs. Q-derived price curve in the Shopbot model at $\gamma = 0.5$. Seller 1 is the Q-learner and seller 2 is myopic. The dashed line and arrows indicate how the prices will evolve over time using these pricing policies, starting from a particular price pair, indicated by the filled circle.

Figure 3 plots the myopic price curve of seller 2 against the Q-derived price curve (at $\gamma = 0.5$) of seller 1. We can see that both curves have a maximum price of 1 and a minimum price of approximately 0.58. The portion of both curves lying along the diagonal indicates undercutting behavior, in which case the seller will respond to the opponent's price by undercutting by ϵ, the price discretization interval.

The system dynamics for the state (p_1, p_2) in figure 3 can be obtained by alternately applying the two pricing policies. This can be done by a simple iterative graphical construction, in which for any given starting point, one first holds p_2 constant and moves horizontally to the $p_1(p_2)$ curve, and then one holds p_1 constant and moves vertically to the $p_2(p_1)$ curve. We see in this figure that the iterative graphical construction leads to an unending cyclic price war, whose trajectory is indicated by the dashed line. Note that the price-war behavior begins at the price pair (1, 1), and persists until a price of approximately 0.83. At this point, seller 1 abandons the price war, and resets its price to 1, leading once again to another round of undercutting.

The amplitude of this price war is diminished compared to the situation in which both players use a myopic policy. In that case, seller 1's curve would be a mirror image of seller 2's curve, and the price war would persist all the way to the minimum price point, leading to a lower expected utility for both sellers.

4 Multi-agent Q-learning

Let us now turn to the more challenging situation of simultaneous training of Q-functions and policies for both sellers. The procedure studied here is to alternately adjust a random entry in seller 1's Q-function, followed by a random entry in seller 2's Q-function, using the same formalism presented in the previous section. As each seller's Q-function evolves, the seller's pricing policy is correspondingly updated so that it optimizes the agent's current Q-function. In modeling the two-step payoff r to a seller in equation 5, the opponent's current policy is used, as implied by its current Q-function. The parameters in the experiments below were generally set to the same values as in the previous section. In most of the experiments, the Q-functions were initialized to the instantaneous payoff values (so that the policies corresponded to myopic policies), although other initial conditions were explored in a few experiments.

For simultaneous Q-learning in the Price-Quality model, we find robust convergence to a unique pair of pricing policies, independent of the value of γ, as illustrated in figure 4. This solution also corresponds to the solution found by generalized minimax and by generalized DP in (Tesauro and Kephart, 1999). Repeated application of this pair of price curves leads to a dynamical trajectory that eventually converges to a fixed-point located at $(p_1 = 0.9, p_2 = 0.4)$. A detailed analysis of these pricing policies and the fixed-point solution is presented in (Tesauro and Kephart, 1999). In brief, for sufficiently low prices of seller 2, it pays seller 1 to abandon the price war and to charge a very high price, $p_1 = 0.9$. The value of $p_2 = 0.4$ then corresponds to the highest price that seller 2 can charge without provoking an undercut by seller 1, based on a two-step lookahead calculation (seller 1 undercuts, and then seller 2 replies with a further undercut). This fixed point differs from the Nash equilibrium for this game, which was calculated in (Sairamesh and Kephart, 1998) to be $(p_1 = 0.9, p_2 = 0.545)$. The difference is due to two factors: (i) these models assume alternating-turn dynamics, rather than the simultaneous-move dynamics assumed by the Nash calculation; (ii) the game here is an iterated game, whereas the Nash calculation assumes a one-shot game. It was conjectured in (Tesauro and Kephart, 1999) that the solution observed in figure 4 corresponds to a subgame-perfect equilibrium (Fudenberg and Tirole, 1991) rather than a Nash equilibrium.

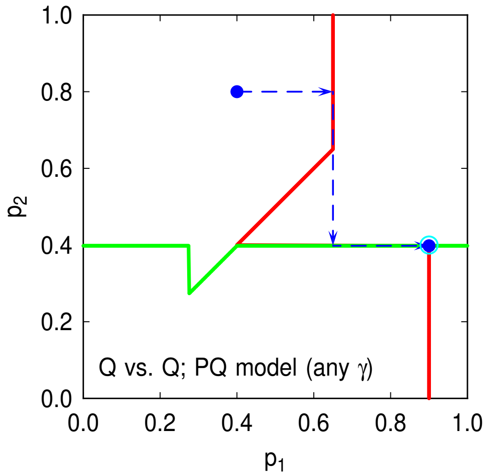

Figure 4: Cross-plot of optimal price curves for seller 1 vs. seller 2 in Price-Quality model obtained by simultaneous Q-learning; the same solution was found for all values of γ. The dashed line indicates how the prices will evolve over time using these pricing policies, starting from a particular price pair, indicated by the filled circle. The price war is eliminated with these pricing curves, and the dynamics instead evolves to a fixed point indicated by an open circle.

The cumulative utilities obtained by the pair of pricing policies in figure 4 are plotted in figure 5. It is interesting that seller 2, the lower-quality seller, actually obtains a significantly higher profit than seller 1, the higher-quality seller. In contrast, with myopic vs. myopic pricing, seller 2 does worse than seller 1.

In the Shopbot model, exact convergence of the Q-functions was not found for all values of γ. However, in those cases where exact convergence was not found, there was very good approximate convergence, in which the Q-functions and policies converged to stationary solutions to within small random fluctuations. Different

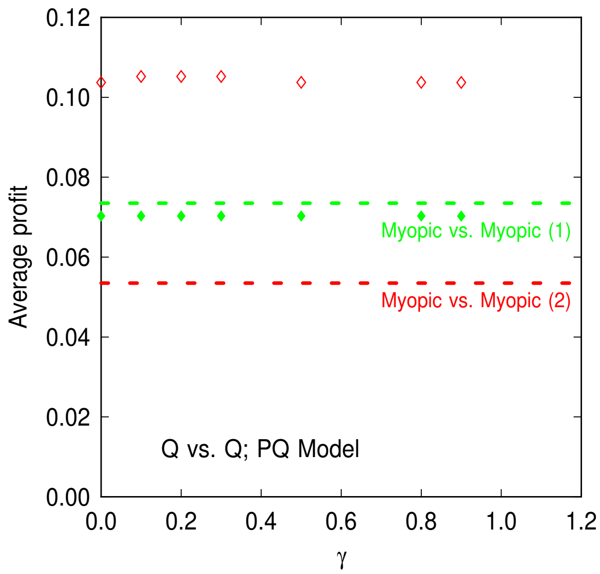

Figure 5: Plot of expected utilities for seller 1 (solid diamonds) and seller 2 (open diamonds) in Price-Quality model obtained by simultaneous Q-learning. By comparison, the utilities for seller 1 and seller 2 in myopic vs. myopic pricing are indicated as dashed lines. Seller 2's utility is higher than seller 1's in the simultaneous Q-learning solution, even though seller 2 has a lower quality parameter.

solutions were obtained at each value of γ. For small γ, a symmetric solution is generally obtained (in which the shapes of $p_1(p_2)$ and $p_2(p_1)$ are identical), whereas a broken symmetry solution, similar to the Price-Quality solution, is obtained at large γ. There was also a range of γ values, between 0.1 and 0.2, where either a symmetric or asymmetric solution could be obtained, depending on initial conditions. The asymmetric solution seems counter-intuitive because we expected that the symmetry of the two sellers' utility functions would lead to a symmetric solution. In hindsight, one can apply the same type of reasoning as in the Price-Quality model to explain the asymmetric solution. Plots of the symmetric and asymmetric solution, obtained at $\gamma = 0$ and $\gamma = 0.9$ respectively, are shown in figures 6 and 7. A plot of the expected utility for both sellers as a function of γ is shown in figure 8.

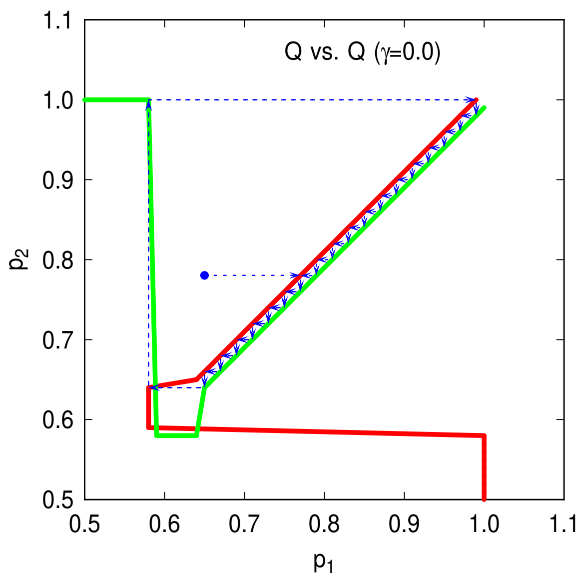

Figure 6: Cross-plot of optimal price curves for seller 1 vs. seller 2 in the Shopbot model obtained by simultaneous Q-learning at $\gamma = 0$. The resulting price dynamics is indicated by the dashed line and arrows.

Finally, in the Information-Filtering model, simultaneous Q-learning produced exact or good approximate convergence for small values of γ ($0 \leq \gamma \leq 0.5$). For large values of γ, no convergence was obtained. The simultaneous Q-learning solutions yielded reduced-amplitude price wars, and montonically increasing profitability for both sellers as a function of γ, at least up to $\gamma = 0.5$. A few data points were examined at $\gamma > 0.5$, and even though there was no convergence, the Q-policies still yielded greater utility for both sellers than in the myopic vs. myopic case. A plot of the Q-derived policies and system dynamics for $\gamma = 0.5$ is shown in figure 9. The expected utilities for both players as a function of γ is plotted in figure 10.

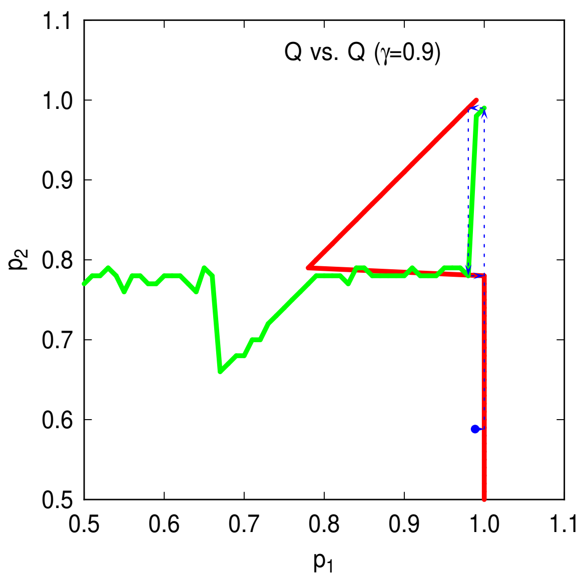

Figure 7: Cross-plot of optimal price curves for seller 1 vs. seller 2 in the Shopbot model obtained by simultaneous Q-learning at $\gamma = 0.9$. The resulting price dynamics is indicated by the dashed line and arrows.

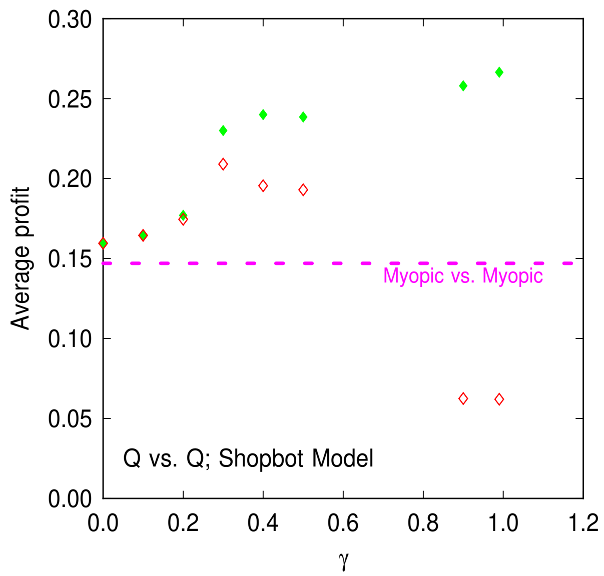

Figure 8: Plot of expected utilities for seller 1 (solid diamonds) and seller 2 (open diamonds) in Shopbot model obtained by simultaneous Q-learning, for values of γ ranging from 0.0 to 0.99. By comparison, the utilities for seller 1 and seller 2 in myopic vs. myopic pricing are indicated as a dashed line.

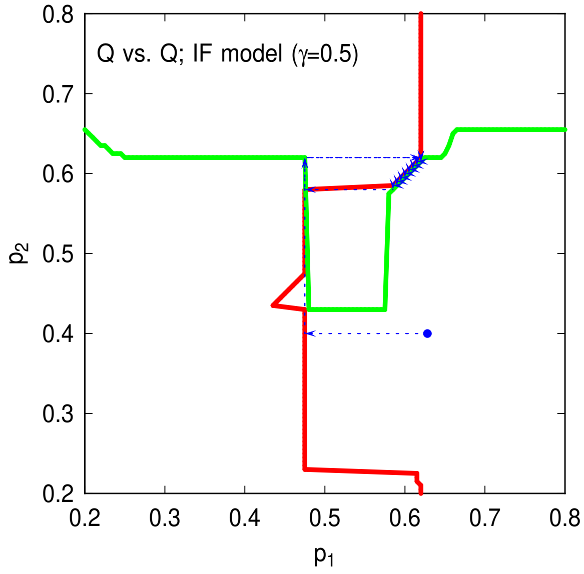

Figure 9: Cross-plot of optimal price curves for seller 1 vs. seller 2 in the Information-Filtering model obtained by simultaneous Q-learning at $\gamma = 0.5$.

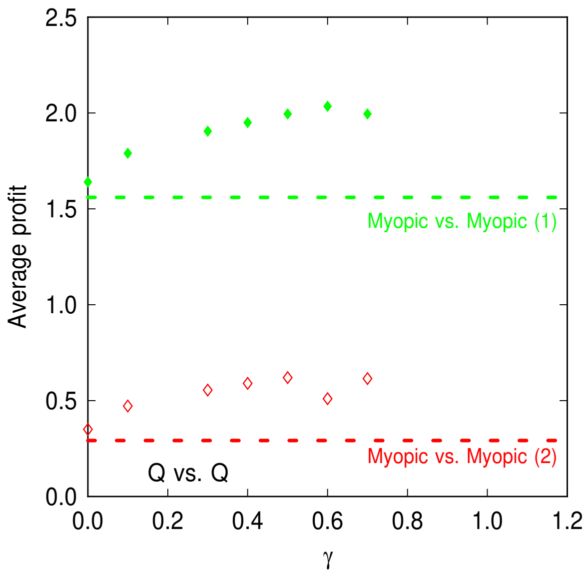

Figure 10: Plot of expected utilities for seller 1 (solid diamonds) and seller 2 (open diamonds) in the Information-Filtering model obtained by simultaneous Q-learning, for values of γ ranging from 0.0 to 0.7. (The data points at $\gamma = 0.6, 0.7$ represent unconverged Q-functions and policies.) By comparison, the utilities for seller 1 and seller 2 in myopic vs. myopic pricing are indicated as dashed lines.

5 Q-learning with neural networks

Using lookup tables to represent Q-functions as described in the previous two sections can only be feasible for small-scale problems. It is likely that the situations that will be faced by software agents in the real world will be too large-scale and complex to tackle via lookup tables, and that some sort of function approximation scheme will be necessary. This section examines the use of multi-layer neural networks to represent the Q-functions in the same economic models studied previously. Some initial results are presented for the case of a single adaptive QNN (Q-learning neural network) agent, training vs. a fixed-strategy myopic agent, as was described previously in section 3.

The neural networks studied here are multi-layer perceptrons (MLPs) as used in back-propagation. The same Q-learning equation 5 is used as previously, however, the quantity $\Delta Q(s, a)$ is interpreted as the output error signal used in a backprop-style gradient calculation of weight changes. As in the previous sections, the state-action pairs (s, a) are chosen by uniform random exploration, although there is some preliminary evidence that somewhat better policies can be obtained by training on actual trajectories. Also in the experiments below, a fixed learning-rate constant $\alpha = 0.1$ was used, rather than the time-varying schedule $\alpha(t)$ described previously. This appears to give a significant speed increase at the cost of only a slight degradation in final network performance.

One of the most important issues in using neural networks is the design of the input state representation scheme. Schemes that incorporate specialized knowledge of the domain can often do better than naive representation schemes. The only knowledge included here is that it is important for a seller to know whether its price is greater than, less than, or equal to the other seller's price. This suggests a coding scheme using five input units. The first two units represent the two seller prices (p_1, p_2) as real numbers, and the remaining three units are binary units representing the three logical conditions $[p_1 < p2], [p_1 = p_2], [p_1 > p_2]$.

In the experiments below, the networks contained a single linear output unit, and a single hidden layer of 10 hidden units, fully connected to the input layer. For the Shopbot model, it was found that the network's ability to approximate the correct Q-function was generally poor, with the worst accuracy obtained at large values of γ. Furthermore, the improvement in approximation error with training was extremely slow, and continued to decrease at an extremely slow rate for as long as the training was continued. Typical training runs lasted for several tens of thousands of sweeps through all possible price pairs. In the case of $\gamma = 0$ an extremely long training run of several million sweeps was performed, after which the approximation accuracy was quite good, but the error was still decreasing at a very slow rate.

The difficulty in obtaining accurate function approximation could have resulted from inaccurate targets

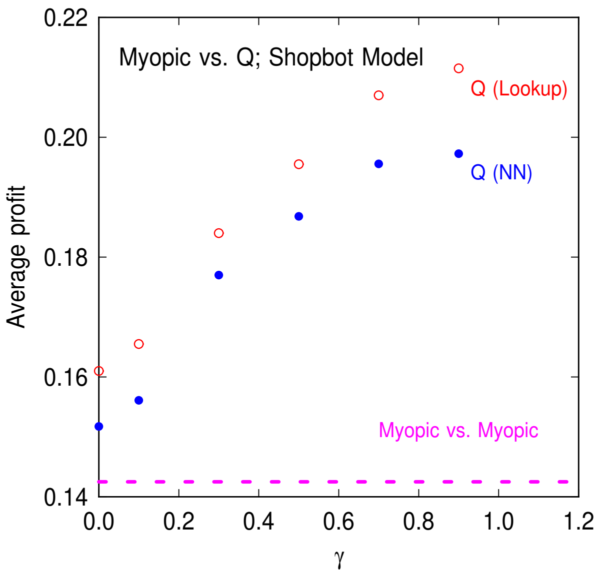

Figure 11: Plot of expected utility for a single Q-learning agent, training against a fixed myopic opponent, in the Shopbot model, as a function of γ. Filled circles represent a neural network Q-learner, while open circles represent a lookup table Q-learner. Each data point represents the best policy obtained during a training run. By comparison, the baseline myopic vs. myopic expected utility is indicated by the dashed line.

$\Delta Q(s, a)$ used in training, or it could be due to intrinsic limitations in the function approximator itself. In a separate series of experiments, neural nets were trained with exact targets provided by the lookup tables, and this yielded no measurable advantage in approximation accuracy, suggesting that the problem is not due to inaccurate heuristic teacher signals in equation 5.

During the neural net training runs, the expected performance of the neural net policy was periodically measured vs. the myopic opponent. While the absolute error in the Q-function improved monotonically, the policy's expected profit was found to reach a peak relatively quickly (usually withing 100-200 sweeps, but longer for large γ) and then either level off or decrease slightly. A plot of the peak expected utility for each training run as a function of γ is shown in figure 11. For comparison, the expected utility of the exact optimal policy, obtained by lookup table Q-learning, is also plotted. It is encouraging to note that, although the absolute accuracy of the neural net Q-function is poor and improves extremely slowly, the resulting policies nevertheless give reasonably decent performance and can be trained relatively quickly. This once again re-emphasizes a point found in other successful applications of neural nets and reinforcement learning: the neural net approach can often give a surprisingly strong policy, even though the absolute accuracy of the value function is poor.

6 Conclusions

This paper has examined single-agent and multi-agent Q-learning in three models of a two-seller economy in which the sellers alternately take turns setting prices, and then instantaneous utilities are given to both sellers based on the current price pair. Such models fall into the category of two-player, alternating-turn, arbitrary-sum Markov games, in which both the rewards and the state-space transitions are deterministic. The game is Markov because the state space is fully observable and the rewards are not history dependent.

In all three models (Price-Quality, Information-Filtering, and Shopbot), large-amplitude cyclic price wars are obtained when the sellers myopically optimize their instantaneous utilities without regard to longer-term impact of their pricing policies. It is found that, in all three models, the use of Q-learning by one of the sellers against a myopic opponent invariably results in exact convergence to the optimal Q-function and optimal policy against that opponent, for all allowed values of the discount parameter γ. The use of the Q-derived policy yields greater expected utility for the Q-learner, with monotonically increasing utility as γ increases. In many cases, it has a side benefit of enhancing social welfare by also giving greater expected utility for the myopic opponent. This comes about by reducing the amplitude of the undercutting price-war regime, or in some cases, eliminating it completely.

The more interesting and challenging situation of simultaneously training Q-functions for both sellers has also been studied. This is more difficult because as each seller's Q-function and policy change, it provides a non-stationary environment for adaptation of the other seller. No convergence proofs exist for such simultaneous Q-learning by multiple agents. Nevertheless, despite the absence of theoretical guarantees, generally good behavior of the algorithm was found. In two of the models (Shopbot and Price-Quality), exact or very good approximate convergence was obtained to simultaneously self-consistent Q-functions and optimal policies for any value of γ, whereas in the Information-Filtering model, simultaneous convergence was found for $\gamma \leq 0.5$. In the Information-Filtering and Shopbot models, monotonically increasing expected utilities for both sellers were also found for small values of γ. In the Price-Quality model, simultaneous Q-learning yields an asymmetric solution, corresponding to the solution found in (Tesauro and Kephart, 1999), that is highly advantageous to the lesser-quality seller, but slightly disadvantageous to the higher-quality seller, when compared to myopic vs. myopic pricing. A similar asymmetric solution is also found in the Shopbot model for large γ, even though the utility functions for both players are symmetric.

For each model, there exists a range of discount parameter values where the solutions obtained by simultaneous Q-learning are self-consistently optimal, and outperform the solutions obtained in (Tesauro and Kephart, 1999). This is presumably because the previously published methods were based on limited lookahead, whereas the Q-functions in principle look ahead infinitely far, with appropriate discounting.

It is intruiging that simultaneous Q-learning works well in our models, despite the lack of theoretical convergence proofs. Sandholm and Crites also found that simultaneous Q-learning generally converged in the Iterated Prisoner's Dilemma game. These empirical findings suggest that a deeper theoretical analysis of simultaneous Q-learning may be worth investigating. There may be some underlying theoretical principles that can explain why simultaneous Q-learning works, for at least certain classes of arbitrary-sum utility functions.

Some initial steps have also been taken in combining nonlinear function approximation, using neural nets, with the Q-learning approach. It was found that a single neural net Q-learner facing a myopic opponent can exhibit reasonably good pricing policies, despite difficulties in obtaining an accurate approximation to the Q-function.

In addition to replacing lookup tables with function approximators, several other important challenges will also be faced in extending our approach to larger-scale, more realistic simulations. First, the three economic models used here quite deliberately ignored frictional effects such as agent search costs. Such effects can damp out price wars, and can lead to different system behaviors such as partial equilibria that support stable price differentiation. Eventually such frictional effects will have to be considered, although it has been argued in prior studies of these models (Sairamesh and Kephart, 1998; Kephart, Hanson and Sairamesh, 1998; Greenwald and Kephart, 1999) that frictional effects in Web-based agent economies will be considerably smaller than in traditional human economies. Also, with many sellers, the concept of sellers taking turns adjusting their prices in a well-defined order becomes problematic. This could lead to an additional combinatorial explosion, if the mechanism for calculating expected reward has to anticipate all possible orderings of opponent responses.

Furthermore, while these economic models have a moderate degree of realism in their utility functions, they are unrealistic in the assumptions of knowledge and dynamics. In the work reported here, the state space was fully observable infinitely frequently at zero cost and with zero propagation delays. The expected consumer response to a given price pair was instantaneous, deterministic and fully known to both players. Indeed, the players' exact utility functions were fully known to both players. It was also assumed that the players would alternately take turns equally often in a well-defined order in adjusting their prices. Under such assumptions of knowledge and dynamics, one could hope to develop an algorithm that could calculate in advance something like a game-theoretic optimal pricing algorithm for each agent.

However, in realistic agent economies, it is likely that agents will have much less than full knowledge of the state of the economy. Agents may not know the details of other agents' utility functions, and indeed an agent may not know its own utility function, to the extent that buyer behavior is unpredictable. The dynamics of buyers and sellers may also be more complex, random and unpredictable than what we have assumed here. There may also be information delays for both buyers and sellers, and part of the economic game may involve paying a cost in order to obtain information about the state of the economy faster and more frequently, and in greater detail. Finally, one may expect that buyer behavior will be non-stationary, so that there will be a more complex co-evolution of buyer and seller strategies.

While such real-world complexities are daunting, there are reasons to believe that learning approaches such as Q-learning may play a role in practical solutions. The advantage of Q-learning is that one does not need a model of either the instantaneous payoffs or of the state-space transitions in the environment. One can simply observe actual rewards and transitions and base learning on that. While the theory of Q-learning requires exhaustive exploration of the state space to guarantee convergence, this may not be necessary when function approximators are used. In that case, after training a function approximator on a relatively small number of observed states, it may then generalize well enough on the unobserved states to give decent practical performance. Several recent empirical studies have provided evidence of this (Tesauro, 1995; Crites and Barto, 1996; Zhang and Dietterich, 1996).

Acknowledgements

The author thanks Jeff Kephart and Amy Greenwald for helpful discussions.

References

[1] R. H. Crites and A. G. Barto, "Improving elevator performance using reinforcement learning." In: D. Touretzky et al., eds., Advances in Neural Information Processing Systems 8, 1017-1023, MIT Press, 1996.

[2] D. Fudenberg and J. Tirole, *Game Theory.* Cambridge, MA: MIT Press, 1991.

[3] A. Greenwald and J. O. Kephart, "Shopbots and pricebots." To appear in Proceedings of IJCAI '99 (International Joint Conferences on Artificial Intelligence), July 31- August 6, 1999, Stockholm, Sweden.

[4] J. Hu and M. P. Wellman, "Self-fulfilling bias in multiagent learning." Proceedings of ICMAS-96, AAAI Press, 1996.

[5] J. Hu and M. P. Wellman, "Multiagent reinforcement learning: theoretical framework and an algorithm." Proceedings of ICML-98, 1998.

[6] J. O. Kephart, J. E. Hanson and J. Sairamesh, "Price-war dynamics in a free-market economy of software agents." In: Proceedings of ALIFE-VI, Los Angeles, 1998.

[7] D. Kreps, *A Course in Microeconomic Theory.* Princeton Univ. Press, Princeton, NJ, 1990.

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

[9] J. Sairamesh and J. O. Kephart, "Dynamics of price and quality differentiation in information and computational markets." Proceedings of the First International Conference on Information and Computation Economics (ICE-98), 28–36, ACM Press, 1998.

[10] T. W. Sandholm and R. H. Crites, "On multiagent Q-Learning in a semi-competitive domain." 14th International Joint Conference on Artificial Intelligence (IJCAI-95), Workshop on Adaptation and Learning in Multiagent Systems, Montreal, Canada, 71–77, 1995.

[11] G. Tesauro, "Temporal difference learning and TD-Gammon." *Comm. of the ACM*, **38:3**, 58-67, 1995.

[12] G. J. Tesauro and J. O. Kephart, "Foresight-based pricing algorithms in an economy of software agents." Proceedings of ICE-98, 37–44, 1998.

[13] G. J. Tesauro and J. O. Kephart, "Foresight-based pricing algorithms in agent economies." *Decision Support Sciences*, to appear, 1999.

[14] J. M. Vidal and E. H. Durfee, "Learning nested agent models in an information economy," J. of Experimental and Theoretical AI, to appear, 1998.

[15] C. J. C. H. Watkins, "Learning from delayed rewards." Ph. D. thesis, Cambridge University, 1989.

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

[17] W. Zhang and T. G. Dietterich, "High-performance job-shop scheduling with a time-delay TD(λ) network." In: D. Touretzky et al., eds., Advances in Neural Information Processing Systems 8, 1024-1030, MIT Press, 1996.