Proceedings of the 2007 IEEE Symposium on Approximate
Dynamic Programming and Reinforcement Learning (ADPRL 2007)

Leader-Follower semi-Markov Decision Problems: Theoretical Framework and Approximate Solution

Kurian Tharakunnel
Department of Information and Decision Sciences
University of Illinois at Chicago
Chicago, Illinois 60607–7124
Email: kthara1@uic.edu

Siddhartha Bhattacharyya
Department of Information and Decision Sciences
University of Illinois at Chicago
Chicago, Illinois 60607–7124
Email: sidb@uic.edu

Abstract— **Leader-follower problems are hierarchical decision problems in which a leader uses incentives to induce certain desired behavior among a set of self-interested followers. Dynamic leader-follower problems extend this structure to multi-period decision situations. In this work we propose a Markov Decision Process (MDP) framework for a class of dynamic leader-follower problems that have important applications and discuss their approximate solution using reinforcement learning (RL). In these problems, the leader makes incentive decisions intermittently while the followers make their decisions in every period. Our theoretical framework and computational approach are based on the observation that such dynamic problems can be thought of as consisting of two coupled sequential decision processes, that of the leader and of the followers. In our formulation, the leader's decision problem that has the structure of a single-agent semi-Markov Decision process (SMDP), and the followers' sequential decision problem structured as a stochastic game (multiagent competitive MDP) operate over the same state space. We call this MDP framework a leader-follower semi-Markov Decision Process (LFSMDP). We consider approximate solution of these problems using RL and demonstrate the solution approach in the special case where the followers' stochastic game is a repeated game.**

I. Introduction

Leader-follower problems [1] are hierarchical decision problems in which a leader uses incentives to induce certain desired behavior among a set of self-interested followers. The leader and the followers make their decisions sequentially, with the leader making the decision first and announcing it to the followers. The followers, after knowing leader's decision, make their individual decisions concurrently and competitively. These decisions are interrelated in the sense that the followers' payoffs are contingent on the leader's decision while the leader's payoff is a function of the followers' actions. The leader, by its decision, tries to influence the decisions of the followers. The leader's decision thus acts as an incentive/threat to induce the followers to behave in a way that maximizes leader's payoff. The problem faced by the leader then is to design an incentive strategy under which the followers while acting to maximize their own individual objectives will maximize the leader's objective as well.

The leader-follower problems described above are models of many decentralized decision making situations encountered in business and government. Some recent applications of this framework include pricing in communication networks [2], regulation of electricity markets [3], pricing in peer-to-peer systems [4], coordination of supply chains [5], and reserve price-based online auctions [6]. In addition, this framework is used in modeling public policy formulation in pollution control, taxation etc. [7].

Dynamic leader-follower problems extend the leader-follower structure to multi-period decision situations. In these problems the leader and the followers make decisions over multiple periods. These dynamic models have important applications in many fields. For example, in almost every applications described above, it is likely that the leader and the followers interact over multiple (possibly infinite) periods of time, repeatedly making similar decisions. Dynamic models can capture the continuing nature of many leader-follower interactions where decisions are taken on an ongoing basis as information become available. Dynamic leader-follower models are important for several reasons:

- First, in many real-life situations, agents are faced with making similar decisions over and over again. The goal of agents in these situations is to achieve some long term measure of success.
- Secondly, the repeated interaction between the leader and the followers can provide better economic results. This is because multiple interactions provide opportunities for intertemporal solutions that do not exist in the static one-shot models. For example, Radner [8] shows how long term relationship achieves efficiency in repeated principal-agent situations.
- A third and important aspect of dynamic leader-follower models is that they can capture the learning effect inherent in multiple interactions. Learning provides agents information about other agents and the environment, enabling them to make better choices in the long run. For example, Kalai and Ledyard [9] discuss the advantages of repeated implementation and illustrate the learning effects in agents in repeated implementation.

In this paper, we study a special class of dynamic leader-

This research was supported by National Science Foundation grant ECS-0601590

follower problems in which the leader makes incentive decisions intermittently whereas the followers make their decisions in every period. We propose a Markov Decision Process (MDP) [10] framework for these class of problems. Our objective is in developing a theoretical framework that can describe these problems, specify the nature of their solution, and also provide a basis for their computational solution. We call the new MDP framework for these problems as leader-follower semi-Markov Decision Processes (LFSMDPs). This new MDP model takes into account the hierarchical and competitive nature of these leader-follower problems. We formulate value functions for the leader and the followers in a LFSMDP for the average reward case and derive optimality equations for the process.

Our model can be considered as an extension of the MDP framework to hierarchical and competitive situations. MDP models for hierarchical but non-competitive decision problems operating over multiple time scales are discussed in [11]. An MDP formulation for principal-agent problems is discussed in [12]. Principal-agent problems are special cases of leader-follower problems discussed in this work in which there is only one follower. A major simplification in principal-agent problems is that there is no need to model the strategic interaction among the followers as there is only one follower. An early work on dynamic leader-follower problems by Saksena and Cruz [13] uses a control theoretic formulation of the problem.

After discussing the LFSMDP formulation we consider the solution of these problems, especially approximate solution using Reinforcement Learning (RL) [14]. Building on recent results from single-agent RL and multiagent RL [15], [16] we propose RL schemes to solve these problems. Our algorithm may be considered as a first step towards developing efficient RL algorithms for dynamic leader-follower problems with intermittent incentive decisions. Further, we discuss the proposed RL approach in a special case of these problems where the followers play a repeated game. We discuss the conditions for the convergence of this algorithm and demonstrate its application in an illustrative example from the literature.

This paper is organized as follows. Section II discusses dynamic leader-follower problems in detail and Section III describes the LFSMDP framework. In Section IV, an RL approach to solving LFSMDP is discussed. Section V provides an illustration of the RL approach in the special case where the followers in the LFSMDP play a repeated game. We conclude the paper in Section VI noting some continuing work.

II. Dynamic Leader-Follower Problems

A dynamic leader-follower problem can be described as follows. At each time period $t = 1, 2, 3, ...$, first, the leader makes an incentive decision after observing the current state of the system and announces it to the followers. The followers after observing the incentive decision by the leader and also the system state, make their respective decisions concurrently and competitively. The system then transits to a new state stochastically under the actions of the leader and the followers, and under the influence of the environment; the leader and the followers realize their respective rewards for their actions and the process moves to the next period. Figure 1 illustrates this process where $u_1, u_2, u_3...$ are the leader's decisions and $S_1, S_2, S_3...$ are the environmental states at different periods. The followers's decisions are shown within circles to emphasize the fact that they are made only after leader's incentive decisions. Given an incentive decision by the leader, the followers decisions constitute a Nash game at each period. The objective of the leader and the followers is to maximize some cumulative measure of rewards received over the time periods.

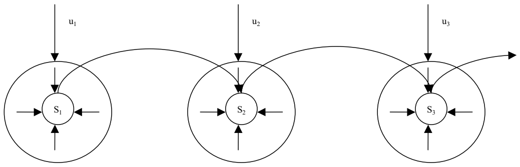

Fig. 1. Dynamic Leader-Follower Problem ($u_1, u_2, u_3, ...$ are leader's decisions and $S_1, S_2, S_3, ...$ are environmental states; arrows inside the big circle represent followers' decisions)

An important observation about the problem described above that plays a crucial role in the development of our MDP framework is that, the decision process described above can be considered as consisting of two coupled sequential decision processes- decision process of the leader and that of the followers. The leader's decision process is a single-agent decision process whereas the followers' sequential decision process is a stochastic game [17]. These two processes are coupled, as incentive decisions of the leader affect the followers' decisions and incentive decision of the leader is a function of followers' actions. An optimal solution of a dynamic leader follower problem then constitutes a *stationary* optimal incentive policy of the leader and the corresponding set of *stationary* equilibrium policies of the followers. This pair of policies then constitute a Stackelberg equilibrium [1] for a dynamic leader-follower problem. It may be noted that, an incentive policy of the leader in a dynamic leader-follower problem is a function of the environmental states also, whereas in the static problem, an incentive policy is just a function of the followers' actions.

In many incentive applications, the incentive decisions are taken at a lower frequency than the decisions of the agents. A large number of applications in regulation and control belong to this class of dynamic leader-follower problems. In this work we focus on these problems. Specifically, in these problems the leader makes an incentive decision only at certain episodic events, for example, in a market regulation problem, the market regulator announcing a new incentive when market price exceeds the price cap. The followers in these problems on the other hand make several rounds of decisions between two consecutive incentive decisions of the leader. Thus, in these problems, the leader's incentive decisions are interspersed with instances of the followers' game such that the time elapsed

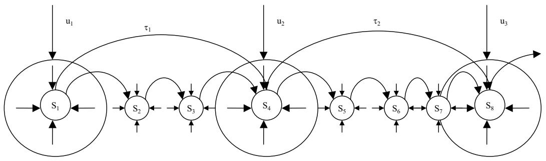

Fig. 2. Leader-Follower semi-Markov Decision Process (u_1, u_2, u_3... are leader's decisions and S_1, S_2, S_3... are environmental states; arrows inside the big circle represent followers' decisions)

between two consecutive incentive decisions is a random variable. The leader tries to design the optimal incentive policy that maximizes its long run expected payoff per period while the followers try to device strategies that maximize their individual long run expected payoffs per period under the announced incentive. In the next section we discuss an MDP framework for this class of leader-follower problems.

III. Leader-Follower Semi-Markov Decision Process

A dynamic leader-follower problems with intermittent incentive decisions is illustrated in Fig. 2 where $u_1, u_2, u_3, ...$ are leader's decisions, $S_1, S_2, S_3, ...$ are environmental states, and $\tau_1, \tau_2, ...$ are the time elapsed between consecutive incentive decisions.

As noted earlier, our MDP framework for this problem is based on the observation that it consists of two coupled sequential decision processes - that of the leader and of the followers, with the incentive constraint coupling the two processes. We note that with intermittent incentive decisions the leader's decision process above resembles a single-agent semi-Markov Decision Process (SMDP). In an SMDP, the agent makes decisions only at certain points in time while the state of the system changes at every time period. An SMDP consists of two processes, a *decision process* and a *natural process* that operate over the same state space. The decision process consists of only those periods where there is a decision whereas the natural process includes all the time periods. Similarly, a dynamic leader-follower problem with intermittent incentive decisions described above consists of two processes, the leader's decision process and the followers' decision process. A key difference here is that in place of the "natural process" of the SMDP, we have another decision process, the decision process of the followers which is a stochastic game. We call the decision process of the dynamic leader-follower problems with intermittent incentive decisions described above as a Leader-Follower semi-Markov Decision Process (LFSMDP). An LFSMDP may be described as a sequential decision process that consists of an SMDP and a stochastic game coupled by an incentive constraint.

Formally, an LFSMDP is a tuple (N, S, U, A, P, R, K, F) where N is the number of followers, S is the discrete state space, U is the set of incentive actions of the leader, A is the set of joint actions $(A_1 \times A_2 \times \ldots \times A_N)$ of the followers (where A_n is the set of actions available for follower n), P is the probability distribution function of state transitions, R is the set of immediate reward functions $r_n (n = 1, 2, \ldots, N)$ of the followers, K is the reward function of the leader and F is the probability distribution function of transition times at each decision epoch of the leader. The reward received by the leader between two successive decision epochs has two parts- a fixed reward $k(s, u)$, received on taking action u at the decision epoch where the state is s (this could also be the fixed cost of implementing the incentive decision), and the accumulated reward until the next decision epoch. In this formulation we assume that the leader's objective is social welfare and so the latter is the aggregate reward across all the followers until the next incentive announcement by the leader. Thus, the immediate reward of the leader for taking action u at state s may be written as

$$K(s, u) = k(s, u) + \sum_{n=1}^{N} \left[E\left(\sum_{t=1}^{\tau} r_n^t(\pi_n(u), \pi_{-n}(u)) \right) \right] \quad (1)$$

where $\pi_n(u)$ and $\pi_{-n}(u)$ are the policy of follower n and the policy of all agents except agent n respectively under the incentive u of the leader , $r_n^t(\pi_n(u), \pi_{-n}(u))$ is the immediate reward of follower n at time t from taking action according to policy $\pi_n(u)$ while other followers follow the policy $\pi_{-n}(u)$, and τ is the transition time to the next decision epoch. Let us denote $\pi^*(u)$ as the equilibrium policy of the followers under the incentive strategy u (we assume that this equilibrium is unique), and $K^*(s, u)$ as the equilibrium reward to the leader.

In this work, we consider an average reward formulation of the above LFSMDP. This means that the objective of the leader and the followers is to maximize their respective average rewards per time step.

The one-step average reward of the leader starting at state s and following an incentive policy ϕ can be written as

$$\rho^\phi(s) = \lim_{T \to \infty} \left[\frac{E^\phi(\sum_{t=1}^{T} K^*(s_t, u_t))}{E^\phi(\sum_{t=1}^{T} \tau_t)} \right] \quad (2)$$

where s_t, u_t, and τ_t are the state, incentive, and transition time respectively at decision epoch t and T is the number of decision epochs . It may be noted that under the assumption of an ergodic process, the one-step average reward $\rho^\phi(s)$ does not vary with the initial state s. The Bellman optimality equations for the leader is

$$\begin{aligned} V^*(s) = \max_u [K^*(s, u) - \rho\tau(s, u, \pi^*(u)) \\ + \sum_{\grave{s}} P^{u,\pi^*(u)}(s, \grave{s}) V^*(\grave{s})] \; \forall s \in S \quad (3) \end{aligned}$$

where$V^*(s)$ is the optimal expected relative value starting from state s, ρ is the optimal one-step average-reward, $\tau(s, u, \pi^*(u))$ is the expected transition time to next decision epoch on taking action u at state s, and $P^{u,\pi^*(u)}(s, \grave{s})$ is the probability of transition from state s to $\grave{s}$ under action u when the followers play the equilibrium strategy $\pi^*(u)$.

However, the optimal incentive policy should also satisfy the incentive compatibility condition which stipulates that the optimal stationary incentive policy of the leader should also result in an equilibrium of the followers' stochastic game in stationary policies of the followers. If we designate such an optimal incentive policy as $\phi^* : S \rightarrow U$, then it can be written as the the solution of the following set of equations.

$$\phi^*(s) = arg \max_u [K^*(s,u) - \rho\tau(s,u,\pi^*(u)) + \sum_{\grave{s}} P^{u,\pi^*(u)}(s,\grave{s})V^*(\grave{s})] \; \forall s \in S \quad (4)$$

The above set of equations represent the Stackelberg equilibrium condition for a dynamic leader-follower problem with intermittent incentive decisions. The solution of an LFSMDP involves determining the optimal stationary policies of the leader and the followers that constitute a Stackelberg equilibrium as above.

Proposition 1. *In leader-follower systems described above, there exists stationary policies for the leader and the followers satisfying the Stackelberg equilibrium condition of* (4).

Proof: We note that for a specific stationary strategy of the leader, the followers' stochastic game has an equilibrium in stationary strategies [17]. Now, if the leader selects the stationary strategy that maximizes its average reward, the corresponding equilibrium stationary strategies of the followers would satisfy the condition of (4).

IV. Solving LFSMDPs

The conventional methods of solving MDPs use iterative dynamic programming algorithms such as *value iteration* and *policy iteration* under the assumption that the transition probabilities and rewards are known a priori. However, in practice, reward information is hard to obtain and transition probabilities are computationally tedious to estimate even if complete information is available. These issues combined with the problem of large state spaces (curse of dimensionality) make exact solution of MDPs difficult in many applications. Hence approximate methods for solving MDPs are being investigated that can provide good solutions. Reinforcement learning (RL) is a machine learning technique used in the approximate solution of MDPs. In the RL approach, the agent interacts with a simulated model of the environment- taking actions and receiving rewards, and incrementally estimates the optimal value function of the MDP from the rewards it receives. During this process, it uses the current estimate of the value function to decide its actions. In this section we discuss an RL approach to solving LFSMDPs that draws from both single-agent RL [18], [14] and multiagent RL [15], [16], [19] research.

A. Reinforcement Learning Approach

As with our LFSMDP framework, the basis of our RL algorithm is the observation that the sequential decision process in a dynamic leader-follower problem consists of leader's SMDP and followers' stochastic game that are coupled by the incentive constraint. Based on this observation our algorithm consists of leader's learning scheme modeled as a single agent RL and followers' learning scheme modeled as a multiagent RL. A crucial point though is, the coupling of these two learning processes so that incentive constraint is satisfied.

Our algorithm (Algorithm 1) is based on the LFSMDP framework discussed above and builds on the Q-learning algorithms proposed in [20] and [21] for average reward SMDPs . The Algorithm 1 consists of a learning scheme for the leader and identical learning schemes for the followers. As we are trying to solve an average reward LFSMDP, the Q-values in this algorithm are written in terms of *relative values*. Relative value of taking an action at a state is the difference between the immediate reward received and the estimated average reward. The learning schemes for the leader and the followers consist of estimating both the Q-values and the average rewards. The λ and β are the learning rates used for these updates. As in [20], to ensure convergence, for each agent the values of these learning rates are set such that $\lim_{t\rightarrow\infty} \frac{\beta_t}{\lambda_t} = 0$. While follower Q-values are updated at every period, leader's Q-values are updated only in those periods where there is an incentive decision. The expolre/exploit action selection for the leader and the followers is implemented using the Boltzmann action selection scheme [14].

Algorithm 1 Q-Learning Algorithm for LFSMDP

```
1. Leader selects an incentive $u$ at the state $s$.
2. Followers play a game
   Each follower $n$ selects an action $a^n$ according to
   explore/exploit action selection scheme
   Next state is $\grave{s}$ and each follower $n$ receives a reward $r^n$
   Each follower updates its $Q$-value, $Q^n$ and
   average reward, $\rho^n$ using the following update schemes
    $Q^n(u,s,a^n) \leftarrow Q^n(u,s,a^n) + \lambda^n[r^n - \rho^n - Q^n(u,s,a^n)$
   $+ \max_b Q^n(u,\grave{s},b)]$
    $\rho^n \leftarrow \rho^n + \beta^n(r^n - \rho^n)$
3. Step 2 is repeated until the next decision epoch,
   which happens after $\tau$ periods
4. Leader receives the aggregate reward $r^l$ since the last
   incentive decision and updates its $Q$-value, $Q^l$ and
   the average reward, $\rho^l$ using the following update schemes
    $Q^l(u,s) \leftarrow Q^l(u,s) + \lambda^l[r^l - \tau\rho^l - Q^l(u,s)$
   $+ \max_w Q^l(w,\grave{s})]$
    $\rho^l \leftarrow \rho^l + \beta^l(\frac{r^l}{\tau} - \rho^n)$
5. If the termination criterion is not met
   The leader selects an incentive $u$ using explore/exploit
   action selection scheme
   Return to Step 2
```

In this algorithm, the leader's learning scheme is a single-agent average reward Q-learning scheme whereas the followers' learning scheme is a multiagent Q-learning scheme for average rewards. Thus, while it is easy to show the convergence of the leader's learning process to optimal incentives, it is difficult to show the convergence of the followers learning scheme to a Nash equilibrium [19]. Consequently, the convergence of this algorithm to a Stackelberg equilibrium is hard to prove. Another difficulty with this algorithm is that being a Q-learning algorithm, the followers' learning scheme can learn only pure strategies. In the next section, we adapt this algorithm for problems where the followers play a repeated game. Further, based on some recent results in multiagent RL, we make modifications to the present algorithm so that the followers' learning scheme can learn mixed strategies.

B. The Repeated Game Case

In this section we consider leader-follower problems with intermittent incentive decisions where the followers play a repeated game. We also assume that the incentive decisions are made at every m periods. Thus, in these problems, after each incentive decision by the leader, the followers repeatedly play the game for m periods. Many applications in coordination and regulation can be modeled by this repeated game version.

Adapting Algorithm 1 to the case of repeated games results in major simplification of the algorithm. This is because, by definition, repeated games are stochastic games with a single state. A major change is in the definition of Q-values. With a single state, the relative value definition of Q-values reduce to the expected reward per time step. Thus, both the leader and the followers in the new algorithm try to learn the expected reward per time step. The leader's learning algorithm then resembles the single agent Q-learning except for the fact that the immediate reward for the leader is the reward accrued over m periods.

The followers' learning scheme in the modified algorithm is based on a Q-learning algorithm for repeated games proposed recently by Leslie and Collins [16]. Their work makes two major contributions. First, their algorithm uses a player dependent learning rate for the update of Q-values of individual agents. This scheme is based on a result from the stochastic approximation theory by Borkar [22]. Borkar has shown that, in the case of two coupled approximation schemes, convergence is achieved if the processes use different step sizes. This is because different step sizes make one of the processes to be faster than the other and thereby the faster process sees the slower one as stationary while the slower one sees the faster one as calibrated to the current value of the slow process. This results in the eventual convergence of the two processes. It may be noted that the Q-learning update scheme is a stochastic approximation process and in the case of multiagent Q-learning, they become N coupled stochastic approximation schemes. Leslie and Collins [23], and [16] extended Borkar's result to the case of N-player systems with the following important assumption.

Assumption 1. *[16] Let the agents be indexed using n such that agents with higher indices are fast learners than those with lower indices. For any n, if the values of slow learning agents $(1, ..., n-1)$ were fixed, then the values of the fast learning agents $(n, ..., N)$ would converge to a unique equilibrium determined by the best response functions of the fast learning agents and the fixed value functions of the slow learning agents.*

With assumption 1, Leslie and Collins [16] show that a learning rate scheme of $\lim_{t\to\infty} \frac{\lambda_t^n}{\lambda_t^{n+1}} = 0$ ensures convergence.

The second important contribution in Leslie and Collins' algorithm is the use of a smooth best response (SBR) [24] scheme for action selection by the agents. This scheme enables the agents to learn mixed strategies. Recall that Q-learning, being an optimum seeking algorithm, can learn only pure strategies. The use of SBR action selection in Leslie collins' algorithm addresses this issue. A SBR scheme maintains a positive probability for every action in an agent's action set to get selected. One way to implement an SBR action selection scheme is to use a Boltzmann action selection scheme with the temperature parameter held constant at a very small value. In Botzmann action selection scheme, the probability of selecting action a when the state is s is $\frac{e^{Q(s,a)/T}}{\sum_{a'} e^{Q(s,a)/T}}$, where $Q(s,a)$ is the current estimate of Q-value for state-action pair (s,a) and T a "temperature" parameter that controls the degree of randomness in action selection. In single agent RL this scheme is used to implement the explore/exploit action selection. There, the parameter T is decreased as the agent learning converges, so that the probability of selecting the optimal action approaches 1. When T is held constant, every action in the action set will always have a positive probability (though small) of getting selected. Leslie and Collins [16] show that, with smooth best responses, an agent's strategy converges toward a Nash distribution which is an approximation of the mixed strategy Nash equilibrium of the game.

The followers' learning scheme in our algorithm incorporates the two features of Leslie and Collins's algorithm discussed above. Hence, under Assumption 1, the followers' learning scheme in our algorithm can be shown to converge to the Nash equilibrium. Besides, in our algorithm, the leader's learning scheme is a single agent Q-learning which is known to converge to the optimal action, which is the optimal incentive in our case. Thus, jointly, these two learning processes in our algorithm converge to the required Stackelberg equilibrium of the dynamic leader-follower problem under the condition that Assumption 1 is satisfied.

It may be noted that, in one-leader-two-follower problems, for a given incentive strategy of the leader, the followers' subgame is a two-player game for which Assumption 1 is trivially satisfied. Thus, in one-leader-two-follower problems the new algorithm will converge to optimal incentives and responses.

Algorithm 2 is the proposed RL algorithm for the repeated game case.

The parameter m decides the number of times the followers play the game before the leader updates its Q value. Note that the update of the leader and the followers happen at

Algorithm 2 Q-learning for Repeated LFSMDP

```
1. Leader starts with an incentive $u$.
2. Followers play a game
   Each follower $n$ selects an action $a^n$ according to
   a smooth best response scheme
   Each follower $n$ receives a reward $r^n$ and updates its
   $Q$ value $Q^n$ using the following update scheme
    $Q^n(u, a^n) \leftarrow Q^n(u, a^n) + \lambda^n \left[r^n - Q^n(u, a^n)\right]$
3. Step 2 is repeated $m$ times
4. Leader receives the aggregate reward $r^l$ since the last
   incentive decision and updates its $Q$ value $Q^l$ using the
   following update scheme
    $Q^l(u) \leftarrow Q^l(u) + \lambda^l \left[\frac{r^l}{m} - Q^l(u)\right]$
5. If the termination criterion is not met
   The leader selects an incentive $u$ using explore/exploit
   action selection scheme
   Return to Step 2
```

different time scales. Follower Q-values are updated in every time period whereas for the leader, the update happens only once in m periods. The followers' maintain separate Q-values for each incentive decision of the leader.

The player dependent learning rates for the followers' learning is implemented using a scheme proposed in [16]. According to this scheme, the learning rate for follower n at time step t is set as $\lambda_t^n = (t+C)^{-\theta^n}$ where $\theta^n \in (0.5, 1]$ and C a constant. By selecting θ^n differently for each follower, the required sequence of learning rates is assured.

Each follower in our algorithm employs a SBR action selection scheme that uses the Boltzmann function with the "temperature" parameter T set at a very small value. As discussed earlier, this enables the followers to learn mixed strategies. We also use the Boltzmann function for explore/exploit action selection of the leader, but with a decaying T as in single agent Q-learning.

V. Illustrative Example

This section describes a well studied incentive problem from the literature that we use to evaluate our proposed learning algorithm. This problem, introduced by Salman and Cruz [25] is a one-leader-two-follower incentive Stackelberg game with linear incentive structure. In this economic model of duopoly markets, two firms produce an identical product and compete for the same market. Their strategic decision variables are the production quantities for the current period (Cournot competition). The market price of the product is determined by the total quantity produced by the two firms and the government policy. The government influences market price by controlling effective income of potential buyers of the commodity through a subsidy/tax. The objective of the government is to induce the two firms to cooperate and maximize the overall profit. Salman and Cruz showed that there exists an optimal incentive strategy for the government when perfect information about firms' strategies is available to the government.

Let q_1 and q_2 be the quantities produced by the two firms. The total output then is $Q = q_1 + q_2$. Price p depends on the quantity Q and the government's decision u about subsidy/tax:

$$p = a_o - a_1 q_1 - a_2 q_2 + a_3 u$$

where a_i, i = 0, ..., 3 are positive constants.The payoff for the firms is

$$r_i = p q_i - \tfrac{1}{2} c_i q_i^2 \quad i = 1, 2$$

where c_i represents a cost parameter for the firms. The government's payoff is given by

$$R = r_1 + r_2 - \tfrac{1}{2} c_0 u^2$$

where c_0 is a parameter representing government's coordination cost. In this model, both the government and the firms have a quadratic cost structure.

The firms have to decide their respective production quantities q_1 and q_2 while the problem faced by the government is to select u so that coordination is achieved.

A. Simulation

In this section, we use a simulation model of the duopoly market described in the previous section to illustrate the working of the proposed learning algorithm. For comparison, we set the parameter values of the model as those used by Salman and Cruz in their paper: $c_0 = 10, c_1 = c_2 = 10, a_0 = 5$, and $a_1 = a_2 = a_3 = 0.5$

In the simulation model, the agents use the learning algorithm of previous section to learn optimal actions. The government tries to learn the optimal incentive while the firms try to learn the optimal production quantities. The simulation starts with the government setting an incentive. On knowing the government's decision, the firms decide about their production quantities using explore/exploit action selection. The resulting market price and the profit for each firm are determined according to the equations given in the previous section. The agents representing the firms update their Q-values and the time is advanced to the next period. The firms continue this game for m periods, at the end of which the agent representing the government receives the aggregate reward from these m games and updates its Q-value. This agent then sets a new incentive based on the SBR action selection scheme described earlier.

For the follower agents, the smooth best response (SBR) was implemented using a Boltzmann function with the value of T set as 0.1. For the leader agent, to implement explore/exploit action selection, the T parameter in the Boltzmann function was varied using the scheme $T = \varepsilon^t T_{max}$ where t is the number of periods and $0 < \varepsilon < 1$. We used a T_{max} of 6 and ε of 0.99995. No exploration was performed when T is below 1. The learning rates for the agents were implemented according to $\lambda_t^n = (t+C)^{-\theta^n}$ with the constant C set to a value of 100 and the parameter θ set to $0.9, 0.8$ and 0.7 for the leader and

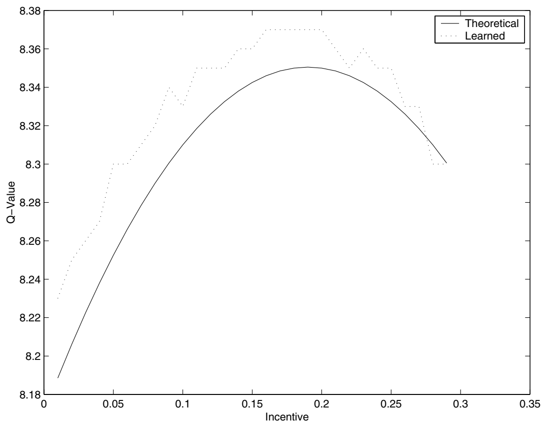

Fig. 3. The learned Q-values of the leader after 50000 steps approximate the theoretical values.

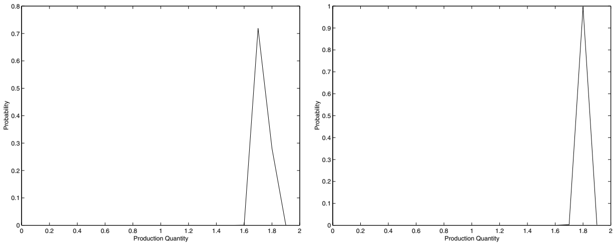

Fig. 4. The SBR probabilities of Firm1 (left-hand side) and Firm2 (right-hand side) after 50000 simulation steps is close to the pure strategy of producing a quantity of approximately 1.7

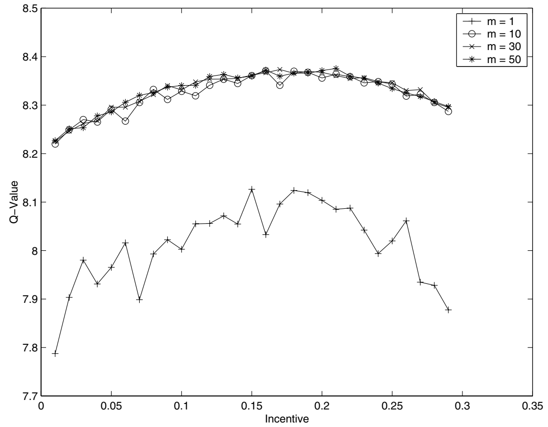

Fig. 5. The Q-values learned by the agent representing the government for different values of m.

the followers respectively. The parameter m in the algorithm that decides the frequency of update for the leader was set to a value of 30.

According to Salman and Cruz, the optimal incentive for the government in this numerical example is 0.17. Also, under this optimal incentive, both firms have an optimal production quantity of 1.7. The two firms have identical production quantities as their cost structures are identical. Fig. 3 shows the rewards learned by the agent representing the government for different incentive values after 50000 simulation steps (all the simulation results presented in this paper are averages over 10 runs). For comparison, this is shown along with the theoretical values. It can be seen that the learned reward function closely follows the theoretical one with a peak near an incentive of 0.17. In this example, the equilibrium behavior of the firms under the optimal incentive consists of pure strategies. Hence, one would expect the learned smooth best response probabilities to approximate a pure strategy. The learned smooth best response probability distributions of the two firms shown in Fig. 4 are seen to closely approximate the pure strategies of producing a quantity of 1.7 each. The results demonstrate that under the proposed learning algorithm, the agents are able to learn the optimal behavior very closely- the agent representing the government learning the optimal incentive and the agents representing the firms learning the optimal production quantities.

In our algorithm, the parameter m decides how often the leader updates its Q-value. This parameter primarily affects the performance of leader's learning, as higher values of m provide better estimates of the leader's immediate reward by averaging over a larger number of periods of the followers' learning. Figure 5 shows the effect of m on the learning performance of the leader. The graph plots the learned Q-values for the incentive levels, when using different m values. With $m = 1$, the leader updates in every period leading to poor learning. As m is made larger, the learning performance improves, but with diminishing marginal improvement. This is encouraging as it shows that large m values after all may not be necessary for efficient learning. It also points to the possibility of existence of an optimal update frequency for the leader.

Since the algorithm consists of two coupled learning processes, it is interesting to see how well these two learning processes work together. The leader's learning process depends on the noisy learning process of the followers for its reward estimates. Figure 6 shows the evolution of leader's Q-values as the simulation progresses. These results show that the leader is able to learn the relative ranking of the incentives at a very early stage of the simulation. This relative ranking of the incentives can help in providing an approximately optimal incentive decision. This shows that the proposed algorithm can provide good approximate solutions quickly.

To test the scaling of the algorithm to larger problems we examined an extended case of the illustrative problem involving 4 firms (oligopoly) keeping the parameters the same as in earlier experiments. The Q-values and the SBR probabilities obtained for this setting are shown in Fig. 7. As in the duopoly case, the plot of Q-values of the leader shows a single peak, indicating convergence to an optimal incentive. The plot of SBR probabilities of the followers shows that all the followers converge to identical pure strategies, which is expected as all the followers have identical cost parameters.

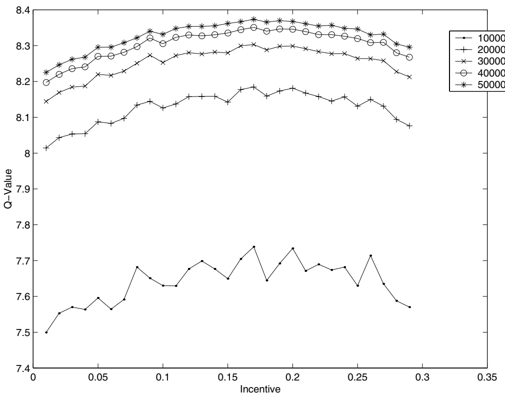

Fig. 6. The evolution of Q-values of the leader.

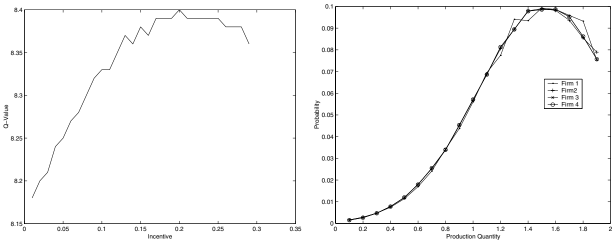

Fig. 7. The Q-values of the leader (left-hand side) and SBR probabilities of the followers(right-hand side) in the oligopoly example

VI. Conclusion

In this paper we discussed a dynamic version of leader-follower problems where the leader makes incentive decisions intermittently. We proposed an MDP framework called LFS-MDP for such problems. We showed how this framework helps in describing these problems and specifying their solution and also provides a basis for the approximate solution of these problems using RL. Based on the proposed LFSMDP framework an RL approach for their solution is described. Our algorithm is a first cut at developing efficient RL algorithms for this class of problems. In continuing work we are investigating the convergence properties of the proposed RL algorithm and its applications in varied contexts.

References

[1] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. London: Academic Press, 1995.

[2] H. Shen and T. Basar, "Incentive-based pricing for network games with complete and incomplete information," *Annals of Dynamic Games*, vol. 8, no. 1, 2006, to appear.

[3] A. Keyhani, "Leader-follower framework for control of energy services," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 837–841, 2003.

[4] Y.-M. Li, Y. Tan, and P. De, "Pricing peer-to-peer networks: Content provision and search intermediary," in *Proceedings of the Fourteenth Annual Workshop on Information Technologies and Systems*, A. Dutta and P. Goes, Eds., 2004, pp. 194–199.

[5] F. Bernstein, F. Chen, and A. Federgruen, "Coordinating supply chains with simple pricing schemes: The role of vendor managed inventories," *Working paper, The Fuqua School of Business, Duke University, Durham, NC and Graduate School of Business, Columbia University, New York.*, August 2003.

[6] D. Garg and Y. Narahari, "Design of incentive compatible mechanisms for stackelberg problems," in *Proceedings of the 1st Workshop on Internet and Network Economics, WINE 2005, December 15-17,2005, Hong Kong*. Springer Verlag LNCS series 3828, 2005, pp. 718–727.

[7] H. Ehtamo, M. Kitti, and P. R. Hämäläinen, "Recent studies on incentive design problems in game theory and management science," in *Optimal Control and Differential Games, Essays in Honor of Steffen Jorgensen*, G. Zaccour, Ed., 2002, pp. 121–134.

[8] R. Radner, "Repeated principal-agent games with discounting," *Econometrica*, vol. 53, no. 5, pp. 1173–1198, 1985.

[9] E. Kalai and J. O. Ledyard, "Repeated implementation," *Journal of Economic Theory*, vol. 83, pp. 308–317, 1998.

[10] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley and Sons, Inc, 1994.

[11] H. S. Chang, P. Fard, S. I. Marcus, and M. Shayman, "Multi-time scale markov decision processes," *IEEE Transactions on Automatic Control*, vol. 48, pp. 976–987, 2003.

[12] E. Plambeck and S. Zenios, "Performance-based incentives in a dynamic principal-agent model," *Manufacturing and Service Operations Management*, vol. 2, pp. 240–263, 2000.

[13] V. Saksena and J. Cruz, "Optimal and near-optimal incentive strategies in the hierarchical control of markov chains," *Automatica*, vol. 21, no. 2, pp. 181–191, 1985.

[14] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

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

[16] D. S. Leslie and E. J. Collins, "Individual q-learning in normal form games," *submitted to SIAM J. of Control and Optimization*, 2004.

[17] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. New York: Springer-Verlag, 1997.

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[19] J. Hu and M. P. Wellman, "Nash q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, 2003.

[20] A. Gosavi, "Reinforcement learning for long-run average cost," *European Journal of Operational Research*, vol. 155, pp. 654–674, 2004.

[21] T. K. Das, A. Gosavi, S. Mahadevan, and N. Marchalleck, "Solving semi-markov decision problems using average reward reinforcement learning," *Management Science*, vol. 45, no. 4, pp. 560–574, 1999.

[22] V. S. Borkar, "Stochastic approximation with two timescales," *System Control Letters*, vol. 29, pp. 291–294, 1997.

[23] D. S. Leslie and E. J. Collins, "Convergent multiple-timescales reinforcement learning algorithms in normal form games," *Annals of Applied Probability*, vol. 13, no. 4, pp. 1231–1251, 2003.

[24] D. Fudenberg and D. K. Levin, *The Theory of Learning in Games*. Cambridge, MA: MIT Press, 1998.

[25] M. A. Salman and J. B. Cruz, "An incentive model of duopoly with government coordination," *Automatica*, vol. 17, no. 6, pp. 821–829, 1981.