PAPER

Model-Based Reinforcement Learning in Multiagent Systems with Sequential Action Selection

Ali AKRAMIZADEH[†a)], **Ahmad AFSHAR**[†], ***Nonmembers***, **Mohammad Bagher MENHAJ**[†], ***Member***, ***and*** **Samira JAFARI**[†], ***Nonmember***

SUMMARY Model-based reinforcement learning uses the gathered information, during each experience, more efficiently than model-free reinforcement learning. This is especially interesting in multiagent systems, since a large number of experiences are necessary to achieve a good performance. In this paper, model-based reinforcement learning is developed for a group of self-interested agents with sequential action selection based on traditional prioritized sweeping. Every single situation of decision making in this learning process, called extensive Markov game, is modeled as n-person general-sum extensive form game with perfect information. A modified version of backward induction is proposed for action selection, which adjusts the tradeoff between selecting subgame perfect equilibrium points, as the optimal joint actions, and learning new joint actions. The algorithm is proved to be convergent and discussed based on the new results on the convergence of the traditional prioritized sweeping.

key words: *multiagent systems, Markov games, model-based reinforcement learning, extensive form game*

1. Introduction

Various algorithms have been extended from single agent learning to multiagent learning such as evolutionary learning [1], coevolutionary learning [2], [3], and the combination of game theory and reinforcement learning (RL), among which the last one seems to be the most promising solution [4]. RL does not need an explicit model of the environment [5], which is a key benefit in most of the real-life applications.

Initially, single agent reinforcement learning (SRL), [6], was used in multiagent systems without much modification. Such approaches treated other agents in the system as a part of the environment, ignoring the differences between an active agent and the passive environment [7]. These methods failed to converge in some difficult coordination problems. It was experimentally shown that cooperative learning through sharing sensation, episodes and learned policies outperforms the independent learning [8].

Contribution of game theory to multiagent reinforcement learning (MRL) was first implied in MinMax-Q [9]. The algorithm can be used for two successive fully competitive agents. Because of the simplicity, MRL was later developed for agents with simultaneous action selection. Nash-Q was proposed for general-sum Markov games, where agents decide on their actions to reach the presumed unique Nash equilibrium point (NEP) of the current game [10]. It was proved that the algorithm gradually converges to the optimal policy. Unfortunately, its applicability is restricted due to some drawbacks [11], especially with a large number of agents. Littman proposed another method in which some of the presumed limitations in Nash-Q were relaxed by adding some additional (a priori) information about the roles of the agents in the system [12]. Some other modifications were also suggested, which can be reviewed in survey papers [4], [13]–[15].

In some real-life applications, simultaneous decision making needs a number of cumbersome constraints, especially on equal resources, authorities, requirements, and so on. As it was mentioned, sequential action selection was primarily introduced in MinMax-Q, which was later partly modified in Asymmetric-Q [16]. The process of learning in asymmetric-Q was divided into a sequence of 2-person perfect information zero-sum extensive form games, referred to as Stackelberg's duopoly games. In single state repeated games, more general forms of extensive form games were investigated. In [34], value of each action was an index equal to its past average payoff. Proportional to them, actions are selected randomly. Thus, the values converged toward the subgame perfect equilibrium values. More precisely, the sub-sequences of index-maximizing actions converge toward the actual subgame perfect equilibrium, but random actions continue to be played with a fixed positive probability. The results are only applicable to single state repeated games with strictly positive payoffs. Two MRL algorithms for complete information repeated extensive form games with unique subgame perfect equilibrium were proposed in [38]. Their first algorithm, "Multiagent Q-Learning", was somewhat related to the one in [34]. The second of their algorithms, "Multiagent Learning Automata", used the reward obtained at the end of a game episode to reinforce the strategy (by properly updating its probability of being chosen) followed at a node of the game-action pairs instead of reinforcing the values of node. Both of these algorithms are proved to converge to the subgame perfect equilibrium of the game. A more sophisticated rule was considered in [37]: the valuation rule associates to each action a stochastic index equal either to its past payoffs (with a probability proportional to their frequency) or to some random values (with a probability decreasing with the number of occurrences of that action); the decision rule states

Manuscript received May 10, 2010.
Manuscript revised September 6, 2010.

[†]The authors are with the Dept. EE., Computational Intelligence and Large Scale System Lab., Amir Kabir University of Technology, Tehran, Iran.
a) E-mail: akramizadeh@aut.ac.ir

DOI: 10.1587/transinf.E94.D.255

that agent chooses the maximizing action. Here, the process converges (for even a larger class of rules containing the preceding one) toward the subgame perfect equilibrium actions, but not toward the equilibrium values, even if they are recovered by taking the expected value of the random variable. Another similar work was done in [35]. Based on their premier work on normal form games [36], they proposed "cumulative proportional reinforcement" which was to randomly chose actions proportional to a kind of reinforcement signal. Their proposed method was only applicable to single state repeated game with positive payoffs. In their methods, neither delayed reward nor nonstationarity in learning was addressed. In other words, their proposed method cannot be used in dynamic tasks.

Previously, we introduced model-free reinforcement learning for extensive Markov games (EMG). EMG is a sequence of n-person general-sum extensive form games with perfect information [17]. The algorithm presents a sensible convergence property regarding the fact that extensive form games with perfect information have pure strategy Nash equilibrium points [18]. Even though the algorithm gradually convergence to the optimal policy without complex computations, it inefficiently uses the gathered information and needs a great deal of experiences to achieve a good performance. This is the same drawback as in traditional SRL, which was comprehensively addressed in model-based RL (MBRL). MBRL is also interesting in applications, where computations are considered to be cheap while real world experiences are costly [6]. Simultaneous planning and learning is the key feature in MBRL. Planning refers to the implementation of estimated models of the environment to compute state-value functions and thereby optimizing or improving policies. Actually, it makes more than a single update per experience. Henceforth, information is used more efficiently than the traditional SRL [19]. After each real-world experience $\left\langle s^k, a^k, s^{k+1}, r\left(s^k, a^k\right)\right\rangle$, Q-functions as well as the transition function and the reward function are updated. Dynamic programming could be immediately applied for the estimated environment, but online dynamic programming, which updates the complete value function with value iteration after each experience, tends to be computationally very expensive. To speed up the algorithm, Dyna-Q only simulates k samples randomly and learns for them [19]. Although Dyna-Q offers a great improvement on traditional model-free Q-learning, it suffers from being relatively undirected, since it selects its simulated observations randomly. Particularly, it is of no use when the goal is reached or when the agent is stuck in a dead end. It continues to update random state-action pairs, rather than concentrating on "interesting" parts of the state space. The term "interesting" was dealt with in prioritized sweeping (PS) [20] and Queue-Dyna [21], which are two independently developed similar techniques. Some other variants of PS were also introduced such as generalized PS [25] and general prioritized solver [26].

Application of MBRL in multiagent systems was initially examined by a simple modification on PS [22]. Their method can be used for a special class of competitive games with two agents, which can be regarded as an extension to MinMax-Q. Instantiating information about the dynamic objects in the model of the environment and re-planning based on MBRL, whenever this information is changed, was proposed in [23]. Their approach can be categorized as a modified version of SRL, since learning agent did not bear in mind explicitly the other agents during action selection and updating value functions. This especially fails when the other agents are also learning.

In this paper, we develop traditional PS based on game theoretic solvers, subgame perfect equilibrium points (SPE), for n-person general-sum multiagent systems with sequential action selection based on our previous papers [17], [24], [33]. The existing learning process, called EMG, is considered as a set of successive extensive form games with perfect information. Each learning agent is able to observe other agents' actions, and model their preferences through Q-functions. A modified version of backward induction is introduced for action selection based on Boltzmann distribution, which controls the tradeoff between SPE, as the optimal joint action, and learning new joint actions. The pseudo code of the algorithm is proposed that can be easily used for realization. Finally, the proposed algorithm, which is based on value iteration with planning, is proved to be convergent.

The reminder of the paper is organized as follows: in Sect. 2, some preliminary concepts regarding RL and game theory are reviewed in accordance with RL terminologies. In Sect. 3, MBRL is stated for EMGs. Section 4 analytically discusses the algorithm and its convergence proof. Some concluding remarks and future works are given in Sect. 5.

2. Preliminary Concepts

In the following, we briefly review fundamental issues, which have been employed in this paper with the terminologies adopted from the established frameworks in RL.

2.1 Model-Based RL

RL can be used to learn an agent by letting it interact with its environment and learn from the obtained rewards by trial and error. The environment is typically formulated as a finite-state Markov Decision Process (MDP). MDP is a mathematical framework for modeling action selection in situations where outcomes are partly random and partly under the control of an agent.

Definition 1: A Markov Decision Process is a tuple (S, A, R, P), where:

- S is the set of states,
- A is the set of admissible actions,
- $R = \{R | R : S \times A \to \Re\}$ is the reward function,
- $P : S \times A \to \Delta(S)$ is the state transition function, where $\Delta(S)$ is the set of probability distributions over the set S.

Agent's objective is to learn a Markov policy, a mapping from states to probabilities of taking admissible actions $\pi : S \times A \rightarrow [0, 1]$, that maximizes the expected discounted future rewards from state s, called state-value function:

$$\begin{aligned} V^{\pi}(s) &= E^{\pi}\left\{r^{k+1} + \gamma r^{k+2} + \gamma^2 r^{k+3} + \cdots | s^k = s\right\} \\ &= E^{\pi}\left\{r^{k+1} + \gamma V^{\pi}\left(s^{k+1}\right) | s^k = s\right\} \\ &= \sum_{a \in A} \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V^{\pi}(s')\right] \end{aligned}$$

where for all $s, s' \in S,\ a \in A$

r^k	Instantaneous reward,
$V^{\pi}(s)$	Value of state s under policy π
$\pi(s,a)$	Probability of taking action $a \in A$ in state s
$\gamma \in [0,1]$	Discount factor
$P^a_{ss'}$	State transition function.
$R^a_{ss'}$	Reward function

The optimal state-value function gives the value of each state under the optimal policy:

$$\begin{aligned} V^*(s) &= \max_{\pi} V^{\pi}(s) \\ &= \max_{a \in A} \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V^*(s')\right] \end{aligned}$$

The value of taking action a in state s under policy π, denoted by $Q^{\pi}(s,a)$, is the expected discounted future rewards starting in s, taking a, and henceforth following π:

$$\begin{aligned} Q^{\pi}(s,a) &= E\left\{r^{k+1} + \gamma r^{k+2} + \gamma^2 r^{k+3} + \cdots | s^k = s, \pi(s) = a\right\} \\ &= \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma \sum_{a'} \pi(s',a') Q^{\pi}(s',a')\right] \end{aligned}$$

The optimal action-value function is,

$$\begin{aligned} Q^*(s,a) &= \max_{\pi} Q^{\pi}(s,a) \\ &= \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma \max_{\pi} Q^*(s',a')\right] \end{aligned}$$

In RL, generally, there is no model containing the transition and the reward function. Therefore, these function as well as the policy have to be learned from the experiences during the interaction with the environment. Although offline dynamic programming, which recomputes the policy after a complete trial, would be more efficient, online updating is much better for efficient exploration and is especially needed in dynamic environment. To speed up online dynamic programming in complex environments, some sort of efficient update-step management should be performed [23]. This can be done by PS which prioritize updating Q-values of different state-action pairs according to their relative Bellman errors. It benefits from higher performance with respect to traditional SRL and Dyna-Q at the expense of additional information [6], [19]. If the value of state s' has changed by amount Δ, then the immediate predecessors of s' for which there exists an action a such that $\hat{P}(s,a,s') \neq 0$ promotes its priority to $\Delta \cdot \hat{P}(s,a,s')$, unless its priority value already exceeds the threshold. The global behavior of the algorithm is that when a state transition succeed in surprising reward (for example, the agent come across a goal state), then much computation is directed to propagate this new information back to relevant predecessor states. On the other hand, when the real world transition is dull (the actual result is similar to the predicted result), then computation are direct to the more deserving parts of the state space.

2.2 Game Theory

Game theory initially was used for reasoning in economics, which later has been widely utilized in social, political, and behavioral phenomena. It provides the necessary tools to model interactive situations in which self interested agents select their actions to gain more according to their preferences and a set of game rules. In this paper, we only focus on games with perfect and complete information.

Definition 2: An extensive form game with perfect information is a tuple $g = (X, \Sigma, f, Q)$, where [27]:

- $X = \{x_1, x_2, \ldots, x_N\}$ is the set of agents,
- $\Sigma = \{\sigma | \sigma \in \langle A_1, A_2, \ldots, A_N \rangle\}$ is the set of joint actions, where A_i is the set of admissible actions for agent i.
- $f(\hat{\sigma}_i)$ is the agent function that assigns an agent to every subsequence of actions $\hat{\sigma}_i$ in order to determine which agent has to select its action after $\hat{\sigma}_i$[†],
- $Q = \{Q_i | Q_i : \Sigma \rightarrow \Re\}$ assigns the values of joint actions for each agent, referred to as game preferences.

Greedy action selection is optimal in single agent learning. In multiagent cases, however, the optimal joint actions have to be elaborated in the form of NEPs in which all of the agents are satisfied and none of them volunteer to select another action.

Definition 3: A strategy profile σ^* is a NEP if no unilateral deviation in strategy by any single agent is profitable for that agent [27], that is:

$$Q_i(\sigma^*) \geq Q_i(a_i, \sigma^*_{-i}) \quad \forall i, a_i \in A_i$$

where σ_{-i} is a strategy profile of all agents except for agent i.

Ignoring sequential action selection in extensive form games may bring about some equilibrium points that are not robust in steady state [27]. In order to properly define equilibrium points in extensive form games, subgame must be defined. A subgame is any part of a game that can be analyzed as a game itself.

Definition 4: Let g be an extensive form game with perfect information. For any subsequence of actions $\hat{\sigma}_{i-1}$, the subgame ${}^{\hat{\sigma}_{i-1}}\hat{g}_i$ is the following extensive form game [27].

[†]Any sequence $\hat{\sigma}_i = \langle a_1, a_2, \ldots, a_i \rangle$ with respect to joint action $\sigma = \langle a_1, a_2, \ldots, a_N \rangle$ is called a subsequence of actions, where $i < N$.

- Agents, the agents in g.
- Joint actions, the set of all sequences of actions σ' such that $(\hat{\sigma}_{i-1}, \sigma') \in \Sigma$ is a joint action of g.
- Agent function, to each proper subsequence $\bar{\sigma}$ of a subgame and agent is assigned according to $f(\hat{\sigma}_{i-1}, \bar{\sigma})$.
- Preferences, each agent prefers σ' to σ'' if and only if $(\hat{\sigma}_{i-1}, \sigma')$ is preferred to $(\hat{\sigma}_{i-1}, \sigma'')$ in g.

Definition 5: A strategy profile σ^* is an SPE in extensive form game g if it is an NEP for every subgame of g [27].

Markov games [23] are the generalization of both static games and (fully observable) MDPs. We introduced EMG, which can be regarded as an extension to Markov games in which each game state is in extensive form with perfect information [17].

Definition 6: EMG is a tuple $\Psi = \langle G, X, \Sigma, P, R \rangle$, where:

- G is the set of extensive form games with perfect information,
- $X = \{x_1, x_2, \ldots, x_N\}$ is the set of agents with fixed priorities in action selection,
- $\Sigma = \{\sigma | \sigma \in \langle A_1, A_2, \ldots, A_N \rangle\}$ is the set of admissible joint actions, where A_i is the set of admissible actions for agent i,
- $P : G \times \Sigma \to \Delta(G)$ is the game transition function, where $\Delta(G)$ is the set of probability distributions over G.
- $R = \{R_i | R_i : G \times \Sigma \to \Re\}$ is the reward function.

Regarding the aforementioned definitions, we can generalize the notion of NEPs in EMGs as follows.

Definition 7: A policy profile π^* is an NEP in EMG Ψ if no unilateral deviation in policy by any single agent is profitable for that agent in any game $g \in G$, that is:

$$Q_i(g, \pi^*(g)) \geq Q_i(g, \pi_i(g), \pi^*_{-i}(g)) \quad \forall g, i, a_i \in A_i$$

where $Q_i(g, \pi(g))$ is the preferences of agent i in game g with joint policy $\pi(g)$, and $\pi_{-i}(g)$ is a policy profile of all agents except for agent i.

In perfect information games, each agent knows all about the preferences of the other agents. We refer to the set of preferences of all agents as *Extended Q-values*,

$$\bar{Q}_i = [Q_1, \ldots, Q_i, \ldots, Q_N]$$

3. Model-Based Multiagent Reinforcement Learning

3.1 Learning in EMGs

Learning in multiagent systems is the process in which less than fully rational agents look for optimality over time [28]. Consider a general EMG process as depicted in Fig. 1, where there is a sequence of extensive form games $g^k \in G$, $k = 1, \ldots, K$. Every single game can be presented as a finite tree with a set of nodes as agents $x_i \in X$, $i = 1, \ldots, N$ and a set of arcs as actions $a_i \in A_i$. Agent i lead the game to the subgame ${}^{\sigma_{i-1}}\hat{g}_i^k$ that maximizes its expected discounted

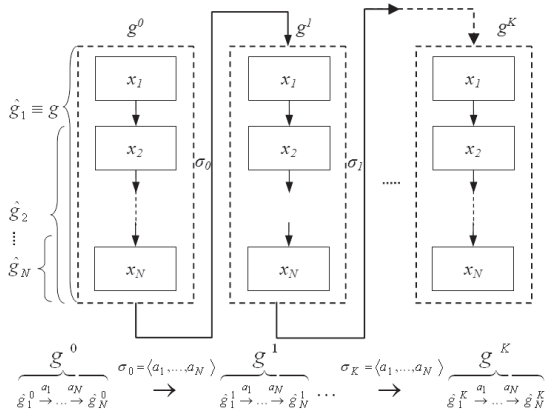

Fig. 1 Extensive Markov games.

future rewards over the set of games according to its game policy:

$$\begin{aligned} V_i^\pi(g) &= E^\pi \left[r_i^{k+1} + \gamma r_i^{k+2} + \gamma^2 r_i^{k+3} + \cdots | g^k = g, \pi \right] \\ &= E^\pi \left[r_i^{k+1} + \gamma V_i^\pi \left(g^{k+1} \right) | g^k = g \right] \\ &= \sum_{\sigma \in \Sigma} \pi(g, \sigma) \sum_{s'} P_{gg'}^\sigma \left[R_{gg'_i}^\sigma + \gamma V_i^\pi(g') \right] \end{aligned}$$

where for all $g, g' \in G$, $\sigma \in \Sigma$,

r_i	Instantaneous rewards for learning agent i,
$V_i^\pi(g)$	Value of game g under policy π,
$\pi(g, \sigma)$	Probability of taking joint action σ in game g,
$\pi_i(g, a_i)$	Policy of agent i to select action a_i in game g,
$\gamma \in [0, 1]$	Discount factor,
$P_{gg'}^\sigma$	$P_{gg'}^\sigma = \Pr\left\{ g^{k+1} = g' \mid g^k = g, \sigma^k = \sigma \right\}$ is the game transition function,
$R_{gg'}^\sigma$	$R_{gg'}^\sigma = E\left\{ r^{k+1} \mid g^k = g, \sigma^k = \sigma, g^{k+1} = g' \right\}$ is the reward function,
$R_{gg'_i}^\sigma$	The i^{th} element of the reward function belonging to agent i.

Thus, the value function of the game g in matrix form is:

$$V^\pi(g) = [V_1^\pi(g) \ \cdots \ V_N^\pi(g)]$$

The optimal value of a game state is the expected infinite discounted sum of future rewards that agent i will gain if it starts in that game and executes the optimal policy.

Each strategy in an NEP is a best response to all other strategies in that equilibrium. The NEP may sometimes appear non-rational in a third-person perspective. This is because it may happen that an NEP is not Pareto optimal. The concept of NEP is completely unrelated to Pareto optimality. Pareto optimality is a measure of efficiency[†]. An outcome of a game is Pareto optimal if there is no other outcome that makes every agent at least as well off and at least one agent

[†]Pareto efficiency is a minimal notion of efficiency and does not necessarily result in a socially desirable distribution of resources, as it makes no statement about equality or the overall well-being of a society.

strictly better off. That is, a Pareto Optimal outcome cannot be improved upon without hurting at least one player. An NEP is not necessarily Pareto Optimal. It implies that values of agents may all be increased yielding to another NEP. In perfect information games, where agents know all about each other, it is clear that rational decision making of self-interested agents will cause NEP points to be "optimal reactions" to "optimal reactions".

On the other hand, SPE is a sharper concept of NEP (Definition 5). It is "time consistent" in that it remains an equilibrium point in whatever truncation of the original game (subgame) the players may find themselves. Thus,

$$\begin{aligned} V_i^*(g) &= \max_{\pi} V_i^{\pi}(g) \\ &= E^{\pi_{NE}}\left[r_i^k + \gamma V_i^*\left(g^{k+1}\right) | g^k = g\right] \\ &= \mathrm{SPEV}_i\left(\sum_{g'} P_{gg'}^{\sigma}\left[R_{gg'}^{\sigma} + \gamma V^*(g')\right]\right) \end{aligned} \tag{1}$$

where $\mathrm{SPEV}_i(.)$ is the value of SPE joint action for agent i. The same can be deduced for action-value function, which is particularly important for learning. The value of taking joint action σ in game g for learning agent i, under joint policy π, denoted by $Q_i^{\pi}(g,\sigma)$, is the expected discounted future reward starting in g, taking σ, and henceforth following π:

$$Q_i^{\pi}(g,\sigma) = \sum_{g'} P_{gg'}^{\sigma}\left[R_{gg'_i}^{\sigma} + \gamma V_i^{\pi}(g')\right]$$

Similarly, optimal action-value function is,

$$Q_i^*(g,\sigma) = \sum_{g'} P_{gg'}^{\sigma}\left[R_{gg'_i}^{\sigma} + \gamma V_i^*(g')\right] \tag{2}$$

These values are the preferences of each agent in a game state over possible joint actions. It is clear that optimal action-value function can be achieved if SPE actions are selected.

The following lemma, formerly proved in [29], substantiates Eq. (2) and clarify the relation between the optimal value of agent i in an EMG and the value of SPE in a game with the preferences $\bar{Q}_i^* = \left[Q_1^*, \ldots, Q_N^*\right]$.

Lemma 1: The following two assertions are equivalent:

- $\pi^* = \left\langle \pi_1^*, \ldots, \pi_N^* \right\rangle$ is an equilibrium point in a discounted EMG with equilibrium values $\left(V_1^*, \ldots, V_N^*\right)$.
- For each $g \in G$, the joint action $\left\langle \pi_1^*(g), \ldots, \pi_N^*(g) \right\rangle$ is an SPE in a game with preferences $\left[Q_1^*(g), \ldots, Q_N^*(g)\right]$ and equilibrium values $\left(V_1^*(g), \ldots, V_N^*(g)\right)$, where:

$$Q^*(g,\sigma) = \sum_{g'} P_{gg'}^{\sigma}\left[R_{gg'}^{\sigma} + \gamma V^*(g')\right] \tag{3}$$

3.2 Estimating World Model

The transition function and the reward function are estimated from the observations received during interaction with the environment. Constructing a model from the experiences can be easily performed by counting the frequency of observing a situation.

There are a number of simple methods for estimating the transition and the reward functions $\hat{P}$ and $\hat{R}$ that converge to the true functions P and R. A maximum likelihood model can be computed as follows [23],

$$\begin{aligned} \hat{P}_{gg'}^{\sigma} &= \frac{C_{gg'}(\sigma)}{C_g(\sigma)} \\ \hat{R}_{gg'}^{\sigma} &= \frac{R_{gg'}(\sigma)}{C_{gg'}(\sigma)} \end{aligned} \tag{4}$$

where,

$C_{gg'}(\sigma)$ Number of transitions from game g to g' after executing joint action σ,
$C_g(\sigma)$ Total number of times that joint action σ is executed in game g,
$R_{gg'}(\sigma)$ Sum of all immediate rewards received after executing σ in game g while game g' is met.

After each observation, both transition and reward functions are updated. In deterministic environments one experience per game-action pair is sufficient to infer the true underlying model. In stochastic environments, however, the effect of an action in a game has to be explored ad infinitum.

3.3 Action Selection

Backward induction (BI) is a well-known algorithm to find SPEs in extensive form games. Each agent tries to predict actions of the lower-level agents, assuming that they are rational and select greedy actions. Greedy actions would be the best idea if agents knew the optimal action-value functions. During learning, however, each agent has only an estimate of action-value functions. There might be that some actions are underestimated and are actually better than the action with the highest value in the current value estimate. At each step, the learning agent must evaluate the direct benefits of choosing the greedy action (exploitation) with the possible benefit of choosing some other actions to find out whether it is not better (Exploration). Regarding the fact that agent i knows all about the previous agents' actions $(a_1, a_2, \ldots, a_{i-1})$, we propose a modified BI algorithm for exploration-exploitation tradeoff which is described compactly for learning agent i as follows:

1. Find the set of optimal actions A_N^* for different subgames $^{\hat{\sigma}_{N-1}}\hat{g}_N$ agent N may play with respect to Q_N,
2. Find the set of optimal actions A_{N-1}^* for different subgames $^{\hat{\sigma}_{N-2}}\hat{g}_{N-1}$ agent $N-1$ may play with respect to Q_{N-1} and A_N^*,
3. Continue the same procedure until agent i is reached.
4. Find the set of agent i's Q-values according to the previous set of actions $(a_1, a_2, \ldots, a_{i-1})$ and the set of optimal joint actions $A_{i+1}^* \times \cdots \times A_N^*$,
5. Select an action probabilistically with respect to Boltzmann distribution.

3.4 Learning Algorithm

At each time step k, the learning agent i decides on its actions by observing subsequent subgames. At the end of each game, it observes its own reward, actions taken by the other agents and their relevant rewards, and the new game state g'. It later updates its world model, including the transition function and the reward function. Finally, it updates its Q-values as well as the others.

During world simulation, the learning agent performs some imaginary observations on different game-action pairs based on its models of the world and the beliefs about the other agents. The priority queue is adjusted according to a quite similar procedure as the one in [20]. It then updates both its own preferences as well as the belief about the preferences of the other agents, since it is assumed that there is no explicit communication among agents. The following algorithm consists of four sub-routines: action selection, updating world model, single update of value function, and world simulation.

MBRL algorithm for learning agent *i*:

- Initialize all Q-tables, functions, and *Queue*;
- **Do forever:**
- $g \leftarrow$current game, $k \leftarrow k+1$
- Select action;
- Wait until the last agent N selects its action;
- Observe new game g' and rewards;
- Update world model (Eq. (4));
- Update Q-values: $Q(g,\sigma) = \sum_{g'} \hat{P}^{\sigma}_{gg'} \left[\hat{R}^{\sigma}_{gg'} + \gamma V(g')\right]$
- While *Queue* is not empty or the maximum number of iterations is not reached (*World simulation*):
  - Select the highest priority game from the *Queue*, call it $\bar{g}$;
  - For all admissible joint action $\bar{\sigma}$ (Perform a Bellman backup to recompute the Q-values)
    - $V' = V(\bar{g})$
    - Update Q-values: $\bar{Q}(\bar{g},\bar{\sigma}) = \sum_{\bar{g}'} \hat{P}^{\bar{\sigma}}_{\bar{g}\bar{g}'} \left[\hat{R}^{\bar{\sigma}}_{\bar{g}\bar{g}'} + \gamma V(\bar{g}')\right]$
    - $V(\bar{g}) = \mathrm{SPEV}\left(\bar{Q}(\bar{g},\bar{\sigma})\right)$
  - End for
  - $\Delta(\bar{g}) = |V(\bar{g}) - V'|$
  - For any predecessors $\tilde{g}$ of $\bar{g}$ for which there exist a joint action $\tilde{\sigma}$ such that $\hat{P}^{\tilde{\sigma}}_{\tilde{g}\bar{g}} = \hat{P}(\tilde{g},\tilde{\sigma},\bar{g}) \neq 0$:
    - $p = \hat{P}^{\tilde{\sigma}}_{\tilde{g}\bar{g}} \cdot \Delta$;
    - If $p > \theta$ then insert $\tilde{g}$ in the queue with priority p;
  - End for
- End while
- **End do**

4. Convergence Proof

In this section we are going to provide a formal convergence proof for the proposed algorithm based on the well-known theorems proved in [30], which have being frequently used in similar papers, such as [31] and [16]. The following assumption is necessary to deduce convergence in RL process, which its feasibility is discussed.

Assumption 1: Every subgame of game state $g \in G$ is reached infinitely often.

Discussion: Consider that a subgame has been reached while modified BI is used for action selection. A learning agent may select its actions randomly proportional to their corresponding values which are non-zero numbers. Thus, each action is almost surely selected an infinite number of times. Recursively, in the higher level subgame, the actions which end to the current subgame is almost surely selected an infinite number of times. Hence, the current subgame (Recall that each game is also a subgame) is reached infinitely often if the root game is initiated an infinite number of times.

The following lemma, which is used in convergence proof, prove that optimal value operator is non-expansion.

Lemma 2: Assume that $\mathcal{B}$ is the space of bounded functions over G. The optimal value operator $T : \mathcal{B}(\mathcal{G}) \rightarrow \mathcal{B}(\mathcal{G})$ is a non-expansion operator, where:

$$TV(g) = \mathrm{SPEV}\left(\sum_{g'} P^{\sigma}_{gg'}\left[R^{\sigma}_{gg'} + \gamma V(g')\right]\right)$$

Proof: We should prove that $\|TV - Tv\| \leq \|V - v\|$, where, $V = [V_i]|_{i=1,\ldots,N}$ and $v = [v_i]|_{i=1,\ldots,N}$. For the space of the uniformly bounded functions $\mathcal{B}$ over G, the appropriate norm is the supremum norm.:

$$B = \left\{ f : \mathcal{X} \rightarrow \mathfrak{R} : \|f\| = \sup_{x \in \mathcal{X}} f(x) < \infty \right\} \tag{5}$$

Thus, for the learning agent i, considering that BI is used to find SPEs, we have,

$$\begin{aligned}
\|TV_i - Tv_i\| &= \left\| \mathrm{SPEV}_i\left(\sum_{g'} P^{\sigma}_{gg'}\left[R^{\sigma}_{gg'_i} + \gamma V_i(g')\right]\right) \right. \\
&\qquad \left. - \mathrm{SPEV}_i\left(\sum_{g'} P^{\sigma}_{gg'}\left[R^{\sigma}_{gg'_i} + \gamma v_i(g')\right]\right) \right\| \\
&= \left\| \max_{a_i \in A_i} \sum_{g'} P^{\sigma^*_{-i} \times a_i}_{gg'}\left[R^{\sigma^*_{-i} \times a_i}_{gg'_i} + \gamma V_i(g')\right] \right. \\
&\qquad \left. - \max_{a_i \in A_i} \sum_{g'} P^{\sigma^*_{-i} \times a_i}_{gg'}\left[R^{\sigma^*_{-i} \times a_i}_{gg'_i} + \gamma v_i(g')\right] \right\| \\
&\xrightarrow{\text{Eq. (5)}} = \max_{g} \left| \max_{a_i \in A_i} \sum_{g'} P^{\sigma^*_{-i} \times a_i}_{gg'}\left[R^{\sigma^*_{-i} \times a_i}_{gg'_i} + \gamma V_i(g')\right] \right.
\end{aligned}$$

$$
\begin{aligned}
&\quad - \max_{a_i \in A_i} \sum_{g'} P_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma v_i(g') \right] \Bigg| \\
&\leq \max_{g} \max_{a_i \in A_i} \left| \sum_{g'} P_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right. \\
&\quad \left. - \sum_{g'} P_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma v_i(g') \right] \right| \\
&\leq \max_{g} \max_{a_i \in A_i} \left| \sum_{g'} P_{gg'}^{\sigma_{-i}^* \times a_i} \left[V_i(g') - v_i(g') \right] \right| \\
&\leq \| V_i - v_i \|
\end{aligned}
$$

(Recall that transition probability over games are less than or equal to 1.) Clearly, in matrix form:

$$\| TV - Tv \| \leqslant \| V - v \|$$

Now, it is possible to prove the convergence of the proposed algorithm.

Lemma 3: Consider a finite EMG such that Assumption 1 holds. The values are updated iteratively by the following rule,

$$V^{k+1}(g) = \begin{cases} \text{SPEV}\left(\sum_{g'} \hat{P}_{gg'}^{\sigma} \left[\hat{R}_{gg'}^{\sigma} + \gamma V(g') \right] \right), & \text{if } g = g^k \\ V^k(g), & \text{otherwise} \end{cases} \tag{6}$$

Assume that the followings hold with probability 1 for all (g, σ, g'):

1. $\hat{R}_{gg'}^{\sigma}$ converges to $R_{gg'}^{\sigma}$,
2. $\hat{P}_{gg'}^{\sigma}$ converges to $P_{gg'}^{\sigma}$, such that:

$$\lim_{k \to \infty} \max_{a_i \in A_i} \left| \sum_{g'} \left[\hat{P}_{gg'}^{\sigma_{-i} \times a_i} - P_{gg'}^{\sigma_{-i} \times a_i} \right] \right| = 0$$

3. $0 \leq \gamma < 1$.

Then, $\left\{ V^k \right\}$ converges to the fixed point of the operator $T : \mathcal{B}(G) \to \mathcal{B}(G)$, where:

$$TV_i(g) = \text{SPEV}\left(\sum_{g'} P_{gg'}^{\sigma} \left[R_{gg'}^{\sigma} + \gamma V(g') \right] \right)$$

Proof: The appropriate approximate dynamic programming operator sequence $\left\{ T^k \right\}$ is defined as,

$$T^k(U, V)(g) = \begin{cases} \text{SPE}^V\left(\sum_{g'} \hat{P}_{gg'}^{\sigma} \left[\hat{R}_{gg'}^{\sigma} + \gamma V(g') \right] \right), & \text{if } g = g^k \\ U(g), & \text{otherwise} \end{cases}$$

Now, we should prove that T^k approximate T in the sense that the recursion generated by $U^{k+1} = T^k\left(U^k, V\right)$ converges to TV in the norm of $\mathcal{B}$ with probability 1. It is clear that we should prove that the following distance converges to zero for learning agent i as $k \to \infty$

$$
\begin{aligned}
D_i^k = &\left| \text{SPEV}_i \left(\sum_{g'} \hat{P}_{gg'}^{\sigma} \left[\hat{R}_{gg'_i}^{\sigma} + \gamma V_i(g') \right] \right) \right. \\
&\left. - \text{SPEV}_i \left(\sum_{g'} P_{gg'}^{\sigma} \left[R_{gg'_i}^{\sigma} + \gamma V_i(g') \right] \right) \right|
\end{aligned} \tag{7}
$$

Consider that BI is used to find the SPE solution for learning agent i,

$$
\begin{aligned}
D_i^k = &\left| \text{SPEV}_i \left(\sum_{g'} \hat{P}_{gg'}^{\sigma} \left[\hat{R}_{gg'_i}^{\sigma} + \gamma V_i(g') \right] \right) \right. \\
&\left. - \text{SPEV}_i \left(\sum_{g'} P_{gg'}^{\sigma} \left[R_{gg'_i}^{\sigma} + \gamma V_i(g') \right] \right) \right| \\
= &\left| \max_{a_i \in A_i} \sum_{g'} \hat{P}_{gg'}^{\sigma_{-i}^* \times a_i} \left[\hat{R}_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right. \\
&\left. - \max_{a_i \in A_i} \sum_{g'} P_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right|
\end{aligned} \tag{8}
$$

where σ_{-i}^* is the sequence of action except a_i, when all agents follow the BI action selection algorithm. Using triangular inequality, we have,

$$
\begin{aligned}
D_i^k \leq &\max_{a_i \in A_i} \left| \sum_{g'} \hat{P}_{gg'}^{\sigma_{-i}^* \times a_i} \left[\hat{R}_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right. \\
&\left. - \sum_{g'} P_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right| \\
\leq &\max_{a_i \in A_i} \left| \sum_{g'} \hat{P}_{gg'}^{\sigma_{-i}^* \times a_i} \left[\hat{R}_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right. \\
&\left. - \sum_{g'} \hat{P}_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right| \\
&+ \max_{a_i \in A_i} \left| \sum_{g'} \hat{P}_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right. \\
&\left. - \sum_{g'} P_{gg'}^{\sigma_{-i}^* \times a_i} \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right| \\
= &\max_{a_i \in A_i} \left| \sum_{g'} \hat{P}_{gg'}^{\sigma_{-i}^* \times a_i} \left[\hat{R}_{gg'_i}^{\sigma_{-i}^* \times a_i} - R_{gg'_i}^{\sigma_{-i}^* \times a_i} \right] \right| \\
&+ \max_{a_i \in A_i} \left| \sum_{g'} \left[\hat{P}_{gg'}^{\sigma_{-i}^* \times a_i} - P_{gg'}^{\sigma_{-i}^* \times a_i} \right] \left[R_{gg'_i}^{\sigma_{-i}^* \times a_i} + \gamma V_i(g') \right] \right|
\end{aligned}
$$

According to assumption 1, $D_i^k \to 0$ as $k \to \infty$ for $i = 1, \ldots, N$. Thus, T^k approximate T.

The rest of the proof is based on the following theorem proved in [30]:

Theorem 1: [30] Let v^* be a fixed point of T and let $\mathcal{T} = \left(T^0, T^1, \ldots \right)$ approximate T at v^*. Let $V^0 \in \mathcal{B}$, and

define $V^{k+1} = T^k\left(V^k, V^k\right)$. If there exist random function $0 \leq \zeta^k(g) \leq 1$ and $0 \leq \xi^k(g) \leq 1$ satisfying the conditions below with probability 1, then V^k converges to v^* with probability 1 in the norm of $\mathcal{B}(G)$:

1. For all U_1 and $U_2 \in \mathcal{B}$, and all $g \in G$, $\left|T^k(U_1, v^*)(g) - T^k(U_2, v^*)(g)\right| \leq \xi^k(g)\left|U_1(g) - U_2(g)\right|$
2. For all U and $V \in \mathcal{B}$, and all $g \in G$, $\left|T^k(U, v^*)(g) - T^k(U, V)(g)\right| \leq \zeta^k(g)\left(\|v^* - V(g)\| + \lambda^k\right)$ where $\lambda^k \to 0$ with probability 1 as $t \to \infty$,
3. For all $j > 0$, $\Pi_{k=j}^{n} \xi^k(g)$ converges to zero uniformly in g as $n \to \infty$,
4. There exist $0 \leq \gamma < 1$ such that for all $g \in G$ and large enough k,

$$\zeta^k(g) \leq \gamma\left(1 - \xi^k(g)\right)$$

Conventionally, the following two functions are selected:

$$\xi^k(g) = \begin{cases} 0, & g = g^k \\ 1, & \text{otherwise} \end{cases}$$

and,

$$\zeta^k(g) = \begin{cases} \gamma, & g = g^k \\ 0, & \text{otherwise} \end{cases}$$

The aforementioned functions together with the result in Lemma 2 satisfy all the conditions of Theorem 1, and the proof is complete.

Another convergence proof for single agent PS was proposed in [32]. It was tried to establish a formal proof of convergence to the optimal value function when they are used as planning algorithms. They proposed not to initialize priority queue to zero. Instead, of randomly initialization, we propose to initialize them to a small positive number. It causes better coordination during action selection since there is no explicit communication among agents. Another condition which is proved to be necessary for convergence is that the Bellman error $\left|V_i^{k+1}(g) - V_i^k(g)\right|$ converges to zero in the limit $k \to \infty$. Regarding that the optimal value operator is non-expansion according to Lemma 2, and the results in Lemma 3, it is easy to verify the necessary condition for Bellman error. Thus, according to the convergence lemma in [32], our proposed multiagent PS is also convergent.

5. Conclusions

The combination of RL and game theoretic solvers represents a promising solution to effective learning in multiagent systems, especially in dynamic environments. Most of the proposed algorithms gradually converge to the optimal policy after a great number of experiences. MBRL offers simultaneously planing and learning to use the gathered information more efficiently. Regarding our previous papers, in this paper, we developed MBRL for EMGs. Every single situation of decision making in this learning process is modeled as an n-person general-sum extensive form game with perfect information. BI is traditionally used in extensive form games to find SPEs, which was proposed to combine with Boltzmann distribution to adjust the tradeoff between exploration and exploitation. During action selection, each agent assumes that it is the only learning agent and all the lower-level agents select their actions rationally. The pseudo code of the algorithm has been given which can be used easily for implementation. The algorithm was proved to be convergent and discussed based on the newly developed work on the convergence of the traditional PS.

In the upcoming paper, we will concentrate on the performance of the algorithm. Controlling the number of updates per simulation and using heuristic approaches may speed up the convergence.

References

[1] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," Autonmous Agents Multi-Agent Systems, vol.11, no.3, pp.387–434, 2005.
[2] M.A. Potter and K.A.D. Jong, "A cooperative coevolutionary approach to function optimization," Lecture Notes in Computer Science, vol.866, pp.249–257, 1994.
[3] S.G. Ficici and J.B. Pollack, "A game-theoretic approach to the simple coevolutionary algorithm," Proc. Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI), Springer Verlag, 2000.
[4] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," IEEE Trans. Syst. Man Cybern. C, Appl. Rev., vol.38, no.2, pp.156–172, 2008.
[5] R.S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.
[6] L.P. Kaelbling, M.L. Littman, and A.W. Moore, "Reinforcement learning: A survey," J. Artificial Intelligence Research, vol.4, pp.237–285, 1996.
[7] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," Proc. Fifteenth National Conference of Artificial Intelligence (AAAI-98)/Proc. Tenth Conference of Innovative Applications of the Artificial Intelligence (IAAI-98), Madison, 1998.
[8] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," Proc. Tenth International Conference on Machine Learning (ICML-93), pp.330–337, Amherst, US, 1993.
[9] M.L. Littman, "Algorithms for sequential decision making," Providence, Rhode Island, 1996.
[10] J. Hu and P. Wellman, "Multiagent reinforcement learning: Theoretical framework and an algorithm," Proc. Fifteenth International Conference on Machine Learning, pp.242–250, 1998.
[11] A. Akramizadeh, A. Afshar, and M.-B. Menhaj, "Different forms of the games in multiagent reinforcement learning: Alternating vs. simultanous movements," 17th Mediterranean Conference on Control and Automation, Thessaloniki, Greece, 2009.
[12] M.L. Littman, "Friend-or-foe Q-learning in general-sum games," Proc. Eighteenth International Conference on Machine Learning (ICML-01), pp.322–328, Williams College, Williams town, US, 2001.
[13] P. Stone and M. Veloso, "Multiagent systems: A survey from the machine learning perspective," Auton. Robots, vol.8, no.3, pp.345–383, 2000.
[14] Y. Shoham, R. Powers, and T. Grenager, "Multi-agent reinforcement learning: A critical survey," AAAI Fall Symposium on Artificial Multi-Agent Learning, 2004.
[15] H.J. van den Herik, M.K.D. Hennes, K. Tuyls, and K. Verbeeck,

"Multi-agent learning dynamics: A survey," Cooperative Information Agents XI, pp.36–56, Berlin, Springer, 2007.
[16] V. Kononen, "Asymmetric multiagent reinforcement learning," Web Intelligence and Agent Systems: An International Journal, pp.105–121, 2004.
[17] A. Akramizadeh, A. Afhsr, and M.B. Menhaj, "Multiagent reinforcement learning in extensive form games with perfect information," J. Applied Science, vol.9, no.11, pp.2056–2066, 2009.
[18] B.L. Slantchev, Game Theory, Department of Political Science, University of California, San Diego, 2008.
[19] R.S. Sutton, "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming," 7th International Conference on Machine Learning, Austin, 1990.
[20] A.W. Moore and C.G. Atkeson, "Prioritized sweeping: Reinforcement learning with less data and less real time," Machine Learning, vol.13, no.1, pp.103–130, 1993.
[21] J. Peng and R.J. Williams, "Efficient learning and planning within the Dyna framework," Adaptive Behavior, pp.437–454, 1993.
[22] W. Uther and M. Veloso, "Adversarial reinforcement learning," Technical report, Carnegie Mellon University, 1997. Unpublished.
[23] M.A. Wiering, "Model-based reinforcement learning in dynamic environments," Utrecht, 2002.
[24] A. Akramizadeh, M.B. Menhaj, and A. Afshar, "Extensive Q-learning, a new approach in multiagent reinforcement learning," Adaptive Dynamic Programming and Reinforcement Learning (ADPRL2009), Series Symposium in Computational Intelligenence (SSCI2009), Nashville, 2009.
[25] D. Andre, N. Friedman, and R. Parr, "Generalized prioritized sweeping," Advances in Neural Information Processing Systems 10, pp.1001–1007, 1998.
[26] D. Wingate and K.D. Seppi, "Prioritization methods for accelerating MDP solvers," J. Machine Learning Research, pp.851–881, 2005.
[27] M.J. Osborne, An Introduction to Game Theory, Oxford University Press, 2000.
[28] D. Fundenberg and D.K. Levine, The Theory of Learning in Games, MIT Press, Cambridge, Massachusetts, 1998.
[29] J. Filar and K. Vrieze, Competitive Markov Decision Process, Springer Verlag, New York, 1997.
[30] C. Szepesvari and M.L. Littman, "A unified analysis of value-function-based reinforcement-learning algorithms," Neural Comput., vol.11, no.8, pp.2017–2059, 1999.
[31] J. Hu and M. Wellman, "Nash Q-learning for general-sum stochastic games," J. Machine Learning Research, vol.4, pp.1039–1069, 2003.
[32] L. Li and M.L. Littman, "Prioritized sweeping converges to the optimal value function," NJ, 2008.
[33] A. Akramizadeh, A. Afshar, M.-B. Menhaj, and S. Jafari, "Model-based reinforcement learning in multiagent systems with hierarchical structure," Accepted to be published in IFAC Large Scale Systems: Theory and Applications, Villeneuve d'Ascq, France, 2010.
[34] P. Jehiel and D. Samet, "Learning to play games in extensive form by valuation," Ecole Nationale des Ponts et Chaussees, 2000.
[35] J.-F. Laslier and B. Walliser, "A reinforcement learning process in extensive form games," International Journal of Game Theory, pp.219–227, 2005.
[36] J.-F. Laslier, R. Topol, and B. Walliser, "A behavioral learning process in games," Games and Economic Behavior, vol.37, no.2, pp.340–366, 2001.
[37] M. Pak, "Reinforcement learning in perfect-information games," working paper, Queen's Economics Department, Canada, 2006.
[38] P. Huang and K. Sycara, "Multi-agent learning in extensive games with complete information," Proc. Second International Joint Conference on Autonomous Agents and Multiagent Systems, pp.701–708, 2003.

Ali Akramizadeh received the M.Sc. degree and the B.Sc. degree from the Ferdowsi University of Mashhad, Iran, in 2000 and 2003, respectively, both in control engineering. Currently, he is working toward the Ph.D. degree at the Computational Intelligence and Large Scale System Research Lab., Faculty of Electrical Engineering, Amir Kabir University of Technology, Tehran, Iran. His current research interests includes reinforcement learning in multiagent systems, approximate reinforcement learning, and adaptive and learning control.

Ahmad Afshar received his Ph.D. degree in Electrical engineering from Manchester University in 1991 and subsequently started his career as a member of staff with University of Petroleum in the department of Electrical Engineering. He joined Amir Kabir University of Technology in 1998 in the Department of Electrical Engineering. He is currently an Assistant Professor in the control engineering group of this department and is the head of the automation and IT research laboratory and jointly conducts the computational intelligence and large scale systems research laboratory. His research fields include large scale systems, intelligent control, multi-agent systems, network based control systems, sensor networks and decision support systems.

Mohammad Bagher Menhaj received his Ph.D. degree from school of electrical and computer engineering at OSU in 1992. After completing one year with OSU as postdoctoral fellow, in 1993, he joined the Amir-Kabir University of technology in Tehran-Iran. December 2000 to Aug. 2003, he was with school of electrical and computer engineering and department of Computer science at OSU as visiting faculty member and research scholar. He is author and co-author of more than 180 technical papers. He is author of three books: Computational Intelligence (vol.1): Fundamentals of Neural Networks, (vol.2): Fundamental of Fuzzy systems, and application of Computational Intelligence in Control (vol.1), all in Persian. His main research interests are: theory of computational intelligence, learning automata, adaptive filtering and their applications in control, power systems, image processing, pattern recognition, and communications, and other areas of interests are: theory of rough set and knowledge discovery.

Samira Jafari received the B.Sc. degree from the Islamic Azad University of Ghazvin, Iran, in 2007 in electronic engineering. Currently, she is working toward the M.Sc. degree in the Islamic Azad University of Tehran (South), Iran. Her current research interests includes reinforcement learning in multiagent systems, web-based learning, and adaptive and learning control.