Multiagent Learning in Adaptive Dynamic Systems

Andriy Burkov
DAMAS Laboratory
Laval University
G1K 7P4, Quebec, Canada
burkov@damas.ift.ulaval.ca

Brahim Chaib-draa
DAMAS Laboratory
Laval University
G1K 7P4, Quebec, Canada
chaib@damas.ift.ulaval.ca

ABSTRACT

Classically, an approach to the multiagent policy learning supposed that the agents, via interactions and/or by using preliminary knowledge about the reward functions of all players, would find an interdependent solution called "equilibrium". Recently, however, certain researchers question the necessity and the validity of the concept of equilibrium as the most important multiagent solution concept. They argue that a "good" learning algorithm is one that is efficient with respect to a certain class of counterparts. Adaptive players is an important class of agents that learn their policies separately from the maintenance of the beliefs about their counterparts' future actions and make their decisions based on that policy and the current belief. In this paper, we propose an efficient learning algorithm in presence of the adaptive counterparts called Adaptive Dynamics Learner (ADL), which is able to learn an efficient policy over the opponents' adaptive dynamics rather than over the simple actions and beliefs and, by so doing, to exploit these dynamics to obtain a higher utility than any equilibrium strategy can provide. We tested our algorithm on a substantial representative set of the most known and demonstrative matrix games and observed that ADL agent is highly efficient against Adaptive Play Q-learning (APQ) agent and Infinitesimal Gradient Ascent (IGA) agent. In self-play, when possible, ADL is able to converge to a Pareto optimal strategy maximizing the welfare of all players.

General Terms

Algorithms, Performance, Experimentation

Keywords

Multiagent learning, Matrix games, Adaptation

1. INTRODUCTION

Classically, an approach to the multiagent policy learning supposed that the agents, by means of interactions and/or by using preliminary knowledge about the reward functions of all players, would find an interdependent solution called "equilibrium". One of the most used concepts of equilibrium is the Nash equilibrium where each agent in a multiagent system (MAS) plays its best response to the other players' strategies and a unilateral deviation of a player from the equilibrium strategy decreases its own utility. There are two basic approaches to finding a Nash equilibrium. The first one is a game theoretic approach, which supposes the complete knowledge of the reward structure of the underlying game by all agents. In such an approach, each agent calculates an equilibrium by using mathematical programming, and all agents play on it. But a problem arises when there are several tantamount equilibria in a game and the agents have calculated the different ones. Another problem is that the agents, by calculating an equilibrium, suppose that the other agents are rational and, thus, they will also follow this solution. But what if certain agents are not rational, or play a fixed strategy, or evolve according to some fixed rules, and what if some agents know (or are able to deduct) this and could exploit this knowledge to augment their utilities? As yet, there is no equilibrium concept which can answer this question.

The second approach to finding an equilibrium is the adaptive one, which supposes that the agents learn by adapting to each other in self-play (i.e., all agents use the same learning algorithm), do not know the reward structure of the game and are only able to make actions and observe their own rewards and, in some approaches, the actions made by the others. It was shown that certain algorithms of this class converge to a Nash equilibrium (or to a utility that is equivalent to the utility of a Nash equilibrium). Among these algorithms the most outstanding are Joint-Action Learning [3], Infinitesimal Gradient Ascent (IGA)[1] [9], Policy Hill-Climbing [1] and Adaptive Play Q-learning [4] (a Q-learning based extension of the Adaptive Play algorithm [11]). The adaptive players[2] learn their policies separately from the maintenance of the beliefs about their counterparts' future actions and make their decisions based on that policy and the current belief. These decisions can be in pure or in mixed strategies depending on the algorithm in question.

[1]IGA is the only algorithm among those listed, which requires a complete knowledge of the reward structure of the game to calculate the gradient.

[2]To discriminate between adaptive player as a member of a class of learning agents and Young's Adaptive Player, we will write "adaptive player" or "adaptive algorithm" (in lower case) to denote the former and Adaptive Player (with a capital letter) for the latter.

Recently, certain researchers question the necessity and the validity of the concept of equilibrium as the most important multiagent solution concept [8]. They rather point out the efficiency of a particular learning algorithm versus a certain class of counterparts. The *adaptive players* is such a class of learning agents, which showed good results with respect to the convergence to a Nash equilibrium, and in this paper, we propose an efficient algorithm of learning in presence of the adaptive counterparts called Adaptive Dynamics Learner (ADL). Our ADL algorithm is able to learn an efficient policy over the adaptation dynamics of its opponents' rather than over the simple actions and beliefs and, by so doing, to exploit these dynamics to obtain a higher utility than any equilibrium strategy can provide. We tested our algorithm on a set of the most known and demonstrative repeated matrix games and the results show that ADL agent is highly efficient in self-play and against APQ and IGA agents (this comparison is correct as soon as APQ and IGA agents have the same or a greater quantity of information about the world as compared to ADL).

The rest of the paper is organized as follows. First, we describe IGA and APQ learning algorithms. Then, we present our novel approach to the learning of the adaptive dynamics in MAS. Then, by comparative testing, we demonstrate that our algorithm exploits IGA and APQ players in adversarial games by remaining rational and highly efficient in other games, versus the stationary opponents and versus itself as well. Again, in self-play it converges to a solution that maximizes the welfare of both players.

2. ADAPTIVE DYNAMICS

In this section, we describe two algorithms, which represent two basic subclasses of adaptive learning algorithms: those that are able to learn a pure strategy and those that are able to learn a mixed one. These are APQ and IGA algorithms respectively. But first, we present the notation we use in this paper. A (normal form) stage game is a tuple $(n, \mathcal{A}^{1\ldots n}, R^{1\ldots n})$, where n is the number of players, $\mathcal{A}^j$ is the strategy space of player $j, j = 1 \ldots n$, and the value function $R^j : \times\mathcal{A}^j \mapsto \mathbb{R}$ defines the utility for player j of a joint action $\mathbf{a} \in \mathbf{A} = \times\mathcal{A}^j$.

A *mixed* strategy for player j is a distribution π^j, where $\pi^j_{a^j}$ is the probability for player j to select some action a^j. A strategy is *pure* if $\pi^j_{a^j} = 1$ for some a^j. A *strategy profile* is a collection $\Pi = \{\pi^j | j = 1 \ldots n\}$ of all players' strategies. A *reduced profile for player* j, $\Pi^{-j} = \Pi\backslash\{\pi^j\}$, is a strategy profile containing strategies of all players except j, and $\Pi^{-j}_{\mathbf{a}^{-j}}$ is the probability for players $k \neq j$ to play a joint action $\mathbf{a}^{-j} \in \mathbf{A}^{-j} = \times\mathcal{A}^{-j}$ where $\mathbf{a}^{-j}$ is $\langle a^k | k = 1 \ldots n, k \neq j\rangle$. Given a player j and a reduced profile Π^{-j}, a strategy $\hat{\pi}^j$ is a *best reply* (BR) to Π^{-j} if the expected utility of the strategy profile $\Pi^{-j} \cup \{\hat{\pi}^j\}$ is maximal for player j. Since a best reply may not to be unique, there is a set of best replies of player j to a reduced profile Π^{-j} which is denoted as $BR^j(\Pi^{-j})$. More formally, the expected utility of a strategy profile Π for a player j is given by:

$$U^j(\Pi) = \sum_{a^j \in \mathcal{A}^j} \pi^j_{a^j} \sum_{a^{-j} \in \mathbf{A}^{-j}} R(\langle a^j, \mathbf{a}^{-j}\rangle)\Pi^{-j}_{\mathbf{a}^{-j}}$$

where Π is $\Pi^{-j} \cup \{\pi^j\}$ and $R(\langle a^j, \mathbf{a}^{-j}\rangle)$ is the value that player j receives if the joint action $\mathbf{a} = \langle a^j, \mathbf{a}^{-j}\rangle$ is played by all players. In this case, a best reply of player j to the reduced profile Π^{-j} is a strategy $\hat{\pi}^j$ such that:

$$U^j(\Pi^{-j} \cup \{\hat{\pi}^j\}) \geq U^j(\Pi^{-j} \cup \{\pi^j\}) \qquad \forall\pi^j \neq \hat{\pi}^j$$

A repeated game (or iterated game) is a game which consists of a certain number of repetitions of some stage game.

2.1 Adaptive Play Q-learning

Formally, each player j playing Adaptive Play saves in memory a history $H^j_t = \{\mathbf{a}^{-j}_{t-p}, \ldots, \mathbf{a}^{-j}_t\}$ of the last p joint actions played by the other players. To select a strategy to play at time $t+1$, each player randomly and irrevocably samples from H^j_t a set of examples of length l, $\hat{H}^j_t = \{\mathbf{a}^{-j}_{k_1}, \ldots, \mathbf{a}^{-j}_{k_l}\}$, and calculates the empiric distribution $\hat{\Pi}^{-j}$ as an approximation of the real reduced profile of strategies played by the other players, using the following:

$$\hat{\Pi}^{-j}_{\mathbf{a}^{-j}} = \frac{C(\mathbf{a}^{-j}, \hat{H}^j_t)}{l} \tag{1}$$

where $C(\mathbf{a}^{-j}, \hat{H}^j_t))$ is the number of times that the joint action $\mathbf{a}^{-j}$ was played by the other players according to the set $\hat{H}^j_t$. Given the probability distribution over the other players' actions, $\hat{\Pi}^{-j}$, the player j plays its best reply, $BR^j(\hat{\Pi}^{-j})$, to this distribution with some exploration. If there are several equivalent best replies, the player j randomly chooses one of them. Young [11] proved the convergence of Adaptive Play to an equilibrium when played in self-play for a big class of games such as the coordination and common interest games. APQ is an extension of Young's Adaptive Play to the multi-state stochastic game context[3]. To do that, the usual single-agent Q-learning update rule was modified to consider multiple agents as follows:

$$\begin{aligned} Q^j(s, \mathbf{a}) \quad \leftarrow \quad & (1-\alpha)Q^j(s,\mathbf{a}) + \alpha[R^j(s,\mathbf{a}) \\ & + \gamma \max_{a^j \in \pi^j(s')} U^j(\hat{\Pi}(s') \cup \{\pi^j(s')\})] \end{aligned}$$

where j is an agent, $\mathbf{a}$ is a joint action played by the agents in state $s \in \mathcal{S}$, $Q^j(s,\mathbf{a})$ is the current value for player j of playing the joint action $\mathbf{a}$ in state s, $R^j(s,\mathbf{a})$ is the immediate reward the player j receives if the joint action $\mathbf{a}$ is played in the state s and $\pi^j(s')$ are all possible *pure* strategies that are available for player j in state s'. In the repeated game context, $|\mathcal{S}| = 1$.

2.2 Infinitesimal Gradient Ascent

Singh et al. [9] examined the dynamics of using the gradient ascent in two-player, two-action repeated games. The problem can be represented with two payoff matrices for the row and column players, r and c, as follows:

$$R^r = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}, \quad R^c = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

The players r and c select simultaneously an action from the set $\mathcal{A}^{r,c} = \{1, 2\}$, the row player r selects an action i, the column player c selects an action j and the payoffs they obtain are R^r_{ij} and R^c_{ij} respectively.

As long as this is a two-action game, a mixed strategy can be represented as a single value. Let $\alpha \in [0,1]$ be a probability the player r selects the action 1 and $1-\alpha$ the probability to play the action 2. Let, similarly, $\beta \in [0,1]$ and $1-\beta$ be the probabilities to play the actions 1 and 2

[3]In stochastic games each system's state s is considered as a matrix game.

respectively by the player c. The expected utility of playing a strategy profile $\Pi = \{\alpha, \beta\}$ is then calculated as follows:

$$\begin{aligned} U^r(\{\alpha,\beta\}) &= r_{11}\alpha\beta + r_{22}(1-\alpha)(1-\beta) \\ &\quad + r_{12}\alpha(1-\beta) + r_{21}(1-\alpha)\beta \\ U^c(\{\alpha,\beta\}) &= c_{11}\alpha\beta + c_{22}(1-\alpha)(1-\beta) \\ &\quad + c_{12}\alpha(1-\beta) + c_{21}(1-\alpha)\beta \end{aligned}$$

To estimate the effect of changing its current strategy, a player can calculate a partial derivative of its expected utility with respect to its mixed strategy:

$$\frac{\partial U^r(\{\alpha,\beta\})}{\partial \alpha} = \beta u - (r_{22} - r_{12})$$

$$\frac{\partial U^c(\{\alpha,\beta\})}{\partial \beta} = \alpha u' - (c_{22} - c_{21})$$

where $u = (r_{11} + r_{22}) - (r_{21} + r_{12})$ and $u' = (c_{11} + c_{22}) - (c_{21} + c_{12})$.

At each time step IGA player adjusts its current strategy in the direction of the gradient to as to maximize its utility:

$$\alpha_{t+1} = \alpha_t + \eta \frac{\partial U^r(\{\alpha_t,\beta_t\})}{\partial \alpha}$$

$$\beta_{t+1} = \beta_t + \eta \frac{\partial U^c(\{\alpha_t,\beta_t\})}{\partial \beta}$$

where η is a step size, usually $0 < \eta \ll 1$. Obviously, the opponent's mixed strategy is supposed to be known by the players.

Singh and colleagues [9] proved the convergence of IGA to a Nash equilibrium (or to the equivalent average reward), when played in self-play, in the case of the infinitesimal step size ($\lim_{\eta\to 0}$), whence the name of the algorithm.

3. ADAPTIVE DYNAMICS LEARNING

Although the adaptive algorithms show an efficient behavior in self-play, Chang and Kaelbling [2] showed that they can be exploited on the example of exploiting a Policy Hill Climbing (PHC) player [1]. Their PHC-Exploiter agent was able to outperform a PHC player in adversarial games by using the knowledge of the structure of PHC's adaptation mechanism. As was later shown by Tesauro [10], it is possible to exploit the adaptive dynamics with a simple knowledge that the opponent is an adaptive player. His Hyper-Q learning algorithm learned explicitly the Q-values of the mixed strategy profiles. To do that, he discretized the probability space with a certain discretization size and empirically showed that Hyper-Q outperforms PHC and IGA players in RockPaperScissors game. But there are three major shortcomings that make this algorithm intractable in real implementations. These are (1) discretization, which creates about 100 thousands of virtual states for a game with merely two players and three actions, such as RockPaperScissors, (2) Hyper-Q agent uses a computationally hard Bayesian belief update operation at each time step and (3) the game of total observability becomes partially observable because the beliefs about other player's strategies are represented in the form of probability distribution over all possible mixed strategies.

On the contrary, we propose here a much simpler (in terms of the amount of computation required per iteration) adaptive dynamics learning algorithm called Adaptive Dynamics Learner (ADL) which associates a Q-value to each experienced game history H of fixed length p and a simple action $a^j \in \mathcal{A}^j$, and then learns these Q-values using a form of Q-learning. This substantially reduces the space of virtual states and actions comparatively to Hyper-Q approach. More formally, ADL player j maintains a table H^j of histories, considered by it as the system's states. To each history $h^j \in H^j$, it associates a Q-value of the form $Q^j(h^j, a^j) \quad \forall a^j \in \mathcal{A}^j$. Being at time step t in the state $h^j_t = \langle a^j_{t-p}\mathbf{a}^{-j}_{t-p}a^j_{t-p+1}\mathbf{a}^{-j}_{t-p+1}\ldots a^j_{t-1}\mathbf{a}^{-j}_{t-1}\rangle$, the agent j searches in H^j the action a^j_t, which maximizes the Q-values for h^j_t. After that the agent j plays a^j_t with some exploration decreasing over time. Having observed the opponents' joint action and its own reward, it updates its state at time $t+1$ by concatenating its own action a^j_t and the opponents' joint-action a^{-j}_t played at time t to h^j_t and by eliminating two very first entries, that is,

$$h^j_{t+1} = \langle a^j_{t-p+1}\mathbf{a}^{-j}_{t-p+1}a^j_{t-p+2}\mathbf{a}^{-j}_{t-p+2}\ldots a^j_t\mathbf{a}^{-j}_t\rangle \qquad (2)$$

Finally, the player j updates the Q-value in h^j_t corresponding to the action a^j_t by using the following Q-learning update rule:

$$\begin{aligned} Q^j(h^j_t, a^j_t) \;&\leftarrow\; (1-\alpha)Q^j(h^j_t, a^j_t) \\ &\quad + \alpha[R^j(h^j_t, \langle a^j_t, a^{-j}_t\rangle) + \gamma U^j(h^j_{t+1})] \quad (3) \end{aligned}$$

where $U^j(h^j_{t+1}) = \max_{a^j\in\mathcal{A}^j} Q^j(h^j_{t+1}, a^j)$ and $R^j(\cdot)$ is the j's reward after playing a^j_t in the previous state h^j_t. The ADL algorithm is formally defined in Algorithm 1.

```
Require: Maximum history length $p$, Maximum time $t_{max}$
 1: Current time $t \leftarrow 0$
 2: Current state $h^j_t \leftarrow EmptySequence$
 3: $a^j_t \leftarrow RandomAction$
 4: while $t \le t_{max}$ do
 5:    Play $a^j_t$ with some decreasing exploration
 6:    Observe the reward $R^j_t$ and the joint-action $\mathbf{a}^{-j}_t$
 7:    Obtain new state $h^j_{t+1}$ using (2) with $h^j_t$, $a^j_t$ and $\mathbf{a}^{-j}_t$
 8:    Find the action $a^j_{t+1}$ maximizing $Q$-values in $h^j_{t+1}$ and
       calculate the utility $U^j(h^j_{t+1})$
 9:    Update $Q^j(h^j_t, a^j_t)$ using equation (3) with $R^j_t$ and
       $U^j(h^j_{t+1})$
10:    Update current state $h^j_t \leftarrow h^j_{t+1}$
11:    Save the action to play $a^j_t \leftarrow a^j_{t+1}$
12:    Increment the time $t \leftarrow t+1$
13: return
```

Algorithm 1: Adaptive Dynamics Learner for player j

4. IMPLEMENTATION AND RESULTS

To test our algorithm, we programmed two adaptive learning algorithms, IGA and APQ. We did not test ADL against PHC (which is, in fact, an adaptive player as well) because it has less knowledge about the world (it does not perceive the opponent's actions, for example), so such a comparison would be incorrect. As shown in [10] and [2], the lack of such a knowledge can be readily exploited by the more informed opponents. In the following subsections, we provide the implementation details for each of the programmed algorithms, including ADL. Then we present the testing results obtained in the games from GAMUT [5], a game theoretic test suite.

4.1 Adaptive Play Q-learning Agent

The APQ agent we used has the following characteristics. The length of the history, p, is 16, the size of sampling, l, is 8, the discount factor, γ, is 0.9, the learning rate, α, is proper for each state-action pair and decreases gradually using the *search-then-converge* schedule depending on the number of updates of the respective Q-value:

$$\alpha_t(h^j, a^j) = \frac{\alpha_0 \tau}{\tau + n_t(h^j, a^j)}$$

where t is the current time step, α_0 is the initial value of the learning rate and $n_t(h^j, a^j)$ is the number of times that the Q-value for the action a^j was updated in state h^j to time t. We set $\alpha_0 = 0.5$ and $\tau = 10,000$.

4.2 Infinitesimal Gradient Ascent Agent

IGA supposes omniscient knowledge by the agents of the mixed strategies of their opponents. However, neither ADL nor APQ agents explicitly play a mixed strategy. On the other hand, their strategies, being pure for them, are mixed for IGA player as soon as they do not act in the same internal state space. Indeed, the current internal states of each agent (counterparts' actions history of APQ, concatenated joint actions of ADL and opponents' current mixed strategies of IGA) are different, though the current external state (the game played) is the same. To permit IGA agent to estimate the strategy of its opponents, we implemented two most used techniques: (i) Adaptive Play's technique and (ii) Exponential Moving Average (EMA). We described Adaptive Play's technique in Subsection 2.1 when we presented APQ algorithm. It consists in using equation (1) to estimate the probability distribution. EMA is a family of similar statistical techniques used to analyze time series data in finance and technical analysis. Typically, EMA assumes that the recent observations are more informative than the older ones, and, thus, as applied to our problem, given a new observed joint action $\mathbf{a}_t^{-j}$, IGA agent j updates a probability distribution $\mathbf{\Pi}_t^{-j}$, represented as a vector, as follows:

$$\mathbf{\Pi}_{t+1}^{-j} \leftarrow (1-\mu)\mathbf{\Pi}_t^{-j} + \mu \mathbf{u}(\mathbf{a}_t^{-j})$$

where $\mathbf{u}(\mathbf{a}_t^{-j})$ is a unit vector representation of the action $\mathbf{a}_t^{-j}$ observed at time t and μ is a small constant, $0 < \mu \ll 1$.

We observed that in almost all runs, IGA agent that used Adaptive Play's estimation technique of probability estimation was more efficient than one that used EMA, therefore in our figures we will only present the results for Adaptive Play's technique based IGA agent.

4.3 Adaptive Dynamics Learner

The only interesting parameter of ADL is p, the history length. What is the length that will permit ADL agent to learn well the dynamics of an opponent? Is there a universal value or it should be adjusted for each opponent individually? Our experiments showed that in many cases the history of the length 2 (that is, the only most recent joint action!) was sufficient to outperform the adaptive agents in adversarial games, but the value of 6 (three most recent actions) was sufficient to perform well regardless of the game played. Thus, in our experiments we set $p = 6$, but the question of how to efficiently determine the best value of p remains open.

4.4 Results

As mentioned above, we tested our algorithm in play versus the other two algorithms and in self-play on a set of games from the GAMUT test suite. First, we examined the behavior of ADL against APQ player (Figure 1). To do that, we observed the evolution of average Bellman error, which is the difference between two successive updates of Q-values, and the changes in the average reward per play[4] (the rewards were averaged over each $10,000$ plays). It is easy to see that the process exhibited good progress toward the convergence, as suggested by progressive reducing of average Bellman error (Figure 1(a)) and substantial positive average reward per time step (Figure 1(b)).

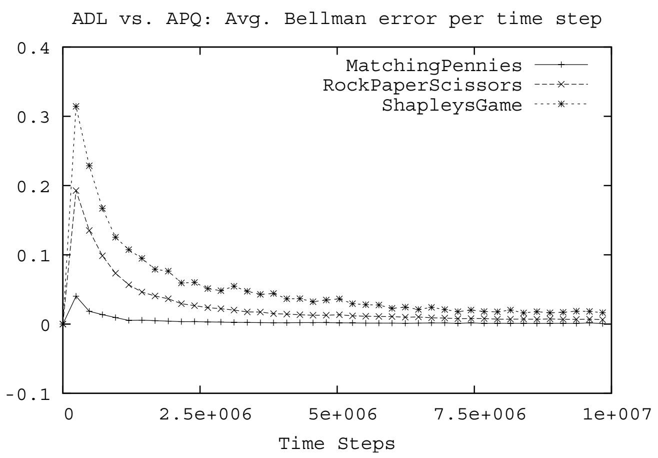
Further, we examined ADL versus IGA player. ADL showed better performance against this opponent, as is seen from the average reward diagram where the converged values are higher than the analogical values obtained in play vs. APQ agent. The average Bellman error decreased slower in that case, which is explained by the fact that IGA is able to play mixed strategies unlike APQ player, which converged directly to a policy in pure strategies.

Finally, we verified whether ADL, by being efficient against the adaptive opponents, remains rational in play against itself (so called self-play) and versus other types of opponents that do not evolve in time and follow a stationary mixed strategy. As for the stationary opponents, we tested the behavior of ADL against mixed strategy players on the example of RockPaperScissors game. Let $Random(x, y, z)$ denote a player playing a mixed strategy where there are nonnegative probabilities of x, y and z, $x+y+z = 1$, that the actions 1, 2 and 3 respectively will be played by that player. As expected, ADL player had a positive average reward close to zero against the $Random(0.33, 0.33, 0.34)$ opponent which played a strategy close to the Nash equilibrium $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, but it performed better as the random players were more distant from the equilibrium: it converged to the average reward of 0.02 against $Random(0.3, 0.3, 0.4)$, to the average reward of 0.25 against $Random(0.25, 0.25, 0.5)$ and to the average reward of 0.4 versus $Random(0.2, 0.2, 0.6)$. Thus, an important conclusion is that ADL algorithm remains rational in that case.

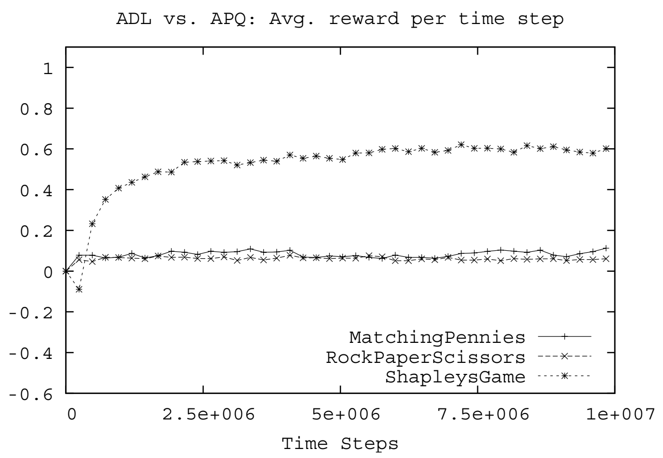
The dynamics of the ADL's self-play in adversarial games is not interesting enough to include it in the paper; as expected, the agents, by being rational, converged to a Nash equilibrium, which brings a zero average reward to both players. The most notable observation is that in games with a Pareto optimal strategy[5] that is not an equilibrium (such as PrisonersDilemma), ADL converges in self-play to a Pareto optimal solution, while the self-played adaptive algorithms may converge to an equilibrium, which is not favorable for both players. Furthermore, if there are two outcomes in a game, one of which is more favorable for the first player and the another one brings higher utility for the second player (such as in GrabTheDollar), then the average rewards obtained by both ADL players in self-play are a mean of these two equilibrium rewards, i.e., the welfare of both is maximized. The adaptive players are able to find just one of these equilibria, thus, as soon as an equilibrium is found, one of the agents will always have a lower utility.

[4]Here plays and time steps are equivalent.

[5]A strategy is said to be Pareto optimal if its utility is maximal for all players.

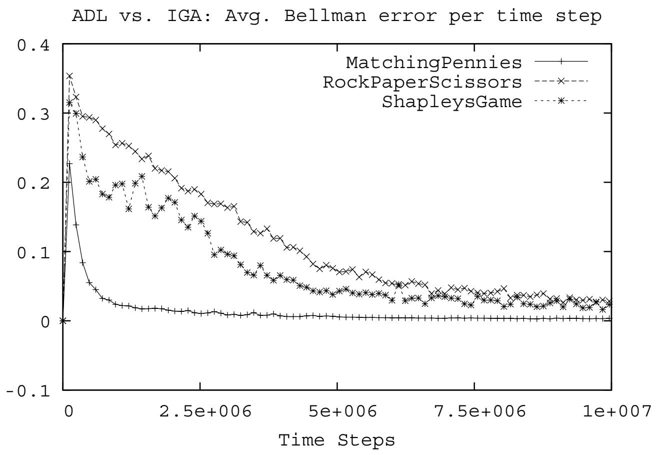


(a)

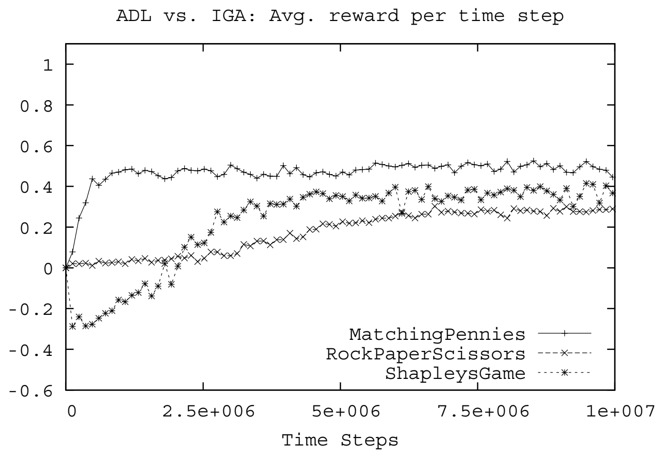


(b)

Figure 1: ADL vs. APQ in the adversarial games.



(a)



(b)

Figure 2: ADL vs. IGA in the adversarial games.

Notice that in ADL's self-play in RandomGame the average reward of the row player is lower than the Nash equilibrium due to the structure of the game that is not favorable for the column player and the emerging tendency of ADL to maximize the welfare of both to the prejudice of its personal utility. One more interesting observation is that in play with APQ in PrisonersDilemma, as well as in self-play, the ADL algorithm also converged to the Pareto optimal solution instead of the Nash equilibrium solution.

In our comparative analysis, we presented the dynamics of the learning process for three adversarial games only because, in our opinion, the adversarial case is the most interesting one. In fact, the curves of the learning dynamics in the other games were trivial: in most cases, almost from the beginning, they became straight lines with some minor fluctuations. The *minimal* of the converged values of average reward of the row player over all runs for all games are presented in Figure 3. The bar graphs "ADL vs. IGA" and "ADL vs. APQ" show the reward of ADL, as the row player, in play versus these opponents. The "Max Nash" and "Min Nash" bar graphs reflect respectively the maximal and minimal average rewards per play, which the row player can gain if a Nash equilibrium is played.

5. DISCUSSION

We studied how our approach to multiagent learning is situated among the other recent algorithms. There have been many new algorithms proposed in the last five years and there is still no common idea to which direction multiagent learning should progress. Three years later after the publication of critical survey of the state-of-the-art multiagent learning trend by Shoham and colleagues [8], where the authors asked about the *question* the researchers should aim at when speaking about multiagent learning, there is still no such question. We would emphasize two common directions of the modern research: (1) heuristic agent composition [7, 6] and (2) direct dynamics learning and exploiting [2, 10]. The main and common traits of the modern research are (i) a classification of algorithms relatively to a class of counterparts, against which they would perform well and (ii) a definition of a lower bound of the utility, which is guaranteed to be reached by that algorithm against an unknown class of opponents. Our algorithm relates to the second direction and possesses the first trait. In our opinion, however, there

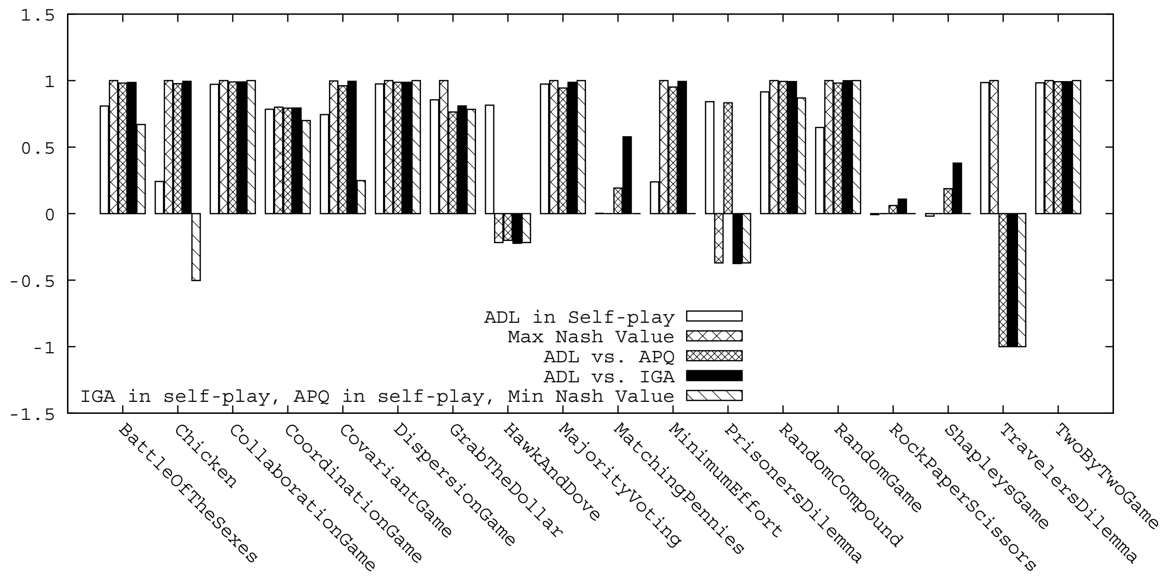


Figure 3: ADL, APQ and IGA players over the games from GAMUT.

should be a more general approach, which is an issue for further investigation.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach to multiagent learning in adaptive dynamic systems. An adaptive dynamic system may be viewed as a two-player game where a goal of a player is to maximize its own long-term utility given that the other agents (considered as one whole agent) may follow an adaptive learning strategy. Our algorithm, called ADL (for Adaptive Dynamics Learner), by interacting with the opponent player, learns the Q-values of internal virtual states that are formed as an ordered set of joint-actions of the past plays of limited-length history. We empirically showed that our algorithm outperforms IGA [9] and APQ [4] algorithms even if a very short history length was used to form the states. While being more general than heuristically composed algorithms, such as Manipulator [6] or MetaStrategy [7], our approach is much simpler (in terms of the amount of computation required in each iteration) than the analogical Hyper-Q algorithm [10] and, thus, is expected to be better scalable. It is also able to maximize the welfare of both players in self-play. Several untested adaptive opponents and certain non-stationary opponents still remain to be compared and we will give attention to that, however considerably more research is needed to develop a theoretical analysis of our approach.

7. REFERENCES

[1] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[2] Y. Chang and L. Kaelbling. Playing is believing: The role of beliefs in multi-agent learning. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'01)*, Canada, 2001.

[3] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*, Menlo Park, CA, 1998. AAAI Press.

[4] O. Gies and B. Chaib-draa. Apprentissage de la coordination multiagent : une méthode basée sur le Q-learning par jeu adaptatif. *Revue d'Intelligence Artificielle*, 20(2-3):385–412, 2006.

[5] E. Nudelman, J. Wortman, K. Leyton-Brown, and Y. Shoham. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of Autonomous Agents and Multiagent Systems (AAMAS'04)*, 2004.

[6] R. Powers and Y. Shoham. Learning against opponents with bounded memory. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*, 2005.

[7] R. Powers and Y. Shoham. New criteria and a new algorithm for learning in multi-agent systems. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2005.

[8] Y. Shoham, R. Powers, and T. Grenager. Multi-agent reinforcement learning: a critical survey. Technical report, Stanford University, 2003.

[9] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI'94)*, pages 541–548, San Francisco, CA, 1994. Morgan Kaufman.

[10] G. Tesauro. Extending Q-learning to general adaptive multi-agent systems. In S. Thrun, L. Saul, and B. Scholkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, Cambridge, MA, 2004. MIT Press.

[11] H. Young. The evolution of conventions. *Econometrica*, 61(1):57–84, 1993.