Multi-agent Learning Dynamics: A Survey

H. Jaap van den Herik, D. Hennes, M. Kaisers, K. Tuyls*, and K. Verbeeck

Adaptive Agents Group, MICC, Maastricht University, The Netherlands
k.tuyls@micc.unimaas.nl

Abstract. In this paper we compare state-of-the-art multi-agent reinforcement learning algorithms in a wide variety of games. We consider two types of algorithms: value iteration and policy iteration. Four characteristics are studied: initial conditions, parameter settings, convergence speed, and local versus global convergence. Global convergence is still difficult to achieve in practice, despite existing theoretical guarantees. Multiple visualizations are included to provide a comprehensive insight into the learning dynamics.

1 Introduction

This article surveys the dynamics and performance of state-of-the-art value iteration and policy iteration reinforcement learning algorithms in multi-agent games. In particular this work studies initial conditions, parameter settings, convergence speed, and local versus global convergence in a wide variety of cooperative and competitive games.

Single-agent reinforcement learning (RL) has been studied extensively in the past [3,13]. It guarantees convergence to the optimal policy assuming sufficient learning cycles and a stationary environment.

Learning and adaptation in a multi-agent context recently has gained a great deal of interest in the Artificial Intelligence research community [1,6,10,11,12,16,4,17,20]. Accomplishing a certain task in highly uncertain environments, in which multiple agents operate, calls for multi-agent learning techniques. These agents involved are not only situated in a non-stationary environment but also need to deal with incomplete information and communication limits. In such non-stationary environments the Markov property does not hold which makes all proofs of convergence inapplicable when considering algorithms from single-agent learning based on the Markov assumption.

Reinforcement learning techniques are subdivided in value iteration and policy iteration. Q-learning and learning automata are examples of each class respectively. Value-based learners estimate a state-action value function that determines the utility of performing a given action in a given state [13]. Once the outcome is established the value function is used to derive a policy that describes the behavior of the agent. Contrary to the value based approach, policy iterators as learning automata learn directly in the policy space.

* Corresponding author.

M. Klusch et al. (Eds.): CIA 2007, LNAI 4676, pp. 36–56, 2007.

The remainder of this paper is organized as follows. Section 2 introduces the state-of-the-art learning algorithms. Section 3 explains the wide variety of games on which the algortihms are tested and concisely explains concepts such as Nash equilibrium and Pareto optimality. We continue in Section 4 with an elaboration on the performance criteria and the method of visualizing the learning dynamics. Section 5 covers the obtained results. A discussion and future research opportunities follow in Section 6. Section 7 concludes the article.

2 State-of-the-Art Learning Algorithms

In this section we shortly describe two different multi-agent reinforcement learning algorithms, viz. value based learners and policy based learners.

The value iteration reinforcement learning algorithms considered are Q-learning and two recent adaptations, i.e., Lenient Q-learning and FMQ-learning [4,7,8]. The policy iteration algorithms considered are Learning Automata and a number of its variants [6,15]. In particular finite action-set learning automata (FALA) and parameterized learning automata (PLA) are studied.

2.1 Value-Based Learners

We start by explaining independent Q-learning, because this is the basis of state-of-the-art value-based algorithms.

Q-Learning. Q-learning was initially introduced for single-agent environments. Each learning step refines a utility-estimation function (the value function) for state-action pairs and generates a new policy from the estimated values to draw the next action to execute. The algorithm bootstraps its estimate for the state-action value $Q_{t+1}(s,a)$ at time $t+1$ upon its estimate for $Q_t(s^{'},a^{'})$ with $s^{'}$ the state where the learner arrives after taking action a in state s:

$$Q_{t+1}(s,a) \leftarrow (1-\alpha)Q_t(s,a) + \alpha(r + \gamma \; max_{a'} Q_t(s^{'},a^{'})) \tag{1}$$

with α the usual step size parameter, γ a discount factor and r the immediate reinforcement.

In the single agent case this algorithm is able to learn optimal behavior in stationary environments [19]. The choice of the action selection mechanism is of utmost importance: it generates actions given the estimated state-action values. An ϵ-greedy action selection assigns the best action with probability $(1-\epsilon)$ and some random action with probability ϵ.

With the Boltzmann distributed exploration mechanism, an action is selected with a probability given by:

$$p_j = \frac{e^{Q_i(s_j)\cdot\tau^{-1}}}{\sum_k e^{Q_i(s_k)\cdot\tau^{-1}}}$$

where an initially high temperature τ promotes exploration and decreasing temperature over time leads to strong exploitation in the final phase.

iteration t					
T	10	-10	-	-	-
M	0	**7**	0	6	0
B	5	0	0	-	-

iteration t + 1					
T	10	-10	-	-	-
M	-	-	-	-	-
B	5	0	0	-	-

Fig. 1. Lenient Q-learning reward register for $L = 5$, example from CG. Q-value of M is updated with $r_i = 7$, maximum of five rewards (left). The next step clears the register (right).

Convergence guarantees are usually lost in multi-agent environments, since other agents' actions can make the environment appear as non-stationary from the viewpoint of a single agent. Still it is possible and sometimes even useful, as we will show later on, to use Q-learning in a multi-agent environment. Agents are called independent when it assumed that they can neither observe the other agents actions nor the rewards they received for them; the agents only act upon the experience collected by experimenting with the environment.

Lenient Q-Learning. In a cooperative multi-agent learning environment it is a good idea to forgive mistakes, especially in the initial learning period. Consider the example of learning in soccer as in [7]. In the initial phase of learning both agents lack the skill for good actions, so even a perfect forward pass may frequently be not rewarded. This leads to the agents converging to actions that work well with a variety of opponents' strategies but it often results in suboptimal behavior.

In order to handle this problem lenient Q-learning collects L rewards for an action before it updates the estimation based on the maximum. Lower rewards are discarded and only the highest reward is used for the update which implies that only $\frac{1}{L} \cdot iterations$ learning steps are executed. Figure 1 depicts the update schematically.

For a detailed description of this algorithm we refer to [7,8].

FMQ-Learning. The FMQ - learner keeps track of the highest reward for each action and its frequency so far [4]. It is used to alter the policy generation which is not based on the Q-values anymore, but on the function $ev(Q_i(s_j))$. Let F be the parameter that describes the persistence to seek the maximal encountered reward $r_i^*(s_j)$ that was observed with frequency $f_i(s_j)$ so far.

$$ev(Q_i(s_j)) = Q_i(s_j) + F \cdot f(s_j) \cdot r_i^*(s_j)$$

The higher F the more the algorithm will alter the policy. This variation works best in combination with another FMQ learner; the ideas is that policies quickly agree on an optimum even if it is surrounded by penalties as it is in the climbing game. If F is large it enforces a quick decision for one action.

2.2 Policy-Based Learners

Rather than building estimated values for states or state-action pairs, policy-based reinforcement learners directly search the policy space for the optimal policy. The two policy-based learners that are considerd in this study are both learning automata algorithms, i.e. *finite action-set learning automata* (FALA) and *parameterized learning automata* (PLA). Both are model free, stateless and independent learners. While these restrictions are not negligible, they allow for simple algorithms that can be discussed analytically. Convergence for learning automata in single and specific multi-agent cases, such as games, has been proved in [5].

Finite Action-Set Learning Automata. The class of finite action-set learning automata (FALA) considers only automata that optimize their policies over a finite action-set $A = \{1, \dots, k\}$ with k some finite integer. One optimization step, called *epoch* from here on, is divided into two steps: action selection and policy update. At the beginning of an epoch t the automaton draws a random action $a(t)$ according to the probability distribution $\pi(t)$, called policy. Based on the action $a(t)$ the environment responds with a reinforcement signal $r(t)$, called reward. Hereafter the automaton uses the reward $r(t)$ to update $\pi(t)$ to the new policy $\pi(t+1)$.

The update rule for FALA is given below.

$$\begin{aligned}
&\text{If } i = a(t) \text{ then}\\
&\pi_i(t+1) = \pi_i(t) + \alpha r(t)(1 - \pi_i(t))\\
&\qquad\qquad - \beta(1 - r(t))\pi_i(t)\\
&\text{otherwise}\\
&\pi_i(t+1) = \pi_i(t) - \alpha r(t)\pi_i(t)\\
&\qquad\qquad + \beta(1 - r(t))[(k-1)^{-1} - p_i(t)]
\end{aligned} \tag{2}$$

Here α and β are in $[0,1]$ are the reward and penalty parameters respectively. Depending on α and β, the update scheme is referred to as *linear reward-penalty (L_{R-P})* if $\alpha = \beta$, for $\beta = 0$ it is called *linear reward-inaction (L_{R-I})*, and if β is chosen to be small compared to α it is called *linear reward-ϵ-penalty ($L_{R-\epsilon P}$)*.

Assuming that r is continuous (called *S-model*[5]) and in the range $[0,1]$, (2) does indeed give a probability distribution satisfying the following two constraints: $\sum_{i=1}^{k} \pi_i(t+1) = 1$ and $\forall i \; \pi_i(t+1) \in [0,1]$.

Parameterized Learning Automata. A learning automaton following the update rule given in (2) is only guaranteed to converge locally [15]. In order to find the global optima the learning algorithm has to be refined.

One solution to this problem is adding a randomization term to the learning rule. Superimposing noise directly on the probability distribution would violate the constraints. Therefore the algorithm presented in [14] uses a probability generating function g mapping an internal state vector u to a valid probability distribution π. This class of algorithms is called parameterized learning automata (PLA). The update rule from [14] for PLA simplifies to

$$\begin{gathered} u_i\left(t+1\right)=u_i\left(t\right)+\alpha r\left(t\right)\frac{\delta \ln g}{\delta u_i}\left(u\left(t\right),a\left(t\right)\right) \\ +\alpha h'\left(u_i\left(t\right)\right)+\sqrt{\alpha}s_i\left(t\right) \end{gathered} \tag{3}$$

where

$$h\left(x\right)=\begin{cases} -K\left(x-L\right)^{2n} & x\geq L \\ 0 & \left|x\right|\leq L \\ -K\left(x+L\right)^{2n} & x\leq -L \end{cases} \tag{4}$$

and $h'(x)=\frac{\delta h(x)}{\delta x}$. Furthermore α is a positive learning parameter and $s_i\left(k\right)$ is a set of IID random variables drawn from a normal distribution with zero mean and variance σ^2.

The difference $u\left(k-1\right)-u\left(k\right)$ is composed of three terms, a gradient, a bound, and a random term. The gradient term includes the probability generating function given by

$$g\left(u,i\right)=\frac{\exp u_i}{\sum_{j=1}^{k}\exp u_j} \tag{5}$$

According to π an action $a(t)$ is selected at every epoch t. The second term uses function h to ensure that the state vector remains within the bound $|u|\leq L$. Constants L, K, and n are all positive; L and K are real values whereas n is an integer. The last term superimposes noise to prevent the algorithm from getting stuck in local optima.

Next, Definition (5) is used to work out the gradient in (3):

$$\begin{aligned} \frac{\delta \ln g}{\delta u_i}\left(u\left(t\right),a\left(t\right)\right) &= \frac{\delta}{\delta u_i}\ln\left(\frac{\exp u_{a(t)}\left(t\right)}{\sum_{j=1}^{k}\exp u_j\left(t\right)}\right) \\ &= \frac{\delta}{\delta u_i}\left(u_{a(t)}\left(t\right)-\ln\left(\sum_{j=1}^{k}\exp u_j\left(t\right)\right)\right) \\ &= \frac{\delta u_{a(t)}}{\delta u_i}-\frac{\exp u_i\left(t\right)}{\sum_{j=1}^{k}\exp u_j\left(t\right)} \\ &= \frac{\delta u_{a(t)}}{\delta u_i}-\pi_i\left(t\right) \\ &= \begin{cases} 1-\pi_i\left(t\right)\, if\ i=a\left(t\right) \\ \quad -\pi_i\left(t\right)\, otherwise \end{cases} \end{aligned}$$

This results in the following update rule, similar to the form seen in (2) for the (L_{R-I}) scheme:

$$\text{If } i = a(t) \text{ then}$$

$$\begin{aligned} u_i(t+1) &= u_i(t) + \alpha r(t)(1 - \pi_i(t)) \\ &\quad + \alpha h' + \sqrt{\alpha} s_i(t) \end{aligned}$$

$$\text{otherwise} \tag{6}$$

$$\begin{aligned} u_i(t+1) &= u_i(t) - \alpha r(t)\pi_i(t) \\ &\quad + \alpha h' + \sqrt{\alpha} s_i(t) \end{aligned}$$

3 Testbed of Games

This section provides background information on multi-agent games used as a benchmark for multi-agent reinforcement learning. Starting with a brief introduction to games, Section 3.1 concisely explains solution concepts, namely Nash equilibrium, Pareto efficiency, and maximum social welfare. Examples of various games for two and more players are provided through Sections 3.2 to 3.4.

3.1 Games

Normal form games are stateless games that make the assumption that players act simultaneously. Each player i participating in the game has a set of actions A^i available. When all agents have played an action they receive a numerical reward r^i.

Since normal form games are stateless, the behavior of player i can be described by a single probability distribution π^i over its action-set A^i. This distribution is called a strategy or policy. If $\pi^i_j = 1$ for any $j \in A^i$ then player i follows a *pure strategy* otherwise a *mixed strategy*. Furthermore let $R^i(\pi^1, \ldots, \pi^n)$ be the expectation of payoff r^i for agent i given the strategies $\pi^1, \ldots, \pi^n$.

Based on the notion of the expected payoff R and the strategy profile $\pi = (\pi^1, \ldots, \pi^n)$ this section gives a formal definition of Nash equilibrium, Pareto efficiency and maximum social welfare profiles.

Definition 1: *Nash equilibrium*
A strategy profile $\pi = (\pi^1, \ldots, \pi^n)$ is a Nash equilibrium if for all players i the following condition holds.

$$R^i(\pi) \geq R^i(\pi^1, \ldots, \pi^{i-1}, \tilde{\pi}^i, \pi^{i+1}, \ldots, \pi^n) \ \forall \tilde{\pi}^i$$

Hence no player can improve its payoff by exclusively changing its strategy to some $\tilde{\pi}$ given fixed strategies for all other agents.

Definition 2: *Pareto efficiency*
A strategy profile π is Pareto efficient (or Pareto optimal) if there is no $\tilde{\pi} \neq \pi$ such that

$$R^i(\tilde{\pi}) \geq R^i(\pi) \ \forall i$$

and for some i

$$R^i(\tilde{\pi}) > R^i(\pi).$$

Thus a Pareto efficient solution implies that no player can improve its expected payoff without making at least one other player worse off.

Definition 3: *Maximum social welfare profile*
The social welfare of an interactive situation is defined by the sum of individual rewards. Hence, the maximum social welfare can be denoted by:

$$\max_{\pi} \ \omega(\pi) = \max_{\pi} \ \sum_{i=1}^{n} R^i(\pi)$$

Furthermore, the strategy profile π^* with

$$\pi^* = \arg\max_{\pi} \ \omega(\pi)$$

is called a *maximum social welfare profile*. In cooperative games Pareto efficient solutions and maximum social welfare profiles are of major interest, whereas in competitive situations Nash equilibria are studied.

The next subsections introduce seven normal form games.

3.2 2 x 2 Matrix Games

Normal form games with two players each choosing from two actions are called 2 x 2 matrix games. The family of 2 x 2 games can be subdivided into three categories according to there payoff matrices [9]: (a) games with one pure equilibrium, (b) games with one mixed equilibrium and (c) games with two pure equilibria and one mixed equilibrium. This subsection presents one example for each class.

The *Prisoners' Dilemma* (PD) is a well studied category (a) game in which the players may *confess* (C) or *deny* (D) [2]. The payoff matrix of the PD game is given below. The single pure Nash equilibrium is located in the bottom-right corner, corresponding to both players playing action C.

	D	C
D	3,3	0,5
C	5,0	1,1

Matching Pennies (MP) is a 2 x 2 game belonging to category (b) and defined by the following payoff matrix:

	H	T
H	1,-1	-1, 1
T	-1, 1	1,-1

Both players chose simultaneously for one side of a penny, either they play *Head* (H) or *Tail* (T). If both pennies show the same face player 1 keeps the coins; for a mismatch player 2 gets rewarded. The mixed equilibrium is reached if both players play strategies $(0.5, 0.5)$ which means that a player selects action H and T each with probability 0.5.

Category (c) is covered by the next example, the *Bach or Stravinsky* (BoS) game, also referred to as the Battle of the Sexes. In this strategic situation the players want to visit a concert together. They can chose between *Bach* (B) or *Stravinsky* (S) but no communication is allowed. Player 1 prefers B whereas player 2 has a preference for S. Strategies corresponding to the joint action pairs (B, B) and (S, S) form pure equilibria; the mixed equilibrium is defined by the strategies $(\frac{2}{3}, \frac{1}{3})$ and $(\frac{1}{3}, \frac{2}{3})$ for player 1 and 2 respectively. Miscoordination results in a zero payoff for both. The payoff matrix of the BoS game is given below.

	B	S
B	2,1	0,0
S	0,0	1,2

3.3 Penalty and Climbing Game

Games with more than one equilibrium are studied to investigate convergence to local or global optima. The BoS game contains multiple pure equilibria, however it is not possible to point out the best due to a conflicting interest. The mixed equilibrium is a fair solution but not in the least optimal with respect to social welfare. Therefore games with equal payoff for both players are studied; these games are called symmetric games. The *penalty game* and the *climbing game* are examples of symmetric games. Payoff matrices for both games are given in Figure 2.

In the penalty game players have to coordinate their actions in order to yield high payoffs (joint actions $(1, 1)$ and $(3, 3)$). Miscoordination leads to punishment by negative rewards. Furthermore the joint action $(2, 2)$ is also an equilibrium but not Pareto efficient.

Two equilibria can be found in the climbing game. Joint actions corresponding to positive non-zero payoffs are points of attraction, where the joint actions $(1, 1)$ and $(2, 2)$ form equilibria. Once again the Pareto optimal Nash equilibrium is surrounded by negative rewards to punish miscoordination. Learners have to virtually climb up to reach the maximum reward.

$$\begin{bmatrix} 10 & 0 & -10 \\ 0 & 2 & 0 \\ -10 & 0 & 10 \end{bmatrix} \begin{bmatrix} 11 & -10 & 0 \\ -10 & 7 & 6 \\ 0 & 0 & 5 \end{bmatrix}$$

Fig. 2. Payoff matrices for the penalty (left) and the climbing game (right)

3.4 Guessing and Dispersion Game

So far the introduced games cover interactions between two players. Since MAS with only two agents are barely seen in practice this subsection defines symmetric games with n players and n actions where n can be any finite integer.

The *guessing game* is well suited for generalization up to n players. Each player i selects an action a^i from the same action set $A = \{1, \dots, n\}$ synchronously. If all players 'guess' the same action the reward will be maximal; if a player exclusively chooses for an action his reward will be minimal. This game is called a coordination game; all players have to coordinate, to 'guess', the same action in order to achieve the maximum payoff.

Closely related is the *dispersion game* also called anti-coordination game. The goal in this particular game is to disperse over the entire action set as much as possible. Just like in the previous game an action set of size n is assumed; which means the maximal dispersion outcome (MDO) is reached if all players choose for sole actions.

Since these two games can be easily scaled up to more than just two players it is more convenient to use a payoff function instead of matrices. Based on the sum of players selecting the same actions these functions are given in (7) and (8) for the guessing game and the dispersion game respectively.

The number of players selecting action j is defined as

$$S(j) = \sum_{i=1}^{n} id(a^i, j)$$

where

$$id(i, j) = \begin{cases} 1 \; if \;\; i = j \\ 0 \; otherwise \end{cases}.$$

Thus the payoff function of the guessing game can be denoted by

$$r^i = \frac{S(a^i)}{n}. \tag{7}$$

For the dispersion game the following payoff function applies:

$$r^i = \begin{cases} 1 \; if \;\; S(a^i) = 1 \\ 0 \; otherwise \end{cases} \tag{8}$$

The payoff differs between the case where one player occupies an action slot alonw (payoff equals 1) and the situation where at least two players have selected the same action (payoff equals 0). Note that the reward $r^i \in [0, 1]$ for all agents $i = 1, \dots, n$.

4 Methodology

In order to explore the performance of a team of learners to its full extent it must be studied under various initial conditions and parameter profiles. The following four subsections present the different means that we will use to approach the given task in Section 5.

4.1 Policy Trajectory Plot

A rather simple way of displaying evolving policies are trajectory plots. In a 2 x 2 game $\pi_2^1 = 1 - \pi_1^1$ and $\pi_2^2 = 1 - \pi_1^2$. Therefore the strategy profile $\pi = \left(\pi^1, \pi^2\right)$ can be reduced to the pair $\left(\pi_1^1, \pi_1^2\right)$ without losing information. The trajectory of this pair is recorded during one single or multiple runs and plotted in a 2D space. To indicate the direction of convergence grayscales are used for trajectory plots in Section 5. With increasing number of epochs t the brightness of the trajectory changes from light to dark.

For the penalty and the climbing game the cardinality of the action-sets equals 3 for both agents. Therefore using the same transformation as above the strategy profile can only be reduced to a 4-tuple and is not displayable in the 2D space. However, the policy trajectories $\pi^1(t)$ and $\pi^2(t)$ can be plotted separately using two simplex plots. The three vertices of a simplex correspond to the pure policies $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$.

4.2 Directional Field Plot

A second visual method to analyze learning dynamics are directional field plots. Again a reduced strategy profile (see Subsection 4.1) is used for analyzing 2 x 2 games. A team of learners start at regular grid points over $[0,1]^2$:

$$\left(\pi_1^1\left(t_0\right), \pi_1^2\left(t_0\right)\right) \in [0,1] \times [0,1]$$

The velocity field of this team can then be denoted by

$$\frac{d\left(v,u\right)}{dt} = \frac{\left(\pi_1^1\left(t_0 + \Delta t\right) - \pi_1^1\left(t_0\right), \pi_1^2\left(t_0 + \Delta t\right) - \pi_1^2\left(t_0\right)\right)}{\Delta t}$$

where Δt is the number of epochs conducted at every grid point and v and u denote the respective strategies of both players. The velocities are displayed by arrows pointing in the direction of $\frac{d(v,u)}{dt}$. The length of the arrows indicates the absolute value $\|\frac{d(v,u)}{dt}\|$.

4.3 Convergence

While 2 x 2 games can easily be studied by graphical analysis, higher dimensional games like the penalty and the climbing game require analytical means. The third method studies convergence in respect to spatial and temporal measures. To measure spatial convergence a metric over the space of strategy profile needs to be defined. Let

$$d\left(\pi, \gamma\right) = \max_i \ \max_j \left|\pi_j^i - \gamma_j^i\right| \tag{9}$$

be the distance of two strategy profiles π and γ. Then a strategy profile π has converged to an optimal π^* if the distance $d(\pi, \pi^*)$ is less than the threshold ϵ at any point in time from epoch T on.

Definition 4: A strategy profile π is called *ϵ-converged* to π^* in T epochs if the following condition holds:

$$d\left(\pi(t), \pi^*\right) < \epsilon \ \forall t, \ t \geq T \tag{10}$$

Note that (9) is used in favor of other metrics since it applies to multi-agent situations without losing the intuitive explanation of threshold ϵ in (10): If a strategy profile is *ϵ-converged* each single action probability diverges at most ϵ from its desired value for all agents.

By means of *ϵ-convergence*, Section 5 studies the convergence to Nash equilibria and Pareto optimal solutions. Multiple runs are conducted in order to estimate the percentage-wise convergence to different points of attraction and the convergence time T. The experiments are repeated to determine confidence intervals.

4.4 Cumulative Reward Plot

The coordination and anti-coordination games (see Subsection 3.4) are cooperative and therefore the maximum social welfare can be used as a performance threshold. Both payoff functions (7) and (8) for the coordination games yield the same maximum social welfare of value n. If the cumulative reward $\sum_{i=1}^{n} r^i(t) = n$ a maximum social welfare profile is being played in t. Section 5 shows cumulative reward plots to indicate if agents successfully converge to these desired profiles.

5 Results

In this section we present the results obtained with the value-iteration learners and the policy-iteration learners in all three type of games. We summarize all learning performances and for some of the most interesting cases we provide visualizations of the learning dynamics. Concerning Learning Automata, the emphasis is placed on FALA, including various update schemes; the explanations only relate to PLA if the results differ significantly.

Not only successful settings are shown, but also conditions and parameters under which the algorithms may fail to converge optimally. Furthermore, a variety of visualizations allow the reader to receive an intuitive impression of the learning dynamics.

5.1 Simple Games: 2 × 2

Policy-Iteration Learners. Table 1 summarizes the learning performance of FALA in 2 x 2 games. The confidence intervals for mean estimates are obtained by gathering 101 samples each averaging 20 runs of a particular game with $T_{max} = 5\,000$ epochs. Thus T-values in the table approaching T_{max} indicate that the learners have not converged (see (10)).

The L_{R-I} update scheme converges in the PD and BoS game to the pure equilibria but fails to find the mixed one in the MP game. Results for the L_{R-P} confirm the finding that the basin of attraction coincides with the equilibrium in the MP game and is located near the mixed one in the BoS game. However, high values for T indicate that both situations are unstable. Thus it cannot be guaranteed that a team of learners stays in a equilibrium once it is reached. Due to the penalty term the action selection persists stochastic, and the team may jump out of the ϵ-convergence region once in a while.

Table 1. Convergence performance of FALA in 2 x 2 games. 95% confidence intervals for mean estimates of of ϵ-convergence percentage with $\epsilon = 0.1$ and mean convergence time T.

FALA L_{R-I} $\alpha = 0.01$, $\beta = 0$

	Nash eq.	Convergence %	T
PD	$(0, 0)$	$99.4\% \pm 0.3\%$	1468.8 ± 31.3
MP	$(\frac{1}{2}, \frac{1}{2})$	$2.4\% \pm 0.7\%$	4408.7 ± 44.2
BoS	$(0, 0)$	$49.2\% \pm 2.3\%$	559.5 ± 17.3
	$(1, 1)$	$50.8\% \pm 2.3\%$	
	$(\frac{2}{3}, \frac{1}{3})$	$0.0\% \pm 0.0\%$	

FALA L_{R-P} $\alpha = \beta = 0.01$

	Nash eq.	Convergence %	T
PD	$(0, 0)$	$0.0\% \pm 0.0\%$	3895.0 ± 35.9
MP	$(\frac{1}{2}, \frac{1}{2})$	$99.1\% \pm 0.4\%$	4821.1 ± 4.1
BoS	$(0, 0)$	$0.0\% \pm 0.0\%$	4550.4 ± 11.2
	$(1, 1)$	$0.0\% \pm 0.0\%$	
	$(\frac{2}{3}, \frac{1}{3})$	$19.8\% \pm 1.7\%$	

FALA $L_{R-\epsilon P}$ $\alpha = 0.01$, $\beta = 0.001$

	Nash eq.	Convergence %	T
PD	$(0, 0)$	$0.0\% \pm 0.0\%$	942.2 ± 19.7
MP	$(\frac{1}{2}, \frac{1}{2})$	$62.0\% \pm 2.3\%$	4813.0 ± 3.9
BoS	$(0, 0)$	$47.5\% \pm 1.9\%$	652.6 ± 22.8
	$(1, 1)$	$48.4\% \pm 1.9\%$	
	$(\frac{2}{3}, \frac{1}{3})$	$0.0\% \pm 0.0\%$	

FALA $L_{R-\epsilon P}$ $\alpha = 0.01$, $\beta = 0.0001$

	Nash eq.	Convergence %	T
PD	$(0, 0)$	$100.0\% \pm 0.0\%$	1301.9 ± 24.9
MP	$(\frac{1}{2}, \frac{1}{2})$	$7.1\% \pm 1.2\%$	4909.2 ± 2.5
BoS	$(0, 0)$	$50.3\% \pm 2.2\%$	559.7 ± 15.4
	$(1, 1)$	$49.7\% \pm 2.2\%$	
	$(\frac{2}{3}, \frac{1}{3})$	$0.0\% \pm 0.0\%$	

The $L_{R-\epsilon P}$ is a trade-off between the previous schemes. Depending on the ratio of reward factor α and penalty factor β, convergence to pure equilibria or near mixed ones is reached. For the parameter setting $\alpha = 0.01$, $\beta = 0.001$ the learners do not ϵ-converge to the equilibrium in the PD, though the T is comparable small to T_{max}. Considering the corresponding field plot (see Figure 3) it becomes clear that the team does converge but to a point near $(0.2, 0.2)$ which is outside the ϵ-convergence region.

It is worth noting that all results obtained for L_{R-I} and $L_{R-\epsilon P}$ in Figure 3 and Table 1 can be reproduced using PLA with appropriate learning rates and zero temperature for L_{R-I} and small values of σ for $L_{R-\epsilon P}$.

These findings are convincingly illustrated by the dynamics of the learning algorithms in Figure 3. All field plots are rendered using small learning rates and parameters $\Delta t = 10$, $r = 100$ as explained in Subsection 4.2. The plots help to localize basins of attraction and show when these coincide with Nash equilibria. We do not show the dynamics of the PD game.

Value-Iteration Learners. Table 2 summarizes the convergence behavior of the studied Q-learning algorithms in simple games. Confidence intervals are computed from 101 samples that average over 20 runs each. Q-values are initialized to corresponding policies that follow a uniform distribution over the policy space.

Table 2 shows convergence of FMQ and lenient Q to the Nash equilibrium in the PD game. Both learners converge to the Pareto optimal strategy (D, D) for all runs that do not converge to the Nash equilibrium (FMQ 25.4% and lenient Q-learner 14.6%). (D, D) is also the maximum social welfare profile.

The MP game yields one mixed NE where both players mix both actions equally. Figure 4 visualizes the learning behavior of the three learners in the MP.

The Battle of Sexes game yields two pure and one mixed equilibrium. Figure 5 shows the learning dynamics of the three learners in this game. All learners converge to the pure Nash equilibrium under $\tau = 0.1$, but not if the temperature is increased to $\tau = 0.5$.

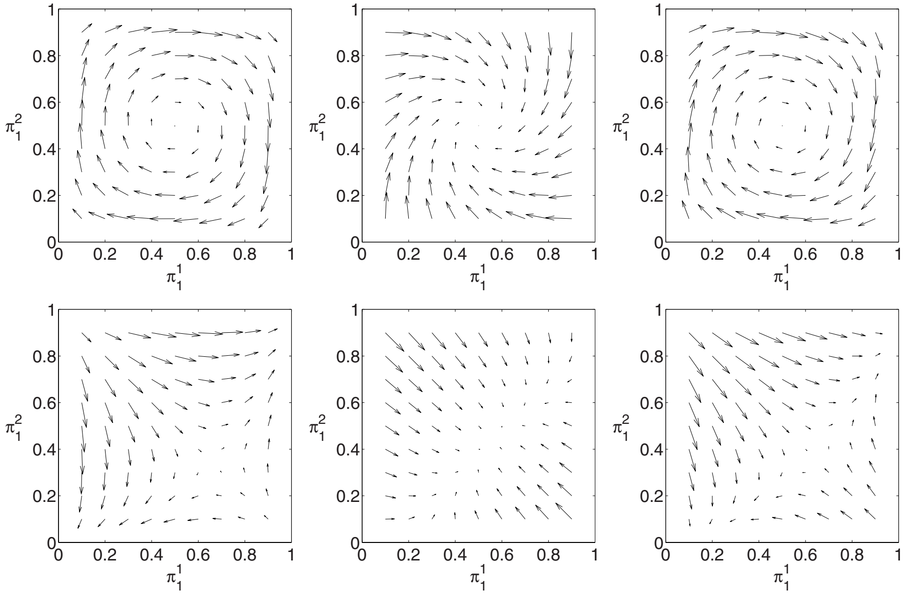

Fig. 3. Overview of directional field plots for FALA in MP (top) and BoS (bottom). Columns correspond to different update schemes; left column: $\alpha = 0.01$, $\beta = 0$ (L_{R-I}), center column: $\alpha = \beta = 0.01$ (L_{R-P}) and right column: $\alpha = 0.01$, $\beta = 0.001$ ($L_{R-\epsilon P}$).

Table 2. ϵ-near convergence with $\epsilon = 0.1$ to equilibria in 2x2 games analyzed after $I = 2000$ iterations. All learners use $\alpha = 0.01$, $\tau = 0.1$ for PD and BoS, $\tau = 0.5$ for PM. Equilibria are given as $(\pi_1(a_{11}), \pi_2(a_{21}))$. Indicated are 95% confidence intervals for convergence percent and mean convergence time.

Q-learner

	NE	Convergence	T
PD	$(0,0)$	99.4% ± 0.4%	1080.1 ± 8.0
MP	$(\frac{1}{2}, \frac{1}{2})$	92.0% ± 1.3%	1862.9 ± 3.2
BoS	$(0,0)$	50.0% ± 2.4%	129.2 ± 2.2
	$(1,1)$	50.0% ± 2.4%	
	$(\frac{2}{3}, \frac{1}{3})$	0.0% ± 0.0%	

FMQ-learner $F = 3$

	NE	Convergence	T
PD	$(0,0)$	74.6% ± 1.9%	5.2 ± 0.6
MP	$(\frac{1}{2}, \frac{1}{2})$	78.7% ± 1.8%	1893.3 ± 2.4
BoS	$(0,0)$	50.6% ± 2.2%	2.5 ± 0.2
	$(1,1)$	49.4% ± 2.2%	
	$(\frac{2}{3}, \frac{1}{3})$	0.0% ± 0.0%	

Lenient Q-learner $L = 3$

	NE	Convergence	T
PD	$(0,0)$	85.4% ± 1.4%	186.3 ± 6.9
MP	$(\frac{1}{2}, \frac{1}{2})$	81.9% ± 1.6%	1547.9 ± 8.8
BoS	$(0,0)$	48.3% ± 2.1%	128.0 ± 4.7
	$(1,1)$	51.7% ± 2.1%	
	$(\frac{2}{3}, \frac{1}{3})$	0.0% ± 0.0%	

5.2 Penalty and Climbing Games

Policy-Iteration Learners. The results for the penalty and the climbing game are summarized by Table 3. The L_{R-P} has not converged to pure policies in these games and is therefore omitted in the overview. Suboptimal convergence for all

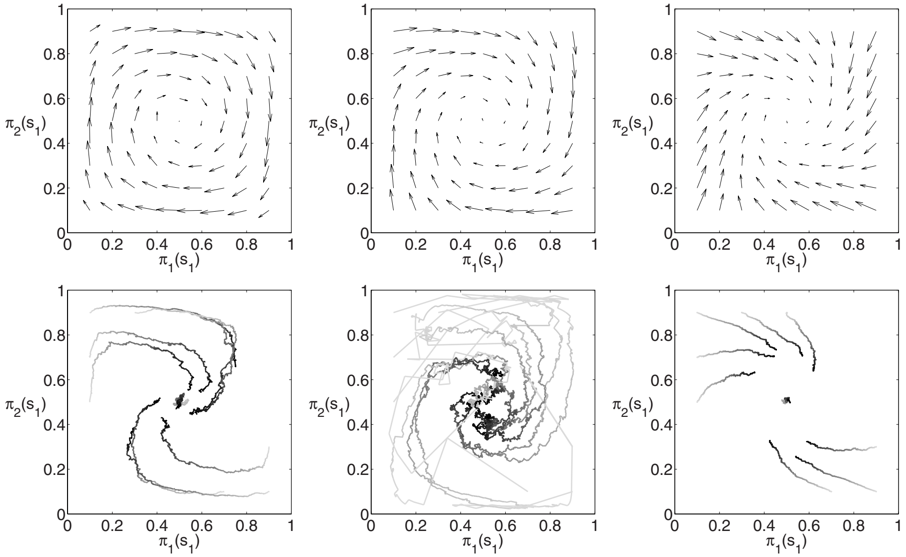

Fig. 4. Directional field plots (top row, $I = 10$) and trajectories (bottom row, $I = 600$) in the MP for the Q-learner (left), FMQ ($F = 3$, center) and lenient Q-learner ($L = 3$, right) under $\alpha = 0.01$ and $\tau = 1$

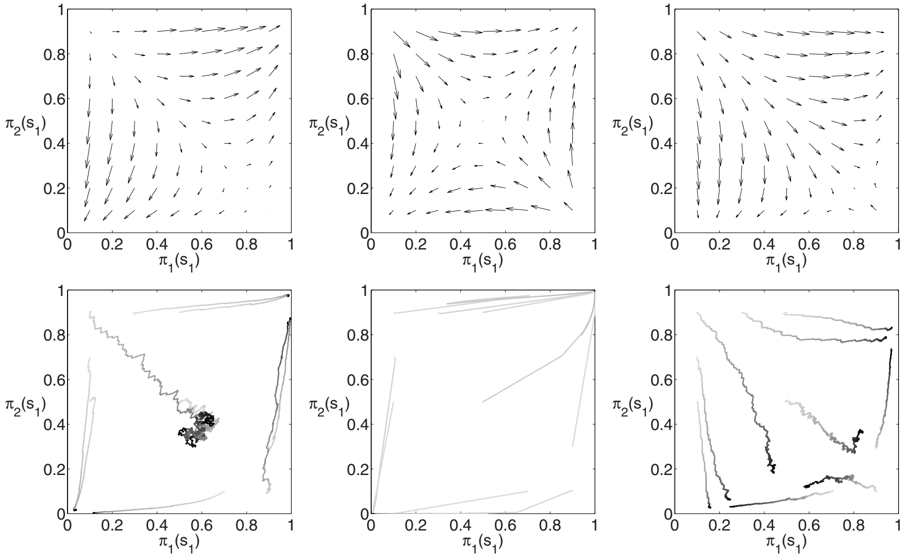

Fig. 5. Directional field plots ($I = 20$, top row) and example trajectories (bottom row) for Q-learner (left), FMQ ($F = 3$, center) and lenient Q-learning ($L = 3$, right) in BoS under $\tau = 0.5$ and $\tau = 0.01$

Table 3. Learning performance in the penalty game (PG) and the climbing game (CG). 95% confidence intervals for mean estimates of ϵ-convergence percentage with $\epsilon = 0.1$ to different joint actions and convergence time T.

FALA L_{R-I} $\alpha = 0.01$, $\beta = 0$

	Joint action	R	Convergence %	T
PG	$(1,1)$	10	46.1% ± 2.3%	1176.7± 36.7
	$(2,2)$	2	5.3% ± 1.1%	
	$(3,3)$	10	47.6% ± 2.3%	
CG	$(1,1)$	11	36.9% ± 1.9%	2623.0± 58.3
	$(2,2)$	7	15.7% ± 1.6%	
	$(2,3)$	6	10.7% ± 1.3%	
	$(3,3)$	5	4.8% ± 1.0%	

FALA L_{R-I} $\alpha = 0.01$, $\beta = 0.001$

	Joint action	R	Convergence %	T
PG	$(1,1)$	10	49.3% ± 2.2%	1247.5± 37.1
	$(2,2)$	2	0.0% ± 0.0%	
	$(3,3)$	10	49.9% ± 2.3%	
CG	$(1,1)$	11	34.5% ± 2.0%	3347.1± 73.8
	$(2,2)$	7	0.3% ± 0.2%	
	$(2,3)$	6	0.0% ± 0.0%	
	$(3,3)$	5	0.0% ± 0.0%	

FALA L_{R-I} $\alpha = 0.01$, $\beta = 0.0001$

	Joint action	R	Convergence %	T
PG	$(1,1)$	10	49.0% ± 2.2%	1160.6± 37.9
	$(2,2)$	2	3.2% ± 0.8%	
	$(3,3)$	10	46.4% ± 2.1%	
CG	$(1,1)$	11	39.7% ± 2.3%	2735.8± 79.1
	$(2,2)$	7	15.1% ± 1.5%	
	$(2,3)$	6	6.2% ± 1.0%	
	$(3,3)$	5	2.0% ± 0.7%	

PLA $\alpha = 1$, $\sigma = 0.05$, $L = 1.8$, $K = 0.5$, $n = 1$

	Joint action	R	Convergence %	T
PG	$(1,1)$	10	41.5% ± 2.0%	4927.5± 3.4
	$(2,2)$	2	0.5% ± 0.3%	
	$(3,3)$	10	41.1% ± 2.1%	
CG	$(1,1)$	11	52.0% ± 1.9%	4943.8± 3.5
	$(2,2)$	7	10.7% ± 1.2%	
	$(2,3)$	6	6.5% ± 1.0%	
	$(3,3)$	5	3.8% ± 0.8%	

other learners are rare in the penalty game, whereas the climbing game is quite challenging. For the first time parameterized learning automata significantly outperform standard FALA. The bound L is chosen small to keep exploration on a constant level whereas the learning rate is set to a high value to approach decisively high payoff situations.

The simplex plots in Figure 6 show example runs for FALA and PLA under two initial conditions. The first condition studies the learning dynamics starting from a strategy profile near $((0,0,1),(0,0,1))$ corresponding to common payoff 5. From this point the learners virtually have to climb up in order to reach the optimal solution, which is in this case a Pareto optimal Nash equilibrium and a maximum social welfare profile. The second initial condition is near another Nash equilibrium yielding a payoff equal to 7. This point challenges the learners as well; in order to improve the common payoff both agents simultaneously have to switch to action 1. If only one agent switches the reward reduces to 6, 0 or -10.

Value-Iteration Learners. The results for regular Q-learning and FMQ learning from [4] as well as the results for lenient Q-learning from [8] are confirmed. Table 4 compares the algorithms' performances in both games. All results of this subsection refer to the games with penalties $c = p = 10$. Confidence intervals are calculated from 101 samples that average over 20 runs.

Experiments in penalty games make use of an iteration dependent temperature and a learning rate of $\alpha = 0.9$. The experiments use a decay factor $s = 0.006$ and an initial temperature $\tau^0 = 500$.

$$\tau^t \leftarrow \left(\tau^0 - 1\right) \cdot e^{-s \cdot t} + 1 \tag{11}$$

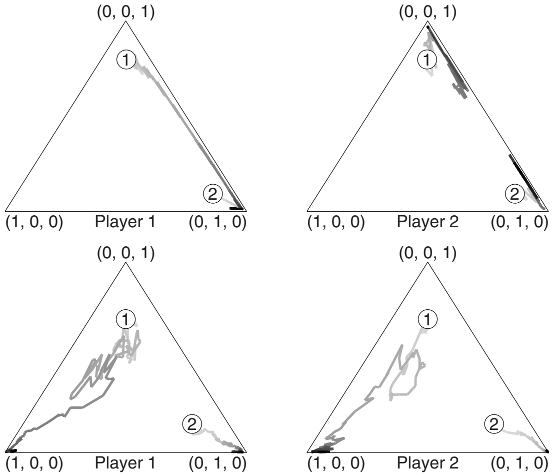

Fig. 6. Simplex plots for FALA (top) and PLA (bottom) in the climbing game shown with two initial conditions each. Convergence after $t = 200$ epochs with $\alpha = 0.1$, $\beta = 0.01$ ($L_{R-\epsilon P}$) for FALA and $\alpha = 0.3$, $\sigma = 0.2$, $L = 4$, $K = n = 1$ for PLA.

Table 4. Average ϵ-near convergence over 2020 runs ($\epsilon = 0.1$) to the maximum social welfare policy for all learners in the CG and PG with penalties $c = p = 10$. All learners under $\alpha = 0.9$ and decreasing τ, $F = 10$ and $L = 10$ for CG and $L = 5$ for PG.

Learner	CG	PG
Q	21.8%	79.6%
FMQ	98.9%	100.0%
Lenient Q	99.9%	99.3%

Table 5 lists the confidence intervals of ϵ-near convergence with $\epsilon = 0.1$ to pure strategy profiles in percentages.

The strategy profile π^* corresponding to (T, L) is a Pareto efficient Nash equilibrium. Furthermore, it is the maximal social welfare profile and as such the desired point of convergence for cooperative players. It also yields the highest individual payoff, so it is as well the best strategy profile for independent learners.

The climbing game cannot be solved satisfactorily by the regular Q-learner. It can be observed, that Q-learning converges to π^* in about 21.8%. Both adaptations outperform this by far, FMQ with $F = 10$ achieves 98.9% while lenient Q-learning with $L = 10$ achieves 99.9%.

Example trajectories of the FMQ-learner are visualized in Figure 7.

5.3 Scaling: Dispersion and Guessing Games

Policy-Iteration Learners. The following scaling experiments are conducted to give an intuition about how well learning automata scale with respect to the number of agents in coordination and anti-coordination games. Therefore the

Table 5. 95% Confidence intervals for ϵ-near convergence with $\epsilon = 0.1$ in CG to pure strategy profiles in percent. Analyzed after $I = 2000$ iterations with $\alpha = 0.9$ and decreasing τ. Q-learner (top, 43.1% not converged to any pure strategy profile), FMQ-learner (middle, $F = 10$, 0.1% n.c.) and lenient Q-learner (bottom, $L = 10$, 0.1% n.c.). Player 1 chooses T, M or B and player two chooses L, C or R. The maximal social welfare profile is (T, L).

Q-learner

	L		C		R	
T	21.8 ±	1.9	0 ±	0	0 ±	0
M	0 ±	0	0.2 ±	0.2	6.0 ±	1.1
B	0 ±	0	0 ±	0	28.9 ±	1.9

FMQ-learner $F = 10$

	L		C		R	
T	98.9 ±	0.4	0 ±	0	0 ±	0
M	0 ±	0	0.6 ±	0.3	0.1 ±	0.1
B	0 ±	0	0 ±	0	0.3 ±	0.3

Lenient Q-learner $L = 10$

	L		C		R	
T	99.9 ±	0.2	0 ±	0	0 ±	0
M	0 ±	0	0 ±	0	0 ±	0
B	0 ±	0	0 ±	0	0 ±	0

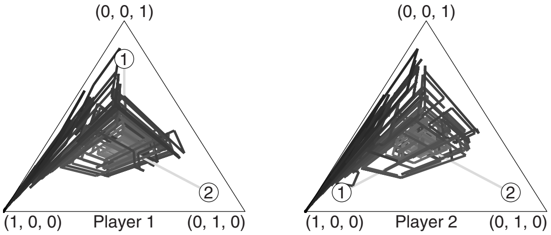

Fig. 7. Two example trajectories for both FMQ-learners with $F = 3$ in the CG show convergence to the global optimum (T, L) starting close to (B, L) in (1) and (M, C) in (2). Initial high exploration causes large policy shifts while eventual exploitation allows convergence.

emphasis lies not on statistical sampling but rather on an intuitive understanding of example runs.

Figure 8 shows the cumulative reward plot for FALA using the $L_{R-\epsilon P}$ update scheme in the dispersion game as well as PLA in the guessing game. In both cases the learners converge to a maximum social welfare profile. Note that the dispersion game has $n!$ distinct maximum social welfare profiles whereas the guessing game has only n. Convergence time T (see (10)) for the two example runs are $T \approx 600$ and $T \approx 6000$ respectively.

For the dispersion game the same facts apply. The payoff function (8) sharply rewards only the case in which an agent has exclusively selected an action. Again the agent has to be decisive in order to learn this action quickly. Furthermore randomness is required to escape from a zero reward situation where no exclusive action slot has been found yet.

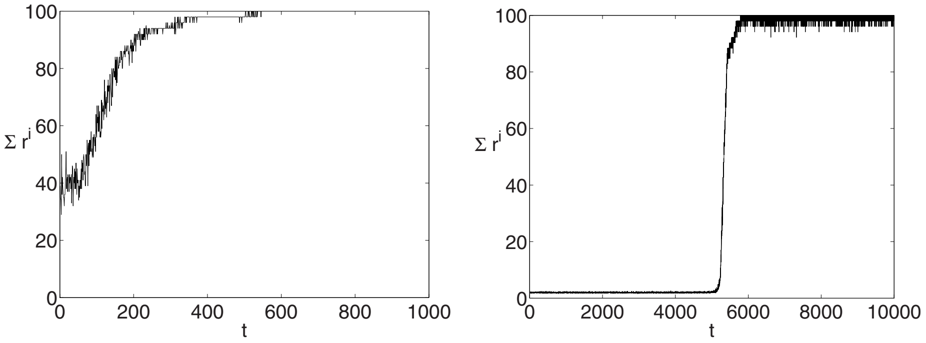

Fig. 8. Cumulative reward plot for FALA in the dispersion game (left) and for PLA in the guessing game (right) with both 100 agents. Learning parameters are $\alpha = 0.1$, $\beta = 0.01$ ($L_{R-\epsilon P}$) and $\alpha = 1$, $\sigma = 0.005$, $L = 10$, $K = n = 1$ for FALA and PLA respectively.

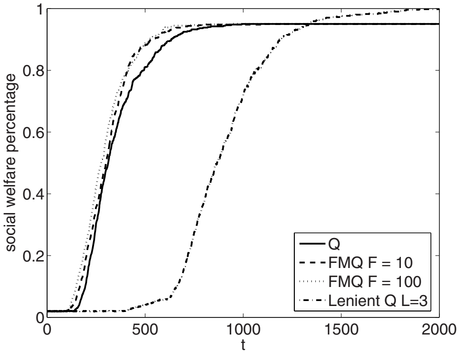

Fig. 9. Social welfare percentage over iterations in the GG for different learners ($n = 100$, $\alpha = 1$ and $\tau = 0.01$; averaged over 10 runs). All learners except the lenient Q-learner converged to the suboptimal solution with two equally sized groups once.

Value-Iteration Learners. The GG with n players has n maximal social welfare profiles while the DG has $n!$ maximal social welfare profiles. As $n!$ is much larger than n the DG can be solved much faster than the GG.

In the GG all agents try to group as quickly as possible. Convergence to suboptimal solutions, e.g. two groups with equally many agents, are not uncommon. Figure 9 shows the speed of convergence for different learners in the guessing game. An increase of the FMQ persistence F shifts the grouping process to an earlier iteration. However, there is a point of diminishing returns. Furthermore, increasing F does not seem to increase the qualitative convergence while lenient Q-learning slows down the learning process but converges to a maximal social welfare equilibrium.

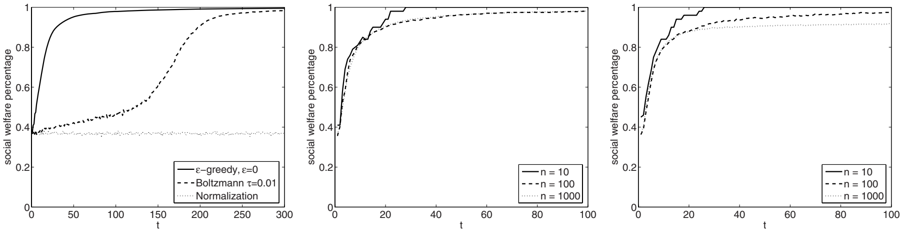

Fig. 10. Social welfare percentage over iterations in the DG (averages of 10 runs). Q-learner under $\alpha = 0.1$, different action selection methods (left, $n = 1000$), different numbers of agents: Q-learner with ϵ-greedy (center, $\epsilon = 0$) and FMQ with Boltzmann distribution (right, $F = 10$, $\tau = 0.01$).

Figure 10 (left) visualizes the impact of the exploration method. Furthermore, scalability of the Q-learner with an ϵ-greedy action selection is compared to the FMQ heuristic with a Boltzmann action selection. An equilibrium can be found within reasonable numbers of iterations using the ϵ-greedy action selection. This action selection method actually allows to scale up to thousand agents without significant deterioration of the performance over the iterations, the lines for $n = 100$ and $n = 1000$ almost coincide in the corresponding plot. FMQ also scales well but has a stronger dependency of the maximal convergence on the number of agents. However, a performance of above 90% is achieved within the first 50 iterations even for $n = 1000$.

6 Discussion and Future Work

The obtained results demonstrate the learning performance of all algorithms considered. Graphical tools help to understand the learning dynamics of the algorithms. Adequate parameter settings for convergence to equilibria have been shown. Overall, it can be observed that all results are quite sensitive to the parameter settings.

In general, low temperatures and accompanying high exploitation lead to convergence to pure strategy profiles while higher temperatures that impose more exploration allow convergence to mixed equilibria. Furthermore, higher convergence to mixed equilibria is achieved by smaller learning rates. In contrast, high learning rates can be applied to overcome penalties in cooperative coordination games. FMQ-learning with high persistence F drives the learning process to pure strategy profiles within few iterations if the temperature is low. Lenient Q-learning finds mixed solutions but requires many iterations to converge. Contrary, FALA are more robust with respect to parameter changes. Scaling down the learning rate generally results in better convergence performance although increasing the required number of iterations. The $L_{R-\epsilon P}$ scheme gives a good trade-off between the two extremes L_{R-I} and L_{R-P}. It unites the ability to find mixed policies with high percentage of global convergence. This result is also confirmed by the various visualizations used in this work.

An second observation from our research is that formal criteria may fail under practical conditions. Thus empirical results as conducted in this work are essential. The PLA superimpose noise on the learning update in order to overcome convergence to suboptimal solutions and for this theoretical guarantees can be found [5]. However, for a challenging task, such as the climbing game, we could not reach it in an experimental setting. Although this may be caused by the limited number of epochs, the central issue is clearly the high dimensional parameter space. The PLA update rule comprises five parameters that all have to be tuned to fit the environment.

Scaling experiments in the Dispersion Game reveal high performance of the Q-learner with an ϵ-greedy action selection and FMQ with a Boltzmann action selection under low temperatures. High exploitation imposed by these action selection methods is required to facilitate a quick dispersion over the actions. For the same reason, a high learning rate is needed for the LA algorithms that also indicate good scaling potential. The Guessing Game requires quick grouping but also more exploration to avoid suboptimal solutions with several, approximately equally sized groups. This implies that a trade-off needs to be chosen between fast convergence and optimal convergence. However, it should be noted that this work presents only example runs for the two coordination games and therefore cannot give any general conclusion on scaling. In future investigations we would like to test the scalability of FALA and PLA more extensively.

7 Conclusion

This research has experimentally studied the learning performance of state-of-the-art value-based and policy-based iterators in multi-agent games. In particular Q-learning, Lenient Q-learning, FMQ, FALA, and PLA are surveyed. A variety of competitive and cooperative games have served as a testbed to analyze their learning performance. Furthermore, various visualization methods are used and interconnected to reveal the complex learning dynamics of LA in games.

From the performances of independent reinforcement learners we may conclude that the learners are highly dependent on the correct parameter tuning. For the value-based methods, high temperatures enhance exploration and enable the convergence to mixed equilibria, while small temperatures enforce exploitation and increase the probability of convergence to pure strategy profiles. Stability of the learning process can be supported by small learning rates and a temperature that decreases over time. In the context of penalty games, the adaptations FMQ and lenient Q-learning outperform the regular Q-learner significantly and converge to the global optimum. For the policy-based methods, results show that the $L_{R-\epsilon P}$ scheme maintains a good trade-off between convergence performance and robustsness.

References

1. Claus, C., Boutilier, C.: The dynamics of reinforcement learning in cooperative multiagent systems. In: Proceedings of the National Conference on Artificial Intelligence, vol. 15, pp. 746–752 (1998)

2. Gibbons, R.: A Primer in Game Theory. Harvester Wheatsheaf (1992)
3. Kaelbling, L.P., Littman, M.L.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)
4. Kapetanakis, S., Kudenko, D.: Reinforcement learning of coordination in cooperative multi-agent systems (2002)
5. Narendra, K., Thathachar, M.A.L.: Learning Automata An Introduction. Prentice-Hall, Englewood Cliffs, NJ (1989)
6. Nowé, A., Verbeeck, K., Peeters, M.: Learning automata as a basis for multi agent reinforcement learning. In: Tuyls, K., 't Hoen, P.J., Verbeeck, K., Sen, S. (eds.) LAMAS 2005. LNCS (LNAI), vol. 3898, pp. 71–85. Springer, Heidelberg (2006)
7. Panait, L., Tuyls, K.: Theoretical advantages of lenient q-learners: An evolutionary game theoretic perspective. In: AAMAS 2007 (2007)
8. Panait, L., Sullivan, K., Luke, S.: Lenience towards teammates helps in cooperative multiagent learning. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems – AAMAS-2006, ACM Press, New York (2006)
9. Redondo, F.V.: Game Theory and Economics. Cambridge University Press, Cambridge, MA (2001)
10. Sen, S., Sekaran, M., Hale, J.: Learning to coordinate without sharing information. In: Twelfth National Conference on Artificial Intelligence, pp. 426–431 (1994)
11. Shoham, Y., Powers, R., Grenager, T.: If multi-agent learning is the answer, what is the question (2006)
12. Stone, P.: Multiagent learning is not the answer. It is the question. To appear in Artificial Intelligence (2007)
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
14. Thathachar, M.A.L., Phansalkar, V.V.: Learning the global maximum with parameterized learning automata. IEEE Transactions on Neural Networks 6(2), 398–406 (1995)
15. Thathachar, M.A.L., Sastry, P.S.: Varieties of learning automata: An overview. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 32(6), 711–722 (2002)
16. Tuyls, K., Parsons, S.: What evolutionary game theory tells us about multiagent learning. Preprint submitted to Elsevier (2007)
17. Tuyls, K., t Hoen, P.J., Vanschoenwinkel, B.: An evolutionary dynamical analysis of multi-agent learning in iterated games. Autonomous Agents and Multi-Agent Systems 12, 115–153 (2006)
18. Verbeeck, K., Nowé, A., Parent, J., Tuyls, K.: Exploring selfish reinforcement learning in repeated games with stochastic rewards, pp. 239–269 (2006)
19. Watkins. Learning from delayed rewards. PhD thesis, King's College, Oxford (1989)
20. t Hoen, P.J., Tuyls, K., Panait, L., Luke, S., la Poutré, H.: An overview of cooperative and competitive multiagent learning. In: LAMAS 2005. LNCS (LNAI), vol. 3898, pp. 1–46. Springer, Heidelberg (2006)